User's Manual

RA4W1 Group

User's Manual: Hardware

32-bit MCU
Renesas Advanced (RA) Family
Renesas RA4 - Efficiency Series

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

Renesas Electronics
www.renesas.com

Rev.1.00 Mar 2020

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

2. Audience

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

3. Renesas Publications

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

Component	Document type	Description
Microcontrollers	Datasheet	Features, overview, and electrical characteristics of the MCU
	User’s Manual: Hardware	MCU specifications such as pin assignments, memory maps, peripheral functions, electrical characteristics, timing diagrams, and operation descriptions
	Application Notes	Technical notes, board design guidelines, and software migration information
	Technical Update (TU)	Preliminary reports on product specifications such as restriction and errata
Software	Datasheet	Functional descriptions and specific performance data for software modules
	User’s Manual: Software	API reference and programming information
	Application Notes	Project files, guidelines for software programming, and application examples to develop embedded software applications
Tools & Kits, Solutions	User’s Manual: Development Tools	User’s manual and quick start guide for developing embedded software applications with Development 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

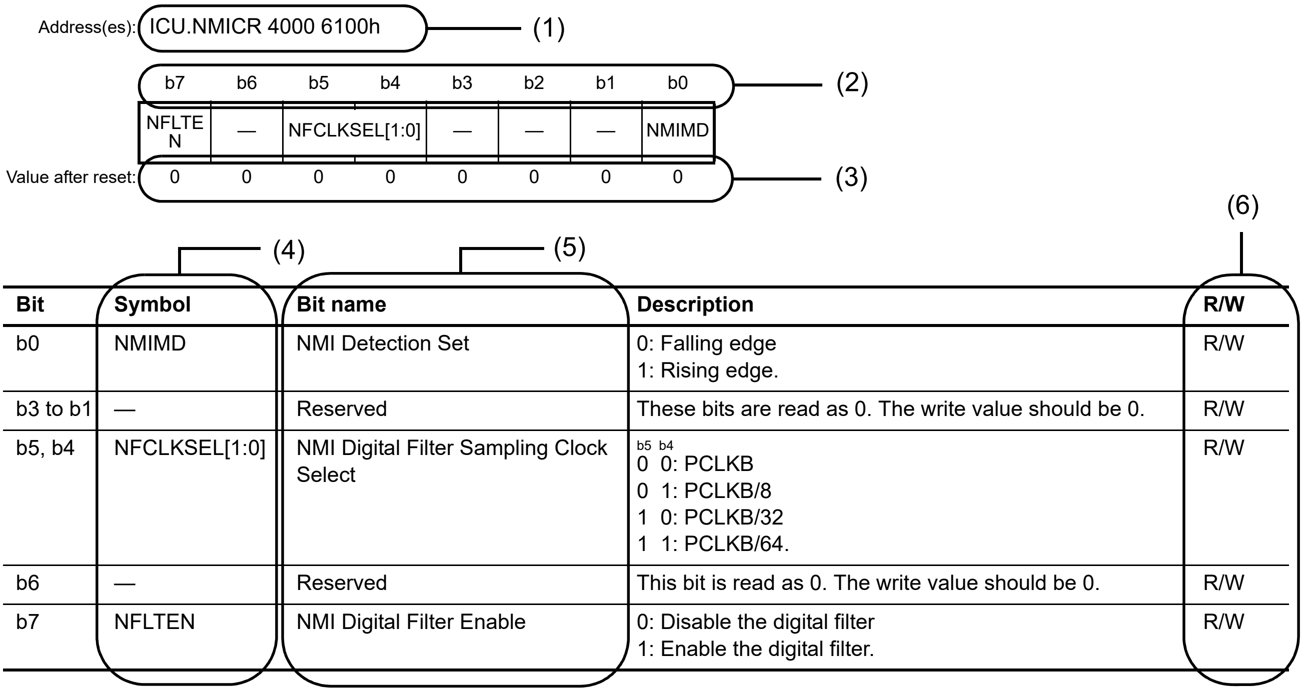
The following terms have special meanings:

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)


Address(es): ICU.NMICR 4000 6100h
(1)
b7 b6 b5 b4 b3 b2 b1 b0
(2)
NFLTEN — NFCLKSEL[1:0] — — — NMIMD
Value after reset: 0 0 0 0 0 0 0 0
(3)
(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.

Bluetooth® is a trademark of the Bluetooth SIG, Inc.

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

Contents

Features 47

1. Overview 48
  - 1.1 Function Outline 48
  - 1.2 Block Diagram 54
  - 1.3 Part Numbering 54
  - 1.4 Function Comparison 56
  - 1.5 Pin Functions 57
  - 1.6 Pin Assignments 60
  - 1.7 Pin Lists 61

2. CPU 64
  - 2.1 Overview 64
    - 2.1.1 CPU 64
    - 2.1.2 Debug 64
    - 2.1.3 Operating Frequency 65
  - 2.2 MCU Implementation Options 66
  - 2.3 Trace Interface 66
  - 2.4 JTAG/SWD Interface 67
  - 2.5 Debug Mode 67
    - 2.5.1 Debug Mode Definition 67
    - 2.5.2 Debug Mode Effects 67
      - 2.5.2.1 Low power mode 67
      - 2.5.2.2 Reset 67
  - 2.6 Programmers Model 68
    - 2.6.1 Address Spaces 68
    - 2.6.2 Cortex-M4 Peripheral Address Map 68
    - 2.6.3 CoreSight ROM Table 69
      - 2.6.3.1 ROM entries 69
      - 2.6.3.2 CoreSight component registers 69
    - 2.6.4 DBGREG Module 70
      - 2.6.4.1 Debug Status Register (DBGSTR) 70
      - 2.6.4.2 Debug Stop Control Register (DBGSTOPCR) 71
      - 2.6.4.3 Trace Control Register (TRACECTR) 71
      - 2.6.4.4 DBGREG CoreSight component registers 72
    - 2.6.5 OCDREG Module 72
      - 2.6.5.1 ID Authentication Code Register (IAUTH0 to 3) 72
      - 2.6.5.2 MCU Status Register (MCUSTAT) 73
      - 2.6.5.3 MCU Control Register (MCUCTRL) 74
      - 2.6.5.4 OCDREG CoreSight component registers 74
  - 2.7 CoreSight ATB Funnel 75

2.8 Flash Patch and Break Unit 75
2.9 SysTick System Timer 75
2.10 CoreSight Time Stamp Generator 75
2.11 OCD Emulator Connection 75
  2.11.1 DBGEN 76
  2.11.2 Unlock ID Code 76
  2.11.3 Restrictions on Connecting an OCD Emulator 76
    2.11.3.1 Starting connection while in low power mode 76
    2.11.3.2 Changing low power mode while in OCD mode 76
    2.11.3.3 Modifying the unlock ID code in OSIS 77
    2.11.3.4 Connecting sequence and JTAG/SWD authentication 77
2.12 References 78

3. Operating Modes 79
3.1 Overview 79
3.2 Details of Operating Modes 79
  3.2.1 Single-Chip Mode 79
  3.2.2 SCI Boot Mode 79
  3.2.3 USB Boot Mode 79
3.3 Operating Mode Transitions 79
  3.3.1 Operating Mode Transitions as Determined by the Mode-Setting Pin 79

4. Address Space 80
4.1 Overview 80

5. Memory Mirror Function (MMF) 81
5.1 Overview 81
5.2 Register Descriptions 81
  5.2.1 MemMirror Special Function Register (MMSFR) 81
  5.2.2 MemMirror Enable Register (MMEN) 82
5.3 Operation 82
  5.3.1 MMF Operation 82
  5.3.2 Setting Example 86

6. Resets 87
6.1 Overview 87
6.2 Register Descriptions 90
  6.2.1 Reset Status Register 0 (RSTSR0) 90
  6.2.2 Reset Status Register 1 (RSTSR1) 91
  6.2.3 Reset Status Register 2 (RSTSR2) 93
6.3 Operation 94
  6.3.1 RES Pin Reset 94
  6.3.2 Power-On Reset 94
  6.3.3 Voltage Monitor Reset 95
  6.3.4 Independent Watchdog Timer Reset 96

6.3.5 Watchdog Timer Reset ... 96
6.3.6 Software Reset ... 97
6.3.7 Determination of Cold/Warm Start ... 97
6.3.8 Determination of Reset Generation Source ... 97
7. Option-Setting Memory ... 99
7.1 Overview ... 99
7.2 Register Descriptions ... 99
7.2.1 Option Function Select Register 0 (OFS0) ... 99
7.2.2 Option Function Select Register 1 (OFS1) ... 102
7.2.3 MPU Registers ... 103
7.2.4 Access Window Setting Control Register (AWSC) ... 104
7.2.5 Access Window Setting Register (AWS) ... 105
7.2.6 OCD/Serial Programmer ID Setting Register (OSIS) ... 106
7.3 Setting Option-Setting Memory ... 107
7.3.1 Allocation of Data in Option-Setting Memory ... 107
7.3.2 Setting Data for Programming Option-Setting Memory ... 107
7.4 Usage Note ... 108
7.4.1 Data for Programming Reserved Areas and Reserved Bits in the Option-Setting Memory ... 108
8. Low Voltage Detection (LVD) ... 109
8.1 Overview ... 109
8.2 Register Descriptions ... 111
8.2.1 Voltage Monitor 1 Circuit Control Register 1 (LVD1CR1) ... 111
8.2.2 Voltage Monitor 1 Circuit Status Register (LVD1SR) ... 111
8.2.3 Voltage Monitor Circuit Control Register (LVCMPCR) ... 112
8.2.4 Voltage Detection Level Select Register (LVDLVLR) ... 112
8.2.5 Voltage Monitor 1 Circuit Control Register 0 (LVD1CR0) ... 113
8.3 VCC Input Voltage Monitor ... 113
8.3.1 Monitoring Vdet0 ... 113
8.3.2 Monitoring Vdet1 ... 113
8.4 Reset from Voltage Monitor 0 ... 114
8.5 Interrupt and Reset from Voltage Monitor 1 ... 114
8.6 Event Link Output ... 116
8.6.1 Interrupt Handling and Event Linking ... 116
9. Clock Generation Circuit ... 117
9.1 Overview ... 117
9.2 Register Descriptions ... 120
9.2.1 System Clock Division Control Register (SCKDIVCR) ... 120
9.2.2 System Clock Source Control Register (SCKSCR) ... 122
9.2.3 PLL Clock Control Register 2 (PLLCCR2) ... 123
9.2.4 PLL Control Register (PLLCR) ... 123

9.2.5 Memory Wait Cycle Control Register (MEMWAIT) 124
9.2.6 Main Clock Oscillator Control Register (MOSCCR) 126
9.2.7 Sub-Clock Oscillator Control Register (SOSCCR) 127
9.2.8 Low-Speed On-Chip Oscillator Control Register (LOCOCR) 128
9.2.9 High-Speed On-Chip Oscillator Control Register (HOCOCR) 129
9.2.10 Middle-Speed On-Chip Oscillator Control Register (MOCOCR) 130
9.2.11 Oscillation Stabilization Flag Register (OSCSF) 130
9.2.12 Oscillation Stop Detection Control Register (OSTDCR) 132
9.2.13 Oscillation Stop Detection Status Register (OSTDSR) 133
9.2.14 Main Clock Oscillator Wait Control Register (MOSCWTCR) 134
9.2.15 High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR) 135
9.2.16 Main Clock Oscillator Mode Oscillation Control Register (MOMCR) 136
9.2.17 Sub-Clock Oscillator Mode Control Register (SOMCR) 136
9.2.18 Segment LCD Source Clock Control Register (SLCDSCKCR) 137
9.2.19 Clock Out Control Register (CKOCR) 138
9.2.20 LOCO User Trimming Control Register (LOCOUTCR) 139
9.2.21 MOCO User Trimming Control Register (MOCOUTCR) 139
9.2.22 HOCO User Trimming Control Register (HOCOUTCR) 140
9.2.23 Trace Clock Control Register (TRCKCR) 140
9.2.24 USB Clock Control Register (USBCKCR) 141
9.3 Main Clock Oscillator 141
9.3.1 Connecting a Crystal Resonator 141
9.3.2 External Clock Input 142
9.3.3 Notes on External Clock Input 142
9.4 Sub-Clock Oscillator 142
9.4.1 Connecting a 32.768-kHz Crystal Resonator 142
9.5 Dedicated Clock Oscillator for Bluetooth 142
9.5.1 Connecting the Oscillator 142
9.5.2 Connecting the Bluetooth-Dedicated Clock Output Pin 143
9.6 Oscillation Stop Detection Function 143
9.6.1 Oscillation Stop Detection and Operation after Detection 143
9.6.2 Oscillation Stop Detection Interrupts 145
9.7 PLL Circuit 146
9.8 Internal Clock 146
9.8.1 System Clock (ICLK) 146
9.8.2 Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD) 147
9.8.3 Flash Interface Clock (FCLK) 148
9.8.4 USB Clock (UCLK) 148
9.8.5 CAN Clock (CANMCLK) 148
9.8.6 CAC Clock (CACCLK) 148
9.8.7 RTC-Dedicated Clock (RTCSCLK, RTCLCLK) 148

9.8.8 IWDT-Dedicated Clock (IWDTCLK) 148
9.8.9 AGT-Dedicated Clock (AGTSCLK, AGTLCLK) 149
9.8.10 SysTick Timer-Dedicated Clock (SYSTICCLK) 149
9.8.11 Segment LCDC Source Clock (LCDSRCCLK) 149
9.8.12 Clock/Buzzer Output Clock (CLKOUT) 149
9.8.13 JTAG Clock (JTAGTCK) 149
9.8.14 Clocks for BLE 149
9.9 Usage Notes 149
9.9.1 Notes on Clock Generation Circuit 149
9.9.2 Notes on Resonator 150
9.9.3 Notes on Board Design 150
9.9.4 Notes on Resonator Connect Pin 150
10. Clock Frequency Accuracy Measurement Circuit (CAC) 151
10.1 Overview 151
10.2 Register Descriptions 152
10.2.1 CAC Control Register 0 (CACR0) 152
10.2.2 CAC Control Register 1 (CACR1) 153
10.2.3 CAC Control Register 2 (CACR2) 154
10.2.4 CAC Interrupt Control Register (CAICR) 155
10.2.5 CAC Status Register (CASTR) 156
10.2.6 CAC Upper-Limit Value Setting Register (CAULVR) 157
10.2.7 CAC Lower-Limit Value Setting Register (CALLVR) 157
10.2.8 CAC Counter Buffer Register (CACNTBR) 157
10.3 Operation 157
10.3.1 Measuring Clock Frequency 157
10.3.2 Digital Filtering of Signals on CACREF Pin 158
10.4 Interrupt Requests 159
10.5 Usage Note 159
10.5.1 Settings for the Module-Stop Function 159
11. Low Power Modes 160
11.1 Overview 160
11.2 Register Descriptions 163
11.2.1 Standby Control Register (SBYCR) 163
11.2.2 Module Stop Control Register A (MSTPCRA) 164
11.2.3 Module Stop Control Register B (MSTPCRB) 164
11.2.4 Module Stop Control Register C (MSTPCRC) 166
11.2.5 Module Stop Control Register D (MSTPCRD) 167
11.2.6 Operating Power Control Register (OPCCR) 168
11.2.7 Sub Operating Power Control Register (SOPCCR) 168
11.2.8 Snooze Control Register (SNZCR) 169
11.2.9 Snooze End Control Register (SNZEDCR) 170

11.2.10 Snooze Request Control Register (SNZREQCR) 171
11.2.11 Flash Operation Control Register (FLSTOP) 173
11.2.12 Power Save Memory Control Register (PSMCR) 173
11.2.13 System Control OCD Control Register (SYOCDCR) 174
11.3 Reducing Power Consumption by Switching Clock Signals 174
11.4 Module-Stop Function 174
11.5 Function for Lower Operating Power Consumption 174
11.5.1 Setting Operating Power Control Mode 174
11.5.2 Operating range 176
11.6 Sleep Mode 179
11.6.1 Transition to Sleep Mode 179
11.6.2 Canceling Sleep Mode 179
11.7 Software Standby Mode 180
11.7.1 Transition to Software Standby Mode 180
11.7.2 Canceling Software Standby Mode 181
11.7.3 Example of Software Standby Mode Application 181
11.8 Snooze Mode 183
11.8.1 Transition to Snooze Mode 183
11.8.2 Canceling Snooze Mode 183
11.8.3 Return to Software Standby Mode 184
11.8.4 Snooze Operation Example 186
11.9 Usage Notes 189
11.9.1 Register Access 189
11.9.2 I/O Port States 190
11.9.3 Module-Stop State of DMAC and DTC 190
11.9.4 Internal Interrupt Sources 190
11.9.5 Transition to Low Power Modes 190
11.9.6 Timing of WFI Instruction 191
11.9.7 Writing WDT/IWDT Registers by DMAC or DTC in Sleep Mode or Snooze Mode 191
11.9.8 Oscillators in Snooze Mode 191
11.9.9 Snooze Mode Entry by RXD0 Falling Edge 191
11.9.10 Using SCI0 in Snooze Mode 191
11.9.11 Conditions of A/D Conversion Start in Snooze Mode 191
11.9.12 Conditions of CTSU in Snooze Mode 191
11.9.13 ELC Event in Snooze Mode 191
11.9.14 Module-Stop Function for ADC140 192
11.9.15 Module-Stop Function for an Unused Circuit 192

12. Battery Backup Function 193
12.1 Overview 193
12.1.1 Features of Battery Backup Function 193
12.1.2 Battery Power Supply Switch 193

12.1.3 VBATT Pin Low Voltage Detection 193
12.1.4 VBATT_R Low Voltage Detection 193
12.1.5 Backup Registers 193
12.1.6 VBATT Wakeup Control Function 193
12.1.7 Time capture Pin Detection 194
12.2 Register Descriptions 196
12.2.1 VBATT Control Register 1 (VBTCR1) 196
12.2.2 VBATT Control Register 2 (VBTCR2) 197
12.2.3 VBATT Status Register (VBTSR) 197
12.2.4 VBATT Comparator Control register (VBTCMPCR) 198
12.2.5 VBATT Pin Low Voltage Detect Interrupt Control Register (VBTLVDICR) 199
12.2.6 VBATT Backup Register (VBTBKRn) (n = 0 to 511) 199
12.2.7 VBATT Wakeup Control Register (VBTWCTLR) 199
12.2.8 VBATT Wakeup I/O 0 Output Trigger Select Register (VBTWCH0OTSR) 200
12.2.9 VBATT Input Control Register (VBTICTLR) 200
12.2.10 VBATT Output Control Register (VBTOCTLR) 201
12.2.11 VBATT Wakeup Trigger Source Enable Register (VBTWTER) 201
12.2.12 VBATT Wakeup Trigger Source Edge Register (VBTWEGR) 202
12.2.13 VBATT Wakeup Trigger Source Flag Register (VBTWFR) 202
12.2.14 Backup Register Access Control Register (BKRACR) 203
12.3 Operation 204
12.3.1 Battery Backup Function 204
12.3.2 VBATT Battery Power Supply Switch Usage 205
12.3.3 VBATT Pin Low Voltage Detection Procedures 206
12.3.4 VBATT Backup Register Usage 207
12.3.5 VBATT Wakeup Control Function Usage 207
12.4 Usage Notes 208

13. Register Write Protection 209
13.1 Overview 209
13.2 Register Descriptions 209
13.2.1 Protect Register (PRCR) 209

14. Interrupt Controller Unit (ICU) 210
14.1 Overview 210
14.2 Register Descriptions 211
14.2.1 IRQ Control Register i (IRQCRi) (i = 0 to 4, 6, 7, 9, 11, 14, 15) 212
14.2.2 Non-Maskable Interrupt Status Register (NMISR) 213
14.2.3 Non-Maskable Interrupt Enable Register (NMIER) 215
14.2.4 Non-Maskable Interrupt Status Clear Register (NMICLR) 217
14.2.5 NMI Pin Interrupt Control Register (NMICR) 218
14.2.6 ICU Event Link Setting Register n (IELSRn) 219
14.2.7 DMAC Event Link Setting Register n (DELSRn) 220

14.2.8 SYS Event Link Setting Register (SELSR0) 221
14.2.9 Wake Up Interrupt Enable Register (WUPEN) 221
14.3 Vector Table 223
14.3.1 Interrupt Vector Table 223
14.3.2 Event Number 224
14.4 Interrupt Operation 229
14.4.1 Detecting Interrupts 229
14.4.2 Selecting Interrupt Request Destinations 230
14.4.2.1 CPU interrupt request 230
14.4.2.2 DTC activation 230
14.4.2.3 DMAC activation 231
14.4.3 Digital Filter 231
14.4.4 External Pin Interrupts 232
14.5 Non-Maskable Interrupt Operation 232
14.6 Return from Low Power Mode 233
14.6.1 Return from Sleep mode 233
14.6.2 Return from Software Standby mode 233
14.6.3 Return from Snooze mode 233
14.7 Using the WFI instruction with Non-Maskable Interrupts 234
14.8 Reference 234
15. Buses 235
15.1 Overview 235
15.2 Description of Buses 236
15.2.1 Main Buses 236
15.2.2 Slave Interface 236
15.2.3 Parallel Operation 236
15.2.4 Restriction on Endianness 237
15.3 Register Descriptions 237
15.3.1 Master Bus Control Register (BUSMCNT<master>) 237
15.3.2 Slave Bus Control Register (BUSSCNT<slave>) 238
15.3.3 Bus Error Address Register (BUSnERRADD) (n = 1 to 4) 239
15.3.4 Bus Error Status Register (BUSnERRSTAT) (n = 1 to 4) 239
15.4 Bus Error Monitoring Section 240
15.4.1 Error Type that Occurs by Bus 240
15.4.2 Operation when a Bus Error Occurs 240
15.4.3 Conditions Leading to Illegal Address Access Errors 241
15.4.4 Timeout 241
15.5 Notes on using Flash Cache 242
15.6 References 242
16. Memory Protection Unit (MPU) 243
16.1 Overview 243

16.2 CPU Stack Pointer Monitor 243
16.2.1 Protection of Registers 246
16.2.2 Overflow/Underflow Error 246
16.2.3 Register Descriptions 246
16.2.3.1 Main Stack Pointer (MSP) Monitor Start Address Register (MSPMPUSA) 247
16.2.3.2 Main Stack Pointer (MSP) Monitor End Address Register (MSPMPUEA) 247
16.2.3.3 Process Stack Pointer (PSP) Monitor Start Address Register (PSPMPUSA) 248
16.2.3.4 Process Stack Pointer (PSP) Monitor End Address Register (PSPMPUEA) 248
16.2.3.5 Stack Pointer Monitor Operation After Detection Register (MSPMPUOAD, PSPMPUOAD) 249
16.2.3.6 Stack Pointer Monitor Access Control Register (MSPMPUCTL, PSPMPUCTL) 249
16.2.3.7 Stack Pointer Monitor Protection Register (MSPMPUPT, PSPMPUPT) 250
16.3 Arm MPU 251
16.4 Bus Master MPU 251
16.4.1 Register Descriptions 252
16.4.1.1 Group A Region n Start Address Register (MMPUSAn) (n = 0 to 15) 253
16.4.1.2 Group A Region n End Address Register (MMPUEAn) (n = 0 to 15) 253
16.4.1.3 Group A Region n Access Control Register (MMPUACAn) (n = 0 to 15) 253
16.4.1.4 Bus Master MPU Control Register (MMPUCTLA) 255
16.4.1.5 Group A Protection of Register (MMPUPTA) 256
16.4.2 Operation 256
16.4.2.1 Memory protection 256
16.4.2.2 Protecting the registers 258
16.4.2.3 Memory protection error 258
16.5 Bus Slave MPU 259
16.5.1 Register Descriptions 260
16.5.1.1 Access Control Register for Memory Bus 3 (SMPUMBIU) 260
16.5.1.2 Access Control Register for Internal Peripheral Bus 9 (SMPUFBIU) 260
16.5.1.3 Access Control Register for Memory Bus 4 (SMPUSRAM0) 261
16.5.1.4 Access Control Register for Internal Peripheral Bus 1 (SMPUP0BIU) 261
16.5.1.5 Access Control Register for Internal Peripheral Bus 3 (SMPUP2BIU) 262
16.5.1.6 Access Control Register for Internal Peripheral Bus 7 (SMPUP6BIU) 263
16.5.1.7 Slave MPU Control Register (SMPUCTL) 263
16.5.2 Functions 264
16.5.2.1 Memory protection 264
16.5.2.2 Protecting the registers 264
16.5.2.3 Memory protection error 264
16.6 Security MPU 264
16.6.1 Register Descriptions (Option-Setting Memory) 265
16.6.1.1 Security MPU Program Counter Start Address Register (SECMPUPCSn) (n = 0, 1) 266

16.6.1.2 Security MPU Program Counter End Address Register (SECMPUPCEn) (n = 0, 1) ... 267
16.6.1.3 Security MPU Region 0 Start Address Register (SECMPUS0) ... 267
16.6.1.4 Security MPU Region 0 End Address Register (SECMPUE0) ... 268
16.6.1.5 Security MPU Region 1 Start Address Register (SECMPUS1) ... 268
16.6.1.6 Security MPU Region 1 End Address Register (SECMPUE1) ... 269
16.6.1.7 Security MPU Region 2 Start Address Register (SECMPUS2) ... 269
16.6.1.8 Security MPU Region 2 End Address Register (SECMPUE2) ... 270
16.6.1.9 Security MPU Region 3 Start Address Register (SECMPUS3) ... 270
16.6.1.10 Security MPU Region 3 End Address Register (SECMPUE3) ... 271
16.6.1.11 Security MPU Access Control Register (SECMPUAC) ... 271
16.6.2 Memory Protection ... 272
16.6.3 Notes on Debug ... 273
16.7 References ... 273
17. DMA Controller (DMAC) ... 274
17.1 Overview ... 274
17.2 Register Descriptions ... 276
17.2.1 DMA Source Address Register (DMSAR) ... 276
17.2.2 DMA Destination Address Register (DMDAR) ... 276
17.2.3 DMA Transfer Count Register (DMCRA) ... 277
17.2.4 DMA Block Transfer Count Register (DMCRB) ... 278
17.2.5 DMA Transfer Mode Register (DMTMD) ... 278
17.2.6 DMA Interrupt Setting Register (DMINT) ... 279
17.2.7 DMA Address Mode Register (DMAMD) ... 280
17.2.8 DMA Offset Register (DMOFR) ... 282
17.2.9 DMA Transfer Enable Register (DMCNT) ... 282
17.2.10 DMA Software Start Register (DMREQ) ... 283
17.2.11 DMA Status Register (DMSTS) ... 284
17.2.12 DMAC Module Activation Register (DMAST) ... 285
17.3 Operation ... 285
17.3.1 Transfer Mode ... 285
17.3.2 Extended Repeat Area Function ... 288
17.3.3 Address Update Function Using Offset ... 290
17.3.4 Activation Sources ... 294
17.3.5 Operation Timing ... 294
17.3.6 Execution Cycles of DMAC ... 295
17.3.7 Activating DMAC ... 295
17.3.8 Starting DMA Transfer ... 297
17.3.9 Registers during DMA Transfer ... 297
17.3.10 Channel Priority ... 298
17.4 Ending DMA Transfer ... 298

17.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations 298
17.4.2 Transfer End by Repeat Size End Interrupt.. 298
17.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow 298
17.4.4 Precautions for the End of DMA Transfer... 299
17.5 Interrupts.. 299
17.6 Event Link 300
17.7 Low Power Consumption Function 300
17.8 Usage Notes 301
17.8.1 Access to Registers during DMA Transfer 301
17.8.2 DMA Transfer to Reserved Areas 301
17.8.3 Setting the DMAC Event Link Setting Register of the Interrupt Controller Unit (ICU.DELSRn) 301
17.8.4 Suspending or Restarting DMA Activation 301

18. Data Transfer Controller (DTC).. 302
18.1 Overview.. 302
18.2 Register Descriptions... 303
18.2.1 DTC Mode Register A (MRA) 304
18.2.2 DTC Mode Register B (MRB) 304
18.2.3 DTC Transfer Source Register (SAR) 305
18.2.4 DTC Transfer Destination Register (DAR) 306
18.2.5 DTC Transfer Count Register A (CRA) 306
18.2.6 DTC Transfer Count Register B (CRB) 307
18.2.7 DTC Control Register (DTCCR) 307
18.2.8 DTC Vector Base Register (DTCVBR) 308
18.2.9 DTC Module Start Register (DTCST) 308
18.2.10 DTC Status Register (DTCSTS) 309
18.3 Activation Sources 309
18.3.1 Allocating Transfer Information and DTC Vector Table 310
18.4 Operation 311
18.4.1 Transfer Information Read Skip Function... 313
18.4.2 Transfer Information Write-Back Skip Function 313
18.4.3 Normal Transfer Mode.. 314
18.4.4 Repeat Transfer Mode.. 315
18.4.5 Block Transfer Mode 316
18.4.6 Chain Transfer.. 317
18.4.7 Operation Timing 318
18.4.8 Execution Cycles of DTC.. 320
18.4.9 DTC Bus Mastership Release Timing 320
18.5 DTC Setting Procedure.. 320
18.6 Examples of DTC Usage 321
18.6.1 Normal Transfer.. 321
18.6.2 Chain Transfer.. 322

18.6.3 Chain Transfer when Counter = 0 324
18.7 Interrupt Source 325
18.8 Event Link 325
18.9 Snooze Control Interface 325
18.10 Module-Stop Function 325
18.11 Usage Notes 326
18.11.1 Transfer Information Start Address 326

19. Event Link Controller (ELC) 327
19.1 Overview 327
19.2 Register Descriptions 328
19.2.1 Event Link Controller Register (ELCR) 328
19.2.2 Event Link Software Event Generation Register n (ELSEGRn) (n = 0, 1) 328
19.2.3 Event Link Setting Register n (ELSRn) (n = 0 to 9, 12, 14 to 18) 329
19.3 Operation 333
19.3.1 Relation between Interrupt Handling and Event Linking 333
19.3.2 Linking Events 333
19.3.3 Example of Procedure for Linking Events 333
19.4 Usage Notes 334
19.4.1 Linking DMAC or DTC Transfer End Signals as Events 334
19.4.2 Setting Clocks 334
19.4.3 Module-Stop Function Setting 334
19.4.4 ELC Delay Time 334

20. I/O Ports 335
20.1 Overview 335
20.2 Register Descriptions 336
20.2.1 Port Control Register 1 (PCNTR1/PODR/PDR) 336
20.2.2 Port Control Register 2 (PCNTR2/EIDR/PIDR) 337
20.2.3 Port Control Register 3 (PCNTR3/PORR/POSR) 338
20.2.4 Port Control Register 4 (PCNTR4/EORR/EOSR) 339
20.2.5 Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) (m = 0 to 5, 9; n = 00 to 15) 340
20.2.6 Write-Protect Register (PWPR) 342
20.3 Operation 342
20.3.1 General I/O Ports 342
20.3.2 Port Function Select 342
20.3.3 Port Group Function for the ELC 343
20.3.3.1 Behavior when ELC_PORT1, 2, 3, or 4 is input from the ELC 343
20.3.3.2 Behavior when an event pulse is output to the ELC 344
20.4 Handling of Unused Pins 345
20.5 Usage Notes 345
20.5.1 Procedure for Specifying the Pin Functions 345
20.5.2 Procedure for Using Port Group Input 345

20.5.3 Port Output Data Register (PODR) Summary 346
20.5.4 Notes on Using Analog Functions 346
20.5.5 I/O Buffer Specification 346
20.5.6 Selecting the USB_DP and USB_DM Pins 346
20.5.7 Pull-up/Pull-down Setting for P914 and P915 using USBFS/GPIO Function 347
20.6 Peripheral Select Settings for each Product 347
21. Key Interrupt Function (KINT) 355
21.1 Overview 355
21.2 Register Descriptions 357
21.2.1 Key Return Control Register (KRCTL) 357
21.2.2 Key Return Flag Register (KRF) 357
21.2.3 Key Return Mode Register (KRM) 357
21.3 Operation 358
21.3.1 Operation When Not Using Key Interrupt Flag (KRMD = 0) 358
21.3.2 Operation When Using Key Interrupt Flag (KRMD = 1) 358
21.4 Usage Notes 360
22. Port Output Enable for GPT (POEG) 361
22.1 Overview 361
22.2 Register Descriptions 362
22.2.1 POEG Group n Setting Register (POEGGn) (n = A, B) 362
22.3 Output-Disable Control Operation 363
22.3.1 Pin Input Level Detection Operation 364
22.3.1.1 Digital filter 364
22.3.2 Output-Disable Request from GPT 364
22.3.3 Output-Disable Control on Detection of Stopped Oscillation 364
22.3.4 Output-Disable Control Using Registers 364
22.3.5 Release from Output Disable 364
22.4 Interrupt Sources 365
22.5 External Trigger Output to GPT 365
22.6 Usage Notes 366
22.6.1 Transition to Software Standby Mode 366
22.6.2 Specifying Pins Associated with the GPT 366
23. General PWM Timer (GPT) 367
23.1 Overview 367
23.2 Register Descriptions 371
23.2.1 General PWM Timer Write-Protection Register (GTWP) 372
23.2.2 General PWM Timer Software Start Register (GTSTR) 372
23.2.3 General PWM Timer Software Stop Register (GTSTP) 373
23.2.4 General PWM Timer Software Clear Register (GTCLR) 373
23.2.5 General PWM Timer Start Source Select Register (GTSSR) 374
23.2.6 General PWM Timer Stop Source Select Register (GTPSR) 376

23.2.7 General PWM Timer Clear Source Select Register (GTCSR) 379
23.2.8 General PWM Timer Up Count Source Select Register (GTUPSR) 381
23.2.9 General PWM Timer Down Count Source Select Register (GTDNSR)...................... 384
23.2.10 General PWM Timer Input Capture Source Select Register A(GTICASR) 386
23.2.11 General PWM Timer Input Capture Source Select Register B(GTICBSR) 389
23.2.12 General PWM Timer Control Register (GTCR) 391
23.2.13 General PWM Timer Count Direction and Duty Setting Register (GTUDDTYC) 393
23.2.14 General PWM Timer I/O Control Register (GTIOR) 395
23.2.15 General PWM Timer Interrupt Output Setting Register (GTINTAD)........................... 398
23.2.16 General PWM Timer Status Register (GTST) 399
23.2.17 General PWM Timer Buffer Enable Register (GTBER)... 403
23.2.18 General PWM Timer Counter (GTCNT) 404
23.2.19 General PWM Timer Compare Capture Register n (GTCCRn) (n = A to F).............. 405
23.2.20 General PWM Timer Cycle Setting Register (GTPR).. 405
23.2.21 General PWM Timer Cycle Setting Buffer Register (GTPBR)................................... 406
23.2.22 General PWM Timer Dead Time Control Register (GTDTCR).................................. 406
23.2.23 General PWM Timer Dead Time Value Register U (GTDVU)................................... 407
23.2.24 Output Phase Switching Control Register (OPSCR) 407
23.3 Operation.. 409
23.3.1 Basic Operation 409
23.3.1.1 Counter operation... 409
23.3.1.2 Waveform output by compare match 415
23.3.1.3 Input capture function... 418
23.3.2 Buffer Operation 420
23.3.2.1 GTPR register buffer operation 420
23.3.2.2 Buffer operation for GTCCRA and GTCCRB 423
23.3.3 PWM Output Operating Mode 428
23.3.3.1 Saw-wave PWM mode 428
23.3.3.2 Saw-wave one-shot pulse mode 431
23.3.3.3 Triangle-wave PWM mode 1 (32-bit transfer at trough) 433
23.3.3.4 Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) 436
23.3.3.5 Triangle-wave PWM mode 3 (64-bit transfer at trough) 438
23.3.4 Automatic Dead Time Setting Function 440
23.3.5 Count Direction Changing Function.. 444
23.3.6 Function of Output Duty 0% and 100% 445
23.3.7 Hardware Count Start/Count Stop and Clear Operation 447
23.3.7.1 Hardware start operation.. 447
23.3.7.2 Hardware stop operation 448
23.3.7.3 Hardware clear operation 451
23.3.8 Synchronized Operation 454
23.3.8.1 Synchronized operation by software 454

23.3.8.2 Synchronized operation by hardware 456
23.3.9 PWM Output Operation Examples 458
23.3.10 Phase Counting Function 464
23.3.11 Output Phase Switching (GPT_OPS) 471
23.3.11.1 Input selection and synchronization of external input signal 474
23.3.11.2 Input sampling 475
23.3.11.3 Input phase decode 475
23.3.11.4 Output selection control 475
23.3.11.5 Output selection control (group output disable function) 477
23.3.11.6 Event Link Controller (ELC) output 477
23.3.11.7 GPT_OPS start operation setting flow 478
23.4 Interrupt Sources 478
23.4.1 Interrupt Sources 478
23.4.2 DMAC/DTC Activation 482
23.5 Operations Linked by ELC 482
23.5.1 Event Signal Output to ELC 482
23.5.2 Event Signal Inputs from ELC 482
23.6 Noise Filter Function 482
23.7 Protection Function 483
23.7.1 Write-Protection for Registers 483
23.7.2 Disabling of Buffer Operation 483
23.7.3 GTIOC Pin Output Negate Control 484
23.8 Initialization Method of Output Pins 485
23.8.1 Pin Settings after Reset 485
23.8.2 Pin Initialization Due to Error during Operation 486
23.9 Usage Notes 486
23.9.1 Module-Stop Function Setting 486
23.9.2 GTCCRn Settings during Compare Match Operation (n = A to F) 486
23.9.3 Setting Range for GTCNT Counter 487
23.9.4 Starting and Stopping the GTCNT Counter 487
23.9.5 Priority Order of Each Event 487
24. Asynchronous General Purpose Timer (AGT) 489
24.1 Overview 489
24.2 Register Descriptions 491
24.2.1 AGT Counter Register (AGT) 491
24.2.2 AGT Compare Match A Register (AGTCMA) 491
24.2.3 AGT Compare Match B Register (AGTCMB) 492
24.2.4 AGT Control Register (AGTCR) 492
24.2.5 AGT Mode Register 1 (AGTMR1) 494
24.2.6 AGT Mode Register 2 (AGTMR2) 495
24.2.7 AGT I/O Control Register (AGTIOC) 495

24.2.8 AGT Event Pin Select Register (AGTISR) 496
24.2.9 AGT Compare Match Function Select Register (AGTCMSR) 497
24.2.10 AGT Pin Select Register (AGTIOSEL) 497
24.3 Operation 498
24.3.1 Reload Register and Counter Rewrite Operation 498
24.3.2 Reload Register and Compare Register A/B Rewrite Operation 499
24.3.3 Timer Mode 500
24.3.4 Pulse Output Mode 501
24.3.5 Event Counter Mode 502
24.3.6 Pulse Width Measurement Mode 503
24.3.7 Pulse Period Measurement Mode 504
24.3.8 Compare Match Function 505
24.3.9 Output Settings for each Mode 507
24.3.10 Standby Mode 507
24.3.11 Interrupt Sources 508
24.3.12 Event Signal Output to ELC 508
24.4 Usage Notes 508
24.4.1 Count Operation Start and Stop Control 508
24.4.2 Access to Counter Register 509
24.4.3 When Changing Mode 509
24.4.4 Digital Filter 509
24.4.5 How to Calculate Event Number, Pulse Width, and Pulse Period 509
24.4.6 When Count is Forcibly Stopped by TSTOP Bit 510
24.4.7 When Selecting AGT0 Underflow as the Count Source 510
24.4.8 Reset of I/O Register 510
24.4.9 When Selecting PCLKB, PCLKB/8, or PCLKB/2 as the Count Source 510
24.4.10 When Selecting AGTLCLK or AGTSCLK as the Count Source 510
24.4.11 When Switching Source Clock 510

25. Realtime Clock (RTC) 511
25.1 Overview 511
25.2 Register Descriptions 513
25.2.1 64-Hz Counter (R64CNT) 513
25.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0) 513
25.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1) 514
25.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2) 515
25.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3) 516
25.2.6 Day Counter (RDAYCNT) 517
25.2.7 Month Counter (RMONCNT) 517
25.2.8 Year Counter (RYRCNT) 518
25.2.9 Second Alarm Register (RSECAR)/Binary Counter 0 Alarm Register (BCNT0AR) 518
25.2.10 Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR) 519

25.2.11 Hour Alarm Register (RHRAR)/Binary Counter 2 Alarm Register (BCNT2AR) 520
25.2.12 Day-of-Week Alarm Register (RWKAR)/Binary Counter 3 Alarm Register (BCNT3AR) 521
25.2.13 Date Alarm Register (RDAYAR)/Binary Counter 0 Alarm Enable Register (BCNT0AER) 522
25.2.14 Month Alarm Register (RMONAR)/Binary Counter 1 Alarm Enable Register (BCNT1AER) 523
25.2.15 Year Alarm Register (RYRAR)/Binary Counter 2 Alarm Enable Register (BCNT2AER) 524
25.2.16 Year Alarm Enable Register (RYRAREN)/Binary Counter 3 Alarm Enable Register (BCNT3AER) 525
25.2.17 RTC Control Register 1 (RCR1) 526
25.2.18 RTC Control Register 2 (RCR2) 527
25.2.19 RTC Control Register 4 (RCR4) 530
25.2.20 Frequency Register (RFRH/RFRL) 531
25.2.21 Time Error Adjustment Register (RADJ) 532
25.2.22 Time Capture Control Register 0 (RTCCR0) 532
25.2.23 Second Capture Register 0 (RSECCP0) /BCNT0 Capture Register 0 (BCNT0CP0) 534
25.2.24 Minute Capture Register 0 (RMINCP0)/BCNT1 Capture Register 0 (BCNT1CP0) 534
25.2.25 Hour Capture Register 0 (RHRCP0) /BCNT2 Capture Register 0 (BCNT2CP0) 535
25.2.26 Date Capture Register 0 (RDAYCP0) /BCNT3 Capture Register 0 (BCNT3CP0) 536
25.2.27 Month Capture Register 0 (RMONCP0) 537
25.3 Operation 537
25.3.1 Outline of Initial Settings of Registers after Power On 537
25.3.2 Clock and Count Mode Setting Procedure 537
25.3.3 Setting the Time 538
25.3.4 30-Second Adjustment 539
25.3.5 Reading 64-Hz Counter and Time 540
25.3.6 Alarm Function 541
25.3.7 Procedure for Disabling Alarm Interrupt 541
25.3.8 Time Error Adjustment Function 542
25.3.8.1 Automatic adjustment 542
25.3.8.2 Adjustment by software 543
25.3.8.3 Procedure for changing the mode of adjustment 543
25.3.8.4 Procedure for stopping adjustment 544
25.3.8.5 Capturing the time 544
25.4 Interrupt Sources 545
25.5 Event Link Output 546
25.5.1 Interrupt Handling and Event Linking 547
25.6 Usage Notes 547
25.6.1 Register Writing during Counting 547

25.6.2 Use of Periodic Interrupts 547
25.6.3 RTCOUT (1-Hz/64-Hz) Clock Output 548
25.6.4 Transitions to Low Power Modes after Setting Registers 548
25.6.5 Notes on Writing to and Reading from Registers 548
25.6.6 Changing the Count Mode 548
25.6.7 Initialization Procedure when the RTC is not to be Used 548
25.6.8 When Switching Source Clock 549

26. Watchdog Timer (WDT) 550
26.1 Overview 550
26.2 Register Descriptions 551
26.2.1 WDT Refresh Register (WDTRR) 551
26.2.2 WDT Control Register (WDTCR) 552
26.2.3 WDT Status Register (WDTSR) 554
26.2.4 WDT Reset Control Register (WDTRCR) 555
26.2.5 WDT Count Stop Control Register (WDTCSTPR) 556
26.2.6 Option Function Select Register 0 (OFS0) 556
26.3 Operation 556
26.3.1 Count Operation in Each Start Mode 556
26.3.1.1 Register start mode 556
26.3.1.2 Auto start mode 558
26.3.2 Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers 559
26.3.3 Refresh Operation 560
26.3.4 Reset Output 561
26.3.5 Interrupt Sources 561
26.3.6 Reading the Down-Counter Value 561
26.3.7 Associations between Option Function Select Register 0 (OFS0) and WDT Registers 562
26.4 Link Operation by ELC 562
26.5 Usage Notes 562
26.5.1 ICU Event Link Setting Register n (IELSRn) Setting 562

27. Independent Watchdog Timer (IWDT) 563
27.1 Overview 563
27.2 Register Descriptions 564
27.2.1 IWDT Refresh Register (IWDTRR) 564
27.2.2 IWDT Status Register (IWDTSR) 565
27.2.3 Option Function Select Register 0 (OFS0) 566
27.3 Operation 568
27.3.1 Auto Start Mode 568
27.3.2 Refresh Operation 569
27.3.3 Status Flags 570
27.3.4 Reset Output 571

27.3.5 Interrupt Sources ... 571
27.3.6 Reading the Down-counter Value ... 571
27.4 Link Operation by ELC ... 571
27.5 Usage Notes ... 572
27.5.1 Refresh Operations ... 572
27.5.2 Clock Division Ratio Setting ... 572
28. USB 2.0 Full-Speed Module (USBFS) ... 573
28.1 Overview ... 573
28.2 Register Descriptions ... 575
28.2.1 System Configuration Control Register (SYSCFG) ... 575
28.2.2 System Configuration Status Register 0 (SYSSTS0) ... 576
28.2.3 Device State Control Register 0 (DVSTCTR0) ... 577
28.2.4 CFIFO Port Register (CFIFO/CFIFOL)
D0FIFO Port Register (D0FIFO/D0FIFOL)
D1FIFO Port Register (D1FIFO/D1FIFOL) ... 579
28.2.5 CFIFO Port Select Register (CFIFOSEL)
D0FIFO Port Select Register (D0FIFOSEL)
D1FIFO Port Select Register (D1FIFOSEL) ... 581
28.2.6 CFIFO Port Control Register (CFIFOCTR)
D0FIFO Port Control Register (D0FIFOCTR)
D1FIFO Port Control Register (D1FIFOCTR) ... 584
28.2.7 Interrupt Enable Register 0 (INTENB0) ... 585
28.2.8 Interrupt Enable Register 1 (INTENB1) ... 586
28.2.9 BRDY Interrupt Enable Register (BRDYENB) ... 587
28.2.10 NRDY Interrupt Enable Register (NRDYENB) ... 588
28.2.11 BEMP Interrupt Enable Register (BEMPENB) ... 588
28.2.12 SOF Output Configuration Register (SOFCFG) ... 589
28.2.13 Interrupt Status Register 0 (INTSTS0) ... 590
28.2.14 Interrupt Status Register 1 (INTSTS1) ... 592
28.2.15 BRDY Interrupt Status Register (BRDYSTS) ... 595
28.2.16 NRDY Interrupt Status Register (NRDYSTS) ... 595
28.2.17 BEMP Interrupt Status Register (BEMPSTS) ... 596
28.2.18 Frame Number Register (FRMNUM) ... 597
28.2.19 USB Request Type Register (USBREQ) ... 598
28.2.20 USB Request Value Register (USBVAL) ... 598
28.2.21 USB Request Index Register (USBINDX) ... 599
28.2.22 USB Request Length Register (USBLENG) ... 600
28.2.23 DCP Configuration Register (DCPCFG) ... 600
28.2.24 DCP Maximum Packet Size Register (DCPMAXP) ... 601
28.2.25 DCP Control Register (DCPCTR) ... 602
28.2.26 Pipe Window Select Register (PIPESEL) ... 605
28.2.27 Pipe Configuration Register (PIPECFG) ... 606
28.2.28 Pipe Maximum Packet Size Register (PIPEMAXP) ... 608

28.2.29 Pipe Cycle Control Register (PIPEPERI) 609
28.2.30 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9) 610
28.2.31 PIPEn Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5) 616
28.2.32 PIPEn Transaction Counter Register (PIPEnTRN) (n = 1 to 5) 617
28.2.33 Device Address n Configuration Register (DEVADDn) (n = 0 to 5) 618
28.2.34 USB Module Control Register (USBMC) 618
28.2.35 BC Control Register 0 (USBBCCTRL0) 619
28.3 Operation 620
28.3.1 System Control 620
28.3.1.1 Setting data to the USBFS-related registers 620
28.3.1.2 Selecting the controller function 620
28.3.1.3 Controlling the USBFS data bus using resistors 620
28.3.1.4 Example of USB external connection circuits 621
28.3.2 Interrupt Sources 625
28.3.3 Interrupt Descriptions 629
28.3.3.1 BRDY interrupt 629
28.3.3.2 NRDY interrupt 632
28.3.3.3 BEMP interrupt 634
28.3.3.4 Device state transition interrupt (device controller mode) 635
28.3.3.5 Control transfer stage transition interrupt (device controller mode) 636
28.3.3.6 Frame update interrupt 637
28.3.3.7 VBUS interrupt 637
28.3.3.8 Resume interrupt 638
28.3.3.9 OVRCR interrupt 638
28.3.3.10 BCHG interrupt 638
28.3.3.11 DTCH interrupt 638
28.3.3.12 SACK interrupt 638
28.3.3.13 SIGN interrupt 638
28.3.3.14 ATTCH interrupt 638
28.3.3.15 EOFERR interrupt 638
28.3.3.16 Portable device detection interrupt 639
28.3.4 Pipe Control 639
28.3.4.1 Pipe control register switching procedures 639
28.3.4.2 Transfer types 640
28.3.4.3 Endpoint number 640
28.3.4.4 Maximum packet size setting 640
28.3.4.5 Transaction counter for pipes 1 to 5 in the receiving direction 640
28.3.4.6 Response PID 641
28.3.4.7 Data PID sequence bit 642
28.3.4.8 Response PID = NAK function 642
28.3.4.9 Auto response mode 642

28.3.4.10 OUT-NAK mode 642
28.3.4.11 Null auto response mode 643
28.3.5 FIFO Buffer Memory 643
28.3.6 FIFO Buffer Clearing 643
28.3.7 FIFO Port Functions 644
28.3.8 DMA Transfers (D0FIFO and D1FIFO Ports) 645
28.3.9 Control Transfers Using DCP 645
28.3.9.1 Control transfers in host controller mode 646
28.3.9.2 Control transfers in device controller mode 646
28.3.10 Bulk Transfers (Pipes 1 to 5) 647
28.3.11 Interrupt Transfers (Pipes 6 to 9) 647
28.3.11.1 Interval counter for interrupt transfers in host controller mode 648
28.3.12 Isochronous Transfers (Pipes 1 and 2) 648
28.3.12.1 Error detection in isochronous transfers 648
28.3.12.2 DATA-PID 649
28.3.12.3 Interval counter 649
28.3.13 SOF Interpolation Function 655
28.3.14 Pipe Schedule 655
28.3.14.1 Conditions for generating transactions 655
28.3.14.2 Transfer schedule 656
28.3.14.3 Enabling USB communication 656
28.3.15 Battery Charging Detection Processing 656
28.3.15.1 Processing in device controller mode 656
28.3.15.2 Processing when host controller is selected 658
28.4 Usage Notes 661
28.4.1 Settings for the Module-Stop State 661
28.4.2 Clearing the Interrupt Status Register on Exiting Software Standby Mode 661
28.4.3 Clearing the Interrupt Status Register after Setting Up the Port Function 661

29. Serial Communications Interface (SCI) 663
29.1 Overview 663
29.2 Register Descriptions 666
29.2.1 Receive Shift Register (RSR) 666
29.2.2 Receive Data Register (RDR) 666
29.2.3 Receive 9-bit Data Register (RDRHL) 667
29.2.4 Receive FIFO Data Register H, L, HL (FRDRH, FRDRL, FRDRHL) 667
29.2.5 Transmit Data Register (TDR) 668
29.2.6 Transmit 9-Bit Data Register (TDRHL) 669
29.2.7 Transmit FIFO Data Register H, L, HL (FTDRH, FTDRL, FTDRHL) 669
29.2.8 Transmit Shift Register (TSR) 670
29.2.9 Serial Mode Register (SMR) for Non-Smart Card Interface Mode (SCMR.SMIF = 0) 670

29.2.10 Serial Mode Register for Smart Card Interface Mode (SMR_SMCI) (SCMR.SMIF = 1) 672

29.2.11 Serial Control Register (SCR) for Non-Smart Card Interface Mode (SCMR.SMIF = 0) 673

29.2.12 Serial Control Register for Smart Card Interface Mode (SCR_SMCI) (SCMR.SMIF = 1) 675

29.2.13 Serial Status Register (SSR) for Non-Smart Card Interface and Non-FIFO Mode (SCMR.SMIF = 0 and FCR.FM = 0) 677

29.2.14 Serial Status Register for Non-Smart Card Interface and FIFO Mode (SSR_FIFO) (SCMR.SMIF = 0 and FCR.FM = 1) 679

29.2.15 Serial Status Register for Smart Card Interface Mode (SSR_SMCI) (SCMR.SMIF = 1) 682

29.2.16 Smart Card Mode Register (SCMR) 684

29.2.17 Bit Rate Register (BRR) 685

29.2.18 Modulation Duty Register (MDDR) 692

29.2.19 Serial Extended Mode Register (SEMR) 694

29.2.20 Noise Filter Setting Register (SNFR) 696

29.2.21 I^{2}C Mode Register 1 (SIMR1) 696

29.2.22 I^{2}C Mode Register 2 (SIMR2) 697

29.2.23 I^{2}C Mode Register 3 (SIMR3) 698

29.2.24 I^{2}C Status Register (SISR) 699

29.2.25 SPI Mode Register (SPMR) 700

29.2.26 FIFO Control Register (FCR) 701

29.2.27 FIFO Data Count Register (FDR) 703

29.2.28 Line Status Register (LSR) 703

29.2.29 Compare Match Data Register (CDR) 704

29.2.30 Data Compare Match Control Register (DCCR) 704

29.2.31 Serial Port Register (SPTR) 706

29.3 Operation in Asynchronous Mode 706

29.3.1 Serial Data Transfer Format 707

29.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode 709

29.3.3 Clock 709

29.3.4 Double-Speed Operation and Frequency of 6 Times the Bit Rate 710

29.3.5 CTS and RTS Functions 710

29.3.6 Address Match (Receive Data Match Detection) Function 711

29.3.7 SCI Initialization in Asynchronous Mode 714

29.3.8 Serial Data Transmission in Asynchronous Mode 715

29.3.9 Serial Data Reception in Asynchronous Mode 720

29.4 Multi-Processor Communications Function 728

29.4.1 Multi-Processor Serial Data Transmission 729

29.4.2 Multi-Processor Serial Data Reception 732

29.5 Operation in Clock Synchronous Mode 737

29.5.1 Clock 738

29.5.2 CTS and RTS Functions 738
29.5.3 SCI Initialization in Clock Synchronous Mode 739
29.5.4 Serial Data Transmission in Clock Synchronous Mode 741
29.5.5 Serial Data Reception in Clock Synchronous Mode 745
29.5.6 Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode 751
29.6 Operation in Smart Card Interface Mode 753
29.6.1 Example Connection 753
29.6.2 Data Format (Except in Block Transfer Mode) 754
29.6.3 Block Transfer Mode 756
29.6.4 Receive Data Sampling Timing and Reception Margin 756
29.6.5 SCI Initialization 757
29.6.6 Serial Data Transmission (Except in Block Transfer Mode) 759
29.6.7 Serial Data Reception (Except in Block Transfer Mode) 761
29.6.8 Clock Output Control 763
29.7 Operation in Simple IIC Mode 764
29.7.1 Generation of Start, Restart, and Stop Conditions 765
29.7.2 Clock Synchronization 766
29.7.3 SDA Output Delay 767
29.7.4 SCI Initialization in Simple IIC Mode 768
29.7.5 Operation in Master Transmission (Simple IIC Mode) 768
29.7.6 Master Reception in Simple IIC Mode 771
29.8 Operation in Simple SPI Mode 773
29.8.1 States of Pins in Master and Slave Modes 773
29.8.2 SS Function in Master Mode 774
29.8.3 SS Function in Slave Mode 774
29.8.4 Relationship between Clock and Transmit/Receive Data 774
29.8.5 SCI Initialization in Simple SPI Mode 775
29.8.6 Transmission and Reception of Serial Data in Simple SPI Mode 775
29.9 Bit Rate Modulation Function 775
29.10 Interrupt Sources 776
29.10.1 Buffer Operations for SCIn_TXI and SCIn_RXI Interrupts (non-FIFO selected) 776
29.10.2 Buffer Operations for SCIn_TXI and SCIn_RXI Interrupts (FIFO selected) 776
29.10.3 Interrupts in Asynchronous, Clock Synchronous, and Simple SPI Modes 776
29.10.4 Interrupts in Smart Card Interface Mode 778
29.10.5 Interrupts in Simple IIC Mode 779
29.11 Event Linking 779
29.12 Address mismatch event output (SCI0_DCUF) 780
29.13 Noise Cancellation Function 780
29.14 Usage Notes 781
29.14.1 Settings for the Module-Stop State 781
29.14.2 SCI Operations during Low Power State 781

29.14.3 Break Detection and Processing ... 786

29.14.4 Mark State and Production of Breaks ... 787

29.14.5 Receive Error Flags and Transmit Operations in Clock Synchronous and Simple SPI Modes ... 787

29.14.6 Restrictions on Clock Synchronous Transmission in Clock Synchronous and Simple SPI Modes ... 787

29.14.7 Restrictions on Using DMAC or DTC ... 788

29.14.8 Notes on Starting Transfer ... 789

29.14.9 External Clock Input in Clock Synchronous and Simple SPI Modes ... 789

29.14.10 Limitations on Simple SPI Mode ... 789

30. I^{2}C Bus Interface (IIC) ... 791

30.1 Overview ... 791

30.2 Register Descriptions ... 794

30.2.1 I^{2}C Bus Control Register 1 (ICCR1) ... 794

30.2.2 I^{2}C Bus Control Register 2 (ICCR2) ... 796

30.2.3 I^{2}C Bus Mode Register 1 (ICMR1) ... 799

30.2.4 I^{2}C Bus Mode Register 2 (ICMR2) ... 800

30.2.5 I^{2}C Bus Mode Register 3 (ICMR3) ... 801

30.2.6 I^{2}C Bus Function Enable Register (ICFER) ... 803

30.2.7 I^{2}C Bus Status Enable Register (ICSER) ... 804

30.2.8 I^{2}C Bus Interrupt Enable Register (ICIER) ... 805

30.2.9 I^{2}C Bus Status Register 1 (ICSR1) ... 806

30.2.10 I^{2}C Bus Status Register 2 (ICSR2) ... 808

30.2.11 I^{2}C Bus Wakeup Unit Register (ICWUR) ... 811

30.2.12 I^{2}C Bus Wakeup Unit Register 2 (ICWUR2) ... 812

30.2.13 Slave Address Register Ly (SARLy) (y = 0 to 2) ... 813

30.2.14 Slave Address Register Uy (SARUy) (y = 0 to 2) ... 814

30.2.15 I^{2}C Bus Bit Rate Low-Level Register (ICBRL) ... 814

30.2.16 I^{2}C Bus Bit Rate High-Level Register (ICBRH) ... 815

30.2.17 I^{2}C Bus Transmit Data Register (ICDRT) ... 816

30.2.18 I^{2}C Bus Receive Data Register (ICDRR) ... 816

30.2.19 I^{2}C Bus Shift Register (ICDRS) ... 817

30.3 Operation ... 817

30.3.1 Communication Data Format ... 817

30.3.2 Initial Settings ... 818

30.3.3 Master Transmit Operation ... 819

30.3.4 Master Receive Operation ... 823

30.3.5 Slave Transmit Operation ... 828

30.3.6 Slave Receive Operation ... 831

30.4 SCL Synchronization Circuit ... 833

30.5 SDA Output Delay Function ... 834

30.6 Digital Noise Filter Circuits ... 834

30.7 Address Match Detection 835
  30.7.1 Slave-Address Match Detection 835
  30.7.2 Detection of General Call Address 837
  30.7.3 Device ID Address Detection 838
  30.7.4 Host Address Detection 839
30.8 Wakeup Function 840
  30.8.1 Normal Wakeup Mode 1 841
  30.8.2 Normal Wakeup Mode 2 844
  30.8.3 Command Recovery Mode/ EEP Response Mode (Special Wakeup Mode) 846
  30.8.4 Precautions for WFI Instruction Execution 849
30.9 Automatic Low-Hold Function for SCL 849
  30.9.1 Function to Prevent Wrong Transmission of Transmit Data 849
  30.9.2 NACK Reception Transfer Suspension Function 850
  30.9.3 Function to Prevent Failure to Receive Data 851
30.10 Arbitration-Lost Detection Functions 852
  30.10.1 Master Arbitration-Lost Detection (MALE Bit) 852
  30.10.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit) 854
  30.10.3 Slave Arbitration-Lost Detection (SALE Bit) 855
30.11 Start, Restart, and Stop Condition Issuing Function 856
  30.11.1 Issuing a Start Condition 856
  30.11.2 Issuing a Restart Condition 856
  30.11.3 Issuing a Stop Condition 858
30.12 Bus Hanging 859
  30.12.1 Timeout Function 859
  30.12.2 Extra SCL Clock Cycle Output Function 860
  30.12.3 IIC Reset and Internal Reset 861
30.13 SMBus Operation 861
  30.13.1 SMBus Timeout Measurement 861
  30.13.2 Packet Error Code (PEC) 862
  30.13.3 SMBus Host Notification Protocol (Notify ARP Master Command) 863
30.14 Interrupt Sources 863
  30.14.1 Buffer Operation for IICn_TXI and IICn_RXI Interrupts 864
30.15 State of Registers when Issuing each Condition 864
30.16 Event Link Output 865
  30.16.1 Interrupt Handling and Event Linking 865
30.17 Usage Notes 865
  30.17.1 Settings for the Module-Stop State 865
  30.17.2 Notes on Starting Transfer 865
31. Controller Area Network (CAN) Module 866
30.1 Overview 866
31.2 Register Descriptions 868

31.2.1 Control Register (CTLR) ... 868
31.2.2 Bit Configuration Register (BCR) ... 871
31.2.3 Mask Register k (MKRk) (k = 0 to 7) ... 873
31.2.4 FIFO Received ID Compare Registers 0 and 1 (FIDCR0 and FIDCR1) ... 873
31.2.5 Mask Invalid Register (MKIVLR) ... 875
31.2.6 Mailbox Register j (MBj_ID, MBj_DL, MBj_Dm, MBj_TS) (j = 0 to 31, m = 0 to 7) ... 875
31.2.7 Mailbox Interrupt Enable Register (MIER) ... 879
31.2.8 Mailbox Interrupt Enable Register for FIFO Mailbox Mode (MIER_FIFO) ... 880
31.2.9 Message Control Registers for Transmit (MCTL_TXj) (j = 0 to 31) ... 881
31.2.10 Message Control Register for Receive (MCTL_RXj) (j = 0 to 31) ... 883
31.2.11 Receive FIFO Control Register (RFCR) ... 884
31.2.12 Receive FIFO Pointer Control Register (RFPCR) ... 886
31.2.13 Transmit FIFO Control Register (TFCR) ... 887
31.2.14 Transmit FIFO Pointer Control Register (TFPCR) ... 888
31.2.15 Status Register (STR) ... 889
31.2.16 Mailbox Search Mode Register (MSMR) ... 891
31.2.17 Mailbox Search Status Register (MSSR) ... 891
31.2.18 Channel Search Support Register (CSSR) ... 892
31.2.19 Acceptance Filter Support Register (AFSR) ... 893
31.2.20 Error Interrupt Enable Register (EIER) ... 894
31.2.21 Error Interrupt Factor Judge Register (EIFR) ... 895
31.2.22 Receive Error Count Register (RECR) ... 897
31.2.23 Transmit Error Count Register (TECR) ... 897
31.2.24 Error Code Store Register (ECSR) ... 898
31.2.25 Time Stamp Register (TSR) ... 899
31.2.26 Test Control Register (TCR) ... 899
31.3 Modes of Operation ... 901
31.3.1 CAN Reset Mode ... 901
31.3.2 CAN Halt Mode ... 902
31.3.3 CAN Sleep Mode ... 903
31.3.4 CAN Operation Mode (Excluding Bus-Off State) ... 903
31.3.5 CAN Operation Mode (Bus-Off State) ... 904
31.4 Data Transfer Rate Configuration ... 905
31.4.1 Clock Setting ... 905
31.4.2 Bit Time Setting ... 905
31.4.3 Data Transfer Rate ... 905
31.5 Mailbox and Mask Register Structure ... 906
31.6 Acceptance Filtering and Masking Functions ... 907
31.7 Reception and Transmission ... 909
31.7.1 Reception ... 910
31.7.2 Transmission ... 911

31.8 Interrupt 912

31.9 Usage Notes 913

31.9.1 Settings for the Module-Stop State 913

31.9.2 Settings for the Operating Clock 913

32. Serial Peripheral Interface (SPI) 914

32.1 Overview 914

32.2 Register Descriptions 917

32.2.1 SPI Control Register (SPCR) 917

32.2.2 SPI Slave Select Polarity Register (SSLP) 919

32.2.3 SPI Pin Control Register (SPPCR) 919

32.2.4 SPI Status Register (SPSR) 920

32.2.5 SPI Data Register (SPDR/SPDR_HA) 923

32.2.6 SPI Sequence Control Register (SPSCR) 927

32.2.7 SPI Sequence Status Register (SPSSR) 927

32.2.8 SPI Bit Rate Register (SPBR) 928

32.2.9 SPI Data Control Register (SPDCR) 929

32.2.10 SPI Clock Delay Register (SPCKD) 931

32.2.11 SPI Slave Select Negation Delay Register (SSLND) 931

32.2.12 SPI Next-Access Delay Register (SPND) 932

32.2.13 SPI Control Register 2 (SPCR2) 932

32.2.14 SPI Command Registers (SPCMDm) (m =0 to 7 for SPI0; m = 0 for SPI1) 933

32.3 Operation 936

32.3.1 Overview of SPI Operations 936

32.3.2 Controlling the SPI Pins 937

32.3.3 SPI System Configuration Examples 938

32.3.3.1 Single master and single slave with the MCU configured as a master 938

32.3.3.2 Single master and single slave with the MCU configured as a slave 938

32.3.3.3 Single-master and multi-slave with the MCU configured as a master 939

32.3.3.4 Single master and multi-slave with the MCU configured as a slave 940

32.3.3.5 Multi-master and multi-slave with the MCU configured as a master 941

32.3.3.6 Master and slave in clock synchronous mode with the MCU configured as a master 942

32.3.3.7 Master and slave in clock synchronous mode with the MCU configured as a slave 943

32.3.4 Data Format 943

32.3.4.1 Operation when parity is disabled (SPCR2.SPPE = 0) 944

32.3.4.2 When parity is enabled (SPCR2.SPPE = 1) 948

32.3.5 Transfer Format 952

32.3.5.1 CPHA = 0 952

32.3.5.2 CPHA = 1 953

32.3.6 Data Transfer Modes 954

32.3.6.1 Full-duplex synchronous serial communications (SPCR.TXMD = 0) 954

32.3.6.2 Transmit-only operations (SPCR.TXMD = 1) 955
32.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts 955
32.3.8 Error Detection 957
32.3.8.1 Overrun errors 958
32.3.8.2 Parity errors 960
32.3.8.3 Mode fault errors 961
32.3.8.4 Underrun errors 961
32.3.9 Initializing the SPI 962
32.3.9.1 Initialization by clearing the SPE bit 962
32.3.9.2 Initialization by system reset 962
32.3.10 SPI Operation 962
32.3.10.1 Master mode operation 962
32.3.10.2 Slave mode operation 973
32.3.11 Clock Synchronous Operation 977
32.3.11.1 Master mode operation 977
32.3.11.2 Slave mode operation 984
32.3.12 Loopback mode 985
32.3.13 Self-Diagnosis of Parity Bit Function 986
32.3.14 Interrupt Sources 987
32.4 Event Link Operation 988
32.4.1 Receive Buffer Full Event Output 989
32.4.2 Transmit Buffer Empty Event Output 989
32.4.3 Mode Fault, Underrun, Overrun, or Parity Error Event Output 989
32.4.4 SPI Idle Event Output 989
32.4.5 Transmission-Completed Event Output 989
32.5 Usage Notes 990
32.5.1 Settings for the Module-Stop State 990
32.5.2 Constraint on Low Power Function 990
32.5.3 Constraint on Starting Transfer 990
32.5.4 Constraint on Mode Fault, Underrun, Overrun, or Parity Error Event Output 990
32.5.5 Constraint on the SPRF and SPTEF Flags 990
33. Cyclic Redundancy Check (CRC) Calculator 991
33.1 Overview 991
33.2 Register Descriptions 992
33.2.1 CRC Control Register 0 (CRCCR0) 992
33.2.2 CRC Control Register 1 (CRCCR1) 992
33.2.3 CRC Data Input Register (CRCDIR/CRCDIR_BY) 993
33.2.4 CRC Data Output Register (CRCDOR/CRCDOR_HA/CRCDOR_BY) 993
33.2.5 Snoop Address Register (CRCSAR) 994
33.3 Operation 994
33.3.1 Basic Operation 994

33.3.2 CRC Snoop 997

33.4 Usage Notes 998

33.4.1 Settings for the Module-Stop State 998

33.4.2 Notes on Transmission 998

34. 14-Bit A/D Converter (ADC14) 999

34.1 Overview 999

34.2 Register Descriptions 1002

34.2.1 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) 1002

34.2.2 A/D Self-Diagnosis Data Register (ADRD) 1006

34.2.3 A/D Control Register (ADCSR) 1007

34.2.4 A/D Channel Select Register A0 (ADANSA0) 1011

34.2.5 A/D Channel Select Register A1 (ADANSA1) 1012

34.2.6 A/D Channel Select Register B0 (ADANSB0) 1012

34.2.7 A/D Channel Select Register B1 (ADANSB1) 1013

34.2.8 A/D-Converted Value Addition/Average Channel Select Register 0 (ADADS0) 1014

34.2.9 A/D-Converted Value Addition/Average Channel Select Register 1 (ADADS1) 1014

34.2.10 A/D-Converted Value Addition/Average Count Select Register (ADADC) 1016

34.2.11 A/D Control Extended Register (ADCER) 1017

34.2.12 A/D Conversion Start Trigger Select Register (ADSTRGR) 1018

34.2.13 A/D Conversion Extended Input Control Register (ADEXICR) 1019

34.2.14 A/D Sampling State Register n (ADSSTRn) (n = 00, 04 to 06, 09, 10, L, T, O) 1021

34.2.15 A/D Disconnection Detection Control Register (ADDISCR) 1022

34.2.16 A/D Group Scan Priority Control Register (ADGSPCR) 1022

34.2.17 A/D Compare Function Control Register (ADCMPCR) 1024

34.2.18 A/D Compare Function Window A Channel Select Register 0 (ADCMPANSR0) 1025

34.2.19 A/D Compare Function Window A Channel Select Register 1 (ADCMPANSR1) 1026

34.2.20 A/D Compare Function Window A Extended Input Select Register (ADCMPANSER) 1026

34.2.21 A/D Compare Function Window A Comparison Condition Setting Register 0 (ADCMPLR0) 1027

34.2.22 A/D Compare Function Window A Comparison Condition Setting Register 1 (ADCMPLR1) 1028

34.2.23 A/D Compare Function Window A Extended Input Comparison Condition Setting Register (ADCMPLER) 1029

34.2.24 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) 1030

34.2.25 A/D Compare Function Window A Channel Status Register 0 (ADCMPSR0) 1032

34.2.26 A/D Compare Function Window A Channel Status Register 1 (ADCMPSR1) 1032

34.2.27 A/D Compare Function Window A Extended Input Channel Status Register (ADCMPSER) 1033
34.2.28 A/D Compare Function Window B Channel Select Register (ADCMPBNSR) 1034
34.2.29 A/D Compare Function Window B Status Register (ADCMPBSR) 1036
34.2.30 A/D Compare Function Window A/B Status Monitor Register (ADWINMON) 1036
34.2.31 A/D High-Potential/Low-Potential Reference Voltage Control Register (ADHVREFCNT) 1037
34.3 Operation 1038
34.3.1 Scanning Operation 1038
34.3.2 Single scan Mode 1039
34.3.2.1 Basic operation 1039
34.3.2.2 Channel selection and self-diagnosis 1040
34.3.2.3 A/D conversion of temperature sensor output or internal reference voltage 1041
34.3.2.4 A/D conversion in double trigger mode 1042
34.3.2.5 Extended operations when double trigger mode is selected 1043
34.3.3 Continuous Scan Mode 1044
34.3.3.1 Basic operation 1044
34.3.3.2 Channel selection and self-diagnosis 1045
34.3.4 Group Scan Mode 1046
34.3.4.1 Basic operation 1046
34.3.4.2 A/D conversion in double trigger mode 1047
34.3.4.3 Operation with group A priority control 1048
34.3.5 Compare Function for Window A and Window B 1055
34.3.5.1 Compare function 1055
34.3.5.2 Event output of compare function 1056
34.3.5.3 Restrictions on the compare function 1058
34.3.6 Analog Input Sampling and Scan Conversion Time 1059
34.3.7 Usage Example of A/D Data Register Automatic Clearing Function 1061
34.3.8 A/D-Converted Value Addition/Average Mode 1062
34.3.9 Disconnection Detection Assist Function 1062
34.3.10 Starting A/D Conversion with an Asynchronous Trigger 1064
34.3.11 Starting A/D Conversion with a Synchronous Trigger from Peripheral Module 1065
34.4 Interrupt Sources and DTC or DMAC Transfer Requests 1065
34.4.1 Interrupt Requests 1065
34.5 Event Link Function 1066
34.5.1 Event Output to the ELC 1066
34.5.2 ADC14 Operation through an Event from the ELC 1066
34.6 Selecting Reference Voltage 1066
34.7 A/D Conversion Procedure when Selecting Internal Reference Voltage as High-Potential Reference Voltage 1066
34.8 Usage Notes 1067
34.8.1 Notes on Reading Data Registers 1067

34.8.2 Notes on Stopping A/D Conversion 1067
34.8.3 A/D Conversion Restarting Timing and Termination Timing 1068
34.8.4 Restrictions on Scan End Interrupt Handling 1068
34.8.5 Settings for the Module-Stop State 1068
34.8.6 Restrictions on Entering Low Power States 1069
34.8.7 Error in Absolute Accuracy when Disconnection Detection Assistance is in Use 1069
34.8.8 ADHSC Bit Rewriting Procedure 1069
34.8.9 Notes on Operating Modes and Status Bits 1069
34.8.10 Notes on Board Design 1069
34.8.11 Notes on Noise Reduction 1069
34.8.12 Port Settings when Using the 14-bit A/D Converter Input 1070
34.8.13 Relationship between the ADC14, OPAMP, and ACMPLP 1070
34.8.14 Notes on Canceling Software Standby Mode 1070

35. 12-Bit D/A Converter (DAC12) 1071
35.1 Overview 1071
35.2 Register Descriptions 1072
35.2.1 D/A Data Register 0 (DADR0) 1072
35.2.2 D/A Control Register (DACR) 1072
35.2.3 DADR0 Format Select Register (DADPR) 1073
35.2.4 D/A A/D Synchronous Start Control Register (DAADSCR) 1073
35.2.5 D/A VREF Control Register (DAVREFCR) 1074
35.3 Operation 1074
35.3.1 Reducing Interference between D/A and A/D Conversion 1075
35.3.2 Notes on Using the Internal Reference Voltage as the Reference Voltage 1077
35.4 Event Link Operation Setting Procedure 1077
35.5 Usage Notes on Event Link Operation 1077
35.6 Usage Notes 1078
35.6.1 Settings for the Module-Stop Function 1078
35.6.2 DAC12 Operation in Module-Stop State 1078
35.6.3 DAC12 Operation in Software Standby Mode 1078
35.6.4 Restriction on Usage when Interference Reduction between D/A and A/D Conversion is Enabled 1078

36. Temperature Sensor (TSN) 1079
36.1 Overview 1079
36.2 Register Descriptions 1079
36.2.1 Temperature Sensor Calibration Data Register H (TSCDRH) 1079
36.2.2 Temperature Sensor Calibration Data Register L (TSCDRL) 1080
36.3 Using the Temperature Sensor 1080
36.3.1 Preparation for Using Temperature Sensor 1080
36.3.2 Procedure for Using the Temperature Sensor 1081

37. Operational Amplifier (OPAMP) 1082
  37.1 Overview 1082
  37.2 Register Descriptions 1083
    37.2.1 Operational Amplifier Mode Control Register (AMPMC) 1083
    37.2.2 Operational Amplifier Trigger Mode Control Register (AMPTRM) 1084
    37.2.3 Operational Amplifier Activation Trigger Select Register (AMPTRS) 1085
    37.2.4 Operational Amplifier Control Register (AMPC) 1085
    37.2.5 Operational Amplifier Monitor Register (AMPMON) 1086
  37.3 Operation 1086
    37.3.1 State Transitions 1086
    37.3.2 Operational Amplifier Control Operation 1088
  37.4 Software Trigger Mode 1092
  37.5 Activation Trigger Mode 1093
  37.6 Activation and A/D Trigger Mode 1094
  37.7 Usage Notes 1094
38. Low Power Analog Comparator (ACMPLP) 1095
  38.1 Overview 1095
  38.2 Register Descriptions 1098
    38.2.1 ACMPLP Mode Setting Register (COMPMDR) 1098
    38.2.2 ACMPLP Filter Control Register (COMPFIR) 1099
    38.2.3 ACMPLP Output Control Register (COMPOCR) 1099
    38.2.4 Comparator Input Select Register (COMPSEL0) 1100
    38.2.5 Comparator Reference Voltage Select Register (COMPSEL1) 1100
  38.3 Operation 1101
  38.4 Noise Filter 1103
  38.5 ACMPLP Interrupts 1104
  38.6 ELC Event Output 1105
  38.7 Interrupt Handling and ELC Linking 1105
  38.8 Comparator Pin Output 1105
  38.9 Usage Notes 1105
    38.9.1 Settings for the Module-Stop State 1105
    38.9.2 Relationship with A/D converter 1105
39. 8-Bit D/A Converter (DAC8) 1106
  39.1 Overview 1106
  39.2 Register Descriptions 1106
    39.2.1 D/A Conversion Value Setting Register n (DACSn) (n = 0, 1) 1106
    39.2.2 D/A Converter Mode Register (DAM) 1107
  39.3 Operation 1107
  39.4 Usage Notes 1107
    39.4.1 Module-Stop State 1107
    39.4.2 Operation of the 8-bit D/A Converter in Module-Stop State 1107
    39.4.3 8-bit D/A Converter in Software Standby Mode Operation 1108

39.4.4 When Not Using the D/A Converter ... 1108

40. Capacitive Touch Sensing Unit (CTSU) ... 1109
40.1 Overview ... 1109
40.2 Register Descriptions ... 1111
40.2.1 CTSU Control Register 0 (CTSUCR0) ... 1111
40.2.2 CTSU Control Register 1 (CTSUCR1) ... 1112
40.2.3 CTSU Synchronous Noise Reduction Setting Register (CTSUSDPRS) ... 1114
40.2.4 CTSU Sensor Stabilization Wait Control Register (CTSUSST) ... 1114
40.2.5 CTSU Measurement Channel Register 0 (CTSUMCH0) ... 1115
40.2.6 CTSU Measurement Channel Register 1 (CTSUMCH1) ... 1116
40.2.7 CTSU Channel Enable Control Register 0 (CTSUCHAC0) ... 1116
40.2.8 CTSU Channel Enable Control Register 1 (CTSUCHAC1) ... 1117
40.2.9 CTSU Channel Enable Control Register 2 (CTSUCHAC2) ... 1117
40.2.10 CTSU Channel Enable Control Register 3 (CTSUCHAC3) ... 1118
40.2.11 CTSU Channel Enable Control Register 4 (CTSUCHAC4) ... 1118
40.2.12 CTSU Channel Transmit/Receive Control Register 0 (CTSUCHTRC0) ... 1119
40.2.13 CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1) ... 1119
40.2.14 CTSU Channel Transmit/Receive Control Register 2 (CTSUCHTRC2) ... 1120
40.2.15 CTSU Channel Transmit/Receive Control Register 3 (CTSUCHTRC3) ... 1120
40.2.16 CTSU Channel Transmit/Receive Control Register 4 (CTSUCHTRC4) ... 1121
40.2.17 CTSU High-Pass Noise Reduction Control Register (CTSUDCLKC) ... 1121
40.2.18 CTSU Status Register (CTSUST) ... 1122
40.2.19 CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register (CTSUSSC) ... 1123
40.2.20 CTSU Sensor Offset Register 0 (CTSUSO0) ... 1124
40.2.21 CTSU Sensor Offset Register 1 (CTSUSO1) ... 1125
40.2.22 CTSU Sensor Counter (CTSUSC) ... 1126
40.2.23 CTSU Reference Counter (CTSURC) ... 1126
40.2.24 CTSU Error Status Register (CTSUERRS) ... 1127
40.3 Operation ... 1127
40.3.1 Principles of Measurement Operation ... 1127
40.3.2 Measurement Modes ... 1129
40.3.2.1 Initial setting flow ... 1131
40.3.2.2 Status counter ... 1132
40.3.2.3 Self-capacitance single scan mode operation ... 1133
40.3.2.4 Self-capacitance multi-scan mode operation ... 1135
40.3.2.5 Mutual capacitance full scan mode operation ... 1137
40.3.3 Parameters Common to Multiple Modes ... 1139
40.3.3.1 Sensor stabilization wait time and measurement time ... 1139
40.3.3.2 Interrupts ... 1140
40.4 Usage Notes ... 1141

40.4.1 Measurement Result Data (CTSUSC and CTSURC Counters) 1141
40.4.2 Constraints on Software Trigger 1141
40.4.3 Constraints on External Trigger 1142
40.4.4 Constraints on Forced Stops 1142
40.4.5 TSCAP Pin 1142
40.4.6 Constraints on Measurement Operation (CTSUCR0.CTSUSTRT Bit = 1) 1142

41. Data Operation Circuit (DOC) 1143
41.1 Overview 1143
41.2 Register Descriptions 1144
41.2.1 DOC Control Register (DOCR) 1144
41.2.2 DOC Data Input Register (DODIR) 1145
41.2.3 DOC Data Setting Register (DODSR) 1145
41.3 Operation 1145
41.3.1 Data Comparison Mode 1145
41.3.2 Data Addition Mode 1146
41.3.3 Data Subtraction Mode 1146
41.4 Interrupt Request and Output to the Event Link Controller (ELC) 1147
41.5 Usage Notes 1147
41.5.1 Settings for the Module-Stop State 1147

42. SRAM 1148
42.1 Overview 1148
42.2 Register Descriptions 1148
42.2.1 SRAM Parity Error Operation After Detection Register (PARIOAD) 1148
42.2.2 SRAM Protection Register (SRAMPRCR) 1149
42.2.3 ECC Operating Mode Control Register (ECCMODE) 1149
42.2.4 ECC 2-Bit Error Status Register (ECC2STS) 1150
42.2.5 ECC 1-Bit Error Information Update Enable Register (ECC1STSEN) 1150
42.2.6 ECC 1-Bit Error Status Register (ECC1STS) 1151
42.2.7 ECC Protection Register (ECCPRCR) 1151
42.2.8 ECC Protection Register 2 (ECCPRCR2) 1152
42.2.9 ECC Test Control Register (ECCETST) 1152
42.2.10 SRAM ECC Error Operation After Detection Register (ECCOAD) 1153
42.3 Operation 1153
42.3.1 Low Power Consumption Function 1153
42.3.2 ECC Function 1153
42.3.3 ECC Error Generation 1154
42.3.4 ECC Decoder Testing 1154
42.3.5 Parity Calculation Function 1155
42.3.6 SRAM Error Sources 1157
42.3.7 Access Cycle 1158
42.4 Usage Notes 1158

42.4.1 Instruction Fetch from SRAM area 1158
42.4.2 Store Buffer of SRAM 1158

43. Flash Memory 1159
43.1 Overview 1159
43.2 Memory Structure 1160
43.3 Flash Cache 1161
43.3.1 Overview 1161
43.3.2 Register Descriptions 1162
43.3.2.1 Flash Cache Enable Register (FCACHEE) 1162
43.3.2.2 Flash Cache Invalidate Register (FCACHEIV) 1162
43.3.2.3 Data Flash Control Resister (DFLCTL) 1163
43.4 Operation 1163
43.4.1 Notice to use Flash Cache 1163
43.5 Operating Modes Associated with the Flash Memory 1164
43.5.1 ID Code Protection 1164
43.6 Overview of Functions 1165
43.6.1 Configuration Area Bit Map 1167
43.6.2 Startup Area Select 1167
43.6.3 Protection by Access Window 1168
43.7 Programming Commands 1169
43.8 Suspend Operation 1169
43.9 Protection 1169
43.10 Serial Programming Mode 1169
43.10.1 SCI Boot Mode 1170
43.10.2 USB Boot Mode 1170
43.11 Using a Serial Programmer 1171
43.11.1 Serial Programming 1171
43.11.2 Programming Environment 1171
43.12 Self-Programming 1171
43.12.1 Overview 1171
43.12.2 Background Operation 1172
43.13 Reading the Flash Memory 1172
43.13.1 Reading the Code Flash Memory 1172
43.13.2 Reading the Data Flash Memory 1172
43.14 Usage Notes 1172
43.14.1 Erase Suspended Area 1172
43.14.2 Suspension by Erase Suspend Commands 1173
43.14.3 Constraints on Additional Writes 1173
43.14.4 Reset during Programming and Erasure 1173
43.14.5 Non-Maskable Interrupt Disabled during Programming and Erasure 1173
43.14.6 Location of Interrupt Vectors during Programming and Erasure 1173

43.14.7 Programming and Erasure in Low-Speed Operating Mode 1173
43.14.8 Abnormal Termination during Programming and Erasure 1173
43.14.9 Actions Prohibited during Programming and Erasure 1173

44. Segment LCD Controller (SLCDC) 1174
44.1 Overview 1174
44.2 Register Descriptions 1175
44.2.1 LCD Mode Register 0 (LCDM0) 1175
44.2.2 LCD Mode Register 1 (LCDM1) 1176
44.2.3 LCD Clock Control Register 0 (LCDC0) 1177
44.3 LCD Display Data Registers 1177
44.4 Selection of LCD Display Data Register 1178
44.4.1 A-Pattern Area and B-pattern Area Data Display 1178
44.4.2 Blinking Display (Alternately Displaying A-Pattern and B-Pattern Area Data) 1178
44.5 Setting LCD Controller/Driver 1179
44.6 Operation Stop Procedure 1180
44.7 Supplying LCD Drive Voltages VL1, VL2, and VL4 1181
44.7.1 External Resistance Division Method 1181
44.8 Common and Segment Signals 1182
44.9 Display Modes 1187
44.9.1 Four-Time-Slice Display Example 1187

45. Secure Cryptographic Engine (SCE5) 1192
45.1 Overview 1192
45.2 Operation 1193
45.2.1 Encryption Engine 1193
45.2.2 Encryption and Decryption 1194
45.3 Usage Notes 1195
45.3.1 Software Standby Mode 1195
45.3.2 Settings for the Module-Stop Function 1195

46. Bluetooth Low Energy (BLE) 1196
46.1 Overview 1196
46.2 Operation 1198
46.2.1 State Transitions 1198
46.3 Interrupts 1199
46.4 Usage Notes 1199
46.4.1 RF Transceiver Power-Supply 1199
46.4.2 Wireless Standards 1200
46.4.3 Notes on Board Design 1201

47. Internal Voltage Regulator 1202
47.1 Overview 1202
47.2 Operation 1202

48. Electrical Characteristics 1203
48.1 Absolute Maximum Ratings 1204
48.2 DC Characteristics 1206
48.2.1 Tj/Ta Definition 1206
48.2.2 I/O VIH, V_{IL} 1206
48.2.3 I/O I_{OH}, I_{OL} 1208
48.2.4 I/O V_{OH}, V_{OL}, and Other Characteristics 1209
48.2.5 I/O Pin Output Characteristics of Low Drive Capacity 1210
48.2.6 I/O Pin Output Characteristics of Middle Drive Capacity 1212
48.2.7 P409 I/O Pin Output Characteristics of Middle Drive Capacity 1214
48.2.8 IIC I/O Pin Output Characteristics 1216
48.2.9 Operating and Standby Current 1217
48.2.10 VCC Rise and Fall Gradient and Ripple Frequency 1226
48.3 AC Characteristics 1227
48.3.1 Frequency 1227
48.3.2 Clock Timing 1229
48.3.3 Reset Timing 1232
48.3.4 Wakeup Time 1233
48.3.5 NMI and IRQ Noise Filter 1237
48.3.6 I/O Ports, POEG, GPT, AGT, KINT, and ADC14 Trigger Timing 1238
48.3.7 CAC Timing 1239
48.3.8 SCI Timing 1240
48.3.9 SPI Timing 1246
48.3.10 IIC Timing 1251
48.3.11 CLKOUT Timing 1252
48.4 USB Characteristics 1254
48.4.1 USBFS Timing 1254
48.5 ADC14 Characteristics 1256
48.6 DAC12 Characteristics 1264
48.7 TSN Characteristics 1266
48.8 OSC Stop Detect Characteristics 1266
48.9 POR and LVD Characteristics 1267
48.10 VBATT Characteristics 1271
48.11 CTSU Characteristics 1273
48.12 Segment LCD Controller Characteristics 1274
48.12.1 Resistance Division Method 1274
48.13 Comparator Characteristics 1274
48.14 OPAMP Characteristics 1275
48.15 Flash Memory Characteristics 1276
48.15.1 Code Flash Memory Characteristics 1276
48.15.2 Data Flash Memory Characteristics 1277
48.16 Joint Test Action Group (JTAG) 1278

48.16.1 Serial Wire Debug (SWD).......... 1280
48.17 BLE Characteristics 1281
48.17.1 Transmission Characteristics 1281
48.17.2 Reception Characteristics (2 Mbps) 1282
48.17.3 Reception Characteristics (1 Mbps) 1282
48.17.4 Reception Characteristics (500 kbps).......... 1283
48.17.5 Reception Characteristics (125 kbps).......... 1283

Appendix 1. Port States in each Processing Mode.......... 1284

Appendix 2. Package Dimensions.......... 1286

Appendix 3. I/O Registers.......... 1288
3.1 Peripheral Base Addresses 1288
3.2 Access Cycles 1289
3.3 Register Descriptions.......... 1291

Revision History 1317

RA4W1 Group

User's Manual

High efficiency 48-MHz Arm® Cortex®-M4 core, 512-KB code flash memory, 96-KB SRAM, Segment LCD Controller, Capacitive Touch Sensing Unit, Bluetooth Low Energy, USB 2.0 Full-Speed, 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, and ETB
- CoreSight™ debug port: JTAG-DP and SW-DP

■ Memory
- 512-KB code flash memory
- 8-KB data flash memory (100,000 erase/write cycles)
- 96-KB SRAM
- Flash Cache (FCACHE)
- Memory Protection Units
- Memory Mirror Function (MMF)
- 128-bit unique ID

■ Connectivity
- Bluetooth Low Energy
  - Bluetooth 5.0 core specification compliant BLE transceiver and link layer
  - Supporting LE 1M, 2M and Coded PHY, and LE Advertising extension
  - Dedicated AES-CCM (128-bit blocks) encryption circuit
- USB 2.0 Full-Speed (USBFS) module
  - On-chip transceiver
  - Compliant with USB Battery Charging Specification 1.2
- Serial Communications Interface (SCI) × 4
  - UART
  - Simple IIC
  - Simple SPI
- Serial Peripheral Interface (SPI) × 2
- I^{2}C bus interface (IIC) × 2
- Controller Area Network (CAN) module

■ Analog
- 14-bit A/D Converter (ADC14)
- 12-bit D/A Converter (DAC12)
- 8-bit D/A Converter (DAC8) ×2 (for ACMPLP)
- Low Power Analog Comparator (ACMPLP) × 2
- Operational Amplifier (OPAMP) × 1
- Temperature Sensor (TSN)

■ Timers
- General PWM Timer 32-bit (GPT32) × 4
- General PWM Timer 16-bit (GPT16) × 3
- Asynchronous General-Purpose Timer (AGT) × 2
- 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 9 segments × 4 commons
  - Capacitive Touch Sensing Unit (CTSU)

■ Multiple Clock Sources
- Main clock oscillator (MOSC)
  (1 to 20 MHz when VCC = 2.4 to 3.6 V)
  (1 to 8 MHz when VCC = 1.8 to 2.4 V)
- Sub-clock oscillator (SOSC) (32.768 kHz)
- High-speed on-chip oscillator (HOCO)
  (24, 32, 48, 64 MHz when VCC = 2.4 to 3.6 V)
  (24, 32, 48 MHz when VCC = 1.8 to 3.6 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 35 input/output pins
  - Up to 3 CMOS input
  - Up to 32 CMOS input/output
    - Up to 4 input/output 5 V tolerant
    - Up to 1 high current (20 mA)

■ Operating Voltage
- VCC: 1.8 to 3.6 V

■ Operating Temperature and Packages
- Ta = –40°C to +85°C
  - 56-pin QFN (7 mm × 7 mm, 0.4 mm pitch)

1. Overview

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

The MCU in this series incorporates a low-power and high-performance Arm Cortex®-M4 32-bit core running up to 48 MHz, with the following features:

- 512-KB code flash memory
- 96-KB SRAM
- Bluetooth Low Energy (BLE)
- 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 core	• Maximum operating frequency: up to 48 MHz • Arm Cortex-M4 core: - 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 protect regions • SysTick timer: - Driven by SYSTICCLK (LOCO) or ICLK.

Table 1.2 Memory

Feature	Functional description
Code flash memory	Maximum 512 KB of code flash memory. See section 43, Flash Memory.
Data flash memory	8 KB of data flash memory. See section 43, 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 42, 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 • SRAM parity error reset • SRAM ECC error reset • Bus master MPU error reset • Bus slave MPU error reset • Stack pointer error reset • Software reset. See section 6, Resets.
Low Voltage Detection (LVD)	The Low Voltage Detection (LVD) 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 • Bluetooth-dedicated clock oscillator • Bluetooth-dedicated low-speed 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 Mode	Power consumption can be reduced in multiple ways, such as by setting clock dividers, 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, SOSC, LOCO, wakeup control, backup memory, VBATT_R low voltage detection, and switch between VCC and VBATT. During normal operation, the battery powered area is powered by the main power supply, which is the VCC pin. When a VCC voltage fall 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 because of software errors. See section 13, Register Write Protection.

Table 1.3 System (2 of 2)

Feature	Functional description
Memory Protection Unit (MPU)	Four Memory Protection Units (MPUs) and a CPU stack pointer monitor function are provided for memory protection. See section 16, Memory Protection Unit (MPU).
Watchdog Timer (WDT)	The Watchdog Timer (WDT) is a 14-bit down-counter that can be used to reset the MCU when the counter underflows because the system has run out of control and is unable to refresh the WDT. In addition, a non-maskable interrupt or interrupt can be generated by an underflow. The refresh-permitted period can be set to refresh the counter and used as the condition for detecting when the system runs out of control. See section 26, Watchdog Timer (WDT).
Independent Watchdog Timer (IWDT)	The Independent Watchdog Timer (IWDT) consists of a 14-bit down-counter that must be serviced periodically to prevent counter underflow. It can be used 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, 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 Event Link Controller (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 Data Transfer Controller (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 DMA Controller (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 Timers

Feature	Functional description
General PWM Timer (GPT)	The General PWM Timer (GPT) is a 32-bit timer with 4 channels and a 16-bit timer with 3 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 Port Output Enable for GPT (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 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 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 Realtime Clock (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 1.7 Communication interfaces (1 of 2)

Feature	Functional description
Serial Communications Interface (SCI)	The Serial Communication 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. SCI0 and SCI1 have FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator. See section 29, Serial Communications Interface (SCI).
I^{2}C bus interface (IIC)	The 2-channel I2C bus interface (IIC) conforms with and provides a subset of the NXP I^{2}C (Inter-Integrated Circuit) bus interface functions. See section 30, I^{2}C Bus Interface (IIC).
Serial Peripheral Interface (SPI)	Two independent Serial Peripheral Interface (SPI) channels are capable of high-speed, full-duplex synchronous serial communications with multiple processors and peripheral devices. See section 32, Serial Peripheral Interface (SPI).
Controller Area Network (CAN) module	The Controller Area Network (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 31, Controller Area Network (CAN) Module.
USB 2.0 Full-Speed (USBFS) module	The USB 2.0 Full-Speed (USBFS) module can operate as a host controller or device controller. The module supports full-speed and low-speed (only for the host controller) 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. See section 28, USB 2.0 Full-Speed Module (USBFS).

Table 1.7 Communication interfaces (2 of 2)

Feature	Functional description
Bluetooth low energy(BLE)	• On-chip RF transceiver and link layer compliant with the Bluetooth 5.0 Low Energy specification • Bit rates: 1 Mbps, 2 Mbps, 500 kbps, and 125 kbps • LE Advertising extension support • Includes an RF transceiver power supply (selectable as a DC-to-DC converter or linear regulator) • On-chip matching circuit to help reduce the number of external parts • Transmission power: +4 dBm support

Table 1.8 Analog

Feature	Functional description
14-bit A/D Converter (ADC14)	A 14-bit successive approximation A/D converter is provided. Up to 8 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 34, 14-Bit A/D Converter (ADC14).
12-bit D/A Converter (DAC12)	The 12-bit D/A Converter (DAC12) converts data and includes an output amplifier. See section 35, 12-Bit D/A Converter (DAC12).
8-bit D/A Converter (DAC8) for ACMPLP	The 8-bit D/A Converter (DAC8) converts data and does not include an output amplifier. The DAC8 is used only as the reference voltage for ACMPLP. See section 39, 8-Bit D/A Converter (DAC8).
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 36, Temperature Sensor (TSN).
Low-Power Analog Comparator (ACMPLP)	The Low-Power Analog Comparator (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 voltage can be selected from an input to the CMPREFi(i = 0,1) pin, an internal 8-bit D/A converter output, or 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 38, Low Power Analog Comparator (ACMPLP).
Operational Amplifier (OPAMP)	The Operational Amplifier (OPAMP) can be used to amplify small analog input voltages and output the amplified voltages. A differential operational amplifier unit with two input pins and one output pin are provided. See section 37, Operational Amplifier (OPAMP).

Table 1.9 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 uses an external resistance division method • Automatic output of segment and common signals based on automatic display data register read • The LCD can be made to blink. See section 44, Segment LCD Controller (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 electrode. See section 40, Capacitive Touch Sensing Unit (CTSU).

Table 1.10 Data processing

Feature	Functional description
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. See section 33, Cyclic Redundancy Check (CRC) Calculator.
Data Operation Circuit (DOC)	The Data Operation Circuit (DOC) compares, adds, and subtracts 16-bit data. See section 41, Data Operation Circuit (DOC).

Table 1.11 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 a block diagram of the MCU superset. Some individual devices within the group may have a subset of the features.

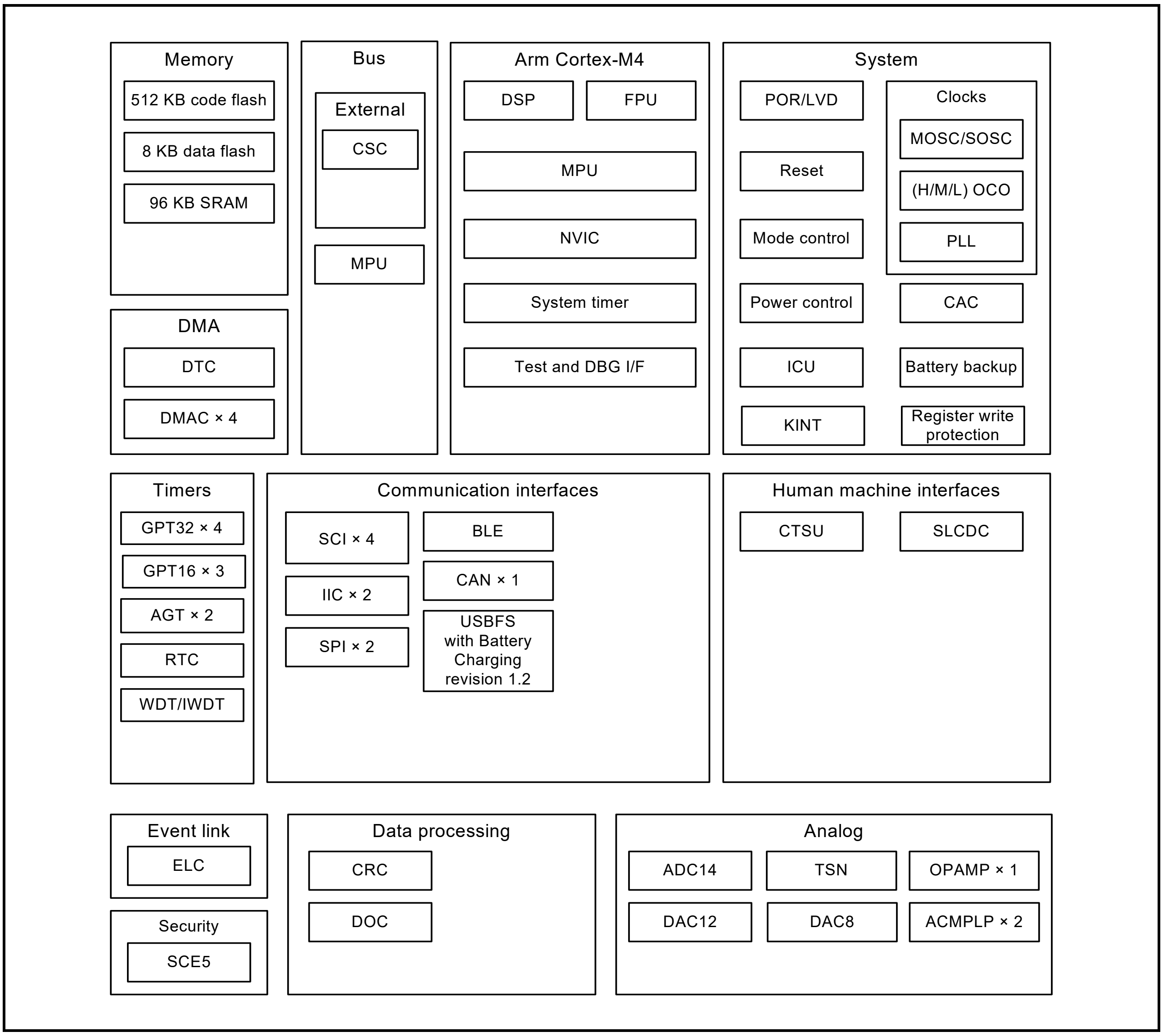

Memory
512 KB code flash
8 KB data flash
96 KB SRAM
Bus
External
CSC
MPU
Arm Cortex-M4
DSP
FPU
MPU
NVIC
System timer
Test and DBG I/F
System
POR/LVD
Reset
Mode control
Power control
ICU
KINT
Clocks
MOSC/SOSC
(H/M/L) OCO
PLL
CAC
Battery backup
Register write protection
DMA
DTC
DMAC × 4
Timers
GPT32 × 4
GPT16 × 3
AGT × 2
RTC
WDT/IWDT
Communication interfaces
SCI × 4
IIC × 2
SPI × 2
BLE
CAN × 1
USBFS with Battery Charging revision 1.2
Human machine interfaces
CTSU
SLCDC
Event link
ELC
Security
SCE5
Data processing
CRC
DOC
Analog
ADC14
TSN
OPAMP × 1
DAC12
DAC8
ACMPLP × 2

Figure 1.1 Block diagram

1.3 Part Numbering

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

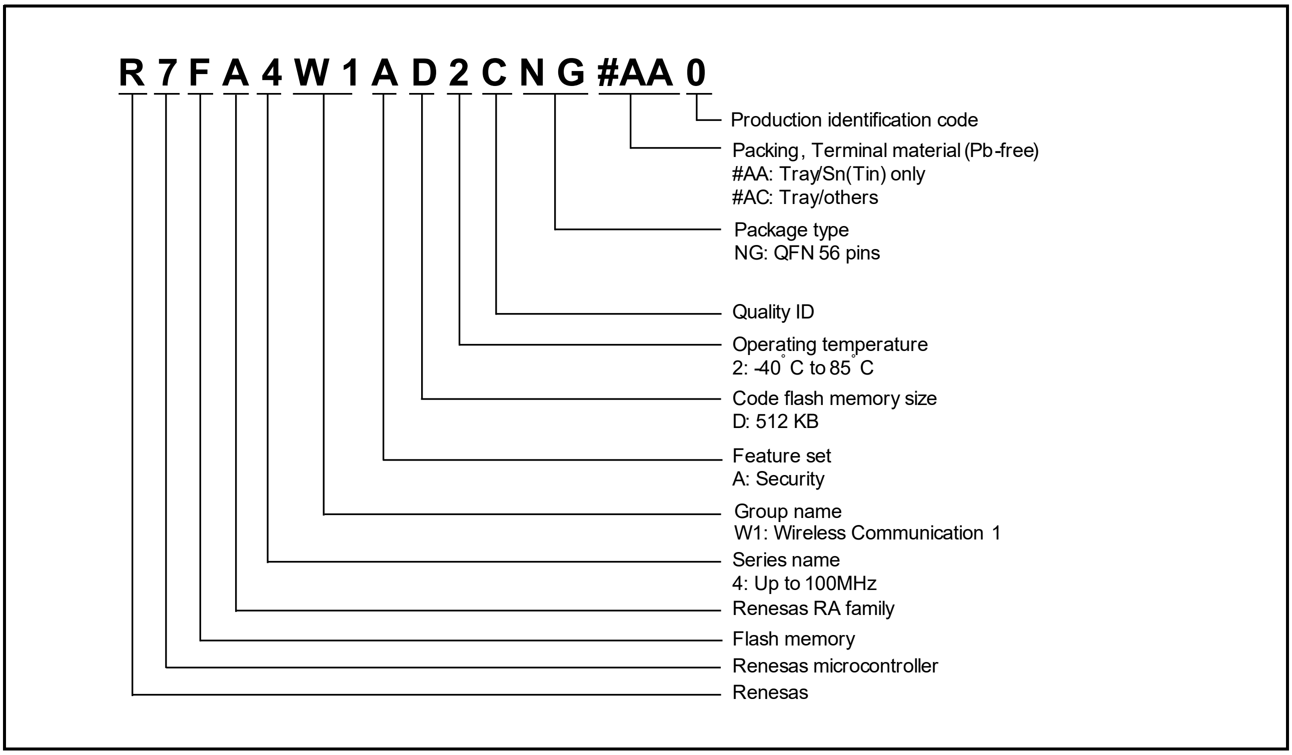

R 7 F A 4 W 1 A D 2 C N G #AA 0
Production identification code
Packing, Terminal material (Pb-free)
#AA: Tray/Sn(Tin) only
#AC: Tray/others
Package type
NG: QFN 56 pins
Quality ID
Operating temperature
2: -40° C to 85° C
Code flash memory size
D: 512 KB
Feature set
A: Security
Group name
W1: Wireless Communication 1
Series name
4: Up to 100MHz
Renesas RA family
Flash memory
Renesas microcontroller
Renesas

Figure 1.2 Part numbering scheme

Table 1.12 Product list

Product part number	Orderable part number	Code flash	Data flash	SRAM	Operating temperature
R7FA4W1AD2CNG	R7FA4W1AD2CNG#AA0	512 KB	8 KB	96 KB	-40 to +85°C

1.4 Function Comparison

Table 1.13 Function comparison

Part numbers		**R7FA4W1AD2CNG**
Pin count		56
Package		QFN
Code flash memory		512 KB
Data flash memory		8 KB
SRAM		96 KB
	Parity	80 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
Timers	GPT32	4
	GPT16	3
	AGT	2
	RTC	Yes
	WDT/IWDT	Yes
Communication	SCI	6
	IIC	2
	SPI	2
	CAN	1
	USBFS	Yes
	BLE	An RF transceiver and link layer compliant with Bluetooth 5.0 low energy specification
Analog	ADC14	8
	DAC12	1
	DAC8	2
	ACMPLP	2
	OPAMP	1
	TSN	Yes
HMI	SLCDC	4 com × 9 seg
	CTSU	11
Data processing	CRC	Yes
	DOC	Yes
Security		SCE5

1.5 Pin Functions

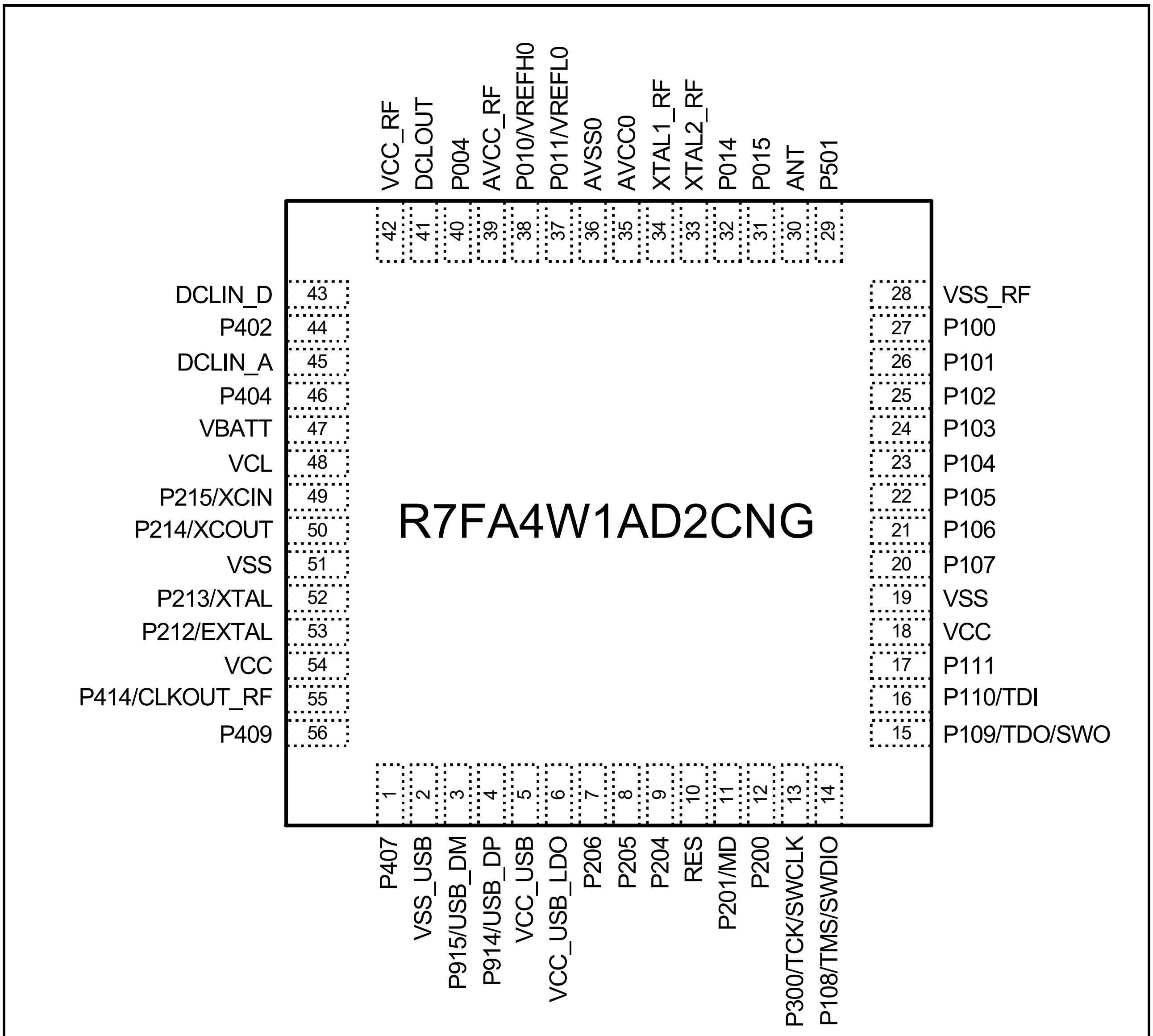
Function	Signal	I/O	Description
Power supply	VCC	Input	Power supply pin. Connect it to the system power supply. Connect this pin to VSS by a 0.1-µF capacitor. The capacitor should be placed close to the pin.
	VCL	Input	Connect this pin to the VSS pin by the smoothing capacitor used to stabilize the internal power supply. Place the capacitor close to the pin.
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
	VBATT	Input	Backup power 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	
	CLKOUT_RF	Output	Bluetooth-dedicated clock output pin for output of a 1-, 2-, or 4-MHz signal
	XTAL1_RF	Input	Pins for connecting the Bluetooth-dedicated clock oscillator. Connect a 32-MHz oscillator to these pins.
	XTAL2_RF	Output	
	CLKOUT	Output	Clock output pin
Operating mode control	MD	Input	Pins for setting the operating mode. The signal levels on these pins must not be changed during operation mode transition at the time of 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 IRQ4, IRQ6, IRQ7, IRQ9, IRQ11, IRQ14, IRQ15	Input	Maskable interrupt request pins
KINT	KR00 to KR07	Input	A key interrupt can be generated by inputting a falling edge to the key interrupt input pins
On-chip debug	TMS	I/O	On-chip emulator 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
Battery backup	VBATWIO0	I/O	Output wakeup signal for the VBATT wakeup control function. External event input for the VBATT wakeup control function.
GPT	GTETRGA, GTETRGB	Input	External trigger input pin
	GTIO0A to GTIOA5A,GTIO8A, GTIO0B to GTIOA5B,GTIO8B	I/O	Input capture, Output capture, or PWM output pin
	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
	AGTIO0, AGTIO1	I/O	External event input and pulse output
	AGTO0, AGTO1	Output	Pulse output
	AGTOB0	Output	Output compare match B output

Function	Signal	I/O	Description
RTC	RTCOUT	Output	Output pin for 1-Hz/64-Hz clock
	RTCIC0, RTCIC2	Input	Time capture event input pins
SCI	SCK0,SCK1,SCK4, SCK9	I/O	Input/output pins for the clock (clock synchronous mode)
	RXD0, RXD1, RXD4, RXD9	Input	Input pins for received data (asynchronous mode/clock synchronous mode)
	TXD0, TXD1, TXD4, TXD9	Output	Output pins for transmitted data (asynchronous mode/clock synchronous mode)
	CTS0_RTS0, CTS1_RTS1, 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, SCL1, SCL4, SCL9	I/O	Input/output pins for the IIC clock (simple IIC)
	SDA0, SDA1, SDA4, SDA9	I/O	Input/output pins for the IIC data (simple IIC)
	SCK0, SCK1, SCK4, SCK9	I/O	Input/output pins for the clock (simple SPI)
	MISO0, MISO1, MISO4, MISO9	I/O	Input/output pins for slave transmission of data (simple SPI)
	MOSI0, MOSI1, MOSI4, MOSI9	I/O	Input/output pins for master transmission of data (simple SPI)
	SS0, SS1,SS4,SS9	Input	Slave-select input pins (simple SPI), active-low
IIC	SCL0 to SCL1	I/O	Input/output pins for clock
	SDA0 to SDA1	I/O	Input/output pins for data
SPI	RSPCKA, RSPCKB	I/O	Clock input/output pin
	MOSIA, MOSIB	I/O	Inputs or outputs data output from the master
	MISOA, MISOB	I/O	Inputs or outputs data output from the slave
	SSLA0, SSLB0	I/O	Input or output pin for slave selection
	SSLA1, SSLA2, SSLA3, SSLB1, SSLB3	Output	Output pin for slave selection
CAN	CRX0	Input	Receive data
	CTX0	Output	Transmit data
USBFS	VSS_USB	Input	Ground pins
	VCC_USB_LDO	Input	Power supply pin for USB transceiver. Apply the same voltage as VCC_USB.
	VCC_USB	I/O	Input: Power supply pin for USB transceiver.
	USB_DP	I/O	D+ I/O pin of the USB on-chip transceiver. This pin should be connected to the D+ pin of the USB bus.
	USB_DM	I/O	D– I/O pin of the USB on-chip transceiver. This pin should be connected to the D– pin of the USB bus.
	USB_VBUS	Input	USB cable connection monitor pin. This pin should be connected 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_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip
	USB_OVRCURA, USB_OVRCURB	Input	External overcurrent detection signals should be connected to these pins.
Analog power supply	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
ADC14	AN004 to AN006, AN009, AN010, AN017, AN019, AN020	Input	Input pins for the analog signals to be processed by the A/D converter
	ADTRG0	Input	Input pins for the external trigger signals that start the A/D conversion, active-low
DAC12	DA0	Output	Output pins for the analog signals to be processed by the D/A converter

Function	Signal	I/O	Description
Comparator output	VCOUT	Output	Comparator output pin
ACMPLP	CMPREF0, CMPREF1	Input	Reference voltage input pins
	CMPIN0, CMPIN1	Input	Analog voltage input pins
OPAMP	AMP2+	Input	Analog voltage input pins
	AMP2-	Input	Analog voltage input pins
	AMP2O	Output	Analog voltage output pins
CTSU	TS00, TS01, TS03, TS10, TS12, TS13, TS18, TS28, TS30, TS31, TS34	Input	Capacitive touch detection pins (touch pins)
	TSCAP	—	Secondary power supply pin for the touch driver
I/O ports	P004, P010, P011, P014, P015	I/O	General-purpose input/output pins
	P100 to P111	I/O	General-purpose input/output pins
	P200	Input	General-purpose input pin
	P201, P204 to P206, P212, P213	I/O	General-purpose input/output pins
	P214, P215	Input	General-purpose input pins
	P300	I/O	General-purpose input/output pins
	P402, P404, P407, P409, P414	I/O	General-purpose input/output pins
	P501	I/O	General-purpose input/output pins
	P914, P915	I/O	General-purpose input/output pins
SLCDC	VL1, VL2, VL4	I/O	Voltage pin for driving the LCD
	COM0 to COM3	Output	Common signal output pins for the LCD controller/driver
	SEG6, SEG9, SEG11, SEG12, SEG20, SEG23, SEG49, SEG52, SEG53	Output	Segment signal output pins for the LCD controller/driver
BLE (Bluetooth Low Energy)	ANT	I/O	RF single I/O pin for the RF transceiver Set the impedance of the signal line to 50 Ω.
	DCLOUT	Output	RF transceiver power-supply output pin
	DCLIN_A	Input	RF transceiver power-supply output connection pin
	DCLIN_D	Input	RF transceiver power-supply output connection pin
	VCC_RF	Input	RF transceiver power supply pin
	AVCC_RF	Input	RF transceiver power supply pin
	VSS_RF	Input	RF transceiver ground pin

1.6 Pin Assignments

Figure 1.3 shows the pin assignments.


R7FA4W1AD2CNG
1 P407
2 VSS_USB
3 P915/USB_DM
4 P914/USB_DP
5 VCC_USB
6 VCC_USB_LDO
7 P206
8 P205
9 P204
10 RES
11 P201/MD
12 P200
13 P300/TCK/SWCLK
14 P108/TMS/SWDIO
15 P109/TDO/SWO
16 P110/TDI
17 P111
18 VCC
19 VSS
20 P107
21 P106
22 P105
23 P104
24 P103
25 P102
26 P101
27 P100
28 VSS_RF
29 P501
30 ANT
31 P015
32 P014
33 XTAL2_RF
34 XTAL1_RF
35 AVCC0
36 AVSS0
37 P011/VREFL0
38 P010/VREFH0
39 AVCC_RF
40 P004
41 DCLOUT
42 VCC_RF
43 DCLIN_D
44 P402
45 DCLIN_A
46 P404
47 VBATT
48 VCL
49 P215/XCIN
50 P214/XCOUT
51 VSS
52 P213/XTAL
53 P212/EXTAL
54 VCC
55 P414/CLKOUT_RF
56 P409

Note: VSS_RF is assigned as the exposed die pad.
For details, refer to Appendix 2, Package Dimensions.

Figure 1.3 Pin assignment for QFN 56-pin (top view)

1.7 Pin Lists

Pin number				Timers				Communication interfaces					Analogs			HMI	
QFN56	Power, System, Clock, Debug, CAC, VBATT	Interrupt	I/O Ports	AGT	GPT_OPS, POEG	GPT	RTC	USBFS, CAN	SCI	IIC	SPI	RF	ADC14	DAC12, OPAMP	ACMPLP	SLCDC	CTSU
1			P407	AGTIO0			RTCOUT	USB_VBUS	CTS4_RTS4/SS4	SDA0	SSLB3		ADTRG0			SEG11	TS3
2	VSS_USB																
3			P915					USB_DM									
4			P914					USB_DP									
5	VCC_USB																
6	VCC_USB_LDO																
7		IRQ0	P206		GTIU			USB_VBUSEN	RXD4/MISO4/SCL4	SDA1	SSLB1					SEG12	TS1
8	CLKOUT	IRQ1	P205	AGTO1	GTIV	GTIOC4A		USB_OVRCURA	TXD4/MOSI4/SDA4/CTS9_RTS9/SS9	SCL1	SSLB0					SEG20	TSCAP
9	CACREF		P204	AGTO1	GTIW	GTIOC4B		USB_OVRCURB	SCK4/SCK9	SCL0	RSPCKB					SEG23	TS0
10	RES																
11	MD		P201														
12		NMI	P200														
13	TCK/SWCLK		P300		GTOUUP	GTIOC0A					SSLB1						
14	TMS/SWDIO		P108		GTOULO	GTIOC0B			CTS9_RTS9/SS9		SSLB0						
15	TDO/SWO/CLKOUT		P109		GTOVUP	GTIOC1A		CTX0	SCK1/TXD9/MOSI9/SDA9		MOSIB					SEG52	TS10
16	TDI	IRQ3	P110		GTOVLO	GTIOC1B		CRX0	RXD9/MISO9/SCL9		MISOB				VCOUT	SEG53	
17		IRQ4	P111			GTIOC3A			SCK9		RSPCKB						TS12
18	VCC																
19	VSS																
20		KR07	P107			GTIOC8A										COM3	
21		KR06	P106			GTIOC8B					SSLA3					COM2	
22		KR05/IRQ0	P105		GTETRGA	GTIOC1A					SSLA2					COM1	TS34
23		KR04/IRQ1	P104		GTETRGB	GTIOC1B			RXD0/MISO0/SCL0		SSLA1					COM0	TS13
24		KR03	P103		GTOWUP	GTIOC2A		CTX0	CTS0_RTS0/SS0		SSLA0		AN019		CMPREF1	VL4	
25		KR02	P102	AGTO0	GTOWLO	GTIOC2B		CRX0	SCK0		RSPCKA		AN020/ADTRG0		CMPIN1		
26		KR01/IRQ1	P101	AGTEE0	GTETRGB	GTIOC5A			TXD0/MOSI0/SDA0/CTS1_RTS1/SS1	SDA1	MOSIA				CMPREF0	VL2	

Pin number				**Timers**				**Communication interfaces**					**Analogs**			**HMI**	
QFN56	**Power, System, Clock, Debug, CAC, VBATT**	**Interrupt**	**I/O Ports**	**AGT**	**GPT_OPS, POEG**	**GPT**	**RTC**	**USBFS, CAN**	**SCI**	**IIC**	**SPI**	**RF**	**ADC14**	**DAC12, OPAMP**	**ACMPLP**	**SLCDC**	**CTSU**
27		KR00/IRQ2	P100	AGTIO0	GTETRGA	GTIOC5B			RXD0/MISO0/SCL0/SCK1	SCL1	MISOA				CMPIN0	VL1	
28												VSS_RF					
29		IRQ11	P501	AGTOB0	GTIV	GTIOC2B		USB_OVRCURA					AN017		CMPIN1	SEG49	
30												ANT					
31		IRQ7	P015										AN010				TS28
32			P014										AN009	DA0			
33												XTAL2_RF					
34												XTAL1_RF					
35	AVCC0																
36	AVSS0																
37	VREFL0	IRQ15	P011										AN006	AMP2+			TS31
38	VREFH0	IRQ14	P010										AN005	AMP2-			TS30
39												AVCC_RF					
40		IRQ3	P004										AN004	AMP2O			
41												DCLOUT					
42												VCC_RF					
43												DCLIN_D					
44	VBATWIO0	IRQ4	P402	AGTIO0/AGTIO1			RTCIC0	CRX0	RXD1/MISO1/SCL1							SEG6	TS18
45												DCLIN_A					
46			P404			GTIOC3B	RTCIC2										
47	VBATT																
48	VCL																

Pin number				Timers				Communication interfaces					Analogs			HMI	
QFN56	Power, System, Clock, Debug, CAC, VBATT	Interrupt	I/O Ports	AGT	GPT_OPS, POEG	GPT	RTC	USBFS, CAN	SCI	IIC	SPI	RF	ADC14	DAC12, OPAMP	ACMPLP	SLCDC	CTSU
49	XCIN		P215														
50	XCOUT		P214														
51	VSS																
52	XTAL	IRQ2	P213		GTETRGA	GTIOC0A			TXD1/MOSI1/SDA1								
53	EXTAL	IRQ3	P212	AGTEE1	GTETRGB	GTIOC0B			RXD1/MISO1/SCL1								
54	VCC																
55		IRQ9	P414			GTIOC0B					SSLA1	CLKOUT_RF					
56		IRQ6	P409		GTOWUP	GTIOC5A		USB_EXICEN								SEG9	

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 reference 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 Write 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 a block diagram of the Cortex-M4 CPU.

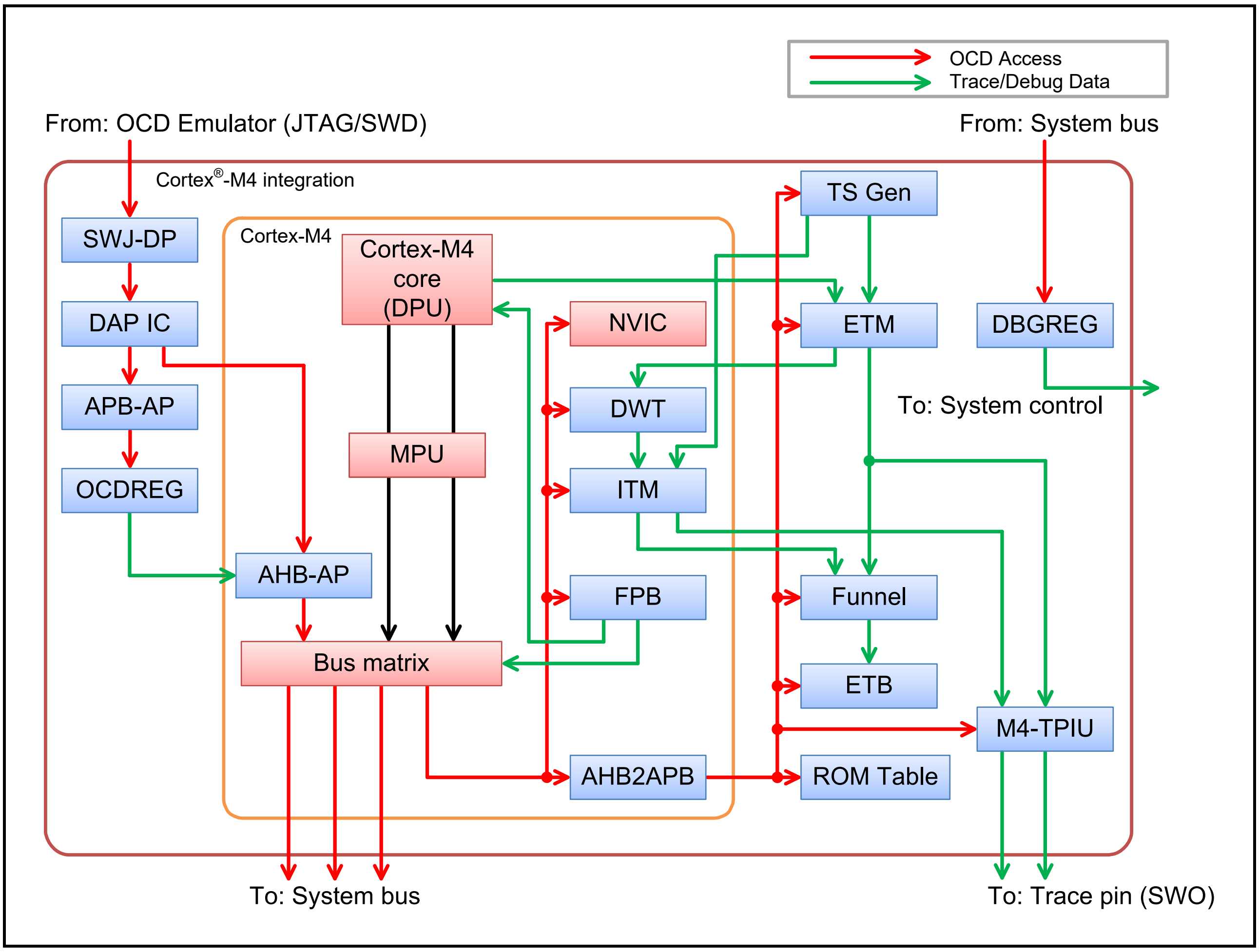

OCD Access
Trace/Debug Data
From: OCD Emulator (JTAG/SWD)
From: System bus
Cortex®-M4 integration
SWJ-DP
DAP IC
APB-AP
OCDREG
Cortex-M4
Cortex-M4 core (DPU)
MPU
NVIC
DWT
ITM
FPB
AHB-AP
Bus matrix
AHB2APB
TS Gen
ETM
DBGREG
To: System control
Funnel
ETB
M4-TPIU
ROM Table
To: System bus
To: Trace pin (SWO)

Figure 2.1 Cortex-M4 CPU block diagram

2.2 MCU Implementation Options

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

Table 2.1 Implementation options

Option	Description
MPU	Included, 8 protect 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*1	Not included
Debug Port	SWJ-DP
FPU	Included
Number of interrupts	32
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 more details, see section 11, Low Power Modes. Note: SCB.SCR.SLEEPDEEP is ignored.
Memory features	Cacheable attribute is utilized in the MCU. See section 15, Buses for the detail.
SysTick Timer	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 Application Interrupt and Reset Control Register causes a CPU reset
Auxiliary fault inputs (AUXFAULT)	Not implemented

Note 1. The ICU can wake up the CPU instead of the Wakeup Interrupt Controller (WIC). For more details, see section 14, Interrupt Controller Unit (ICU).

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/SWD pins.

Table 2.3 JTAG/SWD pins

Name	I/O	P/N	Width	Function	When not in use
TCK/SWCLK	Input	Pos.	1 bit	JTAG clock pin SWD clock pin	Pull-up
TMS/SWDIO	I/O	Neg.	1 bit	JTAG TMS pin SWD I/O pin	Pull-up
TDI	Input	Pos.	1 bit	JTAG TDI pin	Pull-up
TDO/SWO	Output	Neg.	1 bit	JTAG TDO pin Multiplexed with serial wire output	Open

2.5 Debug Mode

2.5.1 Debug Mode Definition

In single chip mode, the debugger state of the connection is defined as OCD mode, and the debugger state of the disconnection 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 the register settings even when the CPU enters Software Standby or Snooze mode. However, AHB-AP cannot respond to On-Chip Debug (OCD) access in these low power modes. The OCD must wait for cancellation of the low power mode to access the CoreSight debug components. To request low power mode cancellation, the OCD can set the DBIRQ bit in the MCUCTRL register. For details, see section 2.6.5.3, MCU Control Register (MCUCTRL).

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 (1 of 2)

Reset or Interrupt name	Control in On-Chip Debug (OCD) mode	
	OCD break mode	**OCD run mode**
RES pin reset	Same as user mode	
Power-on reset	Same as user mode	
Independent watchdog timer reset/interrupt	Does not occur*1	Depends on DBGSTOPCR setting*2

Table 2.5 Reset or interrupt and mode setting (2 of 2)

Reset or Interrupt name	Control in On-Chip Debug (OCD) mode	
	OCD break mode	**OCD run mode**
Watchdog timer reset/interrupt	Does not occur*1	Depends on DBGSTOPCR setting*2
Voltage monitor 0 reset	Depends on DBGSTOPCR setting*3	
Voltage monitor 1 reset/interrupt	Depends on DBGSTOPCR setting*3	
SRAM parity error reset/interrupt	Depends on DBGSTOPCR setting*3	
SRAM ECC error reset/interrupt	Depends on DBGSTOPCR setting*3	
MPU bus master reset/interrupt	Same as user mode	
MPU bus slave reset/interrupt	Same as user mode	
Stack pointer error reset/interrupt	Same as user mode	
Software reset	Same as user mode	

Note: In OCD break mode, the CPU is halted. In OCD run mode, the CPU is in OCD mode and the CPU is not halted.
Note 1. The IWDT and WDT always stop in this mode.
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 a block diagram of the AP connection and address spaces.

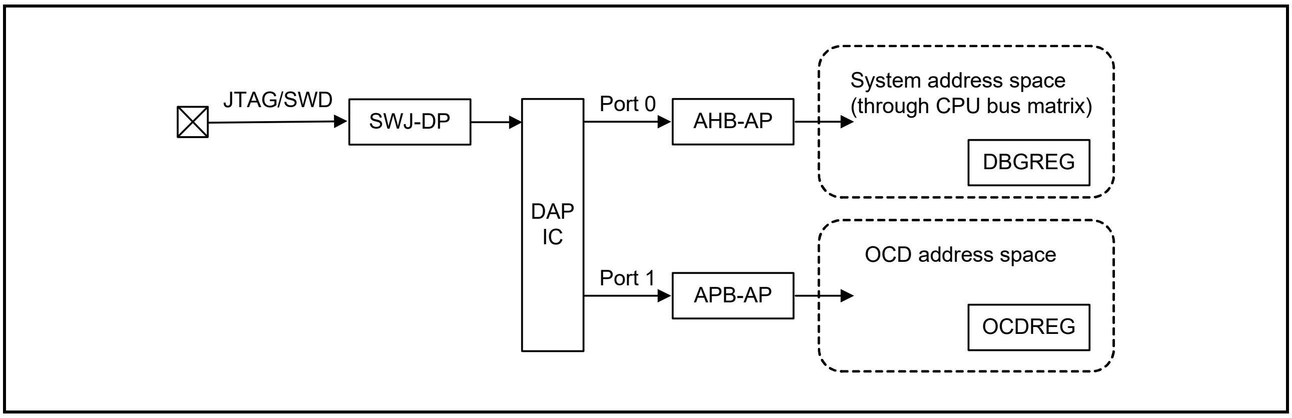


Figure 2.2 JTAG/SWD authentication block diagram

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

2.6.2 Cortex-M4 Peripheral Address Map

In the 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 the Arm components.

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	FFF0 F003h	SCS
1	E00F F004h	32 bits	R	FFF0 2003h	DWT
2	E00F F008h	32 bits	R	FFF0 3003h	FPB
3	E00F F00Ch	32 bits	R	FFF0 1003h	ITM
4	E00F F010h	32 bits	R	FFF4 1003h	TPIU
5	E00F F014h	32 bits	R	FFF4 2003h	ETM
6	E00F F018h	32 bits	R	FFF4 3003h	Funnel
7	E00F F01Ch	32 bits	R	FFF4 4003h	ETB
8	E00F F020h	32 bits	R	FFF4 5003h	TSG
9	E00F F024h	32 bits	R	0000 0000h	(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 shows the registers. See reference 7. for details of each register.

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

Name	Address	Access size	R/W	Initial value
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

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

Name	Address	Access size	R/W	Initial value
PID7	E00F FFDCh	32 bits	R	0000 0000h
PID0	E00F FFE0h	32 bits	R	0000 0013h
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 register module controls the debug functionalities and is implemented as a CoreSight compliant component.

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

Table 2.9 Non-CoreSight DBGREG registers

Name		DAP port	Address	Access size	R/W
Debug Status Register	DBGSTR	Port 0	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[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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	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[1:0]	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
b23 to b18	—	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
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 shows these registers. See reference 7. for details of each register.

Table 2.10 DBGREG CoreSight component registers

Name	Address	Access size	R/W	Initial value
PID4	4001 BFD0h	32 bits	R	00000004h
PID5	4001 BFD4h	32 bits	R	00000000h
PID6	4001 BFD8h	32 bits	R	00000000h
PID7	4001 BFDCh	32 bits	R	00000000h
PID0	4001 BFE0h	32 bits	R	00000005h
PID1	4001 BFE4h	32 bits	R	00000030h
PID2	4001 BFE8h	32 bits	R	0000001Ah
PID3	4001 BFECh	32 bits	R	00000000h
CID0	4001 BFF0h	32 bits	R	0000000Dh
CID1	4001 BFF4h	32 bits	R	000000F0h
CID2	4001 BFF8h	32 bits	R	00000005h
CID3	4001 BFFCh	32 bits	R	000000B1h

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 shows the OCDREG registers other than the CoreSight component registers.

Table 2.11 Non-CoreSight 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

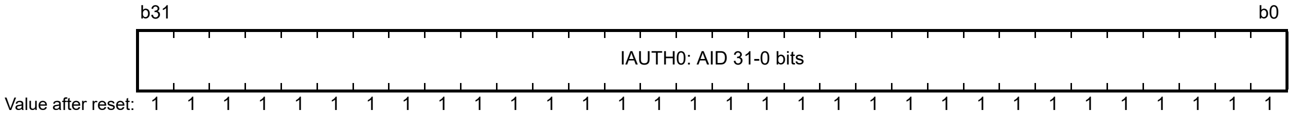
Note: OCDREG is located in the dedicated OCD address space. This address map is independent from the system address map. See section 2.6.2, Cortex-M4 Peripheral Address Map.

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.

Address(es): IAUTH0 8000 0000h



Address(es): IAUTH1 8000 0100h

b31 b0

	IAUTH1: AID 63-32 bits
Value after reset:	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Address(es): IAUTH2 8000 0200h

	b31 … b0
	IAUTH2: AID 95-64 bits
Value after reset:	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Address(es): IAUTH3 8000 0300h

	b31 … b0
	IAUTH3: AID 127-96 bits
Value after reset:	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

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		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. This indicates that the MCU is in Normal mode or Sleep mode 1: CPU clock is stopped. This indicates 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 MCU 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 shows these registers. See reference 7. for details of each register.

Table 2.12 OCDREG CoreSight component registers

Name	Address	Access size	R/W	Initial value
PID4	8000 0FD0h	32 bits	Read only	00000004h
PID5	8000 0FD4h	32 bits	Read only	00000000h
PID6	8000 0FD8h	32 bits	Read only	00000000h
PID7	8000 0FDCh	32 bits	Read only	00000000h
PID0	8000 0FE0h	32 bits	Read only	00000004h
PID1	8000 0FE4h	32 bits	Read only	00000030h
PID2	8000 0FE8h	32 bits	Read only	0000000Ah
PID3	8000 0FECh	32 bits	Read only	00000000h
CID0	8000 0FF0h	32 bits	Read only	0000000Dh
CID1	8000 0FF4h	32 bits	Read only	000000F0h
CID2	8000 0FF8h	32 bits	Read only	00000005h
CID3	8000 0FFCh	32 bits	Read only	000000B1h

2.7 CoreSight ATB Funnel

There is one CoreSight ATB funnel in the MCU. 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 MCU.

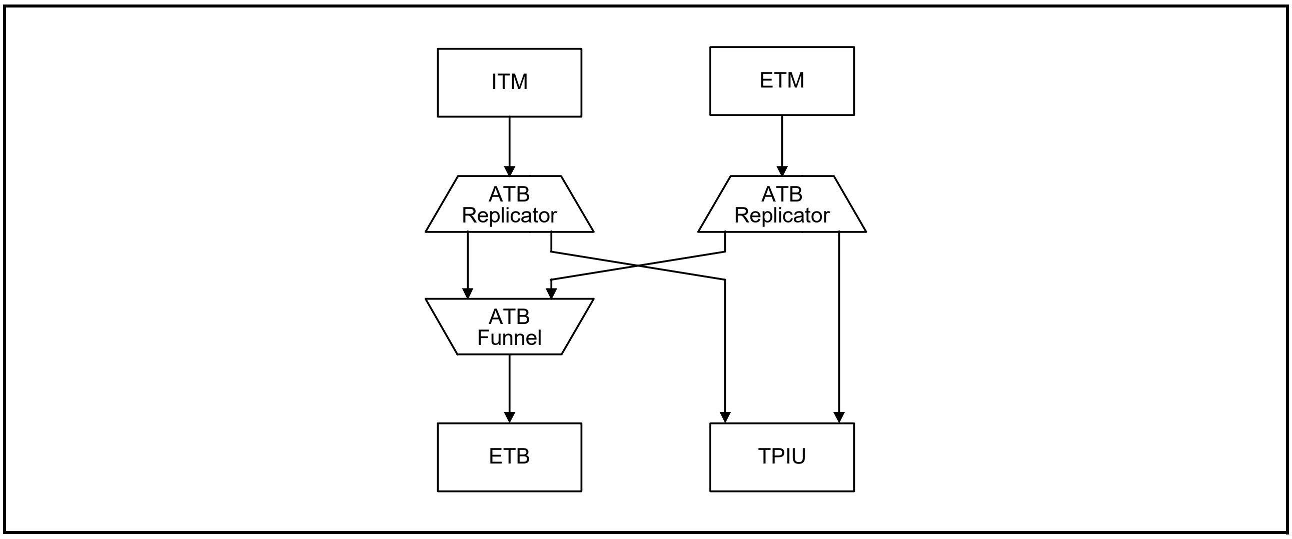


Figure 2.3 CoreSight ATB connection

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 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 SysTick Timer clock (SYSTICCLK). See section 9, Clock Generation Circuit and reference 1.*1 for details.

Note 1. In the reference, 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 a block diagram of the authentication mechanism.

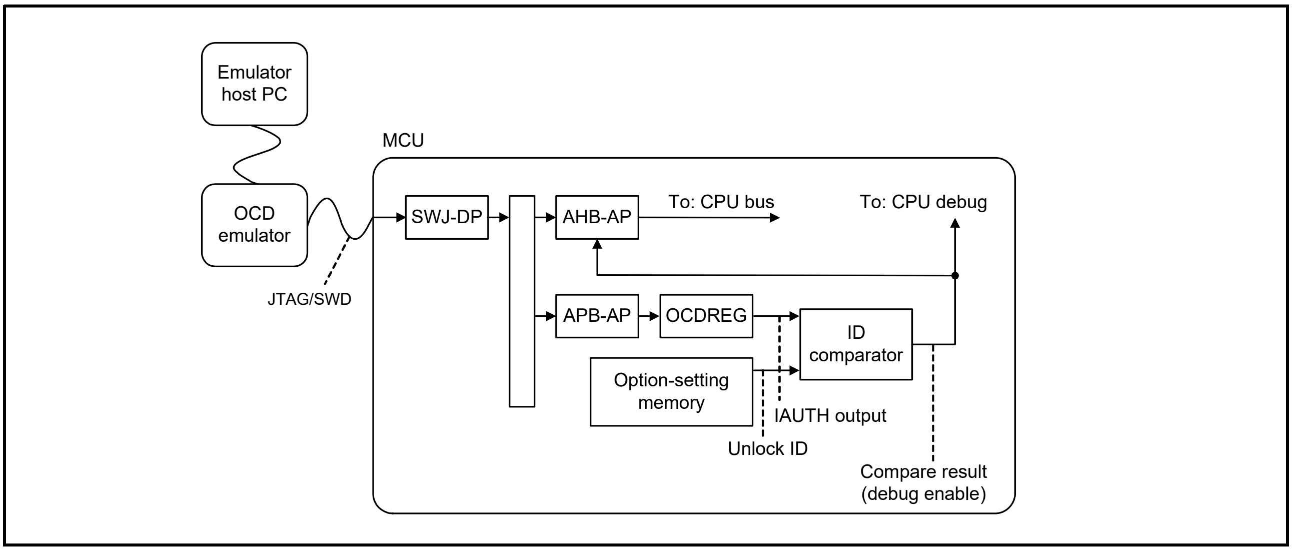


Figure 2.4 Authentication mechanism block diagram

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 it. 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 low power mode

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

2.11.3.2 Changing low power mode while in OCD mode

When the MCU is in OCD mode, the low power mode can be changed. However, system bus access from AHB-AP is prohibited in Software Standby or Snooze mode. Only SWJ-DP, APB-AP, and OCDREG can be accessed from the OCD emulator in these modes. Table 2.14 shows the restrictions.

Table 2.14 Restrictions by mode (1 of 2)

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

Table 2.14 Restrictions by mode (2 of 2)

Active mode	Start OCD emulator connection	Change low power mode	Access AHB-AP and system bus	Access APB-AP and OCDREG
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 Modifying 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 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 required before comparing the OSIS value at cold start.

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

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

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

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

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

(3) When OSIS[127:126] 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 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, then wait until CSDBGPWRUPACK in the same register is asserted.
3. Set up the APB-AP to access OCDREG. The APB-AP is connected to the DAP bus port 1.
4. Write the 128-bit ID code to IAUTH registers 0 to 3 in the OCDREG using the APB-AP.
5. If the 128-bit ID code matches the OSIS value, the AHB-AP is authorized to issue an AHB transaction. The authorization result can be confirmed by the AUTH bit in the MCUSTAT Register or the DbgStatus bit in the AHB-AP Control Status Word Register.
    - When the DbgStatus bit is 1, the 128-bit ID code is 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 the system address space. The AHB-AP is connected to the DAP bus port 0.
7. Start accessing the CPU debug resources using the AHB-AP.

(4) When OSIS[127:126] 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 as when OSIS[127:126] is 10b except for ALeRASE capability.

When IATUH0-3 are ALeRASE in ASCII code, the content of code flash, data flash, and the configuration area are erased at once. See section 43, Flash Memory for details.

ALeRASE sequence

1. Connect the OCD debugger to the MCU through the JTAG or SWD interface.
2. Setup SWJ-DP to access DAP bus. In the set up, the OCD emulator must assert CDBGPWRUPREQ in the SWJDP Control Status Register, then wait until CSDBGPWRUPACK in the same register is asserted.
3. Set the APB-AP to access OCDREG. The APB-AP is connected to 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 (414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFFh), the contents of code flash, data flash, and the configuration area are erased. After that, 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, Details of Operating Modes. Operation starts with the on-chip flash memory 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 Details of Operating Modes

3.2.1 Single-Chip Mode

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

3.2.2 SCI Boot Mode

In this mode, the on-chip flash memory programming routine (SCI boot program), stored in 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 universal asynchronous receiver/transmitter (SCI). For details, see section 43, Flash Memory. The MCU starts 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 43, 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 settings of the MD pin.

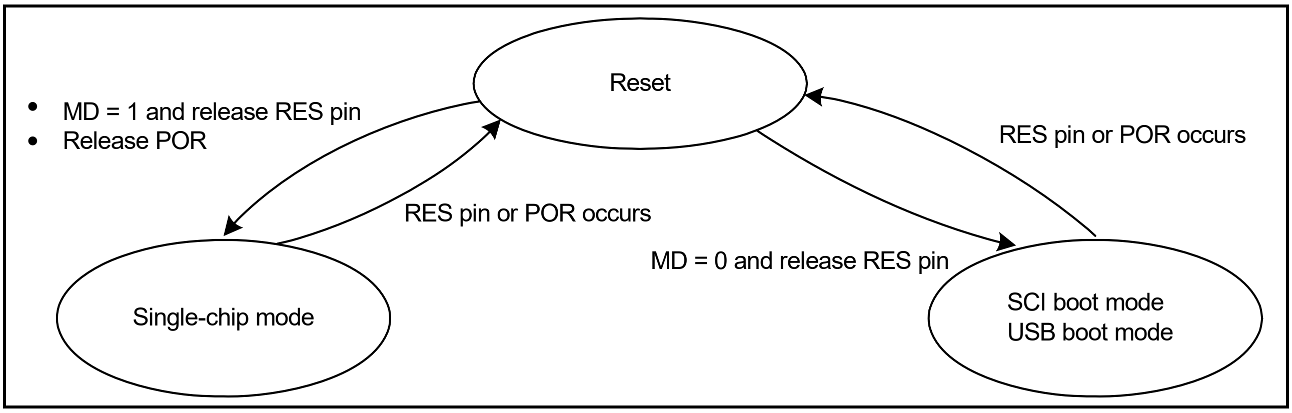
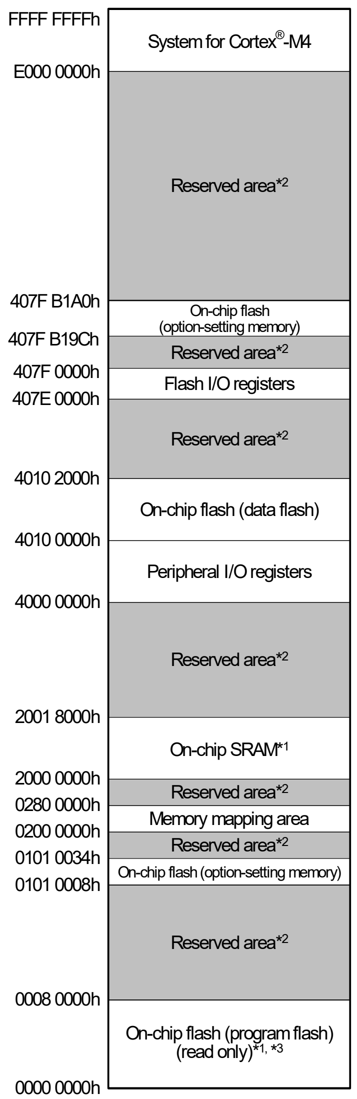


Figure 3.1 Mode-setting pin level and operating mode

4. Address Space

4.1 Overview

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.


FFFF FFFFh
System for Cortex®-M4
E000 0000h
Reserved area*2
407F B1A0h
On-chip flash (option-setting memory)
407F B19Ch
Reserved area*2
407F 0000h
Flash I/O registers
407E 0000h
Reserved area*2
4010 2000h
On-chip flash (data flash)
4010 0000h
Peripheral I/O registers
4000 0000h
Reserved area*2
2001 8000h
On-chip SRAM*1
2000 0000h
Reserved area*2
0280 0000h
Memory mapping area
0200 0000h
Reserved area*2
0101 0034h
On-chip flash (option-setting memory)
0101 0008h
Reserved area*2
0008 0000h
On-chip flash (program flash) (read only)*1, *3
0000 0000h

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

Figure 4.1 Memory map

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 the 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 words 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 bits. Data written to the KEY bits is not saved. These bits are read as 0. The KEY code and MEMMIRADDR 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 words 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 Operation

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 start address of the memory mirror space addresses (0200 0000h) is linked to. Figure 5.2, Figure 5.3, and Figure 5.4 show the operation of the MMF. Figure 5.5 shows the setting procedure of the MMF.

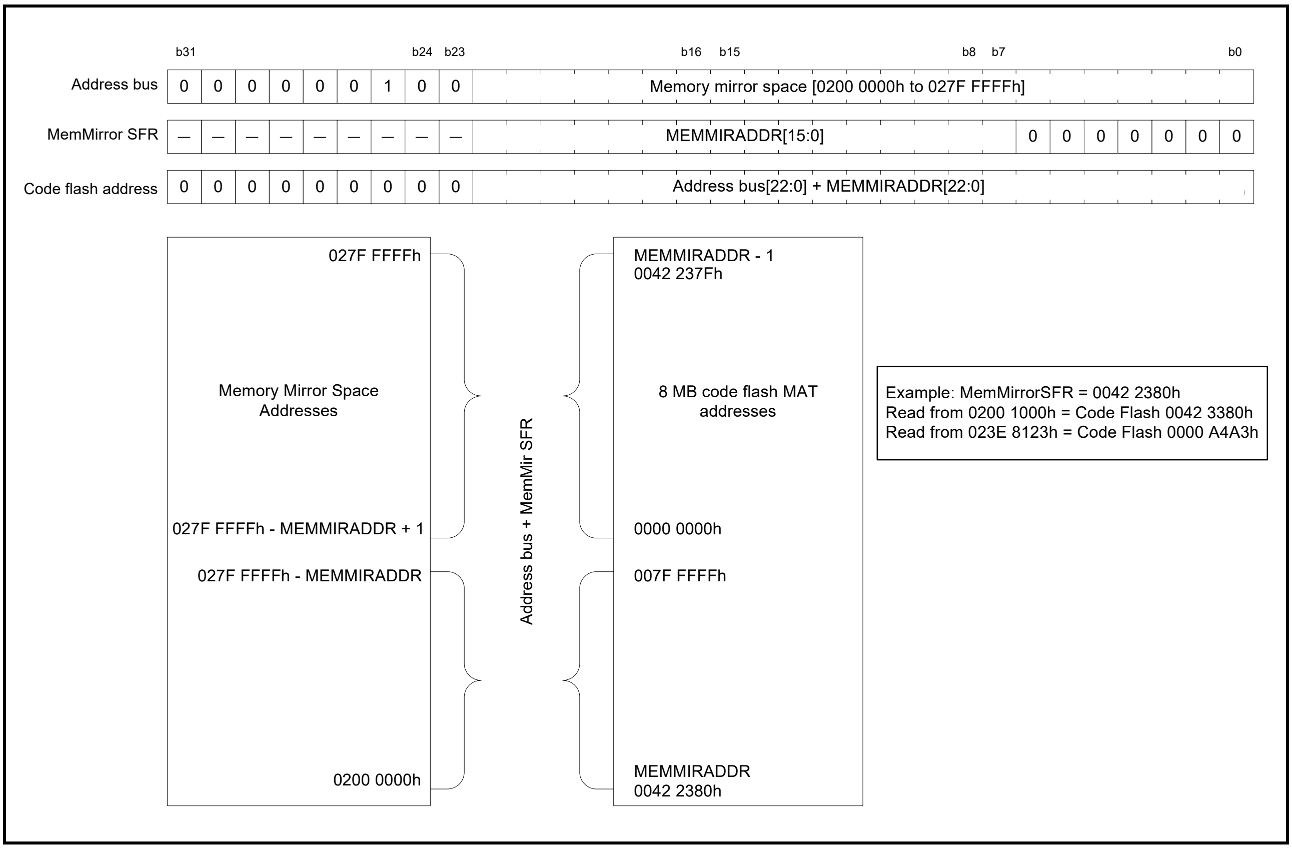
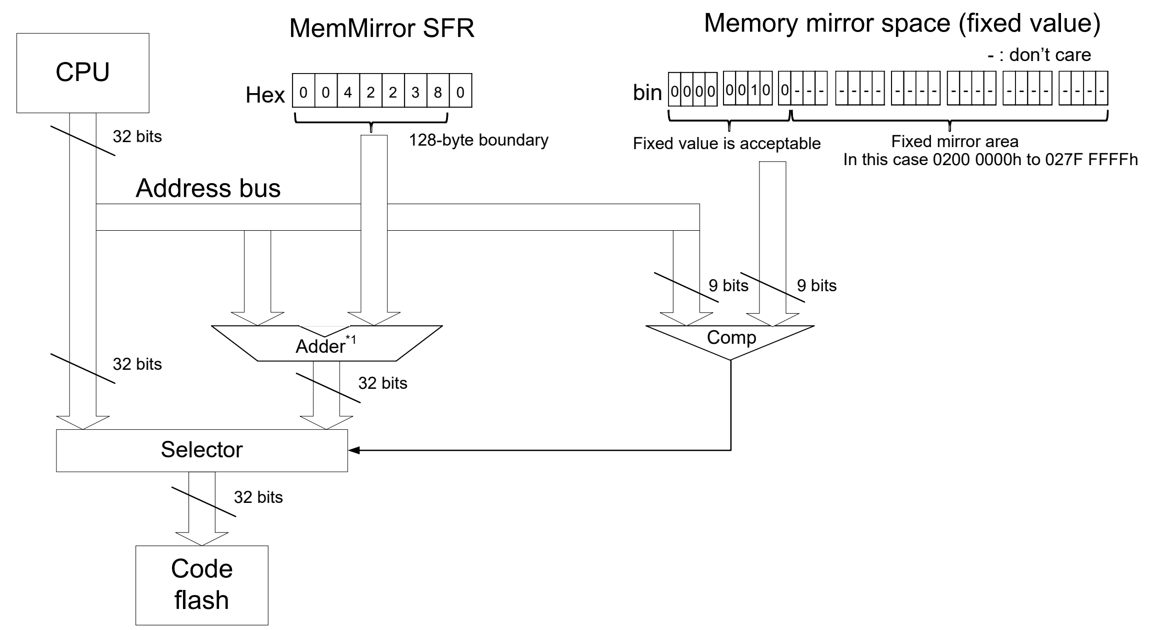

b31
b24
b23
b16
b15
b8
b7
b0
Address bus
0 0 0 0 0 0 1 0 0
Memory mirror space [0200 0000h to 027F FFFFh]
MemMirror SFR
— — — — — — — — —
MEMMIRADDR[15:0]
0 0 0 0 0 0 0
Code flash address
0 0 0 0 0 0 0 0 0
Address bus[22:0] + MEMMIRADDR[22:0]
027F FFFFh
Memory Mirror Space Addresses
027F FFFFh - MEMMIRADDR + 1
027F FFFFh - MEMMIRADDR
0200 0000h
Address bus + MemMir SFR
MEMMIRADDR - 1
0042 237Fh
8 MB code flash MAT addresses
0000 0000h
007F FFFFh
MEMMIRADDR
0042 2380h
Example: MemMirrorSFR = 0042 2380h
Read from 0200 1000h = Code Flash 0042 3380h
Read from 023E 8123h = Code Flash 0000 A4A3h

Figure 5.1 MMF operation


CPU
MemMirror SFR
Hex
0 0 4 2 2 3 8 0
128-byte boundary
Memory mirror space (fixed value)
- : don't care
bin
0 0 0 0 0 0 1 0 0 -
Fixed value is acceptable
Fixed mirror area
In this case 0200 0000h to 027F FFFFh
32 bits
Address bus
9 bits
9 bits
Adder*1
Comp
32 bits
32 bits
Selector
32 bits
Code flash

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.

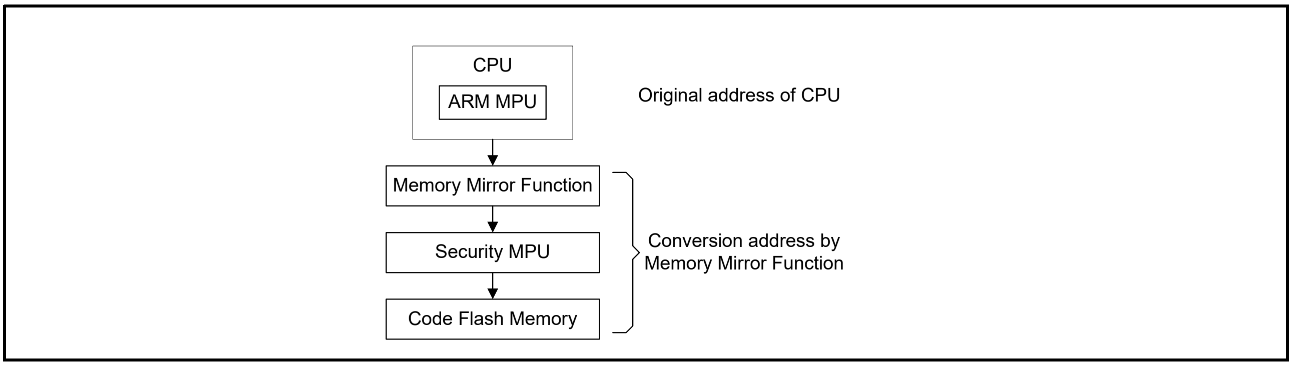

CPU
ARM MPU
Original address of CPU
Memory Mirror Function
Security MPU
Code Flash Memory
Conversion address by Memory Mirror Function

Figure 5.3 MMF address handling

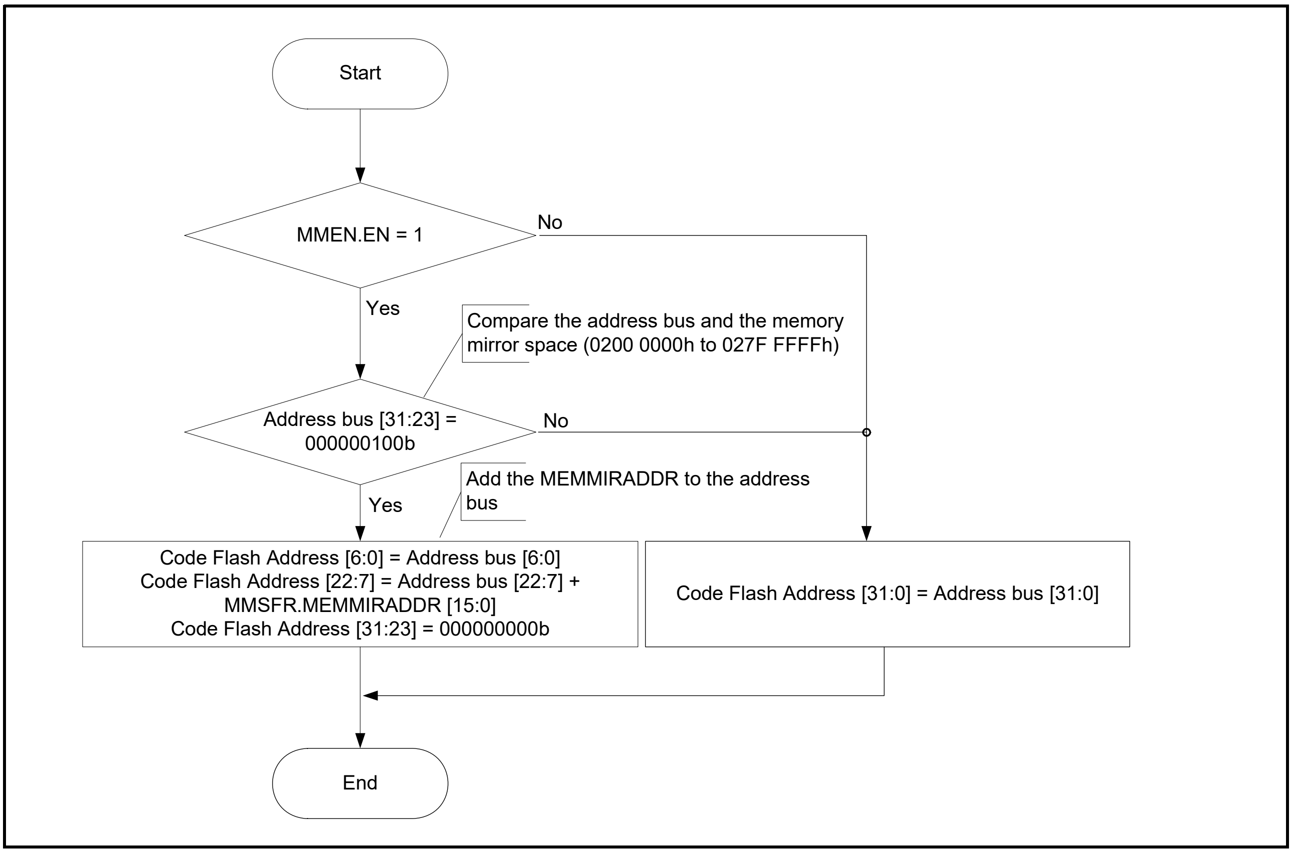

Start
MMEN.EN = 1
No
Yes
Compare the address bus and the memory mirror space (0200 0000h to 027F FFFFh)
Address bus [31:23] = 000000100b
No
Add the MEMMIRADDR to the address bus
Yes
Code Flash Address [6:0] = Address bus [6:0]
Code Flash Address [22:7] = Address bus [22:7] + MMSFR.MEMMIRADDR [15:0]
Code Flash Address [31:23] = 000000000b
Code Flash Address [31:0] = Address bus [31:0]
End

Figure 5.4 MMF operation flow

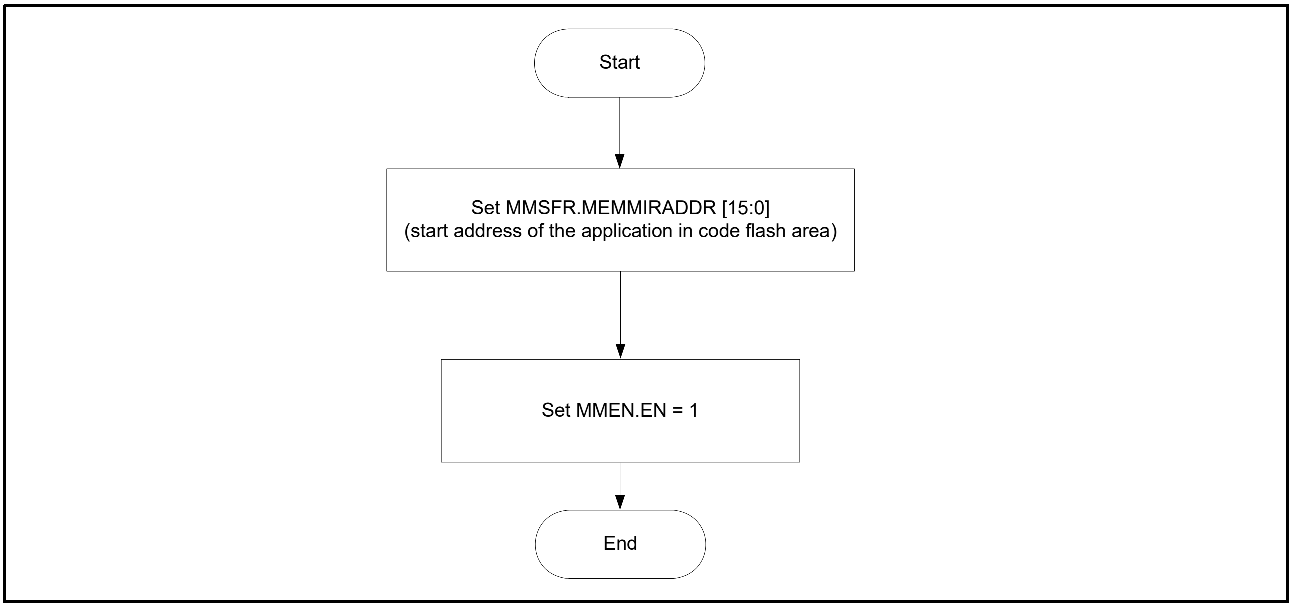

Start
Set MMSFR.MEMMIRADDR [15:0]
(start address of the application in code flash area)
Set MMEN.EN = 1
End

Figure 5.5 MMF setup flow

5.3.2 Setting Example

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

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

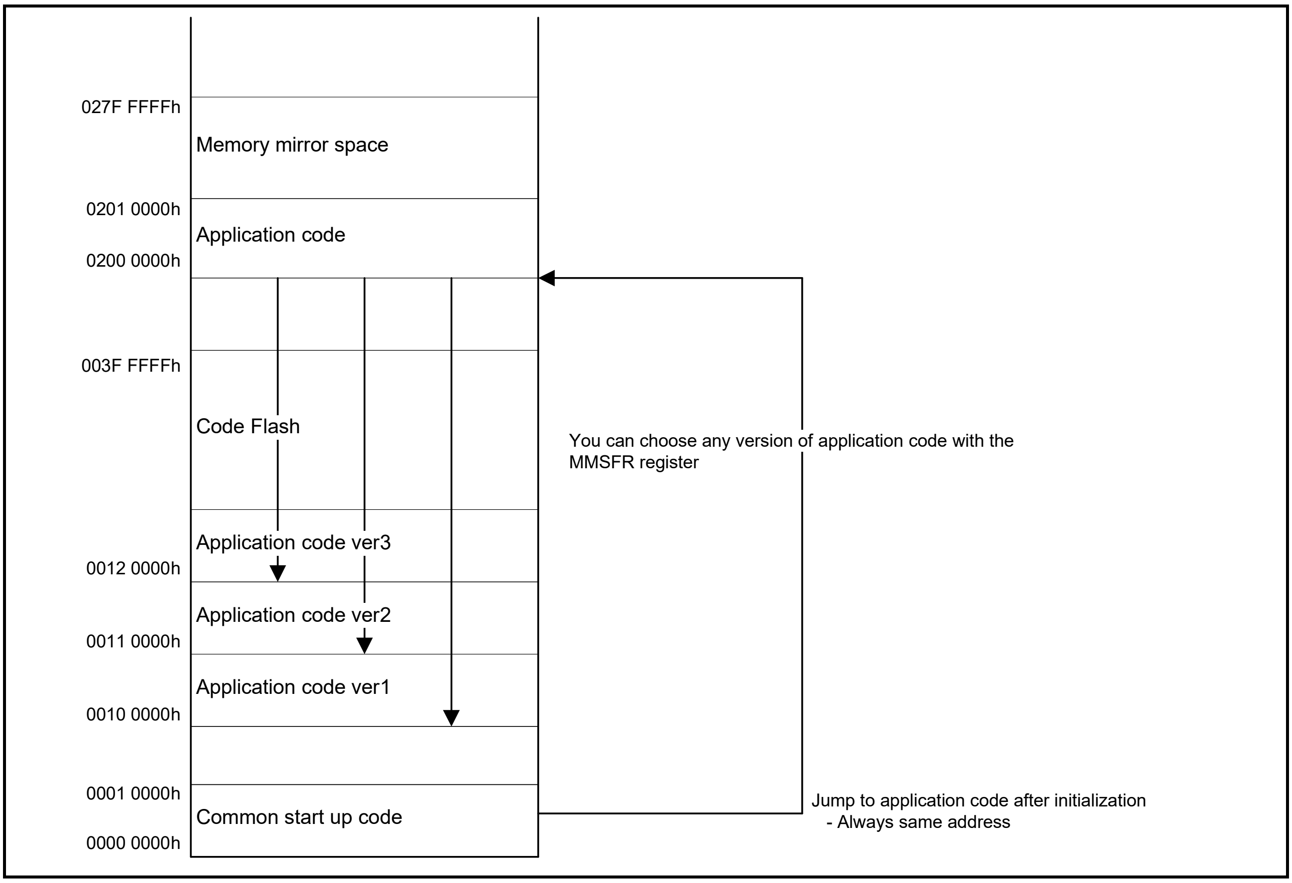

027F FFFFh
Memory mirror space
0201 0000h
Application code
0200 0000h
003F FFFFh
Code Flash
You can choose any version of application code with the MMSFR register
Application code ver3
0012 0000h
Application code ver2
0011 0000h
Application code ver1
0010 0000h
0001 0000h
Common start up code
Jump to application code after initialization
- Always same address
0000 0000h

Figure 5.6 MMF setting example

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

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

Setting 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
- SRAM parity error reset
- SRAM ECC error reset
- Bus master MPU error reset
- Bus slave MPU error reset
- 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]
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
Stack pointer error reset	Stack pointer error detection
Software reset	Register setting (use the Arm® software reset bit AIRCR.SYSRESETREQ)

Note 1. For details on the voltages to be monitored (V_{POR}, V_{det0}, V_{det1} and $V_{DETBATT}$), see section 8, Low Voltage Detection (LVD), section 12., Battery Backup Function, and section 48, 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 (1 of 2)

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	Software reset
Power-On Reset Detect Flag (RSTSR0.PORF)	✓	×	×	×	×	×	×

Table 6.2 Reset detect flags initialized by each reset source (2 of 2)

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	Software reset
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)	✓	✓	✓	×	×	×	×
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)	×	×	×	×	×	×
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 (1 of 2)

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	Software reset
Registers related to the Watchdog Timer	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓	✓	✓	✓	✓	✓
Registers related to the voltage monitor function 1	LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL	✓	✓	✓	✓	✓	×	×
	LVD1CR1/LVD1SR	✓	✓	✓	✓	✓	×	×
Register related to the SOSC	SOSCCR	×	×	×	×	×	×	×
	SOMCR	×	×	×	×	×	×	×

Table 6.3 Module-related registers initialized by each reset source (2 of 2)

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	Software reset
Register related to the LOCO	LOCOCR	×	×	×	×	×	×	×
	LOCOUTCR	×	×	×	×	×	×	×
Register related to the MOSC	MOMCR	✓	✓	✓	✓	✓	✓	✓
Register related to the realtime clock*2		×	×	×	×	×	×	×
Register related to the AGT		×	✓	✓	×	×	✓	×
Register related to the MPU		✓	✓	✓	✓	✓	✓	✓
Pin state (except XCIN/XCOUT pin)		✓	✓	✓	✓	✓	✓	✓
Pin state (XCIN/XCOUT pin)		×	×	×	×	×	×	×
Battery backup	VBTCR1	×	✓	×	×	×	×	×
	VBTCR2, VBTSR, VBTCMPCR, VBTLVDICR, VBTWCTLR, VBTWCH0OTSR, VBTICTLR, VBTOCTLR, VBTWTER, VBTWEGR, VBTWFR	×	×	×	×	×	×	×
	VBTBKRn(n = 0 to 511)	×	×	×	×	×	×	×
Registers other than the above, 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
Registers related to the Watchdog Timer	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓	✓	✓	✓	×
Registers related to the voltage monitor function 1	LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL	×	×	×	×	×	×
	LVD1CR1/LVD1SR	×	×	×	×	×	×
Register related to the SOSC	SOSCCR	×	×	×	×	×	✓*1
	SOMCR	×	×	×	×	×	✓
Register related to the LOCO	LOCOCR	×	×	×	×	×	✓
	LOCOUTCR	×	×	×	×	×	✓
Register related to the MOSC	MOMCR	✓	✓	✓	✓	✓	×
Register related to the realtime clock*2		×	×	×	×	×	×
Register related to the AGT		×	×	×	×	×	×
Register related to the MPU		✓	✓	×	×	×	×
Pin state (except XCIN/XCOUT pin)		✓	✓	✓	✓	✓	×
Pin state (XCIN/XCOUT pin)		×	×	×	×	×	✓
Battery backup	VBTCR1	×	×	×	×	×	×
	VBTCR2, VBTSR, VBTCMPCR, VBTLVDICR, VBTWCTLR, VBTWCH0OTSR, VBTICTLR, VBTOCTLR, VBTWTER, VBTWEGR, VBTWFR	×	×	×	×	×	✓
	VBTBKRn(n = 0 to 511)	×	×	×	×	×	×
Registers other than the above, 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.

The RTC is not initialized by any reset source. 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

		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

		Reset source	
		VBATT_POR	**Other**
LOCO	Enable or disable	Initialized to enable	Continue with the state that was selected before the reset occurred
	Oscillation accuracy	Initialized to accuracy before trimming by LOCOUTCR (accuracy: ±15%)	Continue with the accuracy that was trimmed by LOCOUTCR

When a reset is canceled, reset exception handling starts.

Table 6.6 lists the pin related to the reset function.

Table 6.6 Pin related to reset

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
	—	—	—	—	—	LVD1RF	LVD0RF	PORF
Value after reset:	0	0	0	0	x	x^{*1}	x^{*1}	x^{*1}

x: Undefined

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	—	Reserved	The read value is undefined. The write value should be 0.	R/W
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 Vdet0.

[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 the VCC voltage fell below Vdet1.

[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.

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

Bit	Symbol	Bit name	Description	R/W
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 a SRAM parity error reset occurred.

[Setting condition]

- When a 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 a 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. To successfully reset the MCU, the RES pin must be held low for the power supply stabilization time specified at power-on.

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

For details, see section 48, Electrical Characteristics.

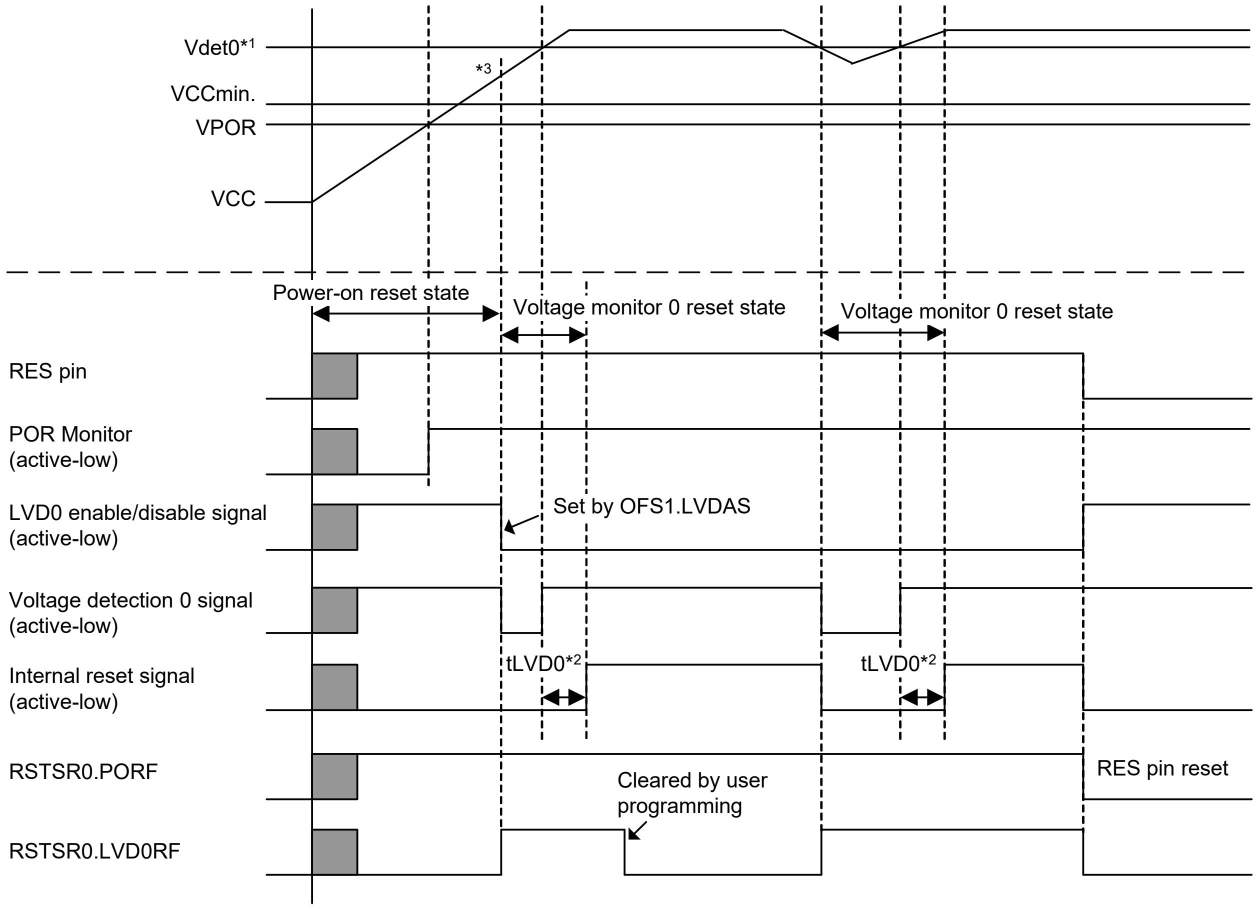
6.3.2 Power-On Reset

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 VPOR 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 a 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 Vdet0, the RSTSR0.LVD0RF flag is set to 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 Vdet0 and the voltage monitor 0 reset time (tLVD0) elapses, the internal reset is canceled and the CPU starts the reset exception handling. The Vdet0 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 example of operations during a power-on reset and voltage monitor 0 reset.


Vdet0*1
VCCmin.
VPOR
VCC
*3
Power-on reset state
Voltage monitor 0 reset state
Voltage monitor 0 reset state
RES pin
POR Monitor
(active-low)
LVD0 enable/disable signal
(active-low)
Set by OFS1.LVDAS
Voltage detection 0 signal
(active-low)
Internal reset signal
(active-low)
tLVD0*2
tLVD0*2
RSTSR0.PORF
Cleared by user
programming
RES pin reset
RSTSR0.LVD0RF

Note: For details on the electrical characteristics, see section 48, Electrical Characteristics.

Note 1. Vdet0 shows a voltage monitor 0 reset detection level, VPOR shows a power-on reset detection level, and VCCmin shows the minimum guaranteed voltage of MCU.

Note 2. tLVD0 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

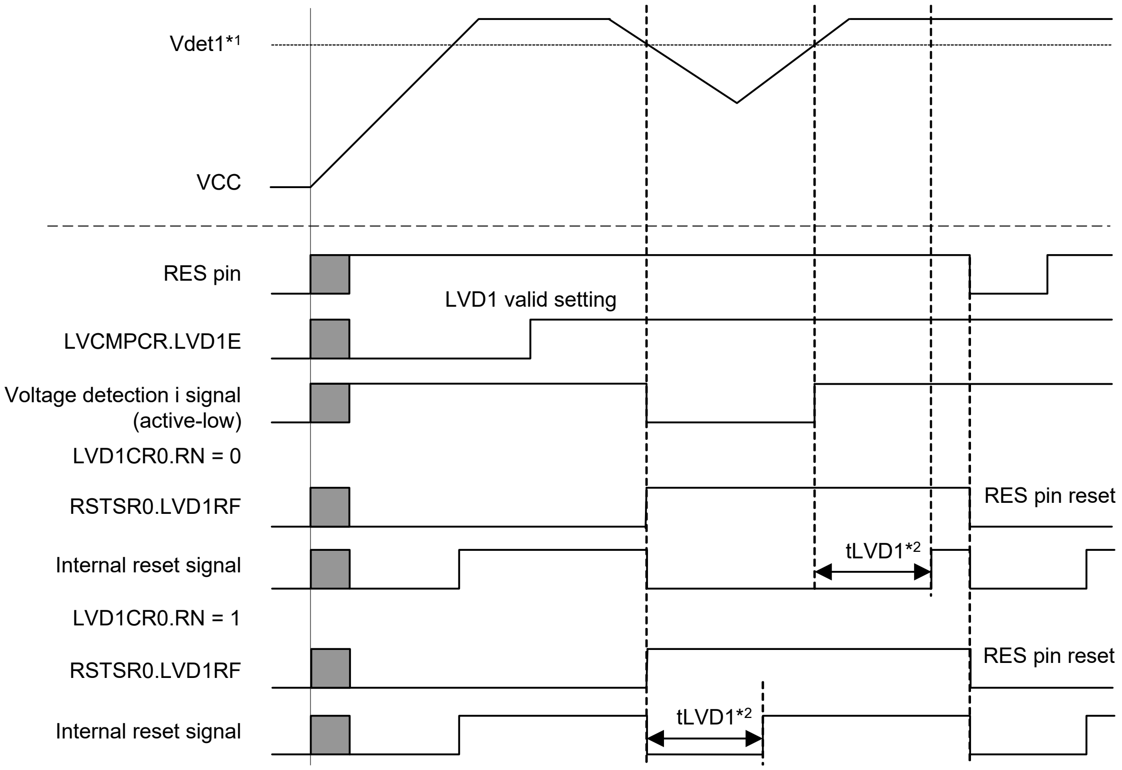
The voltage monitor 0 reset is an internal reset generated by the voltage monitor circuit. If the Voltage Detection 0 Circuit Start (LVDAS) bit in Option Function Select register 1 (OFS1) is 0 (voltage monitor 0 reset is enabled after a reset) and VCC falls below Vdet0, the RSTSR0.LVD0RF flag becomes 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 Vdet0 and the voltage monitor 0 reset time (tLVD0) 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 (LVD1CR0.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 Vdet1.

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 Vdet1, 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 Vdet1. When the LVD1CR0.RN bit is 1 and VCC falls to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling when the LVD1 reset time (tLVD1) elapses.

Detection level Vdet1 can be changed in the Voltage Detection Level Select Register (LVDLVLR).

Figure 6.2 shows example of operations during voltage monitor 1 reset. For details on the voltage monitor 1 reset, see section 8, Low Voltage Detection (LVD).


Vdet1*1
VCC
RES pin
LVD1 valid setting
LVCMPCR.LVD1E
Voltage detection i signal
(active-low)
LVD1CR0.RN = 0
RSTSR0.LVD1RF
RES pin reset
Internal reset signal
tLVD1*2
LVD1CR0.RN = 1
RSTSR0.LVD1RF
RES pin reset
Internal reset signal
tLVD1*2

Note: For details on the electrical characteristics, see section 48, Electrical Characteristics.

Note 1. Vdet1 indicates the detection level of voltage monitor 1 reset

Note 2. tLVD1 indicates the time for voltage monitor 1 reset

Figure 6.2 Example of operation during voltage monitor 1 reset

6.3.4 Independent Watchdog Timer Reset

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

When output of the independent watchdog timer reset is selected, the reset is generated if the IWDT underflows, or if data is written when refresh operation is disabled. When the internal reset time (tRESW2) 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 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 (tRESW2) 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 (tRESW2) 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 a cold/warm start determination operation.

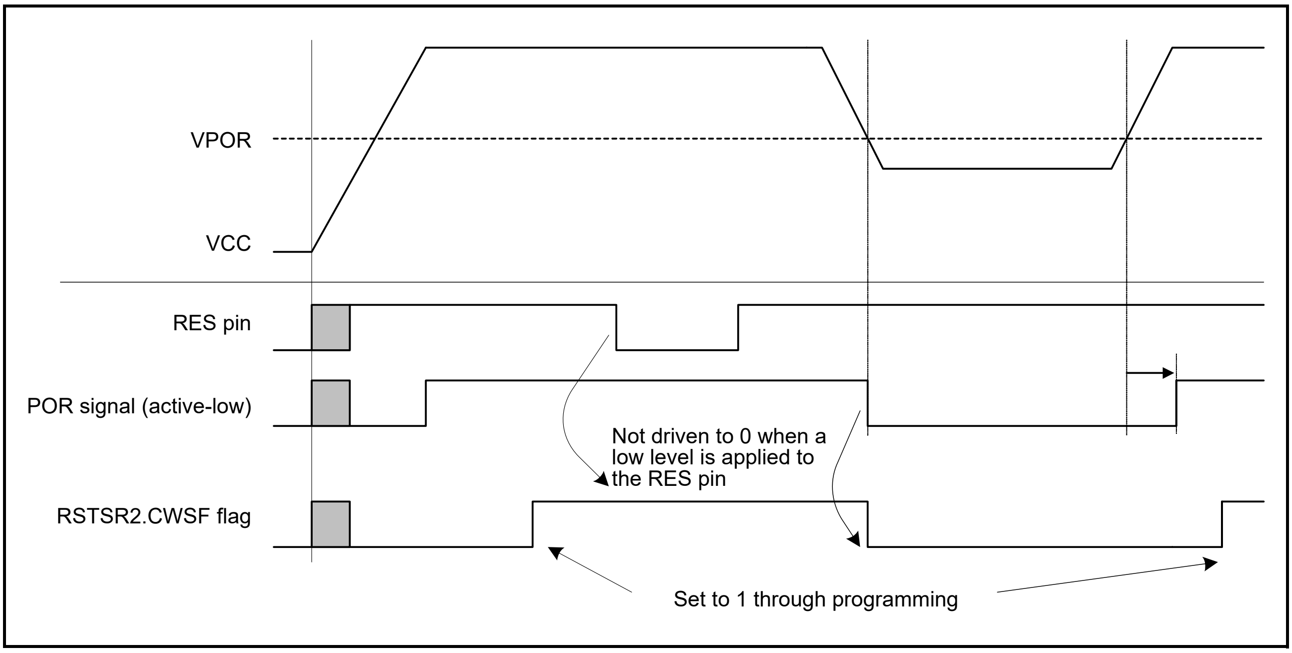
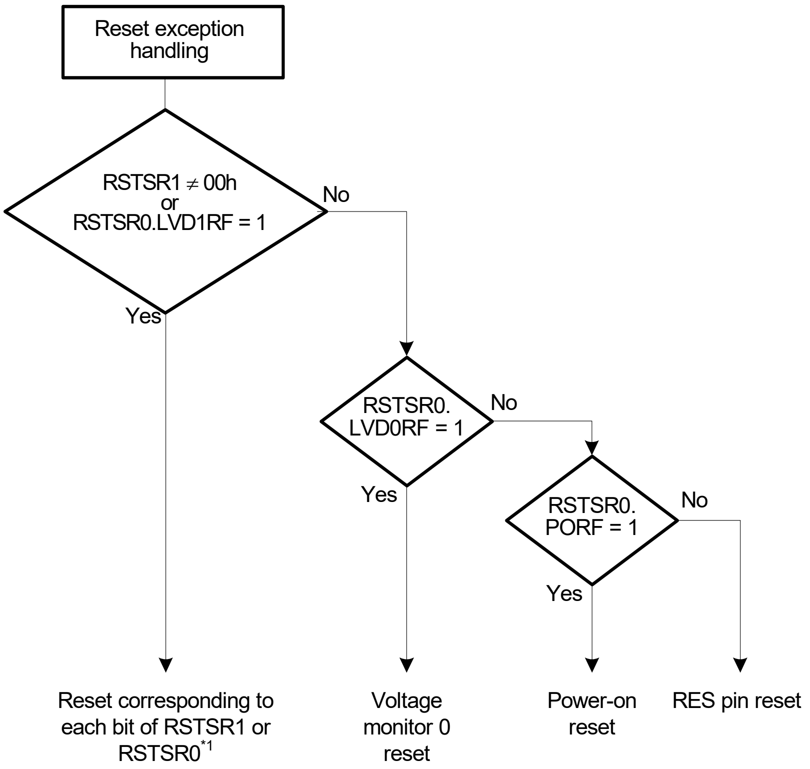

VPOR
VCC
RES pin
POR signal (active-low)
RSTSR2.CWSF flag
Not driven to 0 when a low level is applied to the RES pin
Set to 1 through programming

Figure 6.3 Example of a cold/warm start determination operation

6.3.8 Determination of Reset Generation Source

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


Reset exception handling
RSTSR1 ≠ 00h or RSTSR0.LVD1RF = 1
No
Yes
RSTSR0. LVD0RF = 1
No
Yes
RSTSR0. PORF = 1
No
Yes
Reset corresponding to each bit of RSTSR1 or RSTSR0*1
Voltage monitor 0 reset
Power-on reset
RES pin reset

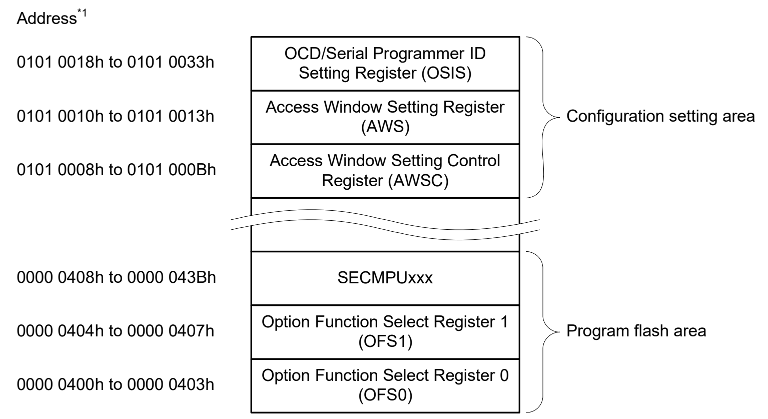
Note 1. If a reset associated with each bit of RSTSR1 or RSTSR0.LVD1RF occurs at the same time, all reset flags are set to 1.

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, and the available methods of setting are different for the two areas. Figure 7.1 shows the option-setting memory area.


Address*1
0101 0018h to 0101 0033h
OCD/Serial Programmer ID Setting Register (OSIS)
0101 0010h to 0101 0013h
Access Window Setting Register (AWS)
0101 0008h to 0101 000Bh
Access Window Setting Control Register (AWSC)
Configuration setting area
0000 0408h to 0000 043Bh
SECMPUxxx
0000 0404h to 0000 0407h
Option Function Select Register 1 (OFS1)
0000 0400h to 0000 0403h
Option Function Select Register 0 (OFS0)
Program flash area

Note 1. The option-setting memory must be allocated to the flash user 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 Selects	b3 b2 0 0: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh).	R

Bit	Symbol	Bit name	Description	R/W
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
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 Selects)

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.

For details, see section 27, Independent Watchdog Timer (IWDT).

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

The IWDTCKS[3:0] bits specify the division ratio of the prescaler for dividing the frequency of the clock for the IWDT as 1/1, 1/16, 1/32, 1/64, 1/128, 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.

For details, see section 27, Independent Watchdog Timer (IWDT).

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

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

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

For details, see section 27, Independent Watchdog Timer (IWDT).

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

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

For details, see section 27, Independent Watchdog Timer (IWDT).

IWDTRSTIRQS bit (IWDT Reset Interrupt Request Select)

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

For details, see section 27, Independent Watchdog Timer (IWDT).

IWDTSTPCTL bit (IWDT Stop Control)

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

For details, see section 27, Independent Watchdog Timer (IWDT).

WDTSTRT bit (WDT Start Mode Select)

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

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

The WDTTOPS[1:0] bits specify the timeout period, the time it takes for the down counter to underflow, as 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set in the WDTCKS[3:0] bits. The number of 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.

For details, see section 26, Watchdog Timer (WDT).

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

The WDTCKS[3:0] bits specify the division ratio of the prescaler 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 4096 to 134217728 PCLKB cycles.

For details, see section 26, Watchdog Timer (WDT).

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

For details, see section 26, Watchdog Timer (WDT).

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

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

For details, see section 26, Watchdog Timer (WDT).

WDTRSTIRQS bit (WDT Reset Interrupt Request Select)

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

For details, see section 26, Watchdog Timer (WDT).

WDTSTPCTL bit (WDT Stop Control)

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

For details, see section 26, Watchdog Timer (WDT).

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]			—	—	—	HOCOEN	—	—	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

Bit	Symbol	Bit name	Description	R/W
b5 to b3	VDSEL1[2:0]	Voltage Detection 0 Level Select	b5 b3 0 0 0: Settings prohibited 0 0 1: Selects 2.82 V 0 1 0: Selects 2.51 V 0 1 1: Selects 1.90 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 bits 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 set immediately 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 shows the registers related to the MPU function. For details, see section 16, Memory Protection Unit (MPU).

The security MPU is disabled on erasure of flash memory. If improper data is written to the 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 (1 of 2)

Register name	Symbol	Function	Address	Size (byte)
Security MPU Program Counter Start Address Register 0	SECMPU PCS0	Specifies the security fetch region of flash or SRAM	0000 0408h	4
Security MPU Program Counter End Address Register 0	SECMPU PCE0	Specifies the security fetch region of flash or SRAM	0000 040Ch	4

Table 7.1 MPU registers (2 of 2)

Register name	Symbol	Function	Address	Size (byte)
Security MPU Program Counter Start Address Register 1	SECMPU PCS1	Specifies the security fetch region of flash or SRAM	0000 0410h	4
Security MPU Program Counter End Address Register 1	SECMPU PCE1	Specifies the security fetch region of flash or SRAM	0000 0414h	4
Security MPU Region 0 Start Address Register	SECMPU S0	Specifies the secure program and flash data	0000 0418h	4
Security MPU Region 0 End Address Register	SECMPU E0	Specifies the secure program and flash data	0000 041Ch	4
Security MPU Region 1 Start Address Register	SECMPUS1	Specifies the secure data of SRAM	0000 0420h	4
Security MPU Region 1 End Address Register	SECMPUE1	Specifies the secure data of SRAM	0000 0424h	4
Security MPU Region 2 Start Address Register	SECMPUS2	Specifies the secure data of security functions	0000 0428h	4
Security MPU Region 2 End Address Register	SECMPUE2	Specifies the secure data of security functions	0000 042Ch	4
Security MPU Region 3 Start Address Register	SECMPUS3	Specifies the secure data of security functions	0000 0430h	4
Security MPU Region 3 End Address Register	SECMPUE3	Specifies the secure data of security functions	0000 0434h	4
Security MPU Access Control Register	SECMPU AC	Security enabled/disabled region is specified	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: First 8-KB area (0000 0000h to 0000 1FFFh) and second 8-KB area (0000 2000h to 0000 3FFFh) are exchanged 1: 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 the temporary boot swap control. When this bit is set to 0, it cannot be changed to 1. 0: Executing the configuration setting command for programming the access window (FAWE[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]*1	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]*1	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 acceptable programming and erasure region defined by the access window. The block address specifies the first address of the block and consists of the address bits [21:10].	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] 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[11:0] 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.

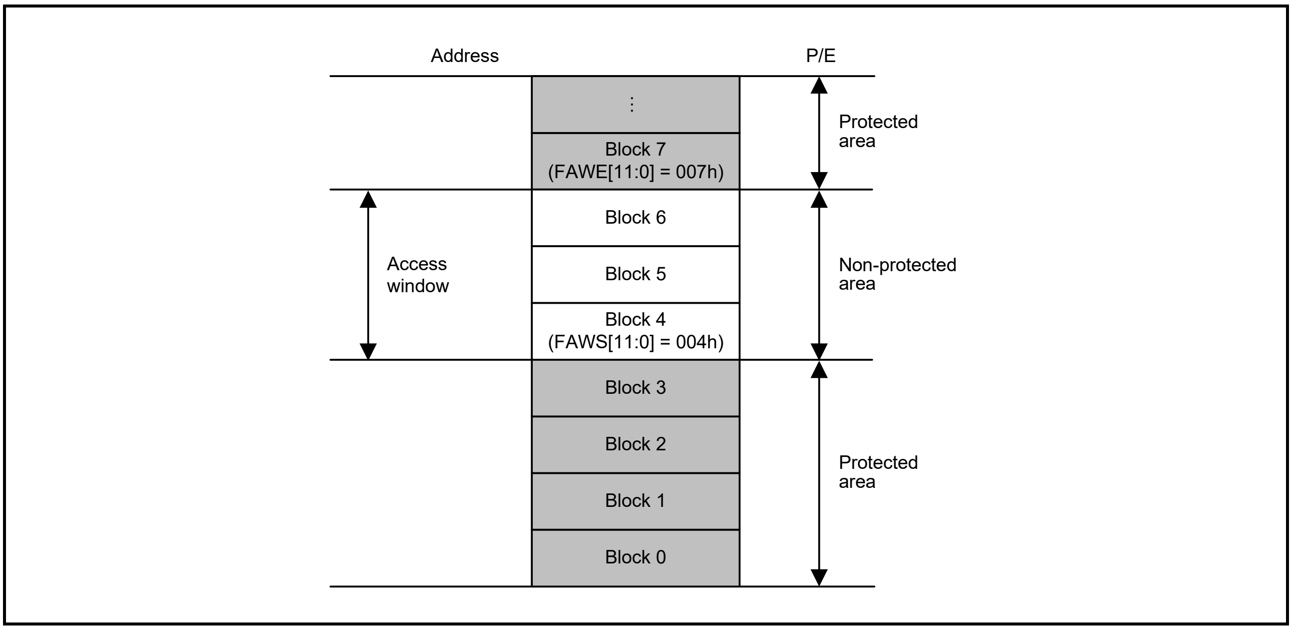

Address
P/E
⋮
Block 7
(FAWE[11:0] = 007h)
Protected area
Access window
Block 6
Block 5
Block 4
(FAWS[11:0] = 004h)
Non-protected area
Block 3
Block 2
Block 1
Block 0
Protected area

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. Use this register to check whether a code transmitted from the OCD/serial programmer matches the ID code in the option-setting memory. When the ID codes match, connection of the OCD/serial programmer is permitted, if not, connection with the OCD/serial programmer is not possible. The OSIS register must be set in 32-bit words.

Address(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 hold the ID for use in ID authentication for the OCD/serial programmer.

ID code bit [127] and bit [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

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 are FFh)	Protection disabled	The ID code is not checked, the ID code always matches, and the connection to the programmer or on-chip debugger is permitted
	Bit [127] = 1, bit [126] = 1, and at least one of the 16 bytes is not FFh	Protection enabled	Matching ID code = authentication is complete and connection to the programmer or the on-chip debugger is permitted. Mismatching ID code = 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_FFFFh), the content of the user flash (code and data) area, and the 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 is complete and connection to the programmer or the on-chip debugger is permitted. Mismatching ID code = 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 to 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.

For details of the programming command, the configuration setting command, and the startup area select function, see section 43, Flash Memory.

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

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

The MCU provides two setting procedures:

- Read the data allocated as described in section 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 as allocated 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 of 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

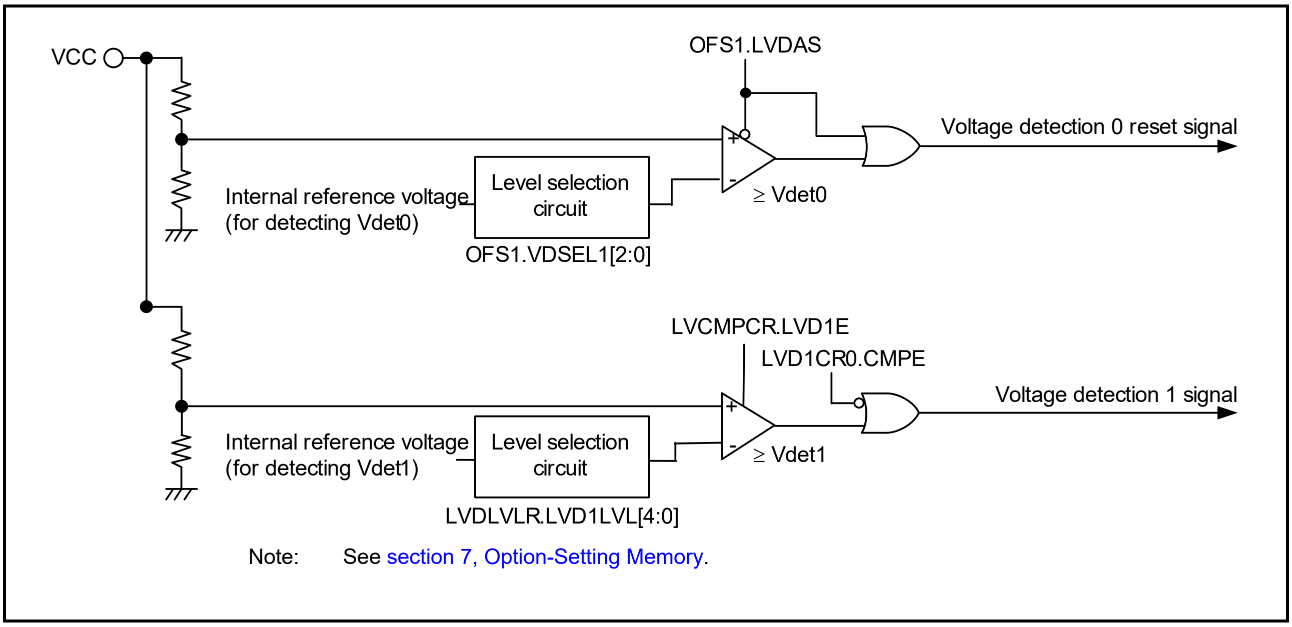
The Low Voltage Detection (LVD) module monitors the voltage level input to the VCC pin, and the detection level can be selected using a software program. The LVD module consists of two separate voltage level detectors, 0 and 1, which measure the voltage level input to the VCC pin. LVD voltage detection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

Each voltage level detector has a voltage monitor associated with it, for example voltage monitor 0, and 1. Voltage monitor registers are used to 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 and 1, Figure 8.2 shows a block diagram of the voltage monitor 1 interrupt/reset circuit.

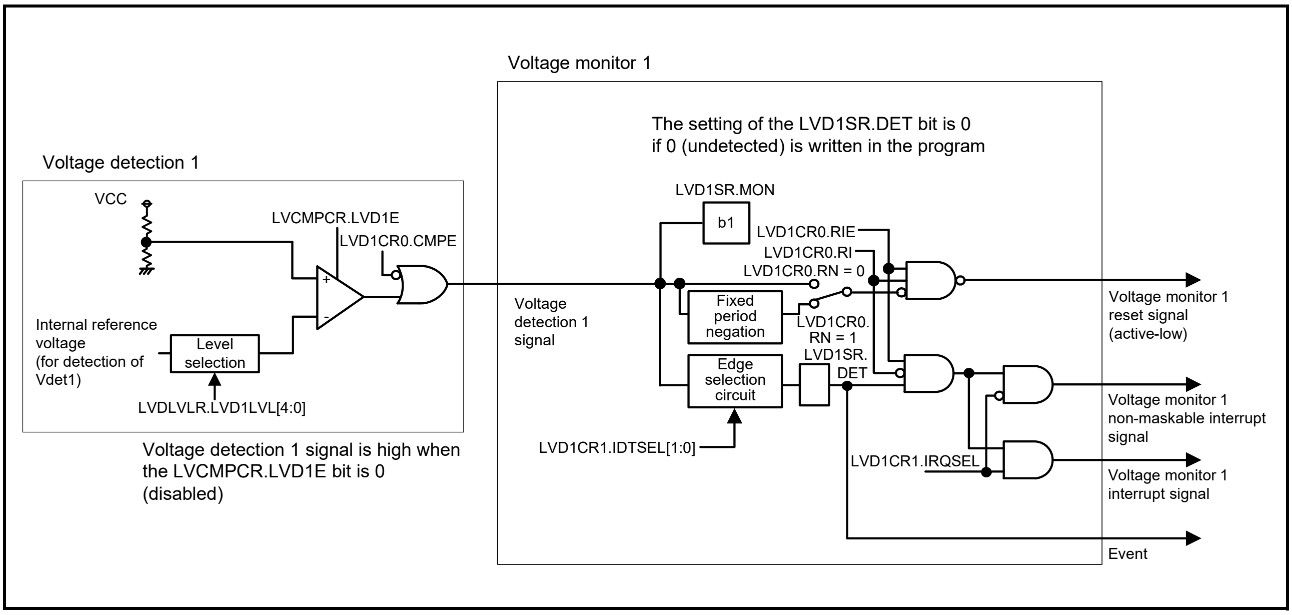
Table 8.1 LVD specifications

Parameter		Voltage monitor 0	Voltage monitor 1
VCC monitoring	Monitored voltage	Vdet0	Vdet1
	Detected event	Voltage falls below Vdet0	Voltage rises or falls past Vdet1
	Detection voltage	Selectable from three different levels in the OFS1.VDSEL1[2:0] bits	Selectable from 10 different levels in the LVDLVLR.LVD1LVL[4:0] bits
	Monitor flag	None	LVD1SR.MON flag: Monitors whether voltage is higher or lower than Vdet1
			LVD1SR.DET flag: Vdet1 passage detection
Process on voltage detection	Reset	Voltage monitor 0 reset	Voltage monitor 1 reset
		Reset when Vdet0 > VCC CPU restart after specified time with VCC > Vdet0	Reset when Vdet1 > VCC CPU restart timing selectable: after specified time with VCC > Vdet1 or Vdet1 > VCC
	Interrupt	No interrupt	Voltage monitor 1 interrupt
			Non-maskable interrupt or maskable interrupt selectable
			Interrupt request issued when Vdet1 > VCC or VCC > Vdet1
Event linking		None	Available Output of event signals on detection of Vdet1 crossings


VCC
OFS1.LVDAS
Voltage detection 0 reset signal
Level selection circuit
Internal reference voltage (for detecting Vdet0)
≥ Vdet0
OFS1.VDSEL1[2:0]
LVCMPCR.LVD1E
LVD1CR0.CMPE
Voltage detection 1 signal
Internal reference voltage (for detecting Vdet1)
Level selection circuit
≥ Vdet1
LVDLVLR.LVD1LVL[4:0]

Note: See section 7, Option-Setting Memory.

Figure 8.1 Voltage detection 0 and 1 block diagram


Voltage monitor 1
Voltage detection 1
VCC
LVCMPCR.LVD1E
LVD1CR0.CMPE
Internal reference voltage (for detection of Vdet1)
Level selection
LVDLVLR.LVD1LVL[4:0]
LVD1SR.MON
b1
LVD1CR0.RIE
LVD1CR0.RI
LVD1CR0.RN = 0
Voltage detection 1 signal
Fixed period negation
LVD1CR0. RN = 1
LVD1SR. DET
Edge selection circuit
LVD1CR1.IDTSEL[1:0]
LVD1CR1.IRQSEL
Voltage monitor 1 reset signal (active-low)
Voltage monitor 1 non-maskable interrupt signal
Voltage monitor 1 interrupt signal
Event

The setting of the LVD1SR.DET bit is 0 if 0 (undetected) is written in the program

Voltage detection 1 signal is high when the LVCMPCR.LVD1E bit is 0 (disabled)

Figure 8.2 Voltage monitor 1 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 ≥ Vdet1 (rise) is detected 0 1: When VCC < Vdet1 (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 value of the NMIER.LVD1EN bit 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: Vdet1 passage detected.	R(/W) *1
b1	MON	Voltage Monitor 1 Signal Monitor Flag	0: VCC < Vdet1 1: VCC ≥ Vdet1 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).

Set the DET flag 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.

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 Circuit Control Register (LVCMPCR)

Address(es): SYSTEM.LVCMPCR 4001 E417h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	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: Voltage detection 1 circuit disabled 1: Voltage detection 1 circuit enabled.	R/W
b7, b6	—	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.

8.2.4 Voltage Detection Level Select Register (LVDLVLR)

Address(es): SYSTEM.LVDLVLR 4001 E418h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LVD1LVL[4:0]				
Value after reset	0	0	0	0	0	1	1	1

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 1 0 0: 3.10 V (Vdet1_4) 0 0 1 0 1: 3.00 V (Vdet1_5) 0 0 1 1 0: 2.90 V (Vdet1_6) 0 0 1 1 1: 2.79 V (Vdet1_7) 0 1 0 0 0: 2.68 V (Vdet1_8) 0 1 0 0 1: 2.58 V (Vdet1_9) 0 1 0 1 0: 2.48 V (Vdet1_A) 0 1 0 1 1: 2.20 V (Vdet1_B) 0 1 1 0 0: 1.96 V (Vdet1_C) 0 1 1 0 1: 1.86 V (Vdet1_D) Other settings are 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 bit (voltage detection 1 circuit disable) is 0.

8.2.5 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.	
b5, 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 Vdet1 passage 1: Enable voltage monitor 1 reset when the voltage falls to and below Vdet1.	R/W
b7	RN	Voltage Monitor 1 Reset Negate Select	0: Negate after a stabilization time (tLVD1) when VCC > Vdet1 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 1 reset 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 after assertion of the LVD1 reset signal), set the MOCOCR.MCSTP bit to 0 (the MOCO operates). In addition, 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 after VCC > Vdet1 is detected). Do not set the RN bit to 1 (negation follows a stabilization time after assertion of the LVD1 reset signal) when this is the case.

8.3 VCC Input Voltage Monitor

8.3.1 Monitoring Vdet0

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

8.3.2 Monitoring Vdet1

Table 8.2 shows the procedure to set up monitoring against Vdet1. After the settings are complete, the comparison results from voltage monitor 1 can be monitored with the LVD1SR.MON flag.

Table 8.2 Procedures to set up monitoring against Vdet1 (1 of 2)

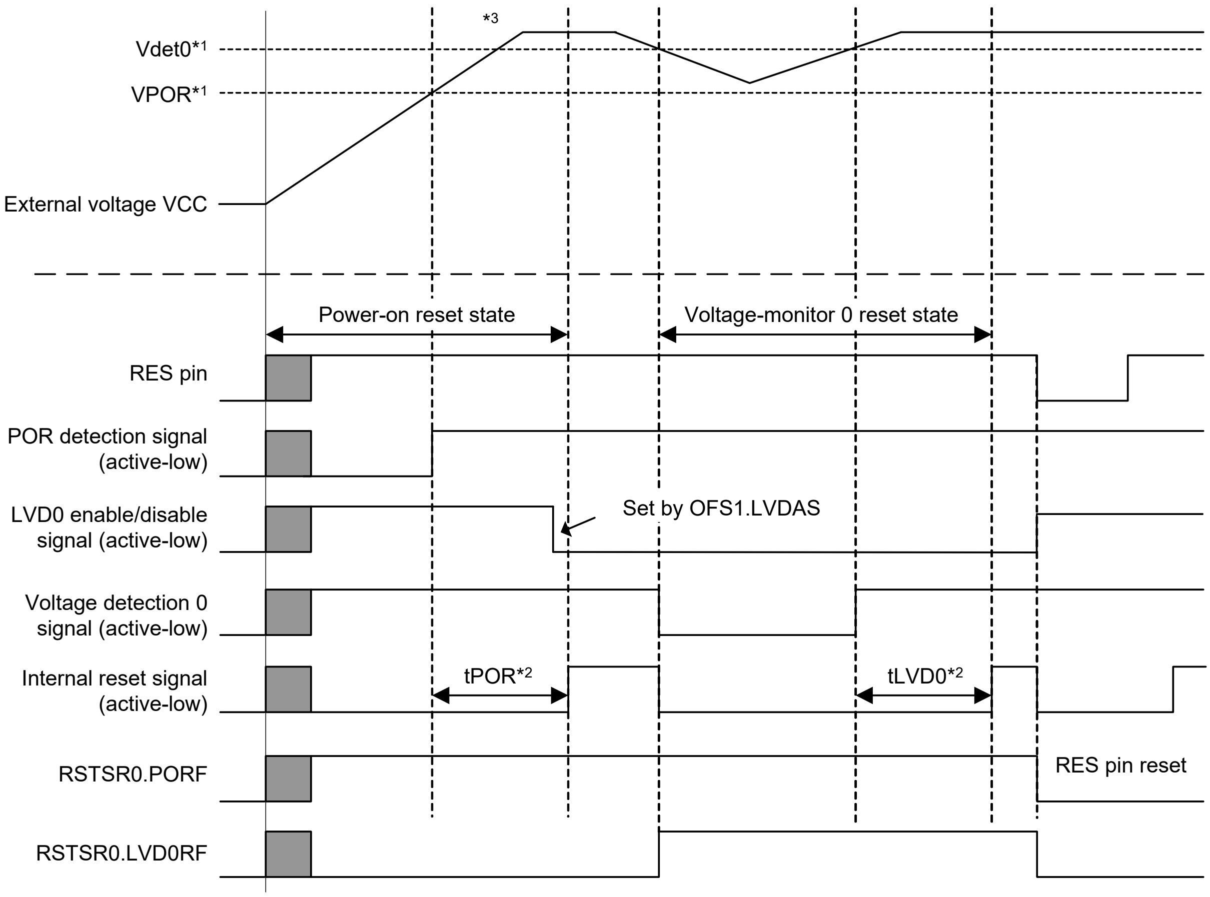
Step		Monitoring the results of comparison from voltage monitor 1
Setting the voltage detection 1 circuit	1	Set LVCMPCR.LVD1E = 0 to disable voltage detection 1 before writing to LVDLVLR register.
	2	Select the detection voltage in the LVDLVLR.LVD1LVL[4:0] bits.
	3	Set LVCMPCR.LVD1E = 1 to enable voltage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

Table 8.2 Procedures to set up monitoring against Vdet1 (2 of 2)

Step		Monitoring the results of comparison from voltage monitor 1
Enabling output	5	Set LVD1CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 1.

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


*3
Vdet0*1
VPOR*1
External voltage VCC
Power-on reset state
Voltage-monitor 0 reset state
RES pin
POR detection signal (active-low)
LVD0 enable/disable signal (active-low)
Set by OFS1.LVDAS
Voltage detection 0 signal (active-low)
Internal reset signal (active-low)
tPOR*2
tLVD0*2
RSTSR0.PORF
RES pin reset
RSTSR0.LVD0RF

Note: For details of the electrical characteristics, see section 48, Electrical Characteristics.
Note 1. VPOR indicates the detection level for a power-on reset and Vdet0 indicates the detection level for a voltage monitor 0 reset.
Note 2. tPOR indicates the period of a power-on reset and tLVD0 indicates the period of a voltage monitor 0 reset.
Note 3. At power-on, raise VCC to the minimum guaranteed voltage before the POR reset is released.

Figure 8.3 Example of voltage monitor 0 reset operation

8.5 Interrupt and Reset from Voltage Monitor 1

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

Table 8.3 shows the procedure for setting bits related to the voltage monitor 1 interrupt/reset so that voltage monitoring operates. Table 8.4 shows the procedure for setting bits related to the voltage monitor 1 interrupt/reset so that voltage monitoring stops. Figure 8.4 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 mode, set up the circuit with the following procedures.

(1) Setting in Software Standby mode

- When VCC > Vdet1 is detected, negate the voltage monitor 1 reset signal (LVD1CR0.RN = 0) following a stabilization time.

Table 8.3 Procedure 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 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 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 48, Electrical Characteristics.
Note 2. Step 8 is not required if only the ELC event signal is to be output.

Table 8.4 Procedure for setting bits related to the voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitor stops

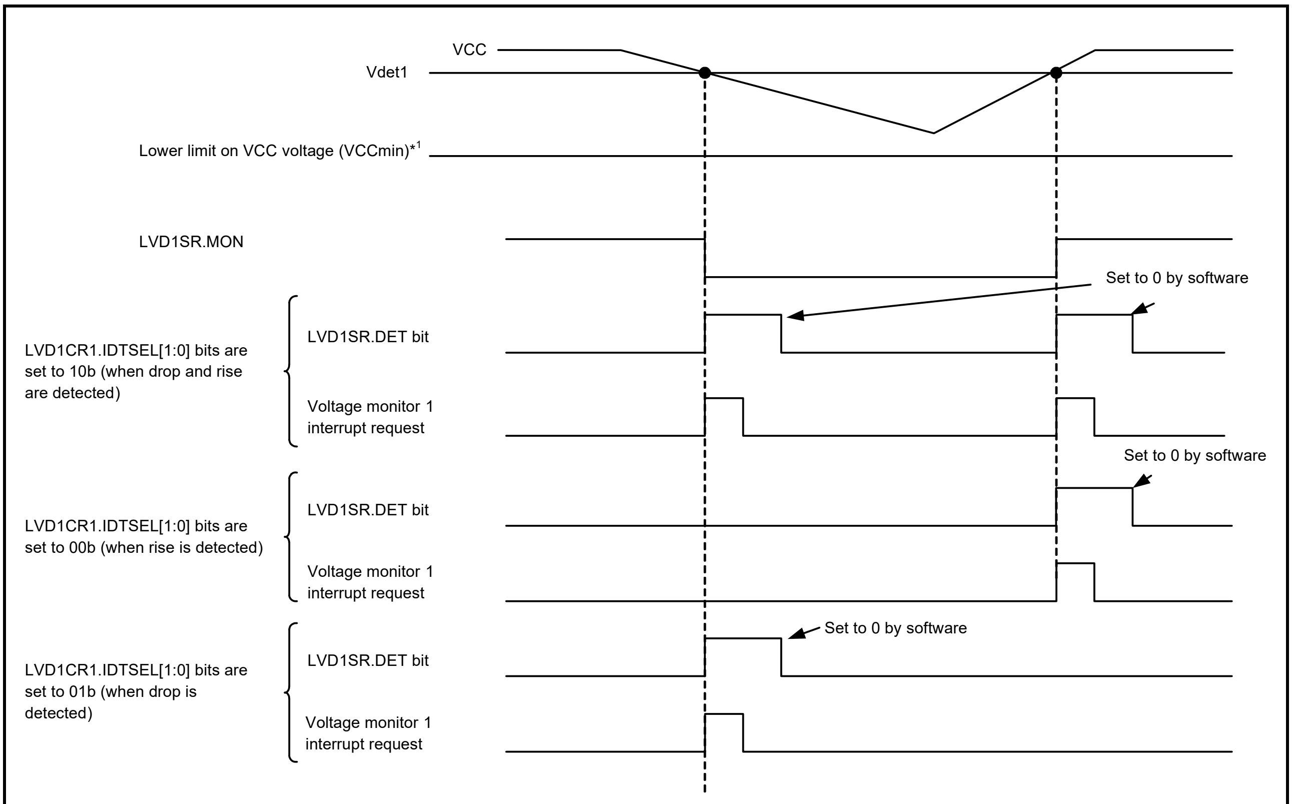
Step		Voltage monitor 1 interrupt (voltage monitor 1 ELC event output), voltage monitor 1 reset
Settings to stop enabling of output	1	Set LVD1CR0.CMPE = 0 to disable output of the comparison results from voltage monitor 1.
	2	Set LVD1CR0.RIE = 0 to disable the voltage monitor 1 interrupt or reset.*1
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 8.4 shows an example of the voltage monitor 1 interrupt operation.


VCC
Vdet1
Lower limit on VCC voltage (VCCmin)*1
LVD1SR.MON
Set to 0 by software
LVD1CR1.IDTSEL[1:0] bits are set to 10b (when drop and rise are detected)
LVD1SR.DET bit
Voltage monitor 1 interrupt request
Set to 0 by software
LVD1CR1.IDTSEL[1:0] bits are set to 00b (when rise is detected)
LVD1SR.DET bit
Voltage monitor 1 interrupt request
Set to 0 by software
LVD1CR1.IDTSEL[1:0] bits are set to 01b (when drop is detected)
LVD1SR.DET bit
Voltage monitor 1 interrupt request

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

Figure 8.4 Voltage monitor 1 interrupt operation example

8.6 Event Link Output

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

(1) Vdet1 Crossing Detection Event

The LVD outputs the event signal when it detects that the voltage has passed the Vdet1 voltage while both the voltage detection 1 circuit and the voltage monitor 1 circuit comparison result output are enabled.

8.6.1 Interrupt Handling and Event Linking

The LVD provides bits to individually enable or disable the voltage monitor 1 interrupt. When an interrupt source is generated and the interrupt is enabled by the interrupt enable bit, the interrupt signal (LVD1CR0.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 interrupt in Software Standby mode. The event signals for the ELC in Software Standby mode are output as follows:

- When a Vdet1 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 Vdet1 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 Vdet1 detection flag

9. Clock Generation Circuit

9.1 Overview

The MCU provides a clock generation circuit.

Table 9.1 and Table 9.2 list the clock generation circuit specifications. Figure 9.1 shows a 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 3.6 V) • 1 MHz to 8 MHz (up to 2.4 V)
	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)
	Output pulse frequency division ratio	Selectable from 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
Bluetooth-dedicated clock oscillator	Resonator frequency	32 MHz
	additional circuit: crystal	Available
	Connection pins : XTAL1_RF, and XTAL2_RF	
Bluetooth-dedicated low speed clock (BLELOCO)	Oscillation frequency	32.768 kHz
	User trimming	Not available

Table 9.2 Clock Generation Circuit Specifications for the internal clocks

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 module (SPI, SCI, SCE5, CRC, 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 module (DAC12, IIC, DOC, CAC, CAN, AGT, POEG, CTSU, ELC, I/O ports, RTC, WDT, IWDT, ADC14, KINT, USBFS, ACMPLP, 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
Flash interface 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
USB clock (UCLK)	HOCO*1/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
Bluetooth-dedicated clock (BLECK)	Bluetooth-dedicated clock oscillator	BLE	32 MHz
Bluetooth-dedicated low speed clock (BLELOCO)	Bluetooth-dedicated low-speed on-chip oscillator	BLE	32.768 kHz

Note: Restrictions on setting the clock frequency: ICLK ≥ PCLKA ≥ PCLKB, PCLKD ≥ PCLKA ≥ PCLKB, ICLK ≥ FCLK
Restrictions on the clock frequency ratio: (N: integer, and up to 64)
ICLK:FCLK = 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: Minimum FCLK frequency is 1 MHz in Programming/Erasure (P/E) mode.

Note 1. Only when USBFS is used as the device controller.

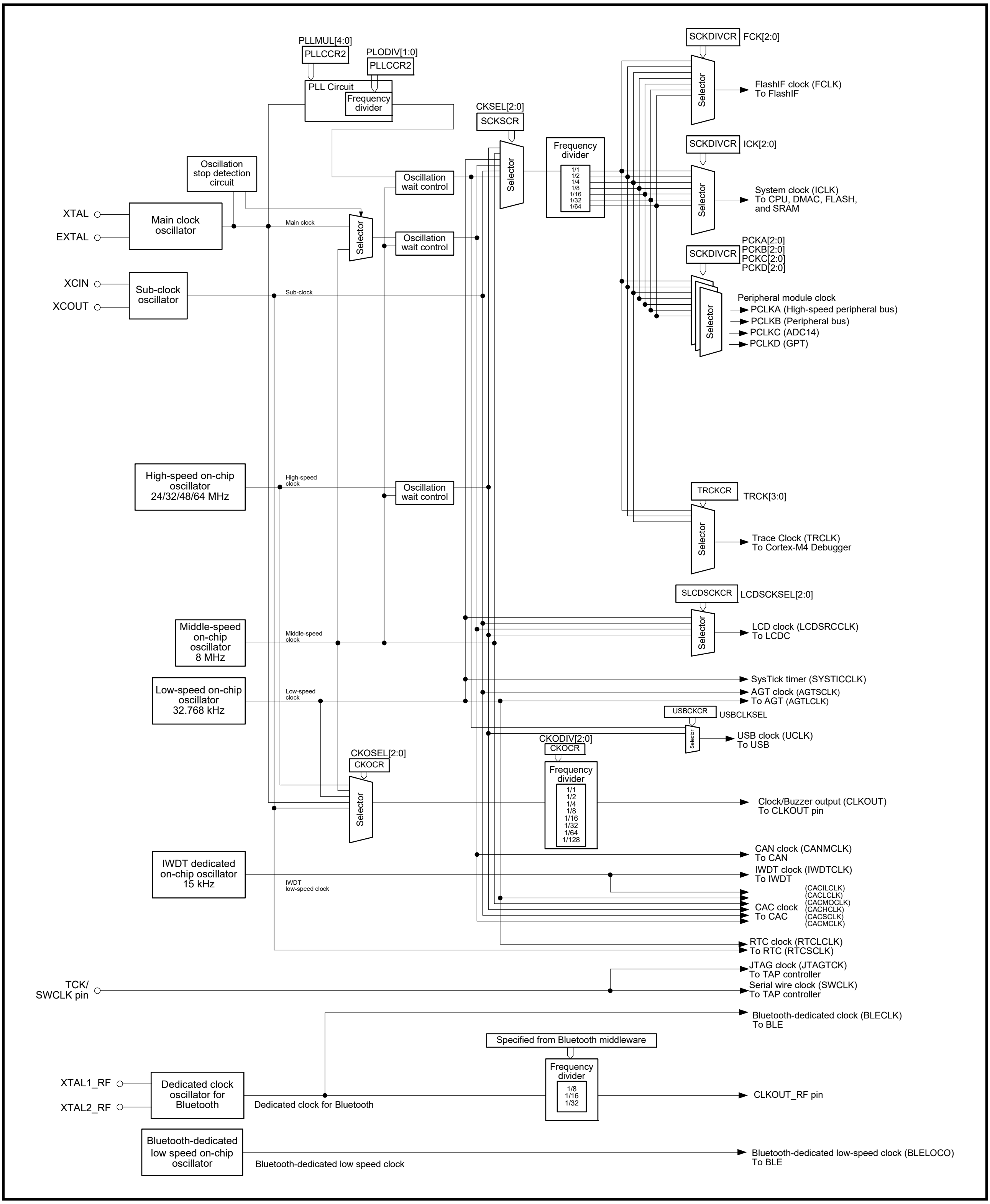

PLLMUL[4:0]
PLLCCR2
PLODIV[1:0]
PLLCCR2
PLL Circuit
Frequency divider
CKSEL[2:0]
SCKSCR
SCKDIVCR
FCK[2:0]
Selector
FlashIF clock (FCLK)
To FlashIF
Oscillation stop detection circuit
Oscillation wait control
Frequency divider
1/1
1/2
1/4
1/8
1/16
1/32
1/64
SCKDIVCR
ICK[2:0]
System clock (ICLK)
To CPU, DMAC, FLASH, and SRAM
XTAL
EXTAL
Main clock oscillator
Main clock
Selector
Oscillation wait control
SCKDIVCR
PCKA[2:0]
PCKB[2:0]
PCKC[2:0]
PCKD[2:0]
XCIN
XCOUT
Sub-clock oscillator
Sub-clock
Peripheral module clock
PCLKA (High-speed peripheral bus)
PCLKB (Peripheral bus)
PCLKC (ADC14)
PCLKD (GPT)
High-speed on-chip oscillator 24/32/48/64 MHz
High-speed clock
Oscillation wait control
TRCKCR
TRCK[3:0]
Trace Clock (TRCLK)
To Cortex-M4 Debugger
SLCDSCKCR
LCDSCKSEL[2:0]
Middle-speed on-chip oscillator 8 MHz
Middle-speed clock
LCD clock (LCDSRCCLK)
To LCDC
SysTick timer (SYSTICCLK)
Low-speed on-chip oscillator 32.768 kHz
Low-speed clock
AGT clock (AGTSCLK)
To AGT (AGTLCLK)
USBCKCR
USBCLKSEL
USB clock (UCLK)
To USB
CKODIV[2:0]
CKOCR
CKOSEL[2:0]
CKOCR
Frequency divider
1/1
1/2
1/4
1/8
1/16
1/32
1/64
1/128
Clock/Buzzer output (CLKOUT)
To CLKOUT pin
CAN clock (CANMCLK)
To CAN
IWDT dedicated on-chip oscillator 15 kHz
IWDT low-speed clock
IWDT clock (IWDTCLK)
To IWDT
CAC clock
To CAC
(CACILCLK)
(CACLCLK)
(CACMOCLK)
(CACHCLK)
(CACSCLK)
(CACMCLK)
RTC clock (RTCLCLK)
To RTC (RTCSCLK)
TCK/ SWCLK pin
JTAG clock (JTAGTCK)
To TAP controller
Serial wire clock (SWCLK)
To TAP controller
Bluetooth-dedicated clock (BLECLK)
To BLE
Specified from Bluetooth middleware
Frequency divider
1/8
1/16
1/32
XTAL1_RF
XTAL2_RF
Dedicated clock oscillator for Bluetooth
Dedicated clock for Bluetooth
CLKOUT_RF pin
Bluetooth-dedicated low speed on-chip oscillator
Bluetooth-dedicated low speed clock
Bluetooth-dedicated low-speed clock (BLELOCO)
To BLE

Figure 9.1 Clock generation circuit block diagram

Table 9.3 lists the input and output pins of the clock generation circuit.

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, see section 9.3.2, External Clock Input.
EXTAL	Input	
XCIN	Input	These pins are used to connect to a 32.768 kHz crystal resonator
XCOUT	Output	
TCK/SWCLK	Input	This pin is used to input the clock for the JTAG
CLKOUT	Output	This pin is used to output the CLKOUT/BUZZER clock
XTAL1_RF	Input	Connect a 32 MHz oscillator
XTAL2_RF	Output	
CLKOUT_RF	Output	Bluetooth-dedicated clock output pin

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]			—	—	—	—	—	—	—	—
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*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. 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*3	b6 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

Bit	Symbol	Bit name	Description	R/W
b10 to b8	PCKB[2:0]	Peripheral Module Clock B (PCLKB) Select*2	b10 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
b14 to b12	PCKA[2:0]	Peripheral Module Clock A (PCLKA) Select*2	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	—	Reserved	These bits should be written the same as PCKB[2:0].	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	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 association between the frequencies of the system clock (ICLK) and the flash interface clock (FCLK) should be ICLK:FCLK = N:1 (N: integer)
If a setting is made where ICLK < FCLK, then that setting is ignored.

Note 2. The association between the frequencies of the system clock (ICLK) and the peripheral module clocks (PCLKA, PCLKB) should be ICLK:PCLKA = N:1, ICLK:PCLKB = N:1 (N: integer)
If a setting is made where ICLK < PCLKA or ICLK < PCLKB, then that setting is ignored.

Note 3. The association between the frequencies of the system clock (ICLK) and the peripheral module clocks (PCLKC, PCLKD) should be ICLK:PCLKC, PCLKD = N:1 or 1:N (N: integer)

Note 4. 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), and the flash interface clock (FCLK).

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

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

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

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

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

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

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

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

ICK[2:0] bits (System Clock (ICLK) Select*1, *2, *3, *4)

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

FCK[2:0] bits (FlashIF Clock (FCLK) Select*1)

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

9.2.2 System Clock Source Control Register (SCKSCR)

Address(es): SYSTEM.SCKSCR 4001 E026h

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

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*1)

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).

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.

The clock sources should be switched when there are no occurring internal asynchronous interrupt.

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 shown 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*1)

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

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

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: PLL is operating*1 1: PLL is stopped.	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, 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 the 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 MCU 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 under 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 has elapsed after PLLSTP is set to 1 (PLL is stopped)
- At least 1 µs has elapsed after PLLMUL[4:0] bits are set (to select the PLL frequency multiplication).

9.2.5 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: Writing 0 to the MEMWAIT bit is prohibited when SCKDIVCR.ICK selects division by 1 and the 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)

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

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

Table 9.4 MEMWAIT bit setting

MEMWAIT bit	MCU operation power control		
		High-speed mode	
	Mode: except 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.

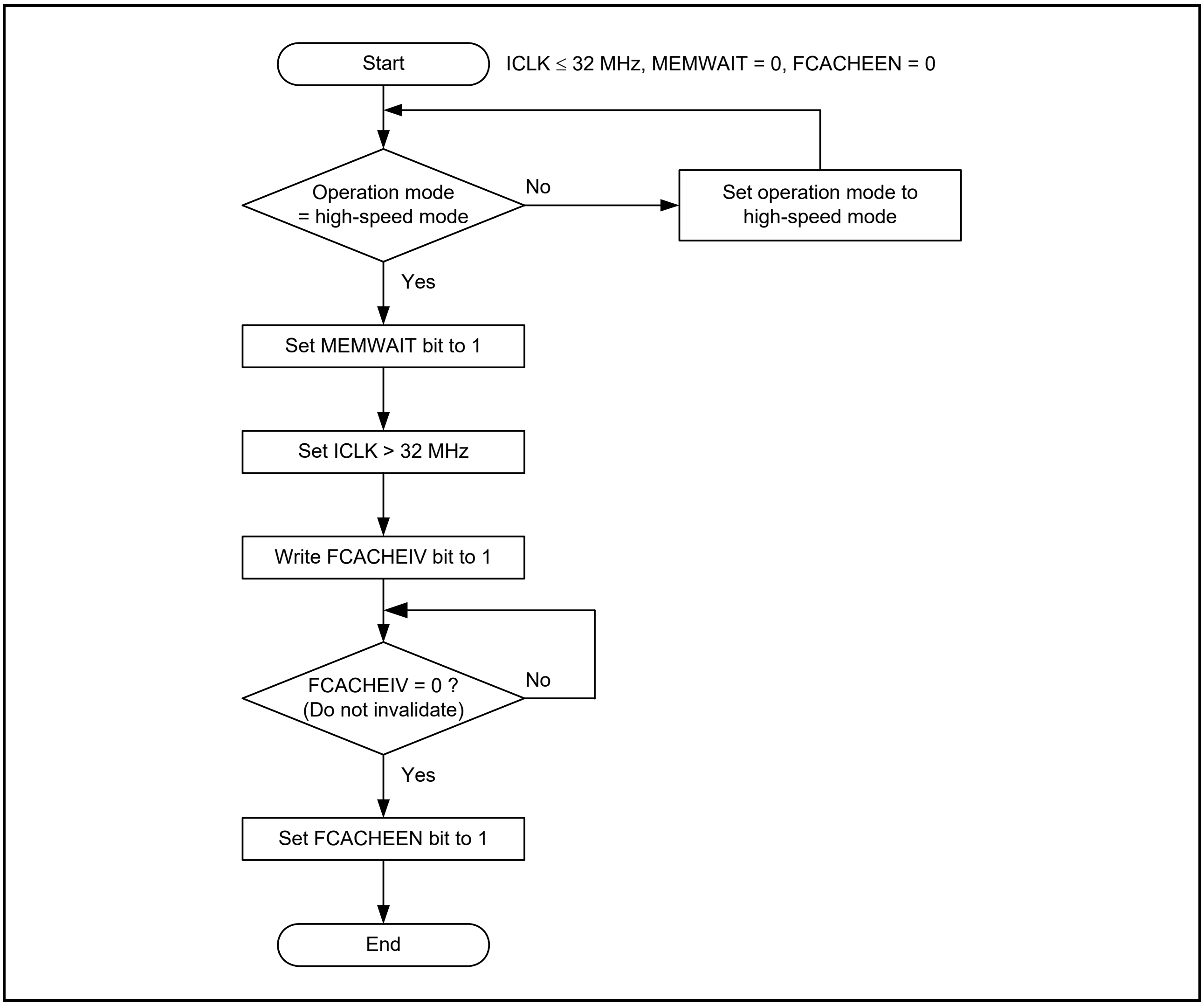

Start
ICLK ≤ 32 MHz, MEMWAIT = 0, FCACHEEN = 0
Operation mode = high-speed mode
No
Set operation mode to high-speed mode
Yes
Set MEMWAIT bit to 1
Set ICLK > 32 MHz
Write FCACHEIV bit to 1
FCACHEIV = 0 ? (Do not invalidate)
No
Yes
Set FCACHEEN bit to 1
End

Figure 9.2 When setting the ICLK > 32 MHz

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

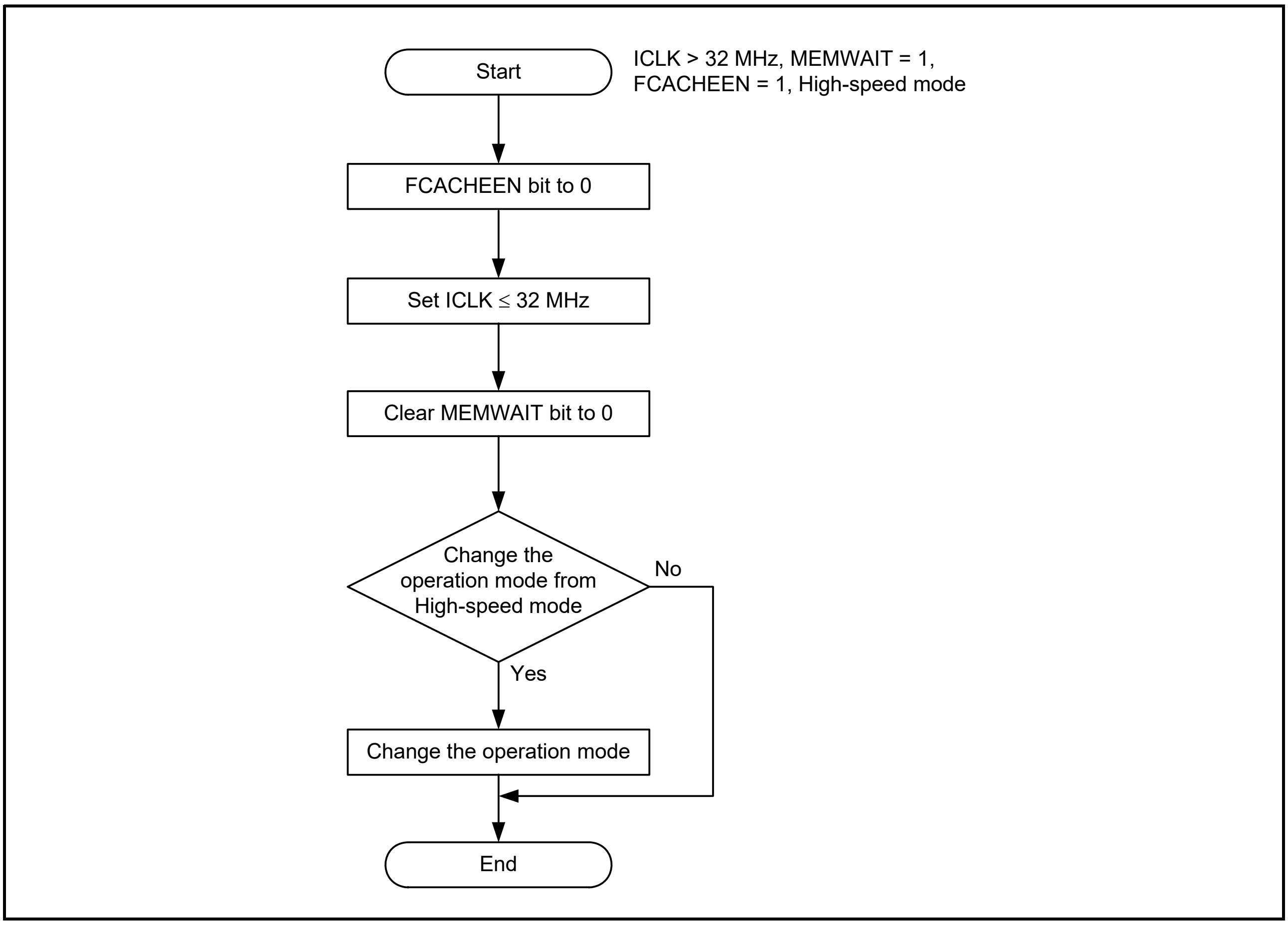

Start
ICLK > 32 MHz, MEMWAIT = 1,
FCACHEEN = 1, High-speed mode
FCACHEEN bit to 0
Set ICLK ≤ 32 MHz
Clear MEMWAIT bit to 0
Change the operation mode from High-speed mode
No
Yes
Change the operation mode
End

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

9.2.6 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 to check 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, only use the main clock after confirming that the OSCSF.MOSCSF bit is set to 1.

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

The following restrictions apply when starting and stopping operation:

- After stopping the main clock oscillator, confirm that the OSCSF.MOSCSF bit is 0 before restarting the main clock oscillator
- Confirm that the main clock oscillator operates and that the OSCSF.MOSCSF bit is 1 before stopping the main clock oscillator
- Regardless of whether the main clock oscillator is selected as the system clock, confirm that the OSCSF.MOSCSF bit is set to 1 before executing a WFI instruction to place the MCU in Software Standby modes
- 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).

9.2.7 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 after reading the bit and checking that its value is updated. When the sub clock is used as the source for some modules for example, RTC, use the SOSTP bit.

When using the sub-clock oscillator, set the Sub Clock Oscillator Mode Control Register (SOMCR) before setting SOSTP to 0. After setting SOSTP to 0, use the sub-clock oscillator only after the sub-clock oscillation stabilization time (tSUBOSCOWT) elapses. A fixed time for stabilization is required for oscillation to become stable after selecting the sub-clock operation with the SOSTP bit. A fixed time is also required for oscillation to actually stop after setting the SOSTP bit.

The following restrictions apply when starting and stopping operation:

- When restarting the sub-clock oscillator after it stops, allow an interval of at least 5 SOSC clock cycles for it to remain stopped.

- Confirm that the sub-clock oscillator is stable when setting the sub-clock oscillator to stop.
- Regardless of whether the sub-clock oscillator is selected as the system clock, ensure that oscillation by the sub-clock oscillator 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 sub-clock oscillator, wait for at least 3 SOSC clock cycles after setting the sub-clock oscillator to stop and before executing the WFI instruction.

Writing 1 to SOSTP is prohibited under the following condition:

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

9.2.8 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 details on VBTCR1.BPWSWSTP, see section 12, Battery Backup Function.

The LOCOCR register controls the LOCO.

LCSTP bit (LOCO Stop)

The LCSTP bit starts or stops the LOCO.

After the LCSTP bit is modified to operate the LOCO, only use the clock after the LOCO clock-oscillation stabilization wait time (tLOCOWT) elapses. A fixed time for stabilization of oscillation is required for oscillation to become stable after setting the LOCO clock to start operation. A fixed time is also required for oscillation to stop after setting the oscillator.

The following restrictions apply when starting and stopping operation:

- When restarting the LOCO after it stops, allow an interval of at least 5 LOCO clock cycles for it to remain stopped
- Ensure that LOCO oscillation is stable before stopping the LOCO clock
- Regardless of whether the LOCO is selected as the system clock, ensure that LOCO oscillation is stable before executing a WFI instruction to place the MCU in Software Standby mode
- When a transition to Software Standby mode is to follow the setting to stop the LOCO clock, wait for at least 3 LOCO cycles before executing the WFI instruction.

Writing 1 to LOSTP is prohibited under the following condition:

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

9.2.9 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, *3 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 while HOCOCR.HCSTP = 0 and OSCSF.HOCOSF = 0 (HOCO is in stabilization wait counting).

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

The HOCOCR register controls the HOCO.

HCSTP bit (HOCO Stop)

The HCSTP bit starts or stops the HOCO. For the HOCO to operate, the High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR) must also be set.

After the HCSTP bit setting is modified to run the HOCO, confirm that the OSCSF.HOCOSF is set to 1 before using the oscillator. When OFS1.HOCOEN is set to 1, confirm that the OSCSF.HOCOSF is also set to 1 before using the HOCO clock. A fixed time for stabilization is required for oscillation to become stable after setting the HOCO operation. A fixed time is also required for oscillation to actually stop after setting the HOCO clock to stop.

The following restrictions apply when starting and stopping operation:

- After stopping the HOCO, confirm that the OSCSF.HOCOSF bit is 0 before restarting the HOCO
- Confirm that the HOCO operates and that the OSCSF.HOCOSF bit is 1 before stopping the HOCO
- 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 after setting the HOCO and before executing the WFI instruction.

Writing 1 to HCSTP is prohibited under the following condition:

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

9.2.10 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 MOCO 1: Stop MOCO.	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.

After setting MCSTP to 0, use the MOCO clock only after the MOCO clock oscillation stabilization time (tMOCOWT) elapses. A fixed time for stabilization of oscillation is required after setting MCSTP to 0. A fixed time is also required for oscillation to stop after setting MCSTP to 1.

The following are limitations when starting and stopping the oscillator:

- When restarting the MOCO after it has been stopped, allow a stop interval of at least 5 MOCO clocks cycle for it to remain stopped
- Ensure that MOCO oscillation is stable when setting the MOCO to stop
- Regardless of whether the MOCO is selected as the system clock, ensure 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 (OSTDCR.OSTDE) in the Oscillation Stop Detection Control Register.

Because the MOCO clock is used to measure the waiting time for other oscillators, the MOCO clock oscillates when the waiting time for other oscillators is measured, regardless of the setting of MOCOCR.MCSTP. The MOCO clock may be unintentionally supplied even when the MCSTP is set to stop.

9.2.11 Oscillation Stabilization Flag Register (OSCSF)

Address(es): SYSTEM.OSCSF 4001 E03Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PLLSF	—	MOSC SF	—	—	HOCO SF
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.	R
b3	MOSCSF	Main Clock Oscillation Stabilization Flag	0: The main clock oscillation is stopped (MOSTP = 1) or is not stable yet*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.	R
b5	PLLSF	PLL Clock Oscillation Stabilization Flag	0: The PLL clock is stopped or oscillation of the PLL clock has not yet become stable 1: Oscillation of 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.	R

Note 1. Value after reset depends on the OFS1.HOCOEN bit setting.
When OFS1.HOCOEN = 1, OSCSF.HOCOSF value becomes 0 after reset is released, and OSCSF.HOCOSF value becomes 1 after the HOCO oscillation stabilization time is 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 contains flags to indicate the operation status of the counters in the oscillation stabilization wait circuits for the individual oscillators.

After oscillation starts, these counters measure the wait time until each oscillator output clock is supplied to the internal circuits. An overflow of a counter indicates that the clock supply is stable and available for the associated circuit.

HOCOSF flag (HOCO Clock Oscillation Stabilization Flag)

The HOCOSF flag indicates the operating state 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, supply of the high-speed clock 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 and then is deactivated because the HOCOCR.HCSTP bit is set to 1.

MOSCSF flag (Main Clock Oscillation Stabilization Flag)

The MOSCSF flag indicates the operating state 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 and then is deactivated because the MOSCCR.MOSTP bit is set to 1.

PLLSF flag (PLL Clock Oscillation Stabilization Flag)

The PLLSF flag indicates the operating state 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 starts 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 operates, it is deactivated when the PLLCR.PLLSTP bit is set to 1.

9.2.12 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 the oscillation stop detection function 1: Enable the 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, set the OSTDIE bit to 0 before OSTDF is set to 0. Wait for at least 2 PCLKB cycles before setting the OSTDIE bit to 1. Depending on the number of cycles required to read a given I/O register, a wait time longer than 2 PCLKB cycles might be required.

OSTDE bit (Oscillation Stop Detection Function Enable)

The OSTDE bit enables the oscillation stop detection function.

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

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

The OSTDE bit must be set to 0 before transitioning to Software Standby mode. To transition to Software Standby mode, first set the OSTDE bit to 0, 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, PCLKA, PCLKB, PCLKC, PCLKD is prohibited
- In low-voltage mode, selecting division by 1, 2 for ICLK, FCLK, PCLKA, PCLKB, PCLKC, PCLKD is prohibited.

9.2.13 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 not detected 1: Main clock oscillation stop detected.	R(/W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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 this stop is detected, the OSTDF 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 and 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).

9.2.14 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/W

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

Set the MSTS[3:0] bits to select the oscillation stabilization wait time for the main clock oscillator.

Set the main clock oscillation stabilization time to a period longer than or equal to the stabilization time recommended by the oscillator manufacturer. When the main clock is input externally, set these bits to 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 starts internally in the MCU, and the OSCSF.MOSCSF flag becomes 1. If the set wait time is short, supply of the main clock starts before oscillation of the clock becomes stable.

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

9.2.15 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 (29.13 µs) When HOCO operating frequency is 24 MHz or 32 MHz, and the operation power control mode is other than low-voltage mode. • Wait time = 287 cycles (35.875 µs) (value after reset) when HOCO operating frequency is 48 MHz and operation power control mode is other than low voltage mode. • Wait time = 679 (84.88 µs) (value after reset) when 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 is calculated at MOCO = 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/W

HOCOWTCR controls the wait time until output of the signal from the high-speed clock oscillator to the internal circuits starts. Only write to HOCOWTCR when the HOCOCR.HCSTP bit is 1 or the OSCSF.HOCOSF flag is 1. Do not write to 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 by counting the number of middle-speed clock cycles set in the HOCOWTCR register.

When the high-speed clock oscillator starts, the oscillation stabilization wait circuit starts counting the number of middle-speed clock cycles set in the HOCOWTCR register. The MCU clock supply is disabled until counting of the set number of cycles is complete. After counting completes, supply of the clock signal in the MCU starts and the OSCSF.HOCOSF flag is set to 1.

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

9.2.16 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

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

MODRV1 bit (Main Clock Oscillator Drive Capability 1 Switching)

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

MOSEL bit (Main Clock Oscillator Switching)

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

9.2.17 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

This register must be modified when SOSCCR.SOSTP is 1 (SOSC is stopped).

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

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

9.2.18 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: LCD source clock out disabled 1: LCD source clock out enabled.	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)

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

When changing these bits, apply the following steps:

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

LCDSCKEN bit (LCD Source Clock Out Enable)

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

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

1. Change this bit.
2. Wait for at least 2 cycles of the source clock selected by 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, apply 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.

9.2.19 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 0: HOCO 0 0 1: MOCO 0 1 0: LOCO 0 1 1: MOSC 1 0 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 0: ×1 0 0 1: /2 0 1 0: /4 0 1 1: /8 1 0 0: /16 1 0 1: /32 1 1 0: /64 1 1 1: /128.	R/W
b7	CKOEN	Clock Out enable	0: Clock Out disabled 1: Clock Out enabled.	R/W

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

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

Set the CKOEN bit to 0 when changing the CLKOUT source clock.

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

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

Set the CKOEN bit to 0 when changing the division ratio. The division ratio of the output clock frequency must be set to a value no higher than the characteristics of the CLKOUT pin output frequency. For details on the characteristics of the CLKOUT pin, see section 48, 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 source 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 source is stopped in that mode.

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

9.2.21 MOCO User Trimming Control Register (MOCOUTCR)

Address(es): SYSTEM.MOCOUTCR 4001 E061h

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

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 time when the frequency is stabilized corresponds to the time when it is stabilized at the start of the MCU operation.

When the ratio of the MOCO frequency to the other oscillation frequency is an integer value, changing the MOCOUTCR value is prohibited.

9.2.22 HOCO User Trimming Control Register (HOCOUTCR)

Address(es): SYSTEM.HOCOUTCR 4001 E062h

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

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 time taken for the frequency to stabilize corresponds to the time taken for the frequency to stabilize at the start of the MCU operation. When USBCKCR.USBCLKSEL = 1, writing any other value except 00h to HOCOUTCR is prohibited.

9.2.23 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: Operation disabled 1: Operation enabled.	R/W

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

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

9.2.24 USB Clock Control Register (USBCKCR)

Address(es): SYSTEM.USBCKCR 4001 E0D0h

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

Bit	Symbol	Bit name	Description	R/W
b0	USBCLKSEL	USB Clock Source Select	0: PLL (Value after reset) 1: HOCO.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

USBCLKSEL bit (USB Clock Source Select)

The USBCLKSEL bit selects the source of the USB clock (UCLK).

- Rewrite the USBCKCR register while the SYSCFG.SCKE bit is 0.
- The USBCKCR.USBCLKSEL bit can only be set to 1 when USBFS is used as the device controller. Set the USBCKSR.USBCLKSEL bit to 0 to use the host controller.
- The user trimming function cannot be used when the USBCKCR.USBCLKSEL bit is 1. To use the HOCO user trimming function, set the bits HOCOUTCR.HOCOUTRM[7:0] to 00h.

9.3 Main Clock Oscillator

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

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

9.3.1 Connecting a Crystal Resonator

Figure 9.4 shows an example of connecting a crystal resonator.

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

When connecting a resonator to supply the clock, the frequency of the resonator must be in the frequency range of the resonator for the main clock oscillator as described in Table 9.1.

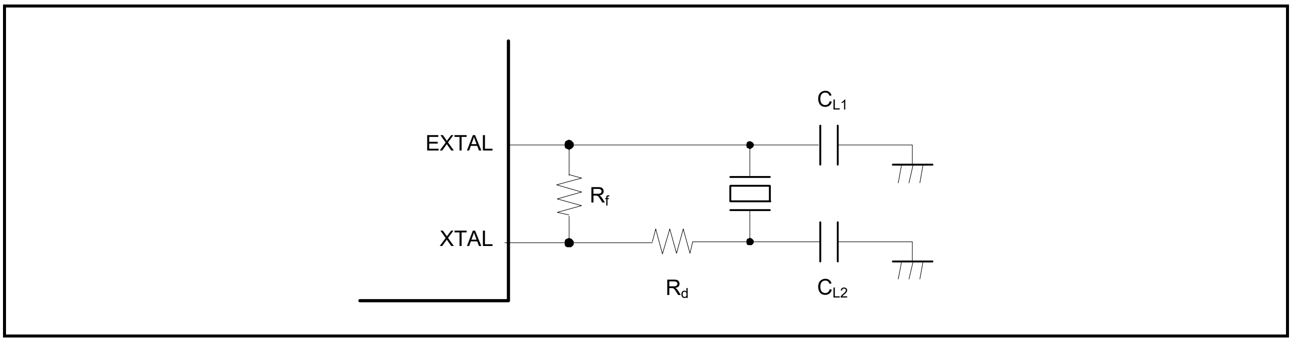


Figure 9.4 Example of crystal resonator connection

9.3.2 External Clock Input

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

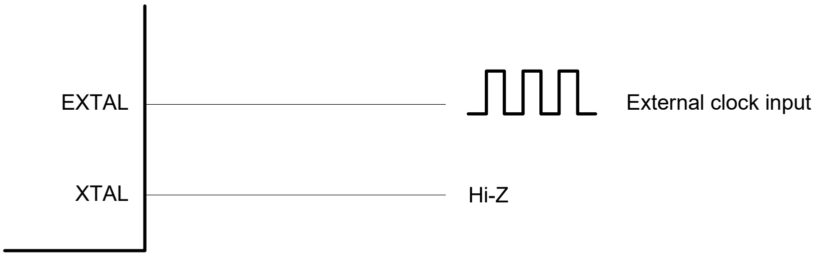


Figure 9.5 Equivalent circuit for external clock

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 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 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.

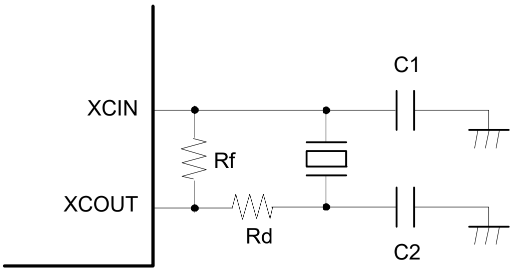


Figure 9.6 Connection example of 32.768-kHz crystal resonator

9.5 Dedicated Clock Oscillator for Bluetooth

Operating Bluetooth requires connection to the 32-MHz clock oscillator.

9.5.1 Connecting the Oscillator

An example of connection of the oscillator is shown in Figure 9.7.

Insert a damping resistor (Rd) as required. Set the resistance to the value recommended by the oscillator manufacturer as the value depends on the oscillator and its driving capability. If the oscillator manufacturer states that the addition of a feedback resistor (Rf) to the oscillator is required, insert Rf between XTAL1_RF and XTAL2_RF according to the instructions.

To control the oscillator, use the Bluetooth middleware provided by Renesas. The Bluetooth middleware is also able to control the settings of the on-chip variable capacitors, CL1 and CL2, to adjust the frequency of oscillation. Adjustment of the frequency of the Bluetooth-dedicated clock oscillator is explained in the application note Procedure for Adjusting the Frequency of the Bluetooth-Dedicated Clock Oscillator (R01AN4887). Obtain the latest version of this document from the Renesas website.

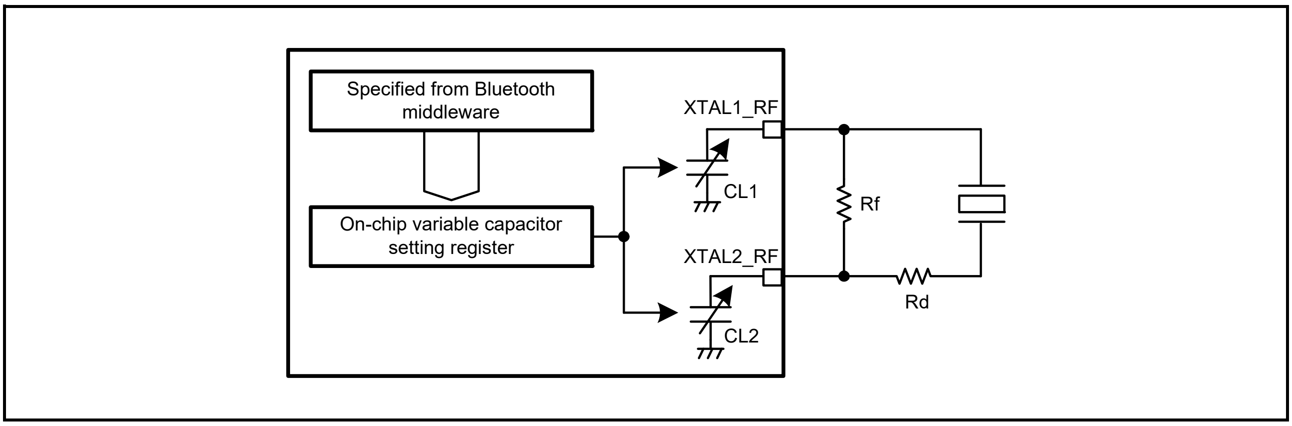

Specified from Bluetooth middleware
On-chip variable capacitor setting register
XTAL1_RF
CL1
XTAL2_RF
CL2
Rf
Rd

Figure 9.7 Example of the Connection of a 32-MHz Crystal Resonator

9.5.2 Connecting the Bluetooth-Dedicated Clock Output Pin

When the Bluetooth middleware sets up the clock output and the BLE shifts from waiting mode to RF power-down mode, the frequency-divided clock for the Bluetooth-dedicated clock is output. If the BLE subsequently shifts to RF low power mode, the RF clock output is disabled. The Bluetooth middleware is able to specify the frequency-divisor for the Bluetooth-dedicated clock.

The Bluetooth-dedicated clock output can also be used as the input of the external clock for the main clock oscillator by externally connecting the CLKOUT_RF and EXTAL pins on the board. Figure 9.8 shows an example of a configuration where the Bluetooth-dedicated clock output is used as the input of the external clock for the main clock oscillator.

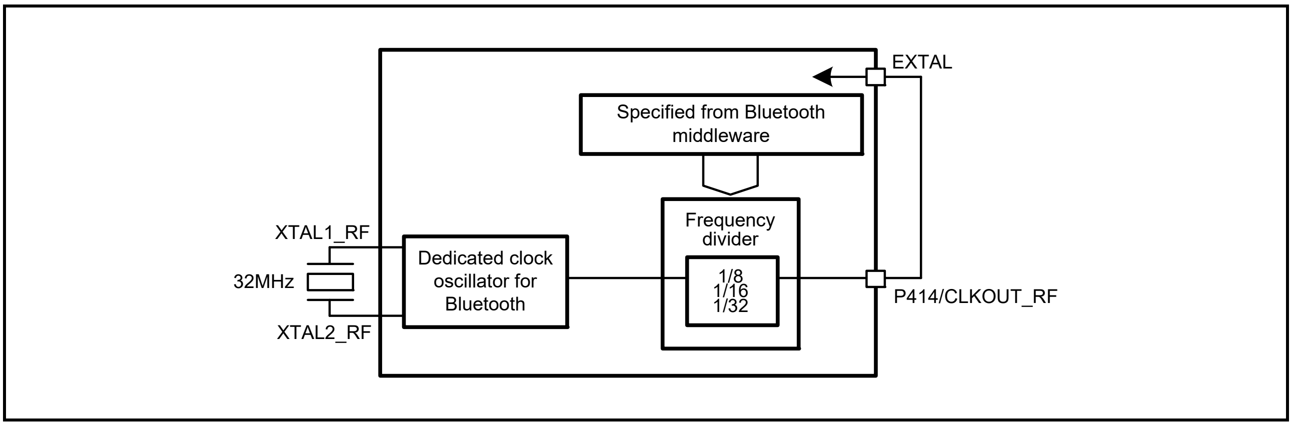

EXTAL
Specified from Bluetooth middleware
Frequency divider
1/8
1/16
1/32
XTAL1_RF
32MHz
XTAL2_RF
Dedicated clock oscillator for Bluetooth
P414/CLKOUT_RF

Figure 9.8 Example of Connection of the Bluetooth-dedicated clock output pin

9.6 Oscillation Stop Detection Function

9.6.1 Oscillation Stop Detection and Operation after Detection

The oscillation stop detection function is used to detect the main clock oscillator stop.

When oscillation stop is detected, the system clock switches as follows:

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

- If an oscillation stop is detected with SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL), the PLL clock remains the system clock source. The frequency becomes a free-running oscillation frequency and the setting of SCKSCR.CKSEL[2:0] is unchanged.

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, due to a malfunction of the main clock oscillator (see section 48, Electrical Characteristics).

Switching between the main clock and MOCO clock or between the PLL clock and 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 MOCO.
  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 the 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 settling time elapses.

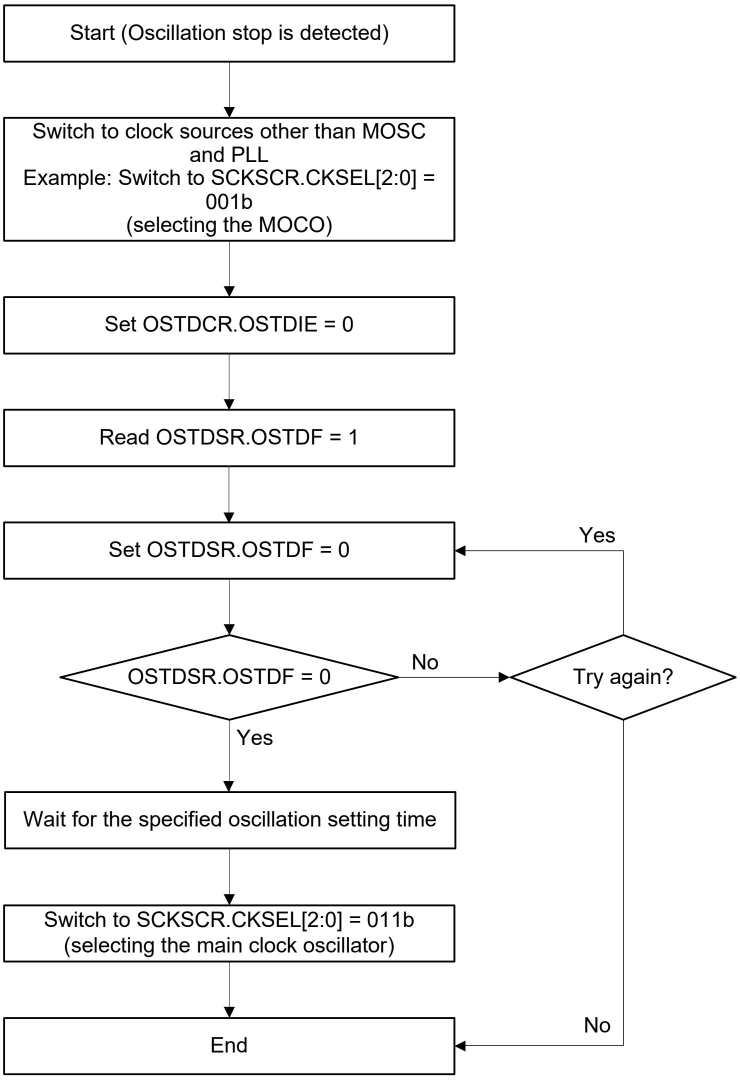
After a reset is released, the main clock oscillator stops and the oscillation stop detection function is disabled. To enable the oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after a specified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

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

The oscillation stop detection function switches the following clocks to the MOCO clock (when system clock is MOSC):

- All clocks that can select the MOSC or PLL except CLKOUT
- The system clock (ICLK) frequency during the MOCO (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 in the System Clock Select bits (SCKDIVCR.ICK[2:0]).

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


Start (Oscillation stop is detected)
Switch to clock sources other than MOSC and PLL
Example: Switch to SCKSCR.CKSEL[2:0] = 001b
(selecting the MOCO)
Set OSTDCR.OSTDIE = 0
Read OSTDSR.OSTDF = 1
Set OSTDSR.OSTDF = 0
Yes
OSTDSR.OSTDF = 0
No
Try again?
Yes
Wait for the specified oscillation setting time
Switch to SCKSCR.CKSEL[2:0] = 011b
(selecting the main clock oscillator)
No
End

Note: On returning 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 9.9 Flow of recovery on detection of oscillator stop

9.6.2 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 the non-maskable interrupts through software before using the oscillation stop detection interrupts. For details, see section 14, Interrupt Controller Unit (ICU).

9.7 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator.

9.8 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 memory, and SRAM — system clock (ICLK)
- Operating clocks for peripheral modules — PCLKA, PCLKB, PCLKC, and PCLKD
- Operating clock for the flash interface — FCLK
- 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.

Operating clocks for the BLE: The Bluetooth-dedicated clock (BLECLK), and the Bluetooth-dedicated low-speed clock (BLELOCO).

For details of the registers used to set the frequencies of the internal clocks, see section 9.8.1, System Clock (ICLK) to section 9.8.13, JTAG Clock (JTAGTCK).

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

9.8.1 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 by the ICK[2:0] bits in SCKDIVCR, the CKSEL[2:0] bits in SCKSCR, the

PLLMUL[4:0] and PLODIV[1:0] bits in PLLCCR2, and the HOCOFRQ1[2:0] bits in OFS1.

When the ICLK clock source is switched, the duration of the ICLK clock cycle becomes longer during the clock source transition period. See Figure 9.10 and Figure 9.11.

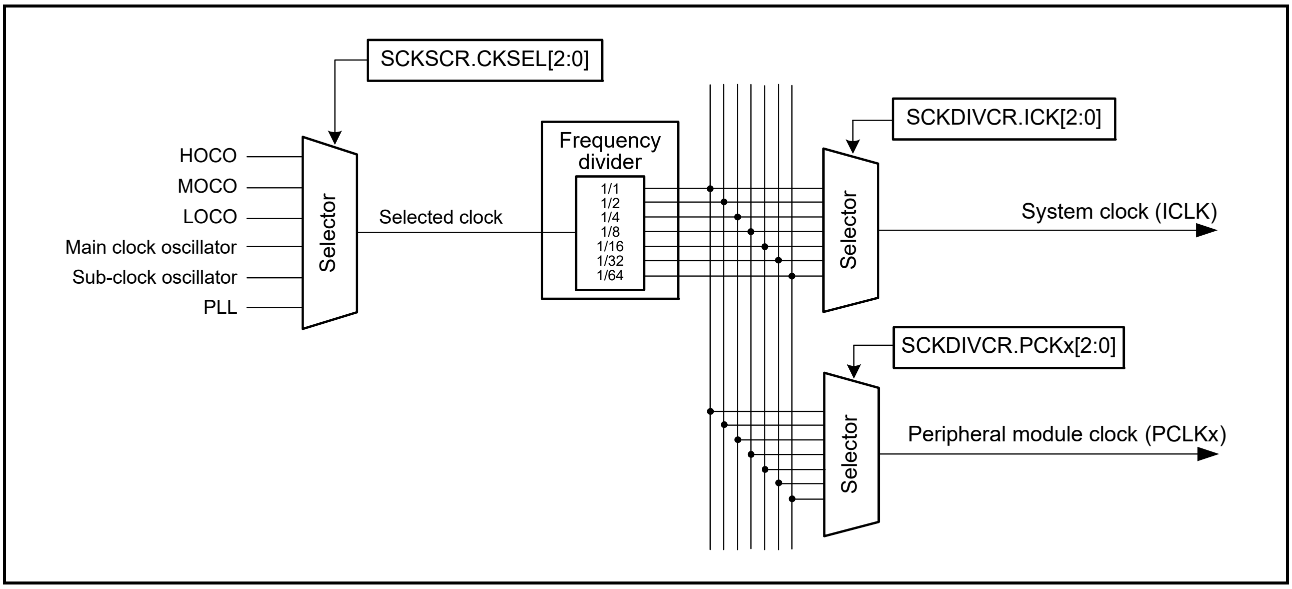

SCKSCR.CKSEL[2:0]
HOCO
MOCO
LOCO
Main clock oscillator
Sub-clock oscillator
PLL
Selector
Selected clock
Frequency divider
1/1
1/2
1/4
1/8
1/16
1/32
1/64
SCKDIVCR.ICK[2:0]
Selector
System clock (ICLK)
SCKDIVCR.PCKx[2:0]
Selector
Peripheral module clock (PCLKx)

Figure 9.10 Clock source selector block diagram

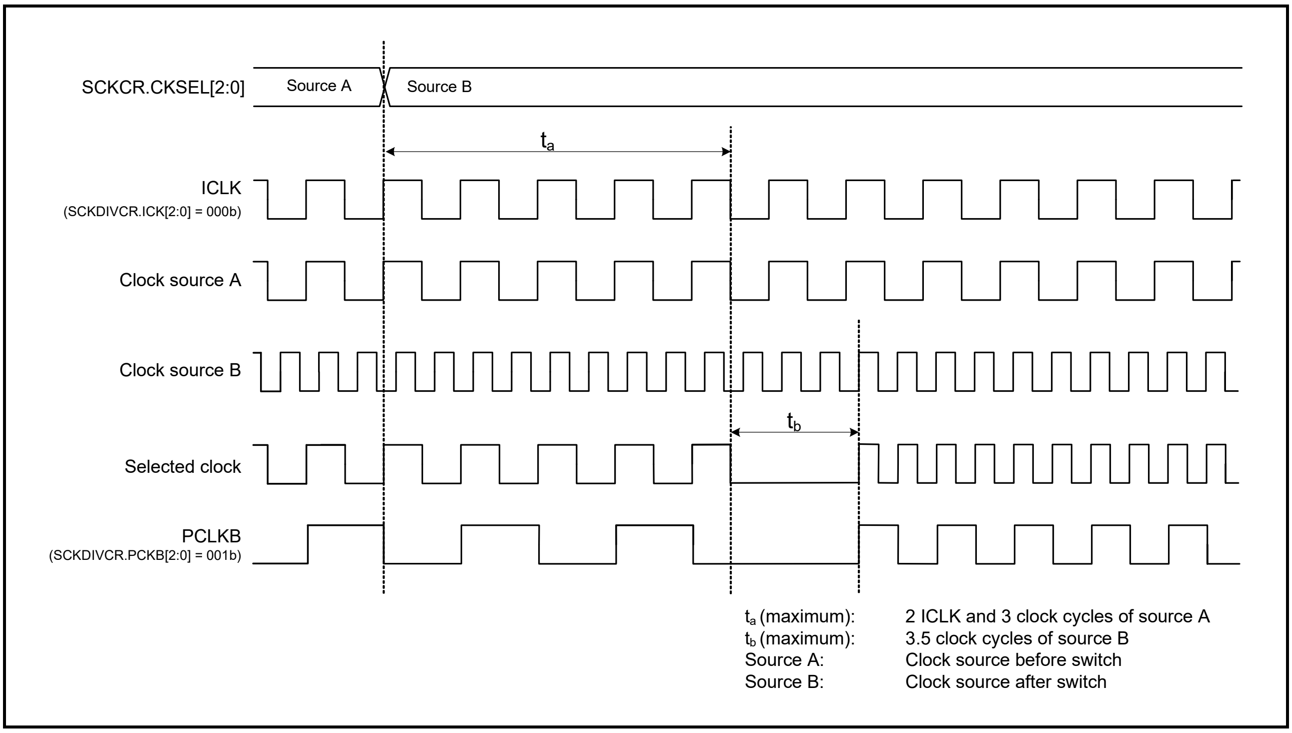

SCKCR.CKSEL[2:0]
Source A
Source B
t_a
ICLK
(SCKDIVCR.ICK[2:0] = 000b)
Clock source A
Clock source B
t_b
Selected clock
PCLKB
(SCKDIVCR.PCKB[2:0] = 001b)
t_a (maximum): 2 ICLK and 3 clock cycles of source A
t_b (maximum): 3.5 clock cycles of source B
Source A: Clock source before switch
Source B: Clock source after switch

Figure 9.11 Clock source switching timing diagram

9.8.2 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] bits in SCKDIVCR

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

When the clock source of the peripheral module clock is switched, the duration of the peripheral module clock cycle becomes longer during the clock source transition period. See Figure 9.10 and Figure 9.11.

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

9.8.4 USB Clock (UCLK)

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

The UCLK frequency is specified in the following bits:

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

9.8.5 CAN Clock (CANMCLK)

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

9.8.6 CAC Clock (CACCLK)

The CAC clock, CACCLK, is the 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.

9.8.7 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.

9.8.8 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.

9.8.9 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.

9.8.10 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.

9.8.11 Segment LCDC Source Clock (LCDSRCCLK)

The Segment LCDC source clock, LCDSRCCLK, is 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.

9.8.12 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. When changing the value of CKODIV[2:0] or CKOSEL[2:0] bits in CKOCR, make sure that the value of CKOCR.CKOEN 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] bits in PLLCCR2
- HOCOFRQ1[2:0] bits in OFS1.

9.8.13 JTAG Clock (JTAGTCK)

The JTAG-dedicated clock, JTAGTCK, is the operating clock for the JTAG.

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

9.8.14 Clocks for BLE

The Bluetooth-dedicated clock (BLECLK) and the Bluetooth-dedicated low-speed clock (BLELOCO) are the operating clocks for the BLE. To control these clocks, use the Bluetooth middleware provided by Renesas.

9.9 Usage Notes

9.9.1 Notes on Clock Generation Circuit

The frequencies of the system clock (ICLK), peripheral module clock (PCLKA to PCLKD), and flash interface clock (FCLK) 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 48, Electrical Characteristics
- The frequencies must not exceed the ranges listed in Table 9.2
- The peripheral modules operate on the PCLKB and PCLKA. As a result, the operating speed of modules such as the timer and SCI varies before and after the frequency is changed.
- The system clock (ICLK), peripheral module clock (PCLKA to PCLKD), flash interface clock (FCLK), must be set according to Table 9.2.

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

9.9.2 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.

9.9.3 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 9.12 to prevent electromagnetic induction from interfering with correct oscillation.

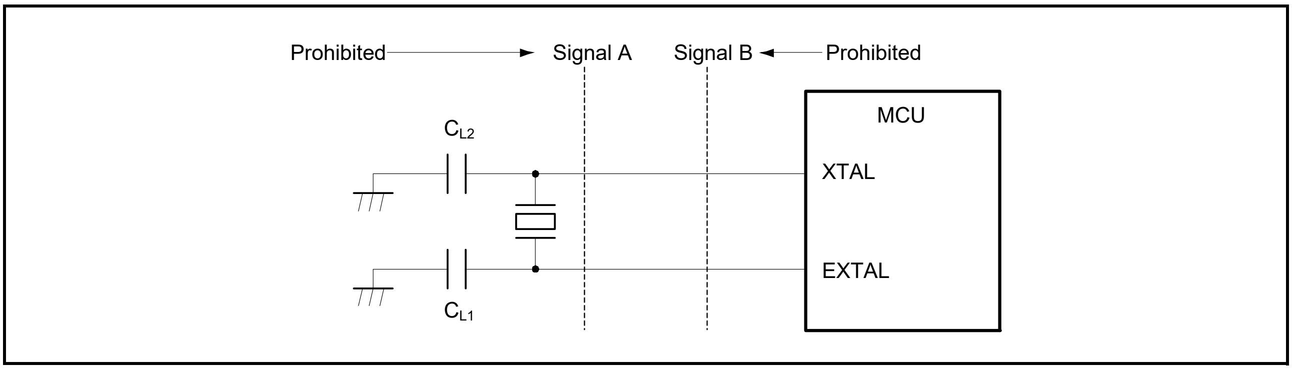


Figure 9.12 Signal routing in board design for oscillation circuit (applicable to the main clock oscillator as well as the sub clock oscillator and bluetooth-dedicated oscillator)

9.9.4 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 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.

Table 10.1 lists the CAC specifications, Figure 10.1 shows a block diagram, and Table 10.2 shows 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

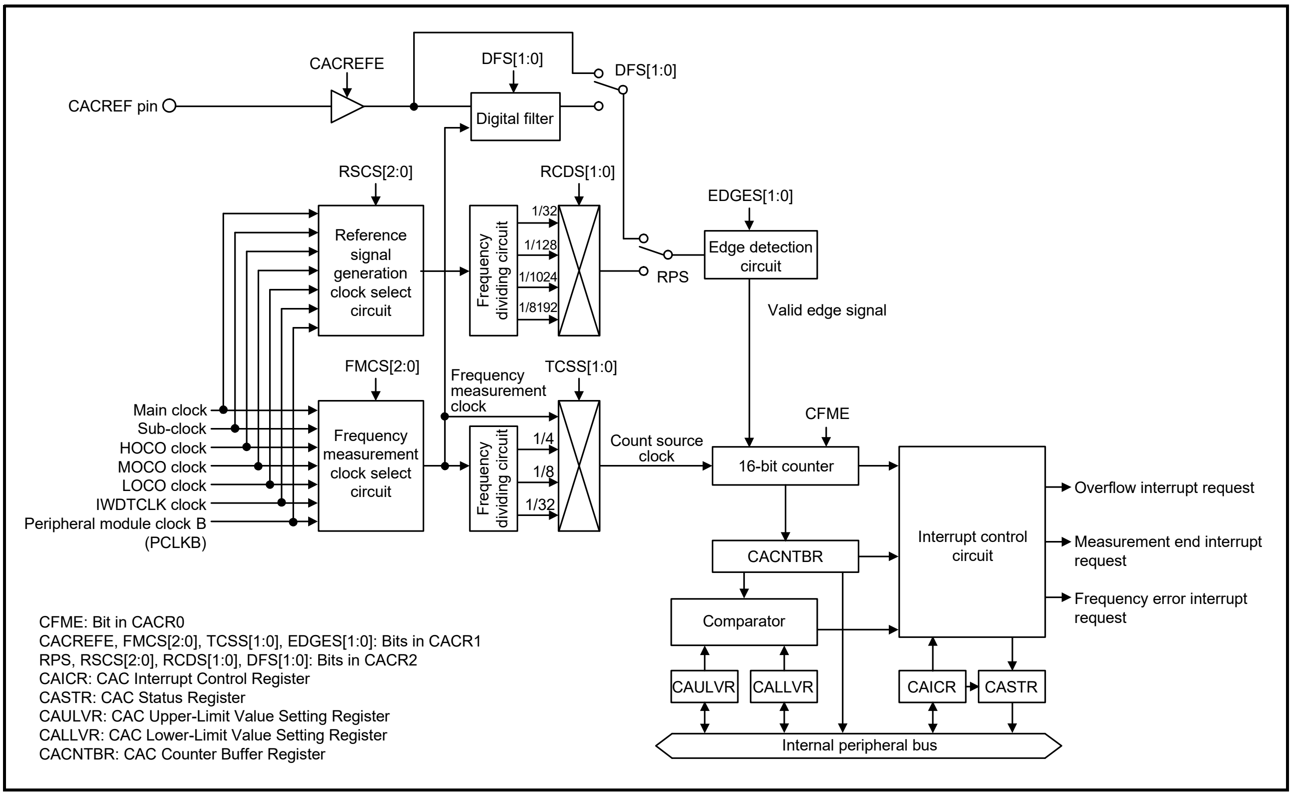

CACREF pin
CACREFE
DFS[1:0]
Digital filter
DFS[1:0]
RSCS[2:0]
RCDS[1:0]
Reference signal generation clock select circuit
Frequency dividing circuit
1/32
1/128
1/1024
1/8192
EDGES[1:0]
Edge detection circuit
RPS
Valid edge signal
FMCS[2:0]
Frequency measurement clock
TCSS[1:0]
Main clock
Sub-clock
HOCO clock
MOCO clock
LOCO clock
IWDTCLK clock
Peripheral module clock B (PCLKB)
Frequency measurement clock select circuit
Frequency dividing circuit
1/4
1/8
1/32
Count source clock
CFME
16-bit counter
Interrupt control circuit
Overflow interrupt request
Measurement end interrupt request
Frequency error interrupt request
CACNTBR
Comparator
CAULVR
CALLVR
CAICR
CASTR
Internal peripheral bus
CFME: Bit in CACR0
CACREFE, FMCS[2:0], TCSS[1:0], EDGES[1:0]: Bits in CACR1
RPS, RSCS[2:0], RCDS[1:0], DFS[1:0]: Bits in CACR2
CAICR: CAC Interrupt Control Register
CASTR: CAC Status Register
CAULVR: CAC Upper-Limit Value Setting Register
CALLVR: CAC Lower-Limit Value Setting Register
CACNTBR: CAC Counter Buffer Register

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 the 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: 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: 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 frequency error interrupt request 1: Enable frequency error interrupt request.	R/W
b1	MENDIE	Measurement End Interrupt Request Enable	0: Disable measurement end interrupt request 1: Enable measurement end interrupt request.	R/W
b2	OVFIE	Overflow Interrupt Request Enable	0: Disable overflow interrupt request 1: Enable overflow interrupt request.	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 enables the frequency error interrupt request.

MENDIE bit (Measurement End Interrupt Request Enable)

The MENDIE bit enables the measurement end interrupt request.

OVFIE bit (Overflow Interrupt Request Enable)

The OVFIE bit enables the overflow interrupt request.

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: Clock frequency is within the allowable range 1: 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)

The MENDF flag indicates the end of measurement.

[Setting condition]

- Measurement ends.

[Clearing condition]

- 1 is written to the MENDFCL bit.

OVFF flag (Overflow Flag)

The OVFF flag indicates that the counter overflowed.

[Setting condition]

- The counter overflowed.

[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

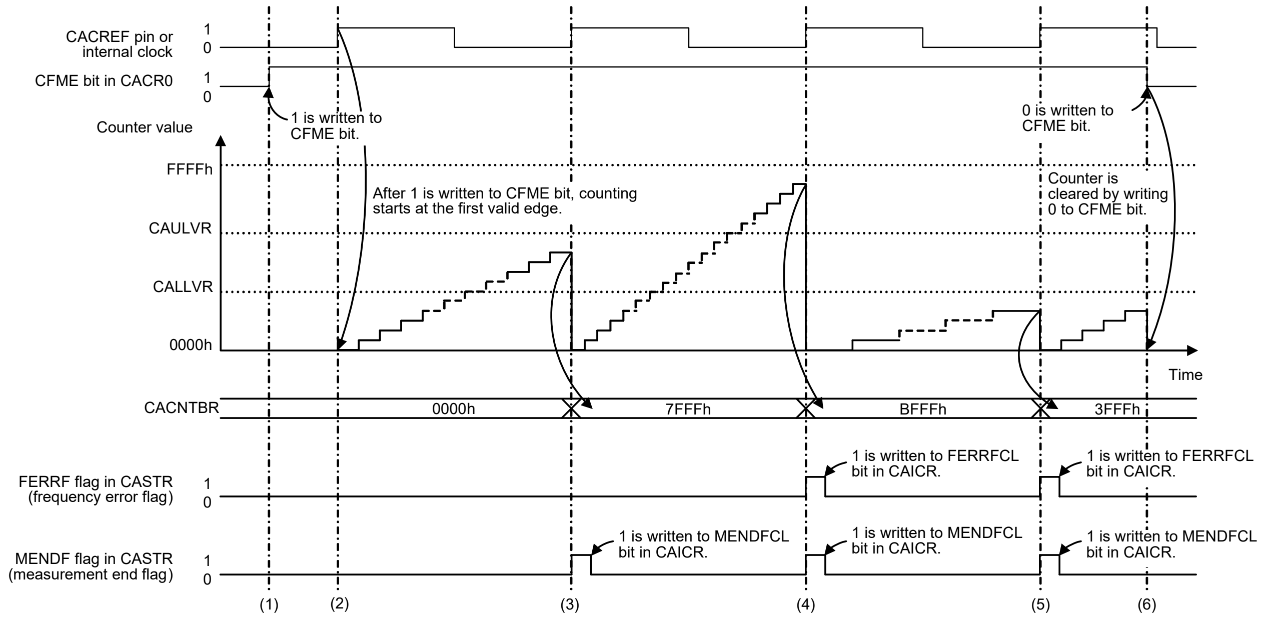
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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 using the CACREF pin input or the internal clock as a reference. Figure 10.2 shows an operating example of the CAC.


CACREF pin or internal clock
1
0
CFME bit in CACR0
1
0
1 is written to CFME bit.
0 is written to CFME bit.
Counter value
FFFFh
CAULVR
CALLVR
0000h
After 1 is written to CFME bit, counting starts at the first valid edge.
Counter is cleared by writing 0 to CFME bit.
Time
CACNTBR
0000h
7FFFh
BFFFh
3FFFh
FERRF flag in CASTR (frequency error flag)
1
0
1 is written to FERRFCL bit in CAICR.
1 is written to FERRFCL bit in CAICR.
MENDF flag in CASTR (measurement end flag)
1
0
1 is written to MENDFCL bit in CAICR.
1 is written to MENDFCL bit in CAICR.
1 is written to MENDFCL bit in CAICR.
(1)
(2)
(3)
(4)
(5)
(6)

When the CACREF pin input is used as a reference:
In CACR1: CACREFE bit = 1, EDGES[1:0] bits = 00b
CAULVR register = AAAAh, CALLVR register = 5555h
When the internal clock is used as a reference:
In CACR1: CACREFE bit = 0, EDGES[1:0] bits = 00b
CAULVR register = AAAAh, CALLVR register = 5555h

Figure 10.2 CAC operating example

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

The counter value transferred 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 using 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 types of 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 with CAULVR and CALLVR is either CACNTBR > CAULVR or CACNTBR < CALLVR
Measurement end interrupt	CAICR.MENDIE	CASTR.MENDF	• Valid edge is input from the CACREF pin or internal clock • Measurement end interrupt does not occur at the first valid edge after writing 1 to the CACR0.CFME bit.
Overflow interrupt	CAICR.OVFIE	CASTR.OVFF	The counter overflows

10.5 Usage Note

10.5.1 Settings for the Module-Stop Function

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

11. Low Power Modes

11.1 Overview

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

Table 11.1 lists the specifications of the low power mode functions. 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 the low power mode functions

Parameter	Specification
Reducing power consumption by switching clock signals	The frequency division ratio can be selected independently for the system clock (ICLK), peripheral module clock (PCLKA, PCLKB, PCLKC, PCLKD) and flash interface clock (FCLK)*1
Module-stop state	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
CPU	Stop (Retained)	Stop (Retained)	Stop (Retained)
SRAM (ECC SRAM included)	Selectable	Stop (Retained)	Selectable
Flash memory	Operating	Stop (Retained)	Stop (Retained)
DMA Controller (DMAC)	Selectable	Stop (Retained)	Operation prohibited
Data Transfer Controller (DTC)	Selectable	Stop (Retained)	Selectable

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

Parameter	Sleep mode	Software Standby mode	Snooze mode*1
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*11
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, 4, 9)	Selectable	Stop (Retained)	Operation prohibited
I^{2}C Bus Interface (IIC0)	Selectable	Selectable	Selectable
I^{2}C Bus Interface (IIC1)	Selectable	Stop (Retained)	Operation prohibited
Event Link Controller (ELC)	Selectable	Stop (Retained)	Selectable*8
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 4, 6, 7, 9, 11, 14, 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	Operating

Note: Selectable means that operating or not operating can be selected in 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 the Option Function Select Register 0 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 ACMPLP uses no digital filter. For details on digital filter, see section 38, Low Power Analog Comparator (ACMPLP).

Note 10. Serial communication modes of SCI0 is only in asynchronous mode.

Note 11. When using the ADC14 in Snooze mode, the ADCMPCR.CMPAE or ADCMPCR.CMPBE bit must be 1.

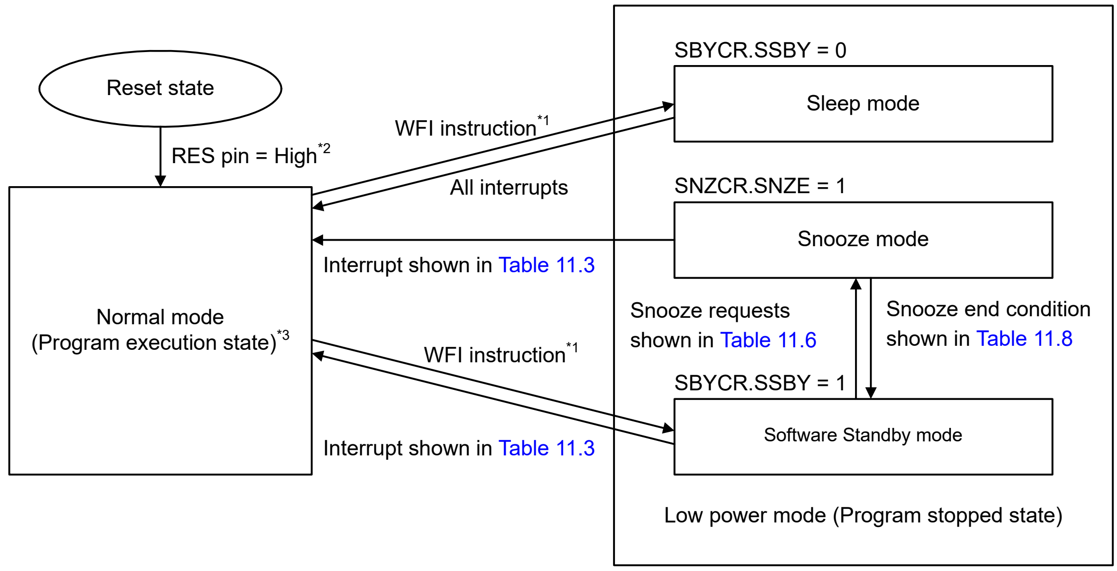
Table 11.3 Interrupt sources to transition to Normal mode from Snooze mode and Software Standby mode

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
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	Yes
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.


Reset state
RES pin = High*2
Normal mode
(Program execution state)*3
WFI instruction*1
All interrupts
Interrupt shown in Table 11.3
WFI instruction*1
Interrupt shown in Table 11.3
SBYCR.SSBY = 0
Sleep mode
SNZCR.SNZE = 1
Snooze mode
Snooze requests
shown in Table 11.6
Snooze end condition
shown in Table 11.8
SBYCR.SSBY = 1
Software Standby mode
Low power mode (Program stopped state)

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 a transition to low power mode.
Note 2. The MOCO is the source of the operating clock following a transition from the reset state to Normal mode.
Note 3. The transition to Normal mode is made from an interrupt in Sleep mode, Software Standby mode, or Snooze mode. The clock source is the same as before entering the low power mode.

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, BC1, and ALE) 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 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
	—	—	—	—	—	—	—	—	—	MSTPA 22	—	—	—	—	—	—
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
	—	—	—	—	—	—	—	—	—	MSTPA 6	—	—	—	—	—	MSTPA 0
Value after reset:	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	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
b5 to b1	—	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
	MSTPB 31	MSTPB 30	—	MSTPB 28	MSTPB 27	—	—	—	—	MSTPB 22	—	—	MSTPB 19	MSTPB 18	—	—
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
	—	—	—	—	MSTPB 11	—	MSTPB 9	MSTPB 8	—	—	—	—	—	MSTPB 2	—	—
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

Bit	Symbol	Bit name	Description	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
b7 to b3	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b8	MSTPB8	I^2C Bus Interface 1 Module Stop	Target module: IIC1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b9	MSTPB9	I^2C Bus Interface 0 Module Stop	Target module: IIC0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
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 Module stop for BLE*3	Target module: SCI for BLE 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b29	—	Reserved	These bits are read as 1. The write value should be 1.	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, 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, then execute a WFI instruction.

Note 3. The Bluetooth middleware executes processing in response to this module stop. Do not change the setting of this bit.

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
	MSTPC 31	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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
	—	MSTPC 14	MSTPC 13	—	—	—	—	—	—	—	—	MSTPC 4	MSTPC 3	—	MSTPC 1	MSTPC 0
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
b12 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	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 to this bit, wait for 2 cycles of the slowest clock from the clocks output by the oscillators, and then execute a WFI instruction.

Note 2. Set the MSTPC31 bit to 0 once at the beginning of the program, to initialize an 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
	MSTPD 31	—	MSTPD 29	—	—	—	—	—	—	—	—	MSTPD 20	MSTPD 19	—	—	MSTPD 16
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
	—	MSTPD 14	—	—	—	—	—	—	—	MSTPD 6	MSTPD 5	—	MSTPD 3	MSTPD 2	—	—
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 169 to 164 Module Stop	Target module: GPT168, GPT165, GPT164 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
b18, b17	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b19	MSTPD19	8-Bit D/A Converter Module Stop*3	Target module: DAC8 0: Cancel the module-stop state 1: Enter the module-stop state.	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
b28 to b21	—	Reserved	These bits are read as 1. The write value should be 1.	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	—	Reserved	This bit is read as 1. The write value should be 1.	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.

Note 3. When using the 8-bit D/A converter (MSTPD19 = 0), set the MSTPD29 bit in ACMPLP to 0.

11.2.6 Operating Power Control Register (OPCCR)

Address(es): SYSTEM.OPCCR 4001 E0A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OPCM TSF	—	—	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 completed 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. Power consumption can be reduced according to the operating frequency and operating voltage used by the OPCCR setting.

For the procedure to change the operating power control modes, see section 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, 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 is not yet 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 becomes 1 when the OPCM[1:0] bits are written, and 0 when mode transition completed. Read this flag and confirm that it 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
	—	—	—	SOPC MTSF	—	—	—	SOPC M
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 completed 1: Transition in progress.	R

Bit	Symbol	Bit name	Description	R/W
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. Setting this register initiates the 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.

The flash cache function should be disabled by setting the CACHEE.FCACHEEN bit to 0 before switching the operating power control mode. For details, see section 43., Flash Memory.

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]) that was active 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 operating power control mode is switched from or to Subosc-speed mode. This flag becomes 1 when the SOPCM bit is written, and 0 when mode transition completed. Read this flag and confirm that it is 0 before proceeding.

Table 11.4 shows each operating power control mode.

Table 11.4 Operating power control mode

Operating power control mode	OPCM[1:0] bits	SOPCM bit	Power consumption
High-speed mode	00b	0	High
Middle-speed mode	01b	0	↓
Low-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 RXD0 falling edge in Software Standby mode 1: Detect RXD0 falling edge in Software Standby mode.	R/W
b1	SNZDTCEN	DTC Enable in Snooze mode	Enable DTC operation in Snooze mode: 0: Disable DTC operation 1: Enable DTC operation.	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 operates 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 specifies whether to enable a transition from Software Standby mode to Snooze mode. To use Snooze mode, set this bit to 1 before entering Software Standby mode. When this bit is set to 1, a trigger as shown in Table 11.6 in Software Standby mode causes the MCU to enter Snooze mode. After the MCU transitions from Software Standby mode or Snooze mode to Normal mode, clear the SNZE bit once 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 a trigger 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 to return to Normal mode from Snooze mode listed in Table 11.3 must not be enabled by SNZEDCR.

AGTUNFED bit (AGT1 Underflow Snooze End Enable)

The AGTUNFED bit specifies whether to enable a transition from Snooze mode to Software Standby mode by an AGT1 underflow. For details of the condition of the trigger, see section 24, Asynchronous General Purpose Timer (AGT).

DTCZRED bit (Last DTC Transmission Completion Snooze End Enable)

The DTCZRED bit specifies whether to enable a transition from Snooze mode to Software Standby mode by completion of the last DTC transmission, that is, CRA or CRB registers in the DTC is 0. For details of the condition of the trigger, see section 18, Data Transfer Controller (DTC).

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

The DTCNZRED bit specifies whether to enable a transition from Snooze mode to Software Standby mode by completion of each DTC transmission, that is, CRA or CRB registers in the DTC is not 0. For details of the condition of the trigger, see section 18, Data Transfer Controller (DTC).

AD0MATED bit (ADC140 Compare Match Snooze End Enable)

The AD0MATED bit specifies whether to enable a transition from Snooze mode to Software Standby mode by an ADC140 event when a conversion result matches the expected data. For details of the condition of the trigger, see section 34, 14-Bit A/D Converter (ADC14).

AD0UMTED bit (ADC140 Compare Mismatch Snooze End Enable)

The AD0UMTED bit specifies whether to enable a transition from Snooze mode to Software Standby mode by an ADC140 event when the conversion result does not match the expected data. For details of the condition of the trigger, see section 34, 14-Bit A/D Converter (ADC14).

SCI0UMTED bit (SCI0 Address Mismatch Snooze End Enable)

The SCI0UMTED bit specifies whether to enable a transition from Snooze mode to Software Standby mode by an SCI0 event when an address received in Software Standby mode does not match the expected data. For details of the condition of the trigger, see section 29, Serial Communications Interface (SCI). Only set this bit to 1 when SCI0 operates 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	—	—	SNZREQEN11	—	SNZREQEN9	—	SNZREQEN7	SNZREQEN6	—	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	Enable IRQ0 pin snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W
b1	SNZREQEN1	Snooze Request Enable 1	Enable IRQ1 pin snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W
b2	SNZREQEN2	Snooze Request Enable 2	Enable IRQ2 pin snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W
b3	SNZREQEN3	Snooze Request Enable 3	Enable IRQ3 pin snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W
b4	SNZREQEN4	Snooze Request Enable 4	Enable IRQ4 pin snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	SNZREQEN6	Snooze Request Enable 6	Enable IRQ6 pin snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W

Bit	Symbol	Bit name	Description	R/W
b7	SNZREQEN7	Snooze Request Enable 7	Enable IRQ7 pin snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9	SNZREQEN9	Snooze Request Enable 9	Enable IRQ9 pin snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W
b10	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11	SNZREQEN11	Snooze Request Enable 11	Enable IRQ11 pin snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W
b13, 12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	SNZREQEN14	Snooze Request Enable 14	Enable IRQ14 pin snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W
b15	SNZREQEN15	Snooze Request Enable 15	Enable IRQ15 pin snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W
b16	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b17	SNZREQEN17	Snooze Request Enable 17	Enable Key Interrupt snooze request: 0: Disable the snooze request 1: Enable the snooze request.	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	Enable ACMPLP0 snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W
b24	SNZREQEN24	Snooze Request Enable 24	Enable RTC alarm snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W
b25	SNZREQEN25	Snooze Request Enable 25	Enable RTC period snooze request: 0: Disable the snooze request 1: Enable the snooze request.	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	Enable AGT1 underflow snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W
b29	SNZREQEN29	Snooze Request Enable 29	Enable AGT1 compare match A snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W
b30	SNZREQEN30	Snooze Request Enable 30	Enable AGT1 compare match B snooze request: 0: Disable the snooze request 1: Enable the snooze request.	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 setting of the WUPEN register always has 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: Code flash and data flash memory operates 1: Code flash and data flash memory stops.	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 completed 1: During transition (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: When changing the value of the FLSTOP bit from 1 to 0 to start flash memory operation, ensure the FLSTPF flag is 0 and OSCSF.HOCOSF is 1 before restarting access to the flash memory. After that, instructions can be executed in the code flash memory.

Note: 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 SRAM 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 SRAM retention area in Software Standby mode is selected with the PSMC[1:0] bits. Supply current can be reduced by setting these bits to 01b (48-KB SRAM in Software Standby mode). A WFI instruction must be executed after setting the PSMC register.

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)

The DBGEN bit enables the on-chip debug mode. This bit must be set to 1 first in the on-chip debugger mode.

[Setting condition]

- Writing 1 to the bit when the debugger is connected.

[Clearing condition]

- Power-on reset is generated
- Writing 0 to the bit.

11.3 Reducing Power Consumption by Switching Clock Signals

When the SCKDIVCR.FCK[2:0], ICK[2:0], PCKA[2:0], PCKB[2:0], PCKC[2:0], and PCKD[2:0] bits are set, the clock frequency changes. The CPU, DMAC, DTC, flash, 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.

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 or the operation of the module is not guaranteed. Also, do not set 1 to the MSTPmi bit while the corresponding module is accessed.

11.5 Function for Lower Operating Power Consumption

By selecting an appropriate operating power consumption control mode according to the operating frequency and operating voltage, power consumption can be reduced in Normal mode, Sleep mode, and Snooze mode.

11.5.1 Setting Operating Power Control Mode

Make sure that the operating condition such as the voltage range and the frequency range is always within the specified range before and after switching the operating power control modes.

This section provides example procedures for switching operating power control modes.

Table 11.5 Available oscillators in each mode

Mode	Oscillator						
	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 3.6 V.

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

Example 1: From High-speed mode to Low-speed mode:

Operation begins in High-speed mode.

1. 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.
3. Set the frequency of each clock lower than the maximum operating frequency in Low-speed mode.
4. Turn off the oscillator that is not required in Low-speed mode.
5. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
6. Set the OPCCR.OPCM bit to 11b (Low-speed mode).
7. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
8. 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 the FCACHEIV.FCACHEIV bit is 0.
    c. Enable the flash cache by setting the FCACHEE.FCACHEEN bit.

Operation is now in Low-speed mode.

Example 2: 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, the main oscillator and PLL.
3. Confirm that all the 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 bit.
  b. Check that FCACHEIV.FCACHEIV bit is 0.
  c. Enable the flash cache by setting FCACHEE.FCACHEEN bit.

Operation is now in Subosc-speed mode.

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

Example 1: From Subosc-speed mode to High-speed mode

Operation begins in Subosc-speed mode.

1. Disable the flash cache by resetting the FCACHEE.FCACHEEN bit 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 the 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: 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 bit.
    b. Check that FCACHEIV.FCACHEIV bit is 0.
    c. Enable the flash cache by setting FCACHEE.FCACHEEN bit.

Operation is now in High-speed mode.

11.5.2 Operating range

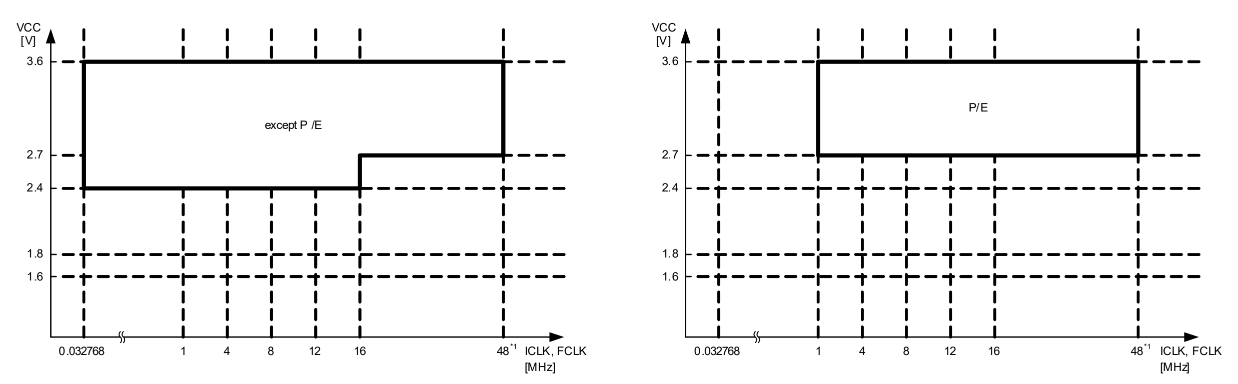
High-speed mode

The maximum operating frequency during flash read is 48 MHz for ICLK and 32 MHz for FCLK. The operating voltage range is 2.4 to 3.6 V during flash read. However, for ICLK and FCLK, the maximum operating frequency 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 and erasure, the operating frequency range is 1 to 48 MHz and the operating voltage range is 2.7 to 3.6 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. Maximum 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 3.6 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 and erasure, the operating frequency range is 1 to 12 MHz and the operating voltage range is 1.8 to 3.6 V. The maximum operating frequency during flash programming and 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.

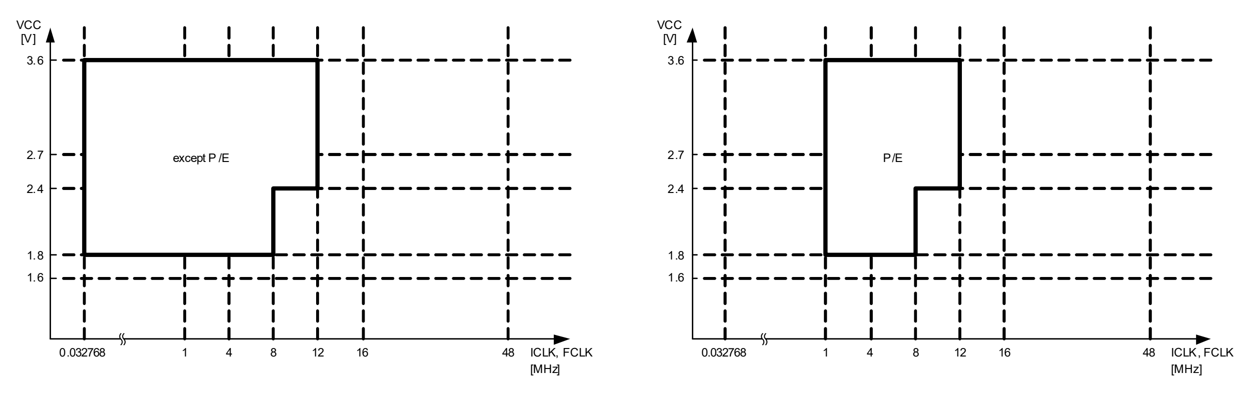


Figure 11.3 Operating voltages and frequencies in Middle-speed mode

Low-voltage mode

After a reset is canceled, operation is started from this mode. Using the PLL is prohibited.

The maximum operating frequency during flash read is 4 MHz for ICLK and FCLK. The operating voltage range is 1.8 to 3.6 V during flash read.

During flash programming and erasure, the operating frequency range is 1 to 4 MHz and the operating voltage range is 1.8 to 3.6 V. Using the PLL is prohibited.

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

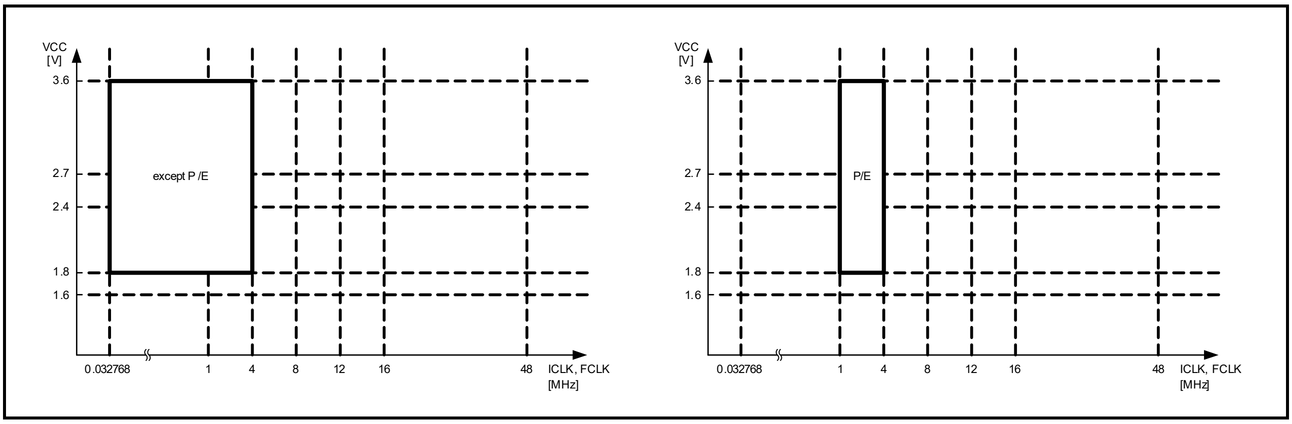


Figure 11.4 Operating voltages and frequencies in Low-voltage mode

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 3.6 V during flash read.

P/E operations for flash memory are prohibited. Using the PLL is prohibited.

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

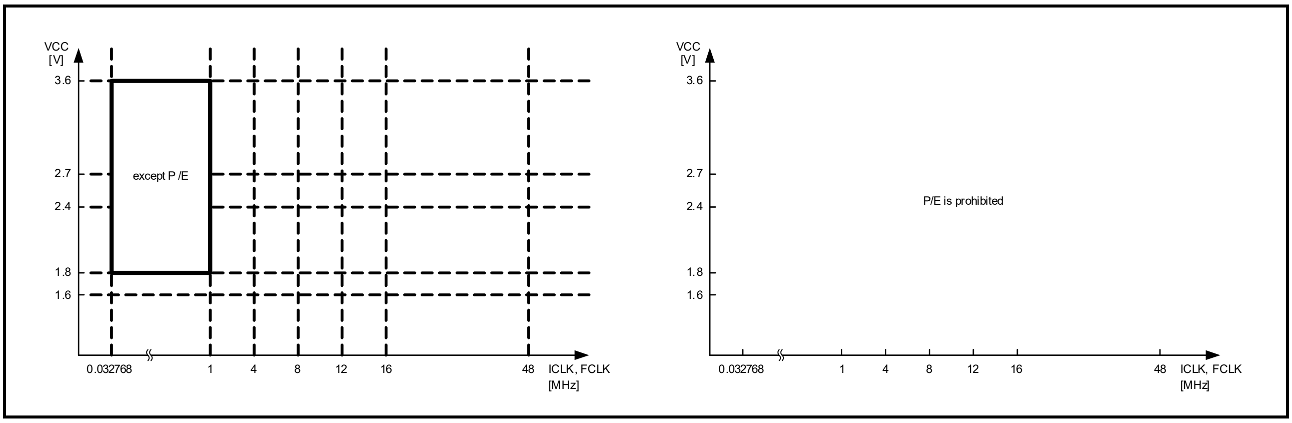


Figure 11.5 Operating voltages and frequencies in Low-speed mode

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 3.6 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 prohibited.

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

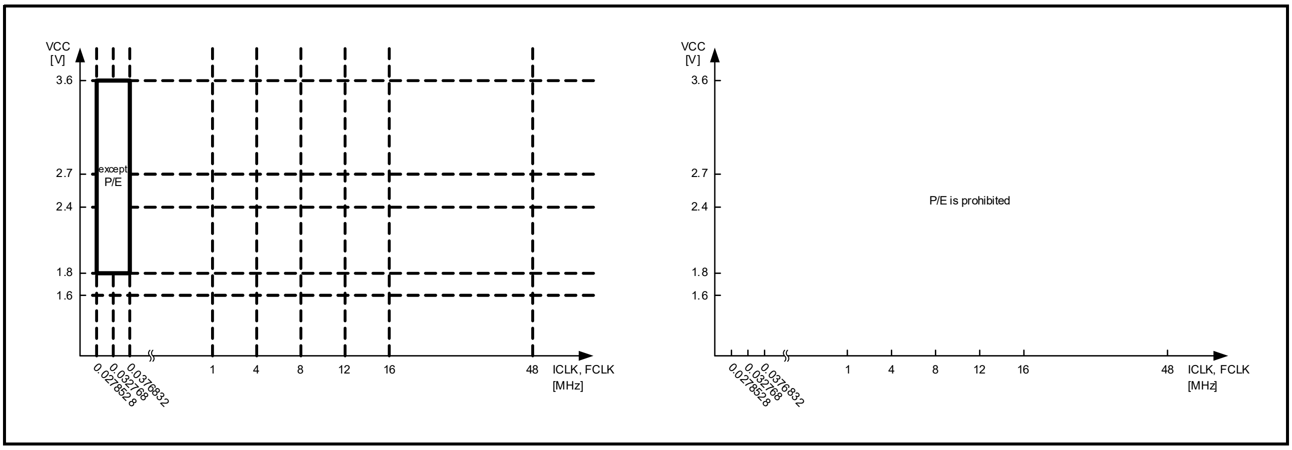


Figure 11.6 Operating voltages and frequencies in Subosc-speed mode

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). Similarly, 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). Similarly, counting by WDT continues when the MCU enters Sleep mode while the WDT is in register start mode and the WDTCSTPR.SLCSTP bit is 0 (WDT does not stop in Sleep mode).

11.6.2 Canceling Sleep Mode

Sleep mode is canceled by:

- An interrupt
- RES pin reset
- A power-on reset
- A voltage monitor reset
- An SRAM parity error reset
- An SRAM ECC error reset
- Bus master MPU error reset
- Bus slave MPU error reset
- A reset caused by an IWDT

- A WDT underflow.

The operations are as follows:

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. Be sure to keep RES pin low for the time period specified in section 48, 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 conditions:
   - OFS0.IWDTSTRT = 0 and OFS0.IWDTSTPCTL = 1.
4. Canceling by WDT reset
   Sleep mode is canceled by an internal reset generated by a WDT underflow and the MCU starts the reset exception handling. However, WDT stops in Sleep mode even when counting in Normal mode and an internal reset for canceling Sleep mode is not generated in the following conditions:
   - OFS0.WDTSTRT = 0 (auto start mode) and OFS0.WDTSTPCTL = 1
   - OFS0.WDTSTRT = 1 (register start mode) and WDTCSTPR.SLCSTP = 1.
5. Canceling by other resets available in Sleep mode
   Sleep mode is canceled by other resets and the MCU starts the reset exception handling.

Note: For details of proper setting of the interrupts, 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 each on-chip peripheral functions and oscillators. Available resets or interrupts in Software Standby mode cause the MCU to cancel Software Standby mode. See Table 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 bit and DTCST.DTCST bit to 0 before executing a WFI instruction, except when using the DTC in Snooze mode. If the 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, or 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 = 1 (oscillation stop detection function is enabled). To enter Software Standby mode, execute a WFI instruction after disabling the oscillation stop detection function (OSTDCR.OSTDE = 0). If executing a WFI instruction while OSTDCR.OSTDE = 1, the MCU enters Sleep mode even when SBYCR.SSBY = 1. In addition, do not enter Software Standby mode while the flash memory performs a programming or erasing procedure. To enter Software Standby mode, execute a WFI instruction after the programming or erasing procedure completes.

11.7.2 Canceling Software Standby Mode

Software Standby mode is canceled by:

- An available interrupt shown in Table 11.3
- RES pin reset
- A power-on reset
- A voltage monitor reset
- A reset caused by an IWDT underflow.

On exiting Software Standby mode, the oscillators that operate before the transition to the mode restart. After all the oscillators are stabilized, the MCU returns to Normal mode from Software Standby mode. See section 14.2.9, Wake Up Interrupt Enable Register (WUPEN) for information on waking up the MCU from Software Standby mode.

You can cancel Software Standby mode from 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 operates before the transition to Software Standby mode restarts. After all the oscillators are stabilized, the MCU returns to Normal mode from Software Standby mode and starts the interrupt handling.
2. Canceling by a RES pin reset
   When RES pin is driven low, the MCU enters the reset state, and the oscillators whose default status is operating, start the oscillation. Be sure to keep the RES pin low for the time period specified in section 48, 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 conditions:
   - OFS0.IWDTSTRT = 0 and OFS0.IWDTSTPCTL = 1.

11.7.3 Example of Software Standby Mode Application

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 by a rising edge of the IRQn pin.

In this example, an IRQn pin interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 01b (falling edge) in Normal mode, and the IRQCRi.IRQMD[1:0] bits set to 10b (rising edge). Follow that, the SBYCR.SSBY bit is set to 1 and a WFI instruction is executed. As a result, entry to Software Standby mode completes and exit from Software Standby mode is initiated by a rising edge of the IRQn pin.

Setting the ICU is also required to exit Software Standby mode. For details, see section 14, Interrupt Controller Unit (ICU).

The oscillation stabilization time in Figure 11.7 is specified in section 48, Electrical Characteristics.

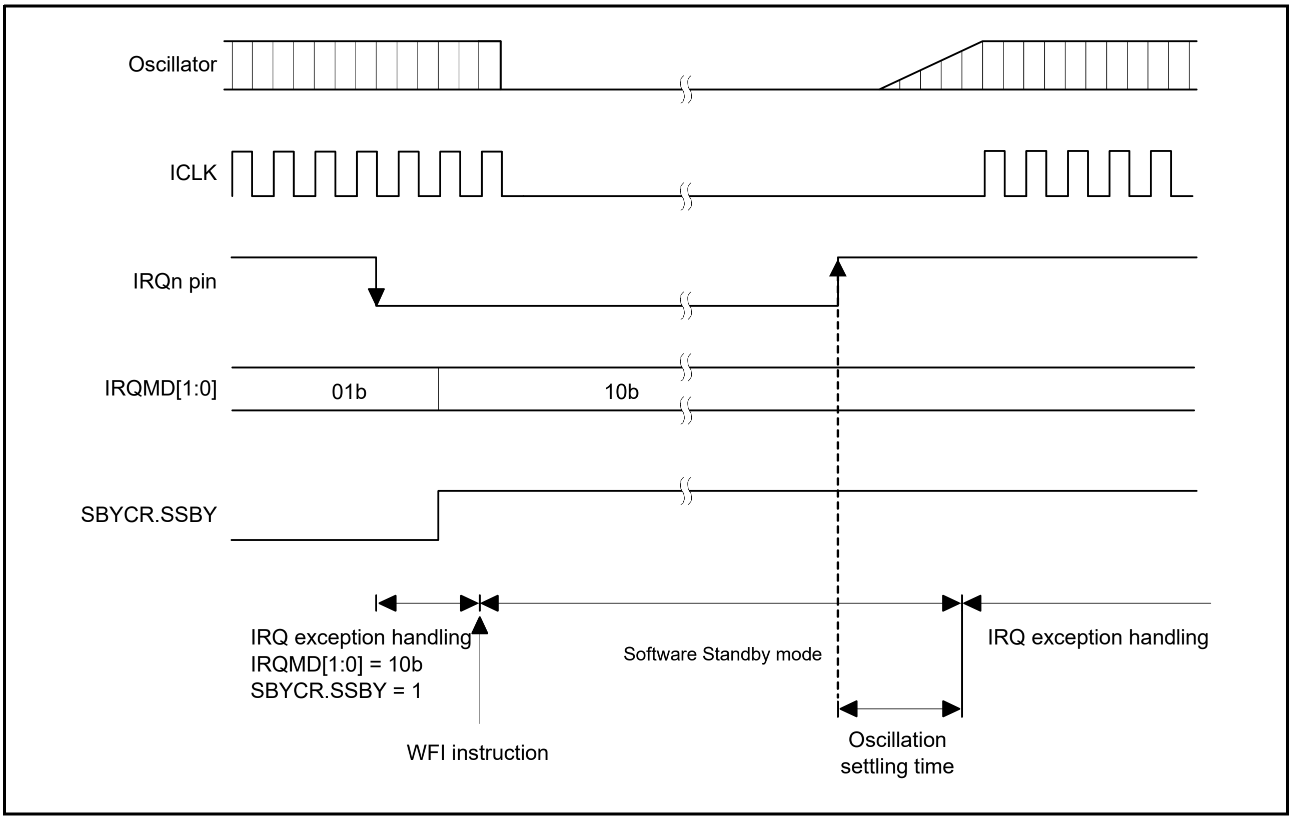

Oscillator
ICLK
IRQn pin
IRQMD[1:0]
01b
10b
SBYCR.SSBY
IRQ exception handling
IRQMD[1:0] = 10b
SBYCR.SSBY = 1
Software Standby mode
IRQ exception handling
WFI instruction
Oscillation
settling time

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. DTC operation in Snooze mode can also be selected by setting the SNZCR.SNZDTCEN bit.

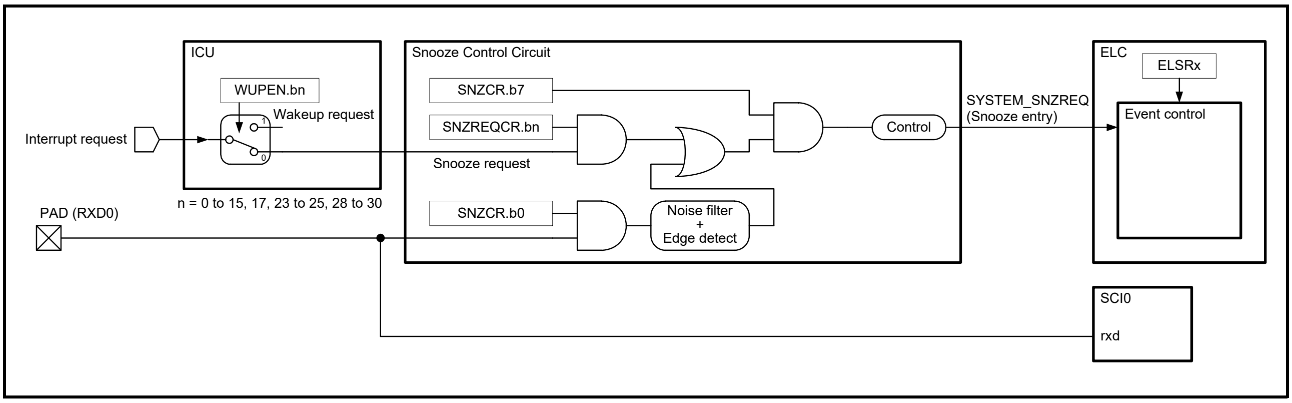

ICU
WUPEN.bn
Wakeup request
Interrupt request
1
0
n = 0 to 15, 17, 23 to 25, 28 to 30
PAD (RXD0)
Snooze Control Circuit
SNZCR.b7
SNZREQCR.bn
Snooze request
SNZCR.b0
Noise filter
+
Edge detect
Control
SYSTEM_SNZREQ
(Snooze entry)
ELC
ELSRx
Event control
SCI0
rxd

Figure 11.8 Snooze mode entry configuration

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

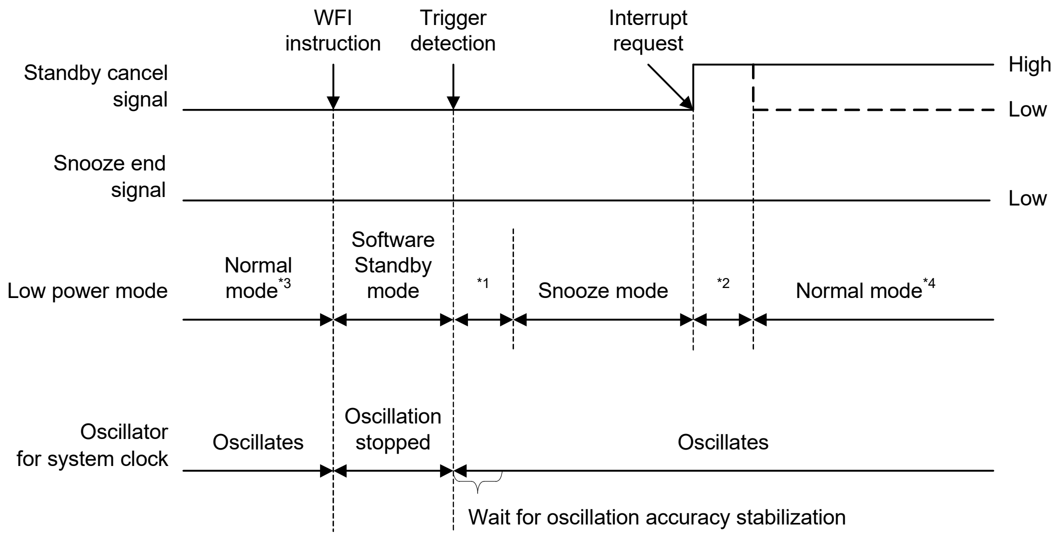
Table 11.6 Available snooze requests to switch to Snooze mode

	Control Register	
Snooze request	**Register**	**Bit**
PORT_IRQn (n = 0 to 4, 6, 7, 9, 11, 14, 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. RXDREQEN bit must not be set to 1 except in asynchronous mode.

11.8.2 Canceling Snooze Mode

Snooze mode is canceled by an interrupt request that is available in Software Standby mode or a reset. Table 11.3 shows the requests that can be used to exit each mode. After canceling the Snooze mode, the MCU enters Normal mode and proceeds with exception processing for the given interrupt or reset. The action triggered by the interrupt requests that is selected by SELSR0, cancels Snooze mode. Interrupt canceling Snooze mode must be selected in IELSRn (n = 0 to 31) to link to the NVIC for the corresponding interrupt handling. See section 14, Interrupt Controller Unit (ICU) for the setting of SELSR0 and IELSRn registers.


WFI instruction
Trigger detection
Interrupt request
Standby cancel signal
High
Low
Snooze end signal
Low
Low power mode
Normal mode*3
Software Standby mode
*1
Snooze mode
*2
Normal mode*4
Oscillator for system clock
Oscillates
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

Note 1. Transition time from Software Standby mode to Snooze mode.
Note 2. Transition time to Normal mode from Snooze mode.
Note 3. Enable Snooze mode (SNZCR.SNZE = 1) immediately before switching to Software Standby mode.
Note 4. Be sure to disable Snooze mode (SNZCR.SNZE = 0) immediately after return to the Normal mode.

Figure 11.9 When interrupt request signal is generated in Snooze mode

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 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 diagram for the transition from Snooze mode to Software Standby mode. This mode transition occurs according to the snooze end requests set in the SNZEDCR register. A snooze request is cleared automatically after the transition to Software Standby mode.

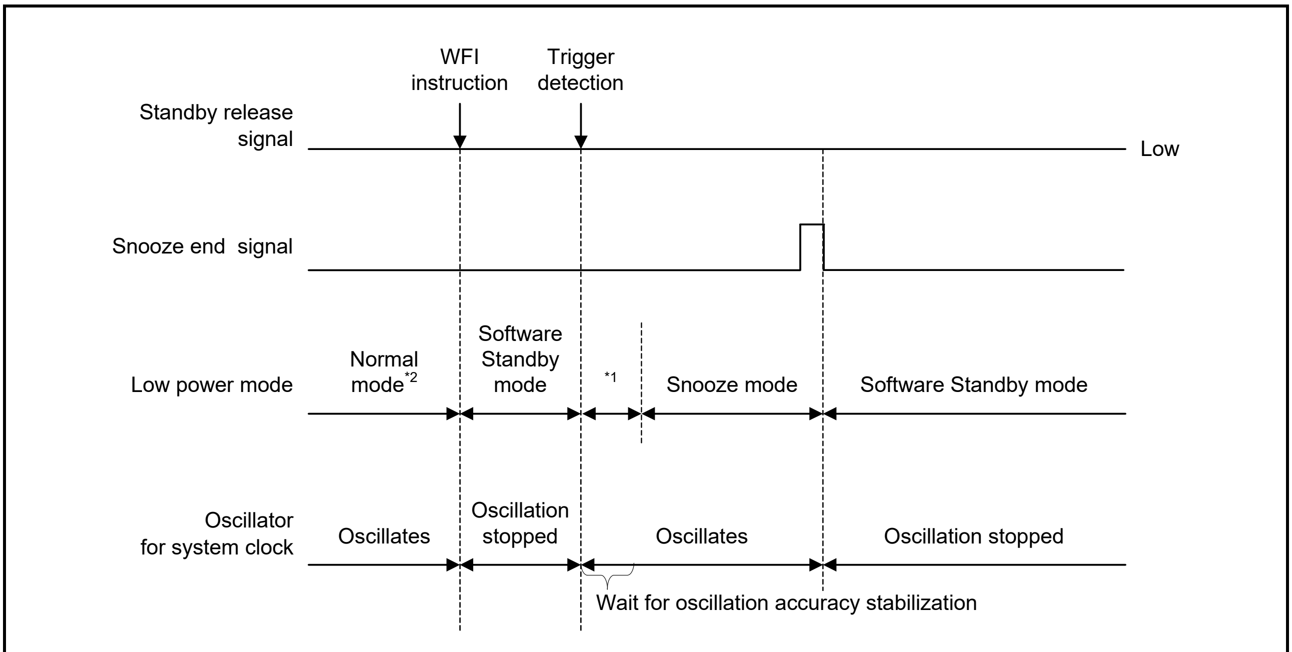
Table 11.7 Available snooze end requests (triggers for transition to Software Standby mode)

	Enable/disable control	
Snooze end request	**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

Operating module when a snooze end request occurs	Snooze end request	
	AGT1 underflow	**Other than AGT1 underflow**
DTC	The MCU transitions to Software Standby mode after all of the modules listed to the left of this column complete operation	The MCU transitions to Software Standby mode after all of the modules complete operation
ADC140		
CTSU		
SCI0	The MCU transitions to Software Standby mode immediately after a snooze end request is generated	
Other than above	The MCU transitions to 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.


WFI instruction
Trigger detection
Standby release signal
Low
Snooze end signal
Low power mode
Normal mode*2
Software Standby mode
*1
Snooze mode
Software Standby mode
Oscillator for system clock
Oscillates
Oscillation stopped
Oscillates
Oscillation stopped
Wait for oscillation accuracy stabilization

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 When interrupt request signal is not generated in Snooze mode

11.8.4 Snooze Operation Example

Figure 11.11 shows an example setting for using ELC in Snooze mode.

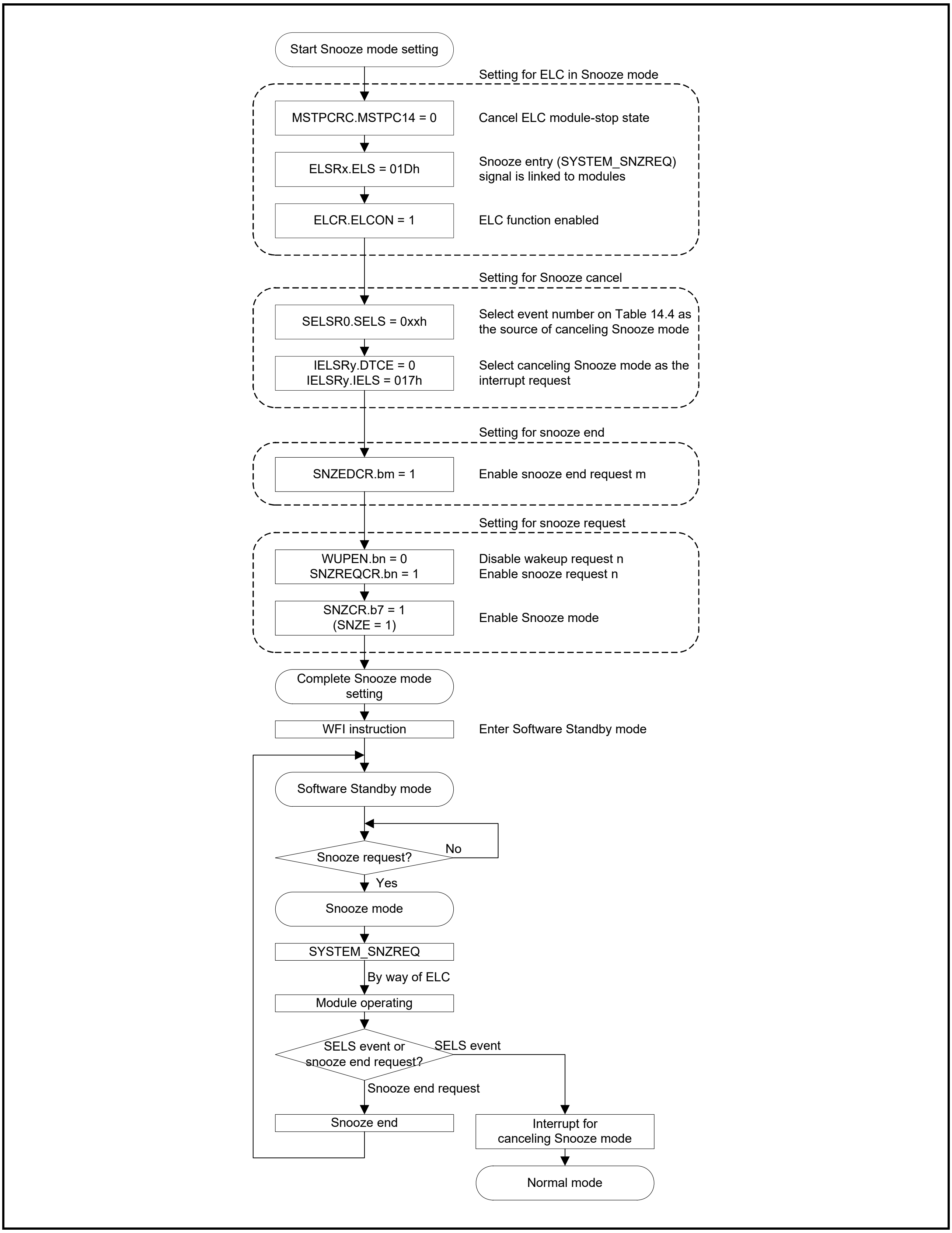

Start Snooze mode setting
Setting for ELC in Snooze mode
MSTPCRC.MSTPC14 = 0
Cancel ELC module-stop state
ELSRx.ELS = 01Dh
Snooze entry (SYSTEM_SNZREQ) signal is linked to modules
ELCR.ELCON = 1
ELC function enabled
Setting for Snooze cancel
SELSR0.SELS = 0xxh
Select event number on Table 14.4 as the source of canceling Snooze mode
IELSRy.DTCE = 0
IELSRy.IELS = 017h
Select canceling Snooze mode as the interrupt request
Setting for snooze end
SNZEDCR.bm = 1
Enable snooze end request m
Setting for snooze request
WUPEN.bn = 0
SNZREQCR.bn = 1
Disable wakeup request n
Enable snooze request n
SNZCR.b7 = 1
(SNZE = 1)
Enable Snooze mode
Complete Snooze mode setting
WFI instruction
Enter Software Standby mode
Software Standby mode
Snooze request?
No
Yes
Snooze mode
SYSTEM_SNZREQ
By way of ELC
Module operating
SELS event or snooze end request?
SELS event
Snooze end request
Snooze end
Interrupt for canceling Snooze mode
Normal mode

Figure 11.11 Setting example of 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
- Low-speed mode.

Do not use Low-voltage mode or Subosc-speed mode.

Table 11.9 and Table 11.10 show the maximum transfer rate of SCI0 in Snooze mode. When SCI0 is used in the Snooze mode, set the bits BGDM to 0, ABCS to 0, and ABCSE to 0. See section 29, Serial Communications Interface (SCI) for details.

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, 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 9600 bps.
Note 2. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 1Eh, SCI0.MDDR = CEh must be used for 9600 bps.
Note 3. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 0Dh, SCI0.MDDR = BAh must be used for 9600 bps.
Note 4. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 32h, SCI0.MDDR = FEh must be used for 9600 bps.
Note 5. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 18h, SCI0.MDDR = F9h must be used for 9600 bps.

High-speed mode, Middle-speed mode, Low-speed mode

Table 11.10 HOCO: ± 2.0% (Ta = -40 to -20°C) **(Unit: bps)**

Maximum division ratio of ICLK, PCLKA, PCLKB, PCLKC, PCLKD, FCLK, 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.

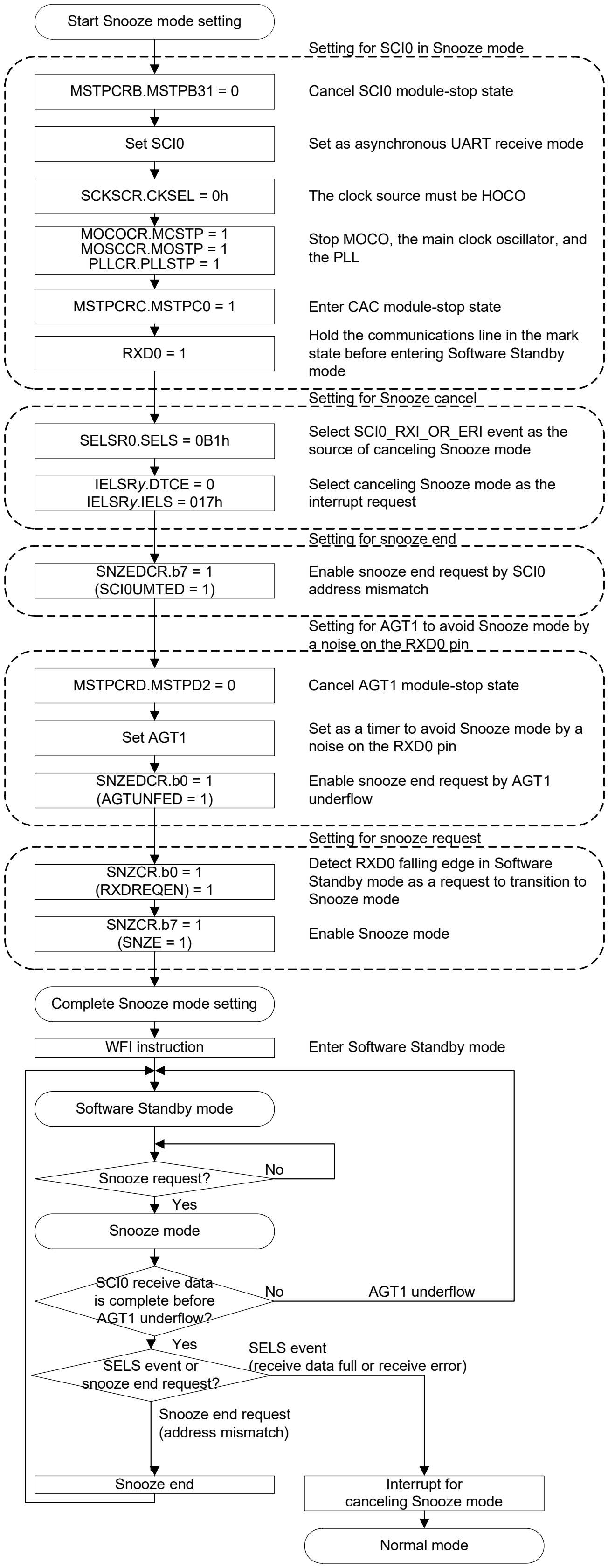

Start Snooze mode setting
Setting for SCI0 in Snooze mode
MSTPCRB.MSTPB31 = 0
Cancel SCI0 module-stop state
Set SCI0
Set as asynchronous UART receive mode
SCKSCR.CKSEL = 0h
The clock source must be HOCO
MOCOCR.MCSTP = 1
MOSCCR.MOSTP = 1
PLLCR.PLLSTP = 1
Stop MOCO, the main clock oscillator, and the PLL
MSTPCRC.MSTPC0 = 1
Enter CAC module-stop state
RXD0 = 1
Hold the communications line in the mark state before entering Software Standby mode
Setting for Snooze cancel
SELSR0.SELS = 0B1h
Select SCI0_RXI_OR_ERI event as the source of canceling Snooze mode
IELSRy.DTCE = 0
IELSRy.IELS = 017h
Select canceling Snooze mode as the interrupt request
Setting for snooze end
SNZEDCR.b7 = 1
(SCI0UMTED = 1)
Enable snooze end request by SCI0 address mismatch
Setting for AGT1 to avoid Snooze mode by a noise on the RXD0 pin
MSTPCRD.MSTPD2 = 0
Cancel AGT1 module-stop state
Set AGT1
Set as a timer to avoid Snooze mode by a noise on the RXD0 pin
SNZEDCR.b0 = 1
(AGTUNFED = 1)
Enable snooze end request by AGT1 underflow
Setting for snooze request
SNZCR.b0 = 1
(RXDREQEN) = 1
Detect RXD0 falling edge in Software Standby mode as a request to transition to Snooze mode
SNZCR.b7 = 1
(SNZE = 1)
Enable Snooze mode
Complete Snooze mode setting
WFI instruction
Enter Software Standby mode
Software Standby mode
Snooze request?
No
Yes
Snooze mode
SCI0 receive data is complete before AGT1 underflow?
No
AGT1 underflow
Yes
SELS event or snooze end request?
SELS event
(receive data full or receive error)
Snooze end request
(address mismatch)
Snooze end
Interrupt for canceling Snooze mode
Normal mode

Figure 11.12 Setting example of using SCI0 in Snooze mode entry

11.9 Usage Notes

11.9.1 Register Access

(1) Do not write to registers listed in this section in any of the following conditions:

[Registers]

- All registers with a peripheral name of SYSTEM.

[Conditions]

- OPCCR.OPCMTSF = 1 or SOPCCR.SOPCMTSF = 1 (during transition of the operating power control mode)
- Time period from executing a WFI instruction to returning to Normal mode
- FENTRYR.FENTRY0 = 1 or FENTRYR.FENTRYD = 1 (flash P/E mode, data flash P/E mode)
- FLSTOP.FLSTPF = 1 (during transition).

(2) Valid setting of 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, otherwise it is ignored. Additionally, each register has some prohibited settings under certain conditions other than the operating power control modes. See section 9, Clock Generation Circuit for other conditions of each register.

Table 11.11 Valid setting for clock-related registers (1)

	Valid setting								
Mode	**SCKSCR.CKSEL[2:0], CKOCR.CKOSEL[2:0]**	**SCKDIVCR.FCK[2:0], SCKDIVCR.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 clock-related registers (2)

	Valid setting	
Operating oscillator	**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) Do not write to registers listed in this section for the following condition:

[Registers]

- SCKSCR, OPCCR.

[Condition]

- SOPCCR.SOPCM = 1 (Subosc-speed mode).

(4) Do not write to registers listed in this section by DTC or DMAC:

[Registers]

- MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD.

(5) Do not write to registers listed in this section in Snooze mode. They must be set before entering Software Standby mode:

[Registers]

- SNZCR, SNZEDCR, SNZREQCR.

(6) Do not set the FLSTOP.FLSTOP bit to 1 in any of the following 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) Do not set the MEMWAIT.MEMWAIT bit to 1 in any of the following 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) Write access to registers listed in this section is invalid when 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, unless modifying in Snooze mode, are the same before entering the modes. Therefore, the supply current 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 setting the module-stop bit while an interrupt request is generated, a CPU interrupt source or a DMAC or DTC startup source cannot be cleared. For this reason, make sure you 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 and CS area. To avoid this problem, it is recommended that you 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 registers in WDT or IWDT by DMAC or DTC while WDT or IWDT stops by entering Sleep mode or Snooze mode.

11.9.8 Oscillators in Snooze Mode

Oscillators that stop by entering Software Standby mode automatically restart when a trigger to switch to Snooze mode is generated. The MCU does not enter Snooze mode until all the oscillators stabilize. If in Snooze mode, make sure to disable oscillators that are not required in Snooze mode before entering Software Standby mode, otherwise it takes longer to transition from Software Standby mode to Snooze mode.

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 transition from Software Standby mode to Snooze mode. Any subsequent RXD0 data can be received in Snooze mode by a 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, and an address mismatch event is not generated, 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 A/D converter can only be triggered by the ELC in Snooze mode. Software trigger or ADTRG0 pin must not be used.

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 Software Standby mode, it is recommended that you set the ADC140 module-stop state to reduce power consumption. In this case, the ADC140 can be available in Snooze mode by releasing the ADC140 module-stop using the DTC. Similarly, set the module-stop state using the DTC before returning to Software Standby mode from 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 this User's Manual), by up to 600 μA. So, initialize the unused circuit using the steps shown in Figure 11.13.

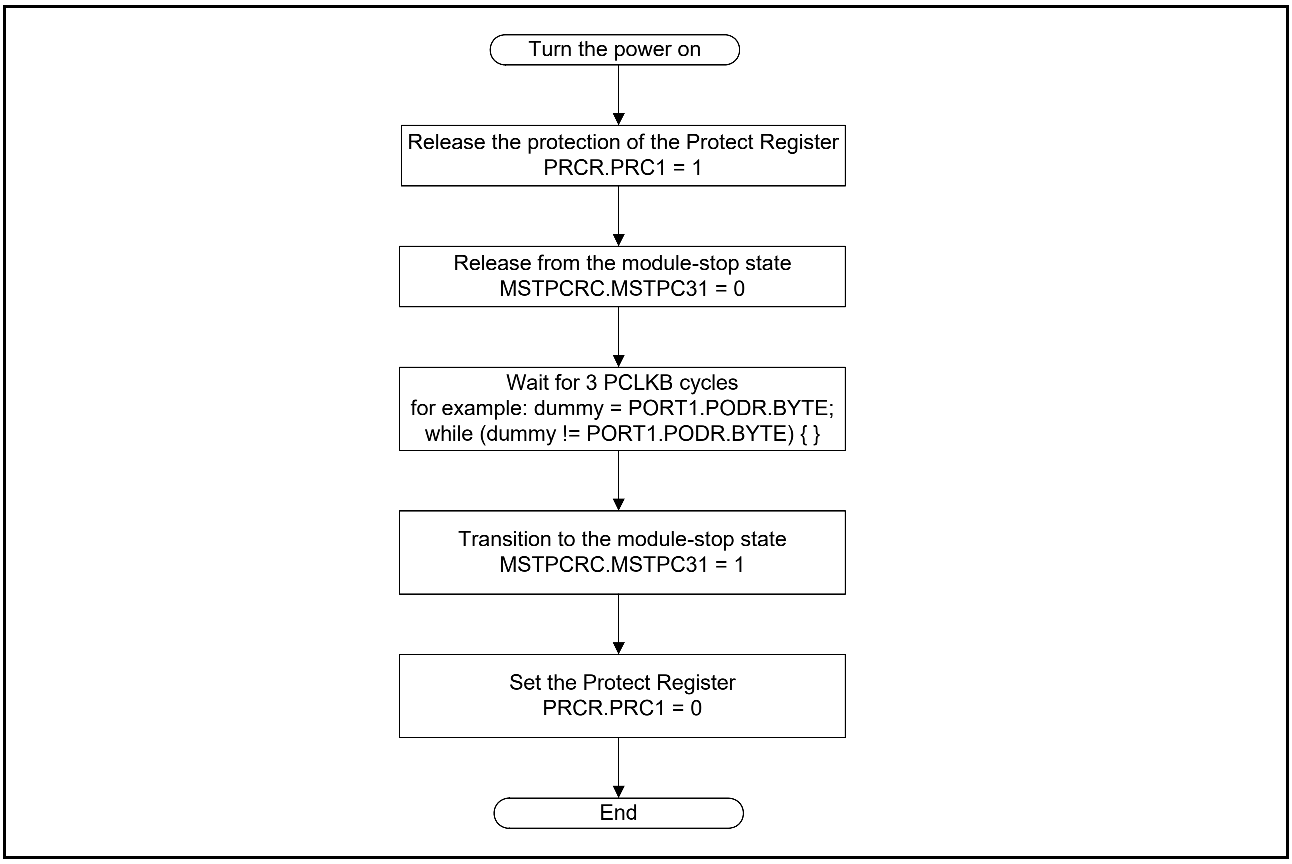

Turn the power on
Release the protection of the Protect Register
PRCR.PRC1 = 1
Release from the module-stop state
MSTPCRC.MSTPC31 = 0
Wait for 3 PCLKB cycles
for example: dummy = PORT1.PODR.BYTE;
while (dummy != PORT1.PODR.BYTE) { }
Transition to the module-stop state
MSTPCRC.MSTPC31 = 1
Set the Protect Register
PRCR.PRC1 = 0
End

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, the battery-powered area includes RTC, SOSC, LOCO, Wakeup Control/Backup Memory, VBATT_R Low Voltage Detection, and VBATT 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
VBATWIO0	Input/Output	Output wakeup signal for the VBATT Wakeup Control function. External event input for the VBATT Wakeup Control function.

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. It is possible to detect a low voltage condition of the power supply. The VBATT status register includes a flag to check for this low voltage detections.

12.1.4 VBATT_R Low Voltage Detection

VBATT_R low voltage detection function supports the battery-powered area. This function monitors the VBATT_R voltage level. VBATT_R is the output voltage of the battery power supply switch. This low voltage detection causes a VBATT_POR reset and initializes the battery-powered area. See details in each register description. The VBATT status register includes a flag to check for this low voltage detections.

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 is a function that can toggle the VBATWIO pin when the RTC alarm, periodic signal is asserted when VBATT_R is powered by the VBATT pin.

Note: The toggle triggered by the wakeup control function does not generate an interrupt at the ICU or a reset to the reset module. The use case of this function is that 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.

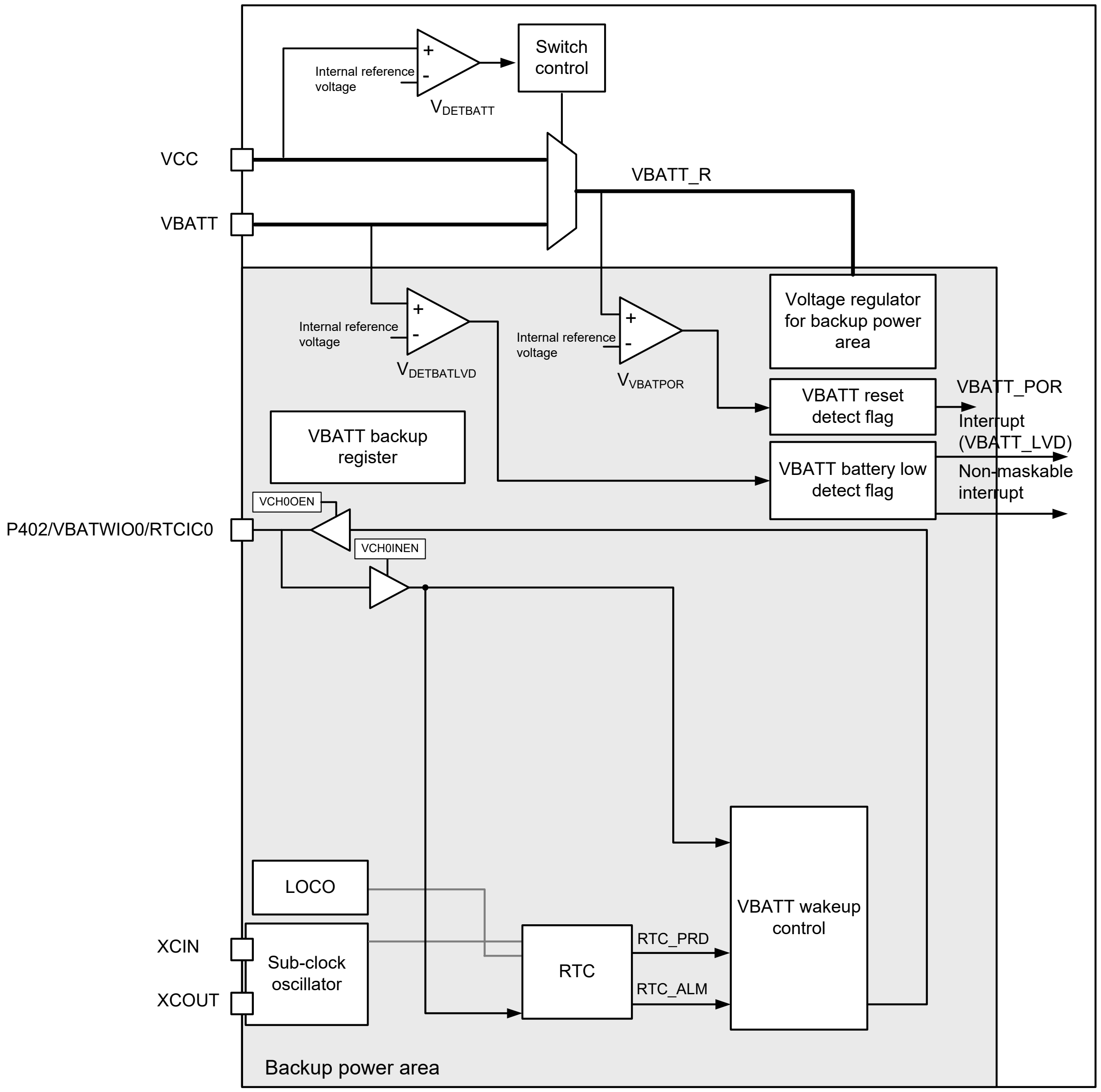
12.1.7 Time capture Pin Detection

The RTC detects input level changes on the time capture pin, RTCIC0.
For the function of the RTCIC0 pin, see section 25, Realtime Clock (RTC). To use RTCIC0 pin, 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 power is turned on, power is not supplied to the RTC, the SOSC (including multiplexed port), or the LOCO before setting the VBTCR1.BPWSWSTP bit to 1. It takes the VBATT_POR reset time tVBATPOR as described in section 48, Electrical Characteristics to supply power to the modules after setting the VBTCR1.BPWSWSTP bit. 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.


Internal reference voltage
+
-
VDETBATT
Switch control
VCC
VBATT
VBATT_R
Internal reference voltage
VDETBATLVD
Internal reference voltage
VVBATPOR
Voltage regulator for backup power area
VBATT reset detect flag
VBATT_POR
Interrupt (VBATT_LVD)
VBATT battery low detect flag
Non-maskable interrupt
VBATT backup register
VCH0OEN
P402/VBATWIO0/RTCIC0
VCH0INEN
LOCO
XCIN
XCOUT
Sub-clock oscillator
RTC
RTC_PRD
RTC_ALM
VBATT wakeup control
Backup power area

VCC	Power supply pin
VBATT	Backup power pin
XCIN	SOSC input pin
XCOUT	SOSC output pin
VBATWIO0	VBATT wakeup output port
RTCIC0	RTC time capture 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: Battery power supply switch enable 1: Battery power supply switch stop.	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 switch the battery backup module supply voltage from VCC to VBATT when the voltage applied to the VCC pin drops. When 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 by 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 other related registers are set when the VBATT function is used.

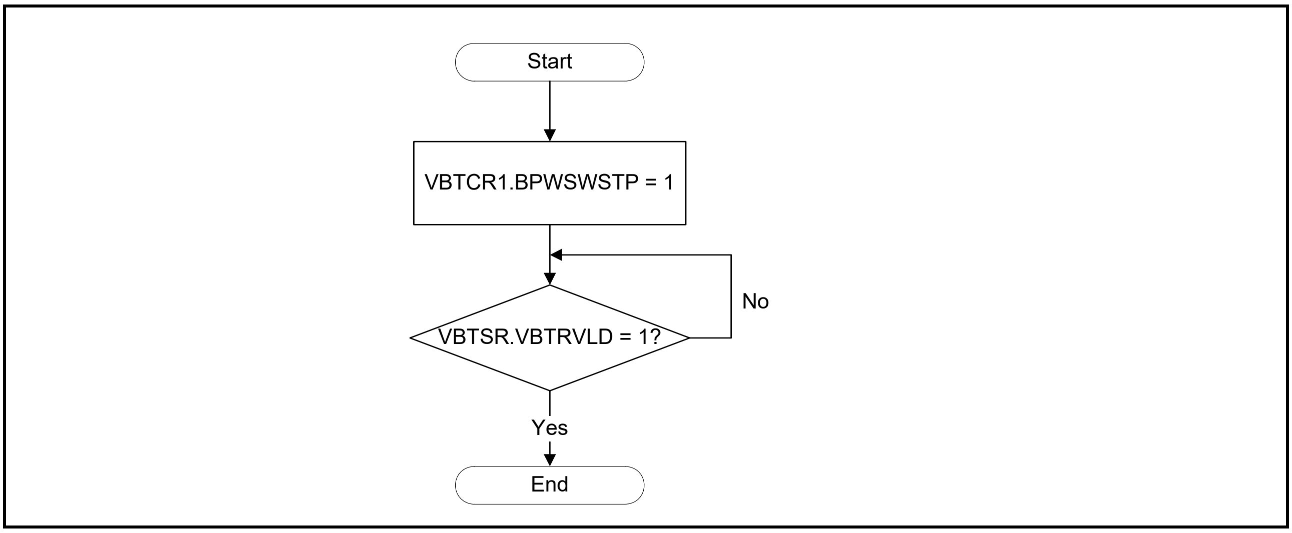


Figure 12.2 Setting flow of the VBTCR1.BPWSWSTP bit

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 48, Electrical Characteristics to exit the loop.
The following registers cannot be accessed when the VBTSR.VBTRVLD bit is 0. 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 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]		—	VBTLV DEN	—	—	—	—
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: VBATT pin low voltage detection disable 1: VBATT pin low voltage detection enable.	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 pin 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
	—	—	—	VBTRV LD	—	—	VBTBL DF	VBTRD F
Value after reset:	0	0	0	0*5	0	0	0*2	1*1

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 set by the VBATT_POR reset.
Note 2. This flag is only reset 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 VBTRDF is read as 1 and 0 is written to VBTRDF.

VBTBLDF flag (VBATT Battery Low Detect Flag)

The VBTBLDF flag indicates that a VBATT pin low voltage detection occurs.

[Setting condition]

- When VBATT pin low voltage detection occurs.

[Clearing condition]

- When VBTBLDF is read as 1 and 0 can be written to VBTBLDF.

VBTRVLD bit (VBATT_R Valid)

Check whether the VBATT area is valid.

The VBTRVLD bit is used to check whether the VBATT_R area is valid. It must 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 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: VBATT pin low voltage detect circuit output disabled 1: VBATT pin low voltage detect circuit output enabled.	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
	—	—	—	—	—	—	VBTLV DISEL	VBTLV DIE
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: VBATT pin low voltage detection interrupt disable 1: VBATT pin low voltage detection interrupt enable.	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 (VBTBKRn) (n = 0 to 511)

Address(es): SYSTEM.VBTBKR0 4001 E500h to SYSTEM.VBTBKR511 4001 E6FFh

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

x: Undefined

VBTBKRn is an 8-bit access read/write register to store data powered by VBATT. The value of this register is retained even in VBATT mode. 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 48, Electrical Characteristics.

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 register controls the VBATT Wakeup function. 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.VCH0OEN bit is set to 1, the VBATWIO0 pin output is low level. When the VWEN bit is set to 1, the output from the VBATWIO0 pin changes to the level specified by the VBTOCTLR.VOUT0LSEL 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:

- VBTWCH0OTSR
- VBTICTLR
- VBTOCTLR
- VBTWTER
- VBTWEGR.

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
	—	—	—	CH0VRTCATE	CH0VRTCTE	—	—	—
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	CH0VRTCTE	VBATWIO0 Output RTC Periodic Signal Enable	0: VBATT wakeup I/O 0 output trigger by the RTC periodic signal is disabled 1: VBATT wakeup I/O 0 output trigger by the RTC periodic signal is enabled.	R/W
b4	CH0VRTCATE	VBATWIO0 Output RTC Alarm Signal Enable	0: VBATT wakeup I/O 0 output trigger by the RTC alarm signal is disabled 1: VBATT wakeup I/O 0 output trigger by the RTC alarm signal is enabled.	R/W
b7 to b5	—	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 this register bit is set to 1 and the associated wakeup trigger flag in the VBTWFR register is set, the VBATWIO0 pin outputs a signal based on the VOUT0LSEL bit in the VBTOCTLR register.

The VBTWCH0OTSR register is initialized by the VBATT_POR signal.

12.2.9 VBATT Input Control Register (VBTICTLR)

Address(es): SYSTEM.VBTICTLR 4001 E4BBh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	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: VBATWIO0, RTCIC0 inputs disabled 1: VBATWIO0, RTCIC0 inputs enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The VBTICTLR register selects VBATT wakeup I/O pin input direction. VBTICTLR is reset by the VBATT_POR signal.

VCH0INEN bit (VBATT Wakeup I/O n Input Enable Bit)

The VCHnINEN bit defines the VBATT wakeup I/O pin input enable. You must set the VBTICTLR register when using only the VBATT wakeup control function but also the time capture function of RTC (RTCIC0). For these functions, see section 25, Realtime Clock (RTC).

12.2.10 VBATT Output Control Register (VBTOCTLR)

Address(es): SYSTEM.VBTOCTLR 4001 E4BCh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	VOUT0LSEL	—	—	VCH0OEN
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: VBATWIO0 output disabled 1: VBATWIO0 output enabled.*1	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	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
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The VBTOCTLR register selects the VBATT wakeup I/O (VBATWIO0) pin output direction and output level. VBTOCTLR is reset by the VBATT_POR signal.

VCH0OEN bit (VBATT Wakeup I/O 0 Output Enable Bit)

The VCH0OEN bit defines the VBATT output enable.

Note 1. When the VCH0OEN bit is set to 1, P402PFS.PMR bit must be 0.

VOUT0LSEL bit (VBATT Wakeup I/O 0 Output Level Selection)

The VOUT0LSEL bit defines the output level from the VBATT wakeup I/O 0 pin. When the VOUT0LSEL bit is set to 0, VBATWIO0 pin outputs low before receiving the VBATT wakeup trigger and high after receiving the VBATT wakeup trigger. When the VOUT0LSEL bit is set to 1, the VBATWIO0 pin outputs high before the VBATT wakeup trigger and low after receiving the VBATT wakeup trigger.

12.2.11 VBATT Wakeup Trigger Source Enable Register (VBTWTER)

Address(es): SYSTEM.VBTWTER 4001 E4BDh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	VRTCAE	VRTCIE	—	—	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: VBATT wakeup triggered by the VBATWIO0 pin is disabled 1: VBATT wakeup triggered by the VBATWIO0 pin is enabled.	R/W
b2 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	VRTCIE	RTC Periodic Signal Enable	0: VBATT wakeup triggered by RTC periodic signal is disabled 1: VBATT wakeup triggered by RTC periodic signal is enabled.	R/W
b4	VRTCAE	RTC Alarm Signal Enable	0: VBATT wakeup triggered by RTC alarm signal is disabled 1: VBATT wakeup triggered by RTC alarm signal is enabled.	R/W

b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The VBTWTER register enables or disables the VBATT wakeup trigger. VBTWTER is reset by the VBATT_POR signal.

Multiple trigger source selection is possible.

12.2.12 VBATT Wakeup Trigger Source Edge Register (VBTWEGR)

Address(es): SYSTEM.VBTWEGR 4001 E4BEh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	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: Wakeup trigger is generated at a falling edge 1: Wakeup trigger is generated at a rising edge.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The VBTWEGR register selects the edge of each VBATT wakeup trigger sources. The VBTWEGR register is reset by the VBATT_POR signal.

12.2.13 VBATT Wakeup Trigger Source Flag Register (VBTWFR)

Address(es): SYSTEM.VBTWFR 4001 E4BFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	VRTCAF	VRTCIF	—	—	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: No wakeup trigger by the VBATWIO0 pin is generated 1: A wakeup trigger by the VBATWIO0 pin is generated.	R/(W)*1
b2 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	VRTCIF	VBATT RTC-Periodic Wakeup Trigger Flag	0: No wakeup trigger by the RTC periodic signal is generated 1: A wakeup trigger by the RTC periodic signal is generated.	R/(W)*1
b4	VRTCAF	VBATT RTC-Alarm Wakeup Trigger Flag	0: No wakeup trigger by the RTC alarm signal is generated 1: A wakeup trigger by the RTC alarm signal is generated.	R/(W)*1
b7 to b5	—	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 flag after reading 1.

The VBTWFR register indicates the triggering factor of the VBATT wakeup control function. This register is protected by the VWEN bit (VBTWCTLR register). VBTWFR is valid 5 PCLKB cycles after writing 1 to VWEN bit enable. Similarly, disabling VBTWFR takes 5 PCLKB cycles after writing 0 to VWEN bit.

Each flag is set to 1 when a trigger request specified by VBTWEGR is generated.

The VBTWFR register is initialized by VBATT_POR.

VCH0F flags (VBATT Wakeup I/O 0 Wakeup Trigger Flag)

These flags indicate that a trigger request by the VBATWIO0 pin is generated.

[Setting condition]

- A trigger request by the VBATWIO0 pin specified by VBTWEGR is generated.

[Clearing condition]

- Each bit is read as 1 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 trigger request by the RTC periodic signal is generated.

[Clearing condition]

- This bit is read as 1 and 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 trigger request by the RTC alarm signal is generated.

[Clearing condition]

- This bit is read as 1 and written as 0.

12.2.14 Backup Register Access Control Register (BKRACR)

Address(es): SYSTEM.BKRACR 4001 E0C6h

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

Bit	Symbol	Bit name	Description	R/W
b2 to b0	BKRACS[2:0]	Backup Register Access cycle Select	b2 b0 0 0 0: Access cycle control disable. When System clock source is SOSC or LOCO. 1 1 0: Access cycle control enable. System clock source is other than SOSC or LOCO. 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 BKRACR register controls the access cycle for the backup register to reduce power consumption. When access cycle control is enabled (110b), the access cycle for the backup register is 64 times that of when it is disabled (000b). The BKRACR is initialized by all the resets except for VBATT_POR.

[Setting Procedure]

To change the system clock from other than SOSC/LOCO to SOSC/LOCO:

1. Change the SCKSCR.CKSEL[2:0] bits.
2. Change the BKRACR.BKRACS[2:0] bits to 000b.

To change the system clock from SOSC/LOCO to other than SOSC/LOCO:

1. Change the BKRACR.BKRACS[2:0] bits to 110b.
2. Change the SCKSCR.CKSEL[2:0] bits.

12.3 Operation

12.3.1 Battery Backup Function

When the voltage at the VCC pin drops, power can be supplied to the RTC, LOCO, and sub-clock oscillator from the VBATT pin. When the power supply drop from the VCC pin is detected, the connection to power is switched from the power supply to the VBATT pin. The power supply from the VCC pin resumes when the voltage at the VCC pin exceeds VDETBATT. This power supply change does not affect the RTC operation. When the voltage level at the VBATT pin voltage drops below the operation-guaranteed voltage, it is possible to monitor the VBTBLDF bit 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 VBATWIO0 by triggering the RTC alarm/periodic signal.

The RTC supports time capture pin detection when the time capture pin input level changes.

The VBATT pin supplies power to the following modules.

- RTC
- Sub-clock oscillator (including XCIN and XCOUT pins)
- VBATWIO0 pin (including RTCIC0)
- 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)

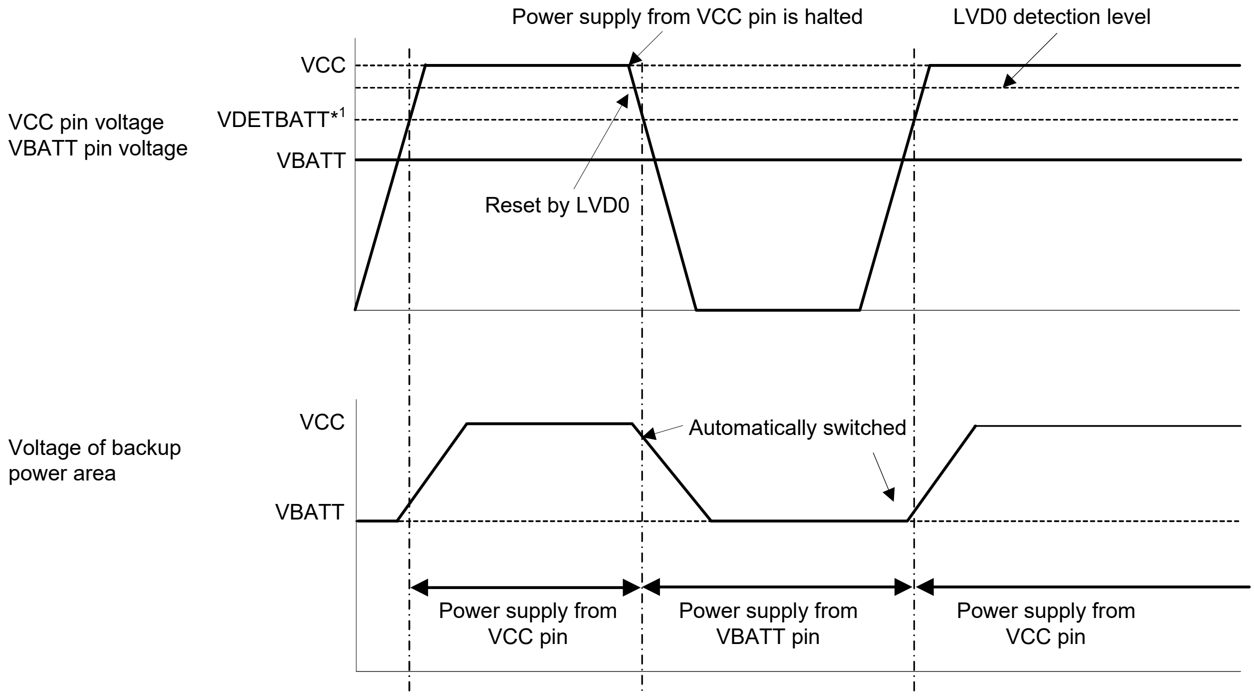
Operating state	VBATT mode
Transition condition	Detection of VCC voltage drop
Canceling method other than reset	Detection of VCC voltage rise
Main clock oscillator	Stopped
Sub-clock oscillator	Operation can be selected by SOSCCR.SOSTP bit. The status of the oscillator is the same as before entering VBATT mode.
High-speed on-chip oscillator	Stopped
Middle-speed on-chip oscillator	Stopped
Low-speed on-chip oscillator	Operation or non-operation can be selected in the LOCOCR.LCSTP bit. The status of the oscillator is the same as before entering VBATT mode.
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 the selecting clock operates as the count source
AGTn (n = 0, 1)	Stopped (undefined)
Low Voltage Detection (LVD)	Stopped

Table 12.2 Operating states in VBATT mode (2 of 2)

Operating state	VBATT mode
Power-on reset circuit	Stopped
Battery backup voltage monitor	Operating
Other peripheral modules	Stopped (undefined)
I/O ports	• RTCIC0 port: Operating • Other than the specified ports: Undefined • VBATWIO0 port: Operating.

Note: Selectable means that operating can be selected in the control register. Some modules are also controlled by the associated 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.


Power supply from VCC pin is halted
LVD0 detection level
VCC
VDETBATT*1
VCC pin voltage
VBATT pin voltage
VBATT
Reset by LVD0
VCC
Automatically switched
Voltage of backup
power area
VBATT
Power supply from
VCC pin
Power supply from
VBATT pin
Power supply from
VCC pin

Note: For details on the electrical characteristics, see section 48, Electrical Characteristics.
Note 1. VDETBATT indicates the threshold level of the power supply change between the VCC pin and the VBATT pin.

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 power supply from the VCC pin to the VBATT pin when the voltage 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.

The BPWSWSTP bit can enable the battery power supply switch which can switch the battery backup module supply voltage from VCC to VBATT when the VCC voltage falls. When the battery power supply switch stops, 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 VDETBATT level (OFS1.VDSEL1[2:0] bits are 001b, or 010b).
Note: This bit can be set without verifying the VBTSR.VBTRVLD bit status.

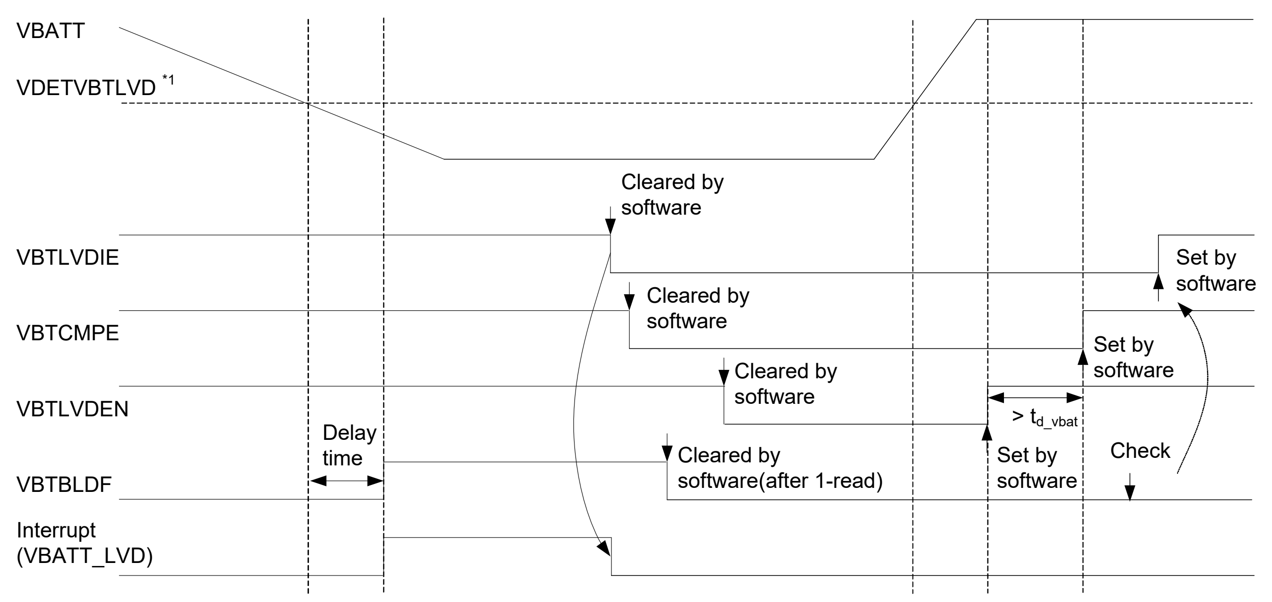
12.3.3 VBATT Pin Low Voltage Detection Procedures

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 the access to this bit is for the first access after a power-on reset.
3. Wait for the VBTSR.VBTRVLD bit to be 1 and ensure 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 for enabling VBATT pin low voltage detection.
7. After waiting for the VBATT comparator operation stabilization time (t_{d_vbat}) as described in section 48, Electrical Characteristics, set the VBTCMPCR.VBTCMPE bit to 1 for the VBATT pin voltage detect circuit to be enabled.
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 output to be enabled.
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.


VBATT
VDETVBTLVD *1
Cleared by software
VBTLVDIE
Set by software
Cleared by software
VBTCMPE
Set by software
Cleared by software
VBTLVDEN
> t_d_vbat
Delay time
Cleared by software(after 1-read)
Set by software
Check
VBTBLDF
Interrupt (VBATT_LVD)

Note 1. VDETVBTLVD is VBATT low voltage detect level selected in the VBTCR2.VBTLVDLVL[1:0] bits.

Figure 12.4 Basic operation of VBATT low voltage detection interrupt

The following procedures show how to unset 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.
3. Set the VBTCMPCR.VBTCMPE bit to 0 for VBATT pin voltage detect circuit output to disable.
4. Set the VBTCR2.VBTLVDEN bit to 0 to disable VBATT pin low voltage output to disable.
5. Modify the setting of bits related to the VBATT pin low voltage detection registers other than

VBTCR2.VBTLVDEN, VBTCMPCR.VBTCMPE, and 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 the following procedure describes:

1. The VBTCR1.BPWSWSTP bit must be set to 1 if the access to this bit is for the first access after a power-on reset.
2. Wait for the VBTSR.VBTRVLD bit to be 1.
3. VBTBKRn 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.

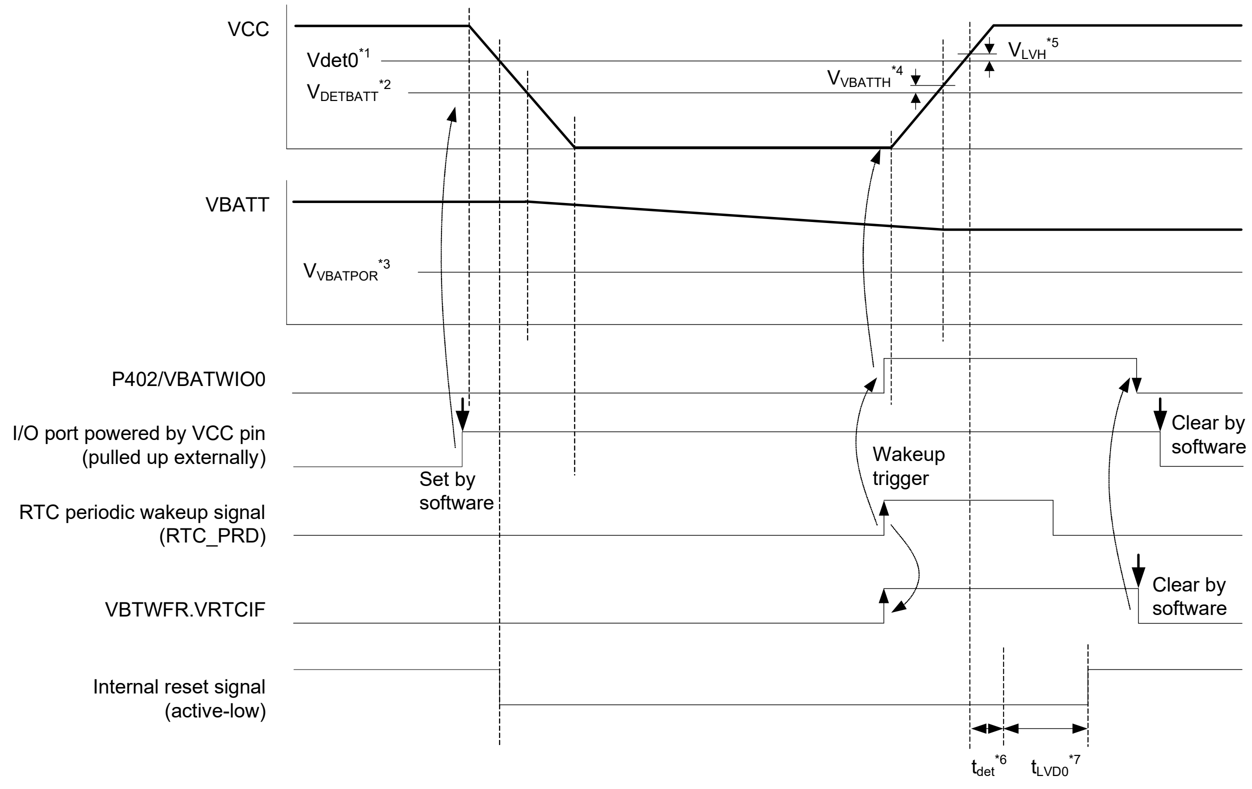
12.3.5 VBATT Wakeup Control Function Usage

The wakeup control function is a function that can toggle the output pin VBATWIO0 when the RTC alarm/periodic signal is asserted when VBATT_R is powered by the VBATT pin.

Note: The toggle that is triggered by the wakeup control function does not generate an interrupt at the ICU or a reset at the reset module.

The following procedures show how to set the registers after the MCU starts up as 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 be sure that the VBTSR.VBTRDF bit is 0.
3. Check the VBATT wakeup trigger source by reading the VBTWFR register. In the example of Figure 12.5, the VBTWFR.VRTCIF bit is set to 1.
4. Clear the corresponding bit in the VBTWFR register to 0, then the toggle output is started on the VBATWIO0 port. In the example of Figure 12.5, it is toggled from high to low on the VBATWIO0 port.
5. Set the I/O registers for the power supply stop control signal to output 0 or 1 to the external power management IC as needed.
6. In case you want to repeat the VBATT wakeup operation, clear the VBTCR1.BPWSWSTP bit to 0 and set the I/O registers for the power supply stop control signal to output 0 or 1 to the external power management IC so as to request stopping the power supply again.
   In case 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.


VCC
Vdet0*1
VDETBATT*2
VVBATTH*4
VLVH*5
VBATT
VVBATPOR*3
P402/VBATWIO0
I/O port powered by VCC pin (pulled up externally)
Set by software
Wakeup trigger
Clear by software
RTC periodic wakeup signal (RTC_PRD)
VBTWFR.VRTCIF
Clear by software
Internal reset signal (active-low)
tdet*6
tLVD0*7

Note 1. V_{det0} is LVD0 voltage detection level selected in the OFS1.VDSEL1[2:0] bits.
Note 2. $V_{DETBATT}$ is voltage level for switching to battery backup.
Note 3. $V_{VBATPOR}$ is voltage detection level VBATT power-on reset (VBAT_POR).
Note 4. V_{VBATTH} is hysteresis width for switching to battery backup.
Note 5. V_{LVH} is hysteresis width of LVD0.
Note 6. t_{det} is response delay time of LVD0.
Note 7. t_{LVD0} is wait time after LVD0 reset cancellation.

Figure 12.5 Timing diagram of VBATT wakeup function

12.4 Usage Notes

1. When the VBATT pin is not in use, connect the VBATT pin to the VCC pin.
2. When the voltage level on 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. If a reset is generated while writing to the registers described in this section, the register values might be lost.
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. The voltage level on the I/O ports powered by VCC pin transits to high-impedance when the power supply is stopped. If these ports are used as the power supply stop control pins for VBATT wakeup function, these ports should be pulled up or pulled 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, USBCKCR
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, VBTICTLR, VBTOCTLR, VBTWTER, VBTWEGR, VBTWFR, VBTBKRn (n = 0 to 511), BKRACR
PRC3	• Registers related to the LVD: LVD1CR1, LVD1SR, LVCMPCR, LVDLVLR, LVD1CR0

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)

The PRCn bits enable or disable writing to the protected registers listed 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 a block diagram of the ICU, and Table 14.2 lists the I/O pins.

Table 14.1 ICU specifications

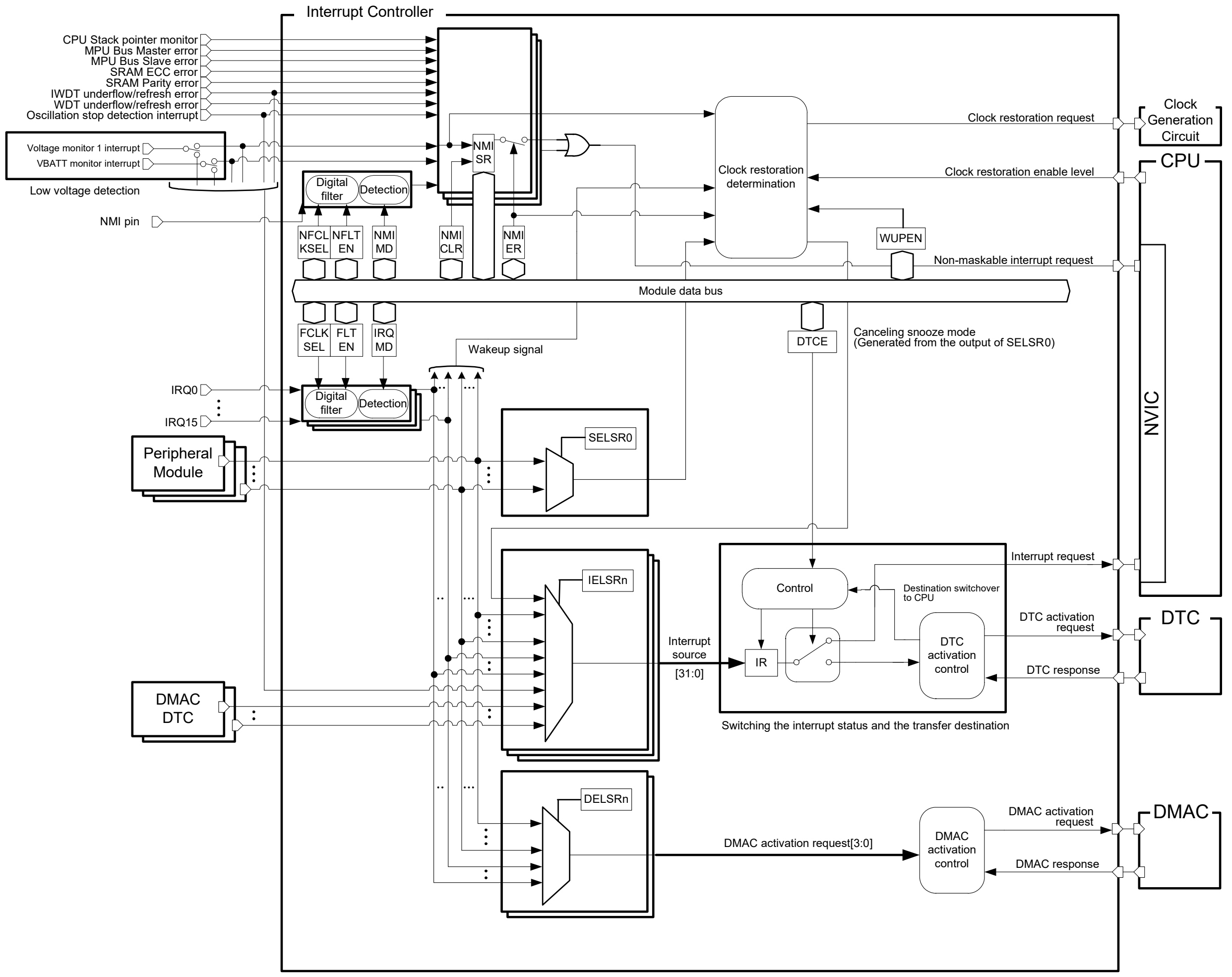
Parameter		Description
Interrupts	Peripheral function interrupts	• Interrupts from peripheral modules Number of sources: 209
	External pin interrupts	• Interrupt detection on low level[*4], falling edge, rising edge, rising and falling edges One of these detection methods can be set for each source. • Digital filter function supported • 11 sources, with interrupts from IRQ0 to IRQ4, IRQ6, IRQ7, IRQ9, IRQ11, IRQ14, IRQ15 pins.
	DTC/DMAC control	The DTC and DMAC can be activated by interrupt sources[*1]
	Interrupt sources for NVIC	• 32 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)
	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. Interrupt can be selected in the WUPEN register • Snooze mode: return is initiated by non-maskable interrupts. Interrupt can be selected in the SELSR0 and WUPEN registers. See section 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, set the LVD1CR1.IRQSEL bit to 1. To enable the VBATT monitor interrupt, set the VBTLVDICR.VBTLVDISEL bit to 1.

Note 4. Low level: interrupt detection is not canceled if you do not clear it after a detection.


Interrupt Controller
CPU Stack pointer monitor
MPU Bus Master error
MPU Bus Slave error
SRAM ECC error
SRAM Parity error
IWDT underflow/refresh error
WDT underflow/refresh error
Oscillation stop detection interrupt
Voltage monitor 1 interrupt
VBATT monitor interrupt
Low voltage detection
NMI pin
NMI SR
Digital filter
Detection
NFCL KSEL
NFLT EN
NMI MD
NMI CLR
NMI ER
Clock restoration determination
Clock restoration request
Clock restoration enable level
WUPEN
Non-maskable interrupt request
Clock Generation Circuit
CPU
Module data bus
FCLK SEL
FLT EN
IRQ MD
Wakeup signal
DTCE
Canceling snooze mode (Generated from the output of SELSR0)
IRQ0
IRQ15
Peripheral Module
SELSR0
NVIC
IELSRn
Interrupt request
Control
Destination switchover to CPU
Interrupt source
[31:0]
IR
DTC activation control
DTC activation request
DTC response
DTC
DMAC DTC
Switching the interrupt status and the transfer destination
DELSRn
DMAC activation request[3:0]
DMAC activation control
DMAC activation request
DMAC response
DMAC

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 4, 6, 7, 9, 11, 14, 15))
FCLKSEL: IRQi Digital Filter Sampling Clock Select (IRQCRi.FCLKSEL (i = 0 to 4, 6, 7, 9, 11, 14, 15))
FLTEN: IRQi Digital Filter Enable (IRQCRi.FLTEN (i = 0 to 4, 6, 7, 9, 11, 14, 15))
SELSR0: SYS Event Link Setting Register 0
WUPEN: Wake Up Interrupt Enable Register
IELSRn: ICU Event Link Setting Register (n = 0 to 31)
IR: Interrupt Status Flag (IELSRn.IR)
DTCE: DTC Activation Enable (IELSRn.DTCE)
DELSRn: DMAC Event Link Setting Register (n = 0 to 3)

Figure 14.1 ICU block diagram

Table 14.2 lists the ICU input/output pins.

Table 14.2 ICU I/O pins

Pin name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ0 to IRQ4, IRQ6, IRQ7, IRQ9, IRQ11, IRQ14, 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 4, 6, 7, 9, 11, 14, 15)

Address(es): ICU.IRQCR0 4000 6000h, ICU.IRQCR1 4000 6001h, ICU.IRQCR2 4000 6002h, ICU.IRQCR3 4000 6003h, ICU.IRQCR4 4000 6004h, ICU.IRQCR6 4000 6006h, ICU.IRQCR7 4000 6007h, ICU.IRQCR9 4000 6009h, ICU.IRQCR11 4000 600Bh, 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: Digital filter disabled 1: Digital filter enabled.	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 31).
  You can change the register values only when the IELSRn.IELS[7:0] bits are 00h.
- 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[7:0] bits are 00h.
- For a wakeup enable signal:
  Change the IRQCRi register setting before setting the target WUPEN.IRQWUPEN[n] (n = 0 to 4, 6, 7, 9, 11, 14, 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 external pin interrupt sources IRQi. 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 external pin interrupt request IRQi, selectable to:

- PCLKB (every cycle)
- PCLKB/8 (once every 8 cycles)
- PCLKB/32 (once every 32 cycles)
- PCLKB/64 (once every 64 cycles).

For the digital filter details, see section 14.4.3, Digital Filter.

FLTEN bit (IRQi Digital Filter Enable)

The FLTEN bit enables the digital filter used for the external pin interrupt sources IRQi. The filter is enabled when the

IRQCRi.FLTEN bit is 1, and disabled when the IRQCRi.FLTEN bit is 0. The IRQi pin level is sampled at the cycle specified in IRQCRi.FCLKSEL[1:0]. When the sampled level matches three times, the output level from the digital filter changes. For the digital filter details, 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	—	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	—	Reserved	This bit is read as 0.	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)

This 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)

This 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)

This 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.

VBATTST flag (VBATT monitor Interrupt Status Flag)

This 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)

This 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)

This 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)

This 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)

This 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)

This 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)

This 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)

This 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	—	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	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
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. You can write 1 to this bit only once after reset. Subsequent write accesses are invalid. Writing 0 to this bit is invalid.
Note 2. Do not write 1 to this bit when the source is used as an event signal.

IWDTEN bit (IWDT Underflow/Refresh Error Interrupt Enable)

The IWDTEN bit enables IWDT underflow/refresh error interrupt as an NMI trigger.

WDTEN bit (WDT Underflow/Refresh Error Interrupt Enable)

The WDTEN bit enables WDT underflow/refresh error interrupt as an NMI trigger.

LVD1EN bit (Voltage monitor 1 Interrupt Enable)

The LVD1EN bit enables voltage monitor 1 interrupt as an NMI trigger.

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	—	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	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	VBATTCLR	VBATT Clear	0: No effect 1: Clear the NMISR.VBATTST flag.	R/(W)^{*1}
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
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

Note 1. Only write 1 to this bit.

IWDTCLR bit (IWDT Clear)

Writing 1 to the IWDTCLR bit clears the NMISR.IWDTST flag. The IWDTCLR bit is read as 0.

WDTCLR bit (WDT Clear)

Writing 1 to the WDTCLR bit clears the NMISR.WDTST flag. The WDTCLR bit is read as 0.

LVD1CLR bit (LVD1 Clear)

Writing 1 to the LVD1CLR bit clears the NMISR.LVD1ST flag. The LVD1CLR bit is read as 0.

VBATTCLR bit (VBATT Clear)

Writing 1 to the VBATTCLR bit clears the NMISR.VBATTST flag. The VBATTCLR bit is read as 0.

OSTCLR bit (OST Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag. The OSTCLR bit is read as 0.

NMICLR bit (NMI Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag. The NMICLR bit is read as 0.

RPECLR bit (SRAM Parity Error Clear)

Writing 1 to the RPECLR bit clears the NMISR.RPEST flag. The RPECLR bit is read as 0.

RECCCLR bit (SRAM ECC Error Clear)

Writing 1 to the RECCCLR bit clears the NMISR.RECCST flag. The RECCCLR bit is read as 0.

BUSSCLR bit (Bus Slave Error Clear)

Writing 1 to the BUSSCLR bit clears the NMISR.BUSSST flag. The BUSSCLR bit is read as 0.

BUSMCLR bit (Bus Master Error Clear)

Writing 1 to the BUSMCLR bit clears the NMISR.BUSMSST flag. The BUSMCLR bit is read as 0.

SPECLR bit (CPU Stack Pointer Monitor Interrupt Clear)

Writing 1 to the SPECLR bit clears the NMISR.SPEST flag. The SPECLR 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: Digital filter is disabled 1: Digital filter is enabled.	R/W

Change the NMICR register settings before enabling NMI pin interrupt (before setting NMIER.NMIEN to 1).

NMIMD bit (NMI Detection Set)

The NMIMD bit selects the detection sensing method for the NMI pin interrupts.

NFCLKSEL[1:0] bits (NMI Digital Filter Sampling Clock Select)

The NFCLKSEL[1:0] bits select the digital filter sampling clock for the NMI pin interrupts, selectable to:

- PCLKB (every cycle)
- PCLKB/8 (once every 8 cycles)
- PCLKB/32 (once every 32 cycles)
- PCLKB/64 (once every 64 cycles).

For details of the digital filter, see section 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 details of the digital filter, see section 14.4.3, Digital Filter.

14.2.6 ICU Event Link Setting Register n (IELSRn)

Address(es): ICU.IELSR0 4000 6300h, ICU.IELSR1 4000 6304h, ICU.IELSR2 4000 6308h, ICU.IELSR3 4000 630Ch……
……ICU.IELSR28 4000 6370h, ICU.IELSR29 4000 6374h, ICU.IELSR30 4000 6378h, ICU.IELSR31 4000 637Ch

	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[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	IELS[7:0]	ICU Event Link Select	b7 b0 00000000: Disable interrupts to the associated NVIC/DTC module 00000001 to 11011001: Event signal number to be linked. For details, see Table 14.4.	R/W
b15 to b8	—	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)*1
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	DTCE	DTC Activation Enable	0: DTC activation is disabled 1: DTC activation is enabled.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register requires halfword or word access.
Note: The Bluetooth middleware executes processing in response to IELSR31 register. Do not change the setting of IELSR31.
Note 1. Writing 1 to the IR flag is prohibited.

The IELSRn register selects the IRQ source used by the NVIC. For details, see Table 14.4, Event table. IELSRn, where n = 0 to 31, corresponds to the NVIC IRQ input source numbers 0 to 31.

IELS[7:0] bits (ICU Event Link Select)

The IELS[7:0] bits link an event signal to the associated NVIC/DTC module.

IR flag (Interrupt Status Flag)

The IR status flag indicates an individual interrupt request from the event specified in IELS[7:0].

[Setting condition]

- When an interrupt request is received from the associated peripheral module or IRQi pin.

[Clearing conditions]

- When 0 is written to the flag. DTCE 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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	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
	—	—	—	—	—	—	—	—	DELS[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	DELS[7:0]	DMAC Event Link Select	b7 b0 00000000: Disable DMA start request to the associated DMAC module is disabled. 00000001 to 11011001: Event signal number to be linked. Other settings are prohibited. For details, see Table 14.4, Event table.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	IR	Interrupt Status Flag for DMAC	0: No interrupt request is generated 1: An interrupt request is generated.	R/W*1
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register requires halfword or word access.

Note 1. Writing 1 to the IR flag is prohibited.

DELS[7:0] bits (DMAC Event Link Select)

The DELS[7:0] bits link an event signal for the DMAC module.

IR flag (Interrupt Status Flag for DMAC)

The IR flag is the status flag of an individual DMA transfer request. This flag corresponds to DELS[7:0] bits of the same register.

[Setting condition]

- The flag is set to 1 when a DMA transfer request is generated from the corresponding peripheral module or IRQi pin.

[Clearing conditions]

- When 0 is written to the flag
- At the start of the DMA transfer after the DMA transfer request is issued.

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[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	SELS[7:0]	SYS Event Link Select	b7 b0 00000000: Disable event output to the associated low-power mode module 00000001 to 11011001: Event signal number to be linked. Other settings are prohibited. For details, see Table 14.4, Event table.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SELSR0 register selects the events that wake up the CPU from Snooze mode. You can only use the events listed in Table 14.4 checked as "Canceling Snooze using SELSR0". Events specified in this register are defined as ICU_SNZCANCEL (017h) in Table 14.4. When 017h is set in IELSRn.IELS, an SELSR0 event interrupt occurs.

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	—	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 interrupt 1: Enable Software standby returns by IRQ interrupt.	R/W
b16	IWDTWUPEN	IWDT Interrupt Software Standby Returns Enable	0: Disable Software standby returns by IWDT interrupt 1: Enable Software standby returns by IWDT interrupt.	R/W
b17	KEYWUPEN	Key Interrupt Software Standby Returns Enable	0: Disable Software standby returns by KEY interrupt 1: Enable Software standby returns by KEY interrupt.	R/W
b18	LVD1WUPEN	LVD1 Interrupt Software Standby Returns Enable	0: Disable Software standby returns by LVD1 interrupt 1: Enable Software standby returns by LVD1 interrupt.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b20	VBATTWUPEN	VBATT Monitor Interrupt Software Standby Returns Enable	0: Disable Software standby returns by VBATT monitor interrupt disabled 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 interrupt 1: Enable Software standby returns by ACMPLP0 interrupt.	R/W

Bit	Symbol	Bit name	Description	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 interrupt 1: Enable Software standby returns by RTC period interrupt.	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 interrupt 1: Enable Software standby returns by USBFS interrupt.	R/W
b28	AGT1UDWUPEN	AGT1 Underflow Interrupt Software Standby Returns Enable	0: Disable Software standby returns by AGT1 underflow interrupt 1: Enable Software standby returns by AGT1 underflow interrupt.	R/W
b29	AGT1CAWUPEN	AGT1 Compare Match A Interrupt Software Standby Returns Enable	0: Disable Software standby returns by AGT1 compare match A interrupt 1: Enable Software standby returns by AGT1 compare match A interrupt.	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 interrupt.	R/W
b31	IIC0WUPEN	IIC0 Address Match Interrupt Software Standby Returns Enable	0: Disable Software standby returns by IIC0 address match interrupt 1: Enable Software standby returns by IIC0 address match interrupt.	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. Only 0 to 4, 6, 7, 9, 11, 14, 15 bits can be used.

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.

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 maskable and non-maskable interrupts. Interrupt priorities are set up in the Arm NVIC. For information about these registers, 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 interrupt vector 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

Table 14.3 Interrupt vector table (2 of 2)

Exception number	IRQ number	Vector offset	Source	Description
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
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

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 the interrupt can be used as a CPU interrupt (IELSRn setting)
Invoke DTC	"✓" indicates the interrupt can be used to request DTC activation (IELSRn setting)
Invoke DMAC	"✓" indicates the interrupt can be used to request DMAC activation (DELSRn setting)
Canceling Snooze mode	"✓" indicates 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 the interrupt can be used to request a return from Software Standby mode

Table 14.4 Event table (1 of 4)

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	✓	✓	✓	✓	✓
007h		PORT_IRQ6	✓	✓	✓	✓	✓
008h		PORT_IRQ7	✓	✓	✓	✓	✓
009h	BLE	BLEIRQ *5	✓	✓	✓	✓	✓
00Ah	Port	PORT_IRQ9	✓	✓	✓	✓	✓
00Ch		PORT_IRQ11	✓	✓	✓	✓	✓
00Fh		PORT_IRQ14	✓	✓	✓	✓	✓
010h		PORT_IRQ15	✓	✓	✓	✓	✓
011h	DMAC0	DMAC0_INT	✓	✓			
012h	DMAC1	DMAC1_INT	✓	✓			
013h	DMAC2	DMAC2_INT	✓	✓			
014h	DMAC3	DMAC3_INT	✓	✓			
015h	DTC	DTC_COMPLETE	✓			✓*4	
017h	ICU	ICU_SNZCANCEL	✓			✓	
018h	FCU	FCU_FRDYI	✓				
019h	LVD	LVD_LVD1	✓			✓	✓
01Bh	VBATT	VBATT_LVD	✓			✓	✓
01Ch	MOSC	MOSC_STOP	✓				
01Dh	Low power mode	SYSTEM_SNZREQ		✓			
01Eh	AGT0	AGT0_AGTI	✓	✓	✓		
01Fh		AGT0_AGTCMAI	✓	✓	✓		
020h		AGT0_AGTCMBI	✓	✓	✓		
021h	AGT1	AGT1_AGTI	✓	✓	✓	✓	✓
022h		AGT1_AGTCMAI	✓	✓	✓	✓	✓
023h		AGT1_AGTCMBI	✓	✓	✓	✓	✓
024h	IWDT	IWDT_NMIUNDF	✓			✓	✓
025h	WDT	WDT_NMIUNDF	✓				
026h	RTC	RTC_ALM	✓			✓	✓
027h		RTC_PRD	✓			✓	✓
028h		RTC_CUP	✓				
029h	ADC140	ADC140_ADI	✓	✓	✓		
02Ah		ADC140_GBADI	✓	✓	✓		
02Bh		ADC140_CMPAI	✓				
02Ch		ADC140_CMPBI	✓				
02Dh		ADC140_WCMPM		✓	✓	✓*4	
02Eh		ADC140_WCMPUM		✓	✓	✓*4	
02Fh	ACMPLP	ACMP_LP0	✓			✓	✓
030h		ACMP_LP1	✓				

Table 14.4 Event table (2 of 4)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC		
031h	USBFS	USBFS_D0FIFO	✓	✓	✓		
032h		USBFS_D1FIFO	✓	✓	✓		
033h		USBFS_USBI	✓				
034h		USBFS_USBR	✓			✓	✓
035h	IIC0	IIC0_RXI	✓	✓	✓		
036h		IIC0_TXI	✓	✓	✓		
037h		IIC0_TEI	✓				
038h		IIC0_EEI	✓				
039h		IIC0_WUI	✓			✓	✓
03Ah	IIC1	IIC1_RXI	✓	✓	✓		
03Bh		IIC1_TXI	✓	✓	✓		
03Ch		IIC1_TEI	✓				
03Dh		IIC1_EEI	✓				
046h	CTSU	CTSU_CTSUWR	✓	✓	✓		
047h		CTSU_CTSURD	✓	✓	✓		
048h		CTSU_CTSUFN	✓			✓*4	
049h	KINT	KEY_INTKR	✓			✓*1	✓*1
04Ah	DOC	DOC_DOPCI	✓			✓*4	
04Bh	CAC	CAC_FERRI	✓				
04Ch		CAC_MENDI	✓				
04Dh		CAC_OVFI	✓				
04Eh	CAN0	CAN0_ERS	✓				
04Fh		CAN0_RXF	✓				
050h		CAN0_TXF	✓				
051h		CAN0_RXM	✓				
052h		CAN0_TXM	✓				
053h	I/O port	IOPORT_GROUP1	✓	✓*2	✓*2		
054h		IOPORT_GROUP2	✓	✓*2	✓*2		
055h		IOPORT_GROUP3	✓	✓*2	✓*2		
056h		IOPORT_GROUP4	✓	✓*2	✓*2		
057h	ELC	ELC_SWEVT0	✓*3	✓			
058h		ELC_SWEVT1	✓*3	✓			
059h	POEG	POEG_GROUP0	✓				
05Ah		POEG_GROUP1	✓				
05Bh	GPT320	GPT0_CCMPA	✓	✓	✓		
05Ch		GPT0_CCMPB	✓	✓	✓		
05Dh		GPT0_CMPC	✓	✓	✓		
05Eh		GPT0_CMPD	✓	✓	✓		
05Fh		GPT0_CMPE	✓	✓	✓		
060h		GPT0_CMPF	✓	✓	✓		
061h		GPT0_OVF	✓	✓	✓		
062h		GPT0_UDF	✓	✓	✓		

Table 14.4 Event table (3 of 4)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC		
063h	GPT321	GPT1_CCMPA	✓	✓	✓		
064h		GPT1_CCMPB	✓	✓	✓		
065h		GPT1_CMPC	✓	✓	✓		
066h		GPT1_CMPD	✓	✓	✓		
067h		GPT1_CMPE	✓	✓	✓		
068h		GPT1_CMPF	✓	✓	✓		
069h		GPT1_OVF	✓	✓	✓		
06Ah		GPT1_UDF	✓	✓	✓		
06Bh	GPT322	GPT2_CCMPA	✓	✓	✓		
06Ch		GPT2_CCMPB	✓	✓	✓		
06Dh		GPT2_CMPC	✓	✓	✓		
06Eh		GPT2_CMPD	✓	✓	✓		
06Fh		GPT2_CMPE	✓	✓	✓		
070h		GPT2_CMPF	✓	✓	✓		
071h		GPT2_OVF	✓	✓	✓		
072h		GPT2_UDF	✓	✓	✓		
073h	GPT323	GPT3_CCMPA	✓	✓	✓		
074h		GPT3_CCMPB	✓	✓	✓		
075h		GPT3_CMPC	✓	✓	✓		
076h		GPT3_CMPD	✓	✓	✓		
077h		GPT3_CMPE	✓	✓	✓		
078h		GPT3_CMPF	✓	✓	✓		
079h		GPT3_OVF	✓	✓	✓		
07Ah		GPT3_UDF	✓	✓	✓		
07Bh	GPT164	GPT4_CCMPA	✓	✓	✓		
07Ch		GPT4_CCMPB	✓	✓	✓		
07Dh		GPT4_CMPC	✓	✓	✓		
07Eh		GPT4_CMPD	✓	✓	✓		
07Fh		GPT4_CMPE	✓	✓	✓		
080h		GPT4_CMPF	✓	✓	✓		
081h		GPT4_OVF	✓	✓	✓		
082h		GPT4_UDF	✓	✓	✓		
083h	GPT165	GPT5_CCMPA	✓	✓	✓		
084h		GPT5_CCMPB	✓	✓	✓		
085h		GPT5_CMPC	✓	✓	✓		
086h		GPT5_CMPD	✓	✓	✓		
087h		GPT5_CMPE	✓	✓	✓		
088h		GPT5_CMPF	✓	✓	✓		
089h		GPT5_OVF	✓	✓	✓		
08Ah		GPT5_UDF	✓	✓	✓		

Table 14.4 Event table (4 of 4)

			IELSRn		DELSRn		
Event number	**Interrupt request source**	**Name**	**Connect to NVIC**	**Invoke DTC**	**Invoke DMAC**	**Canceling Snooze**	**Canceling Software Standby**
09Bh	GPT168	GPT8_CCMPA	✓	✓	✓		
09Ch		GPT8_CCMPB	✓	✓	✓		
09Dh		GPT8_CMPC	✓	✓	✓		
09Eh		GPT8_CMPD	✓	✓	✓		
09Fh		GPT8_CMPE	✓	✓	✓		
0A0h		GPT8_CMPF	✓	✓	✓		
0A1h		GPT8_OVF	✓	✓	✓		
0A2h		GPT8_UDF	✓	✓	✓		
0ABh	GPT	GPT_UVWEDGE	✓				
0ACh	SCI0	SCI0_RXI	✓	✓	✓		
0ADh		SCI0_TXI	✓	✓	✓		
0AEh		SCI0_TEI	✓				
0AFh		SCI0_ERI	✓				
0B0h		SCI0_AM	✓			✓*4	
0B1h		SCI0_RXI_OR_ERI				✓*4	
0B2h	SCI1	SCI1_RXI	✓	✓	✓		
0B3h		SCI1_TXI	✓	✓	✓		
0B4h		SCI1_TEI	✓				
0B5h		SCI1_ERI	✓				
0B6h		SCI1_AM	✓				
0C1h	SCI4	SCI4_RXI	✓	✓	✓		
0C2h		SCI4_TXI	✓	✓	✓		
0C3h		SCI4_TEI	✓				
0C4h		SCI4_ERI	✓				
0C5h		SCI4_AM	✓				
0C6h	SCI9	SCI9_RXI	✓	✓	✓		
0C7h		SCI9_TXI	✓	✓	✓		
0C8h		SCI9_TEI	✓				
0C9h		SCI9_ERI	✓				
0CAh		SCI9_AM	✓				
0CBh	SPI0	SPI0_SPRI	✓	✓	✓		
0CCh		SPI0_SPTI	✓	✓	✓		
0CDh		SPI0_SPII	✓				
0CEh		SPI0_SPEI	✓				
0CFh		SPI0_SPTEND	✓				
0D0h	SPI1	SPI1_SPRI	✓	✓	✓		
0D1h		SPI1_SPTI	✓	✓	✓		
0D2h		SPI1_SPII	✓				
0D3h		SPI1_SPEI	✓				
0D4h		SPI1_SPTEND	✓				

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.
Note 5. The Bluetooth middleware executes processing in response to BLEIRQ. Do not use this source.

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 in either:

- Edges (falling edge, rising edge, or rising and falling edges) of the interrupt signal
- Level (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.

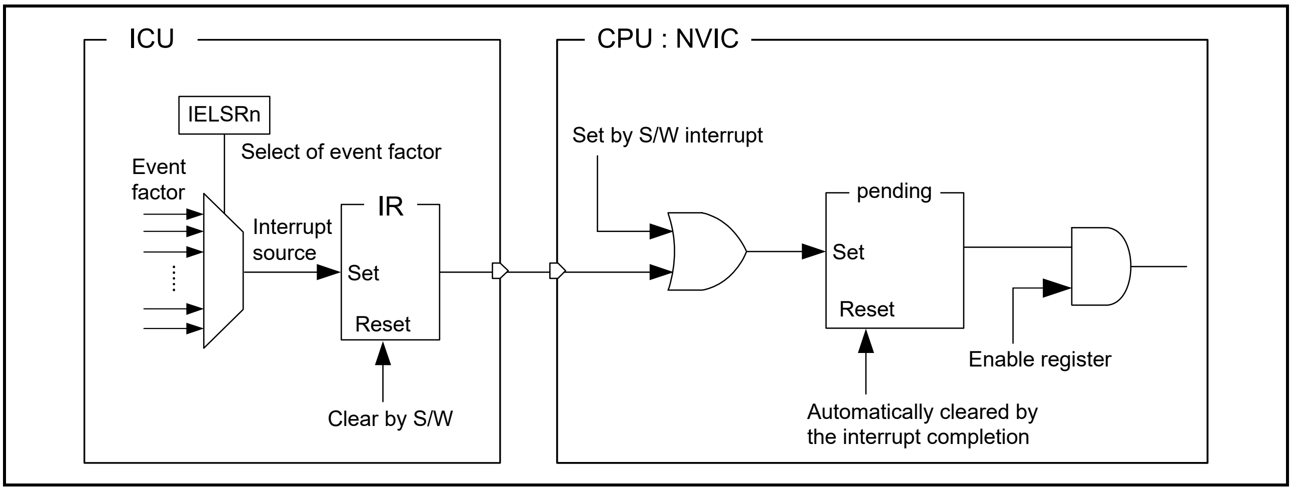

ICU
IELSRn
Select of event factor
Event factor
Interrupt source
IR
Set
Reset
Clear by S/W
CPU : NVIC
Set by S/W interrupt
pending
Set
Reset
Enable register
Automatically cleared by the interrupt completion

Figure 14.2 Interrupt path of the ICU and CPU (NVIC)

Use the following procedures for detecting interrupts:

- 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 with 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 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[7:0] = 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 check, ✓, 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, be sure to 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. Set the IELSRn.IELS bits and IELSRn.DTCE bit 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 activation 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. 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[7:0].
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.

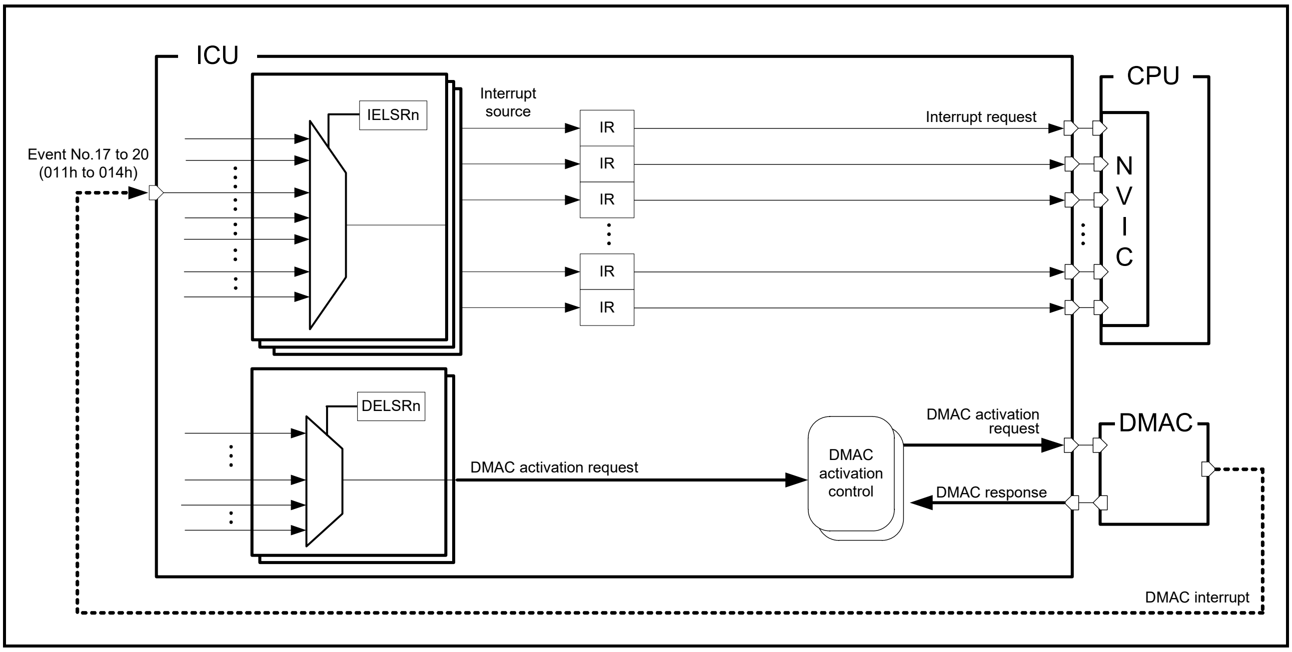

ICU
IELSRn
Interrupt source
Event No.17 to 20 (011h to 014h)
IR
IR
IR
IR
IR
Interrupt request
CPU
NVIC
DELSRn
DMAC activation request
DMAC activation control
DMAC activation request
DMAC response
DMAC
DMAC interrupt

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 4, 6, 7, 9, 11, 14, 15) and the NMI pin interrupt. It samples input signals on the filter PCLKB sampling clock and removes any signal with a pulse width less than 3 sampling cycles.

- To use the digital filter for 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 4, 6, 7, 9, 11, 14, 15).

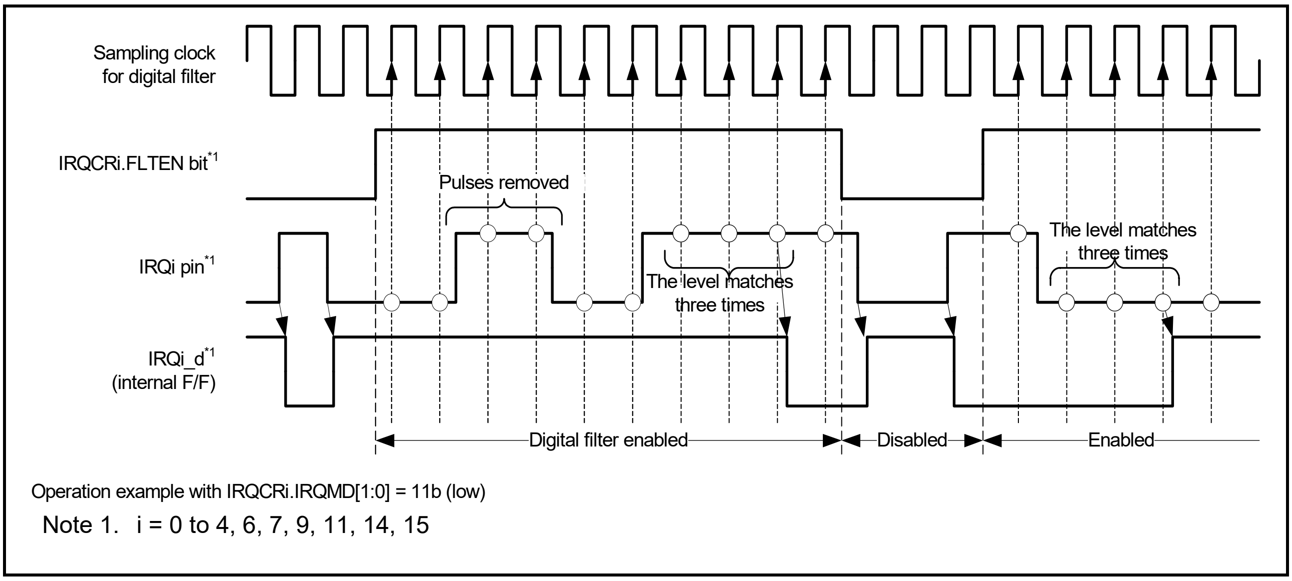
2) Set the IRQCRi.FLTEN bit (i = 0 to 4, 6, 7, 9, 11, 14, 15) to 1 (digital filter enabled).

- 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.


Sampling clock
for digital filter
IRQCRi.FLTEN bit*1
Pulses removed
IRQi pin*1
The level matches
three times
The level matches
three times
IRQi_d*1
(internal F/F)
Digital filter enabled
Disabled
Enabled

Operation example with IRQCRi.IRQMD[1:0] = 11b (low)

Note 1. i = 0 to 4, 6, 7, 9, 11, 14, 15

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 ICU clock 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 (0 to 4, 6, 7, 9, 11, 14, 15) to 0 (digital filter disabled).
2. Set or confirm the I/O port settings.
3. Set the IRQMD[1:0] bits, FCLKSEL[1:0] bits and FLTEN bit of the IRQCRi register.
4. Select the IRQ pin as follows:
    - If the IRQ pin is to be used for CPU interrupt request, set the IELSRn.IELS[7:0] bits and IELSRn.DTCE bit to 0
    - If the IRQ pin is to be used for DTC activation, set the IELSRn.IELS[7:0] 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
- 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. An NMI interrupt cannot be disabled when enabled, except 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 return 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 target 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:

1. Select the interrupt source that enables return from Software Standby.
    - For non-maskable interrupts, use the NMIER register to enable the target interrupt request
    - For maskable interrupts, use the WUPEN register to enable the target interrupt request.
2. Select the CPU as the interrupt request destination.
3. Enable the interrupt in the NVIC.

Interrupt requests through the 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 to Normal mode from Snooze mode using the interrupts provided for this mode.

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 to Normal mode from Snooze mode:

   a. Set the event that you want to trigger a return to Normal mode from Snooze mode in SELSR0.SEL and set the value 017h (ICU_SNZCANCEL) in IELSRn.IELS.
   b. Set the event that you want to trigger a return to Normal mode from Snooze 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 the ICU. If an event selected in IELSRn is detected, the CPU acknowledges the interrupt after returning to Normal mode from Software Standby mode. If an event selected in DELSRn is detected, the DMAC acknowledges 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.
	DMA bus	• Connected to the DMAC/DTC • Connected to on-chip memory, internal peripheral 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
	Internal peripheral bus 1	• Connected to system control related to peripheral modules
	Internal peripheral bus 3	• Connected to peripheral modules (CAC, ELC, I/O ports, POEG, RTC, WDT, IWDT, IIC, CAN, ADC14, DAC12, and DOC)
	Internal peripheral bus 4	• Connected to peripheral modules (SCI, SPI, CRC, and GPT)
	Internal peripheral bus 5	• Connected to peripheral modules (KINT, AGT, USBFS, OPAMP, ACMPLP, DAC8, SLCDC, and CTSU)
	Internal peripheral bus 7	• Connected to SecureIPs
	Internal peripheral bus 9	• Connected to flash memory (in P/E)*1 and data flash memory

Note 1. P/E = Programming/Erasure.

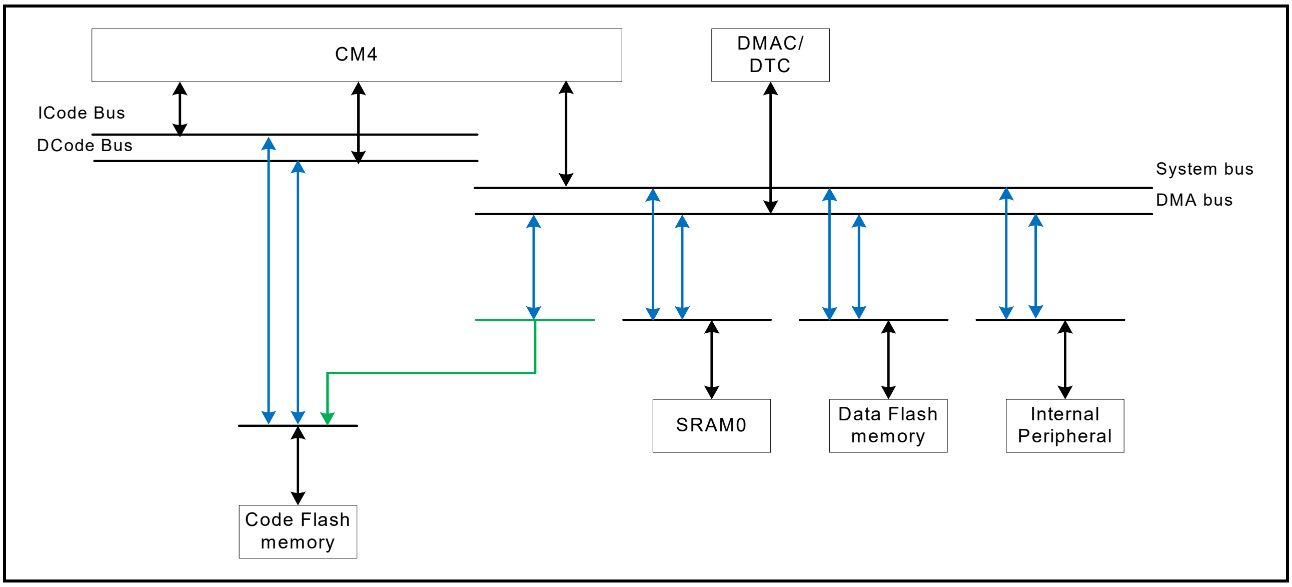


Figure 15.1 Bus configuration

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 7FFFh	Memory bus 4	SRAM0
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

Note 1. P/E = Programming/Erasure.

15.2 Description of Buses

15.2.1 Main Buses

The main buses for the CPU consist of the ICode bus, DCode bus, and system bus.

- The ICode bus and the DCode bus are connected to the 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, the data flash memory, and the internal peripheral bus. The system bus is used for instruction and data accesses 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, data flash memory, internal peripheral 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, then 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

Product 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, the DCode bus, and the 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, the data flash memory, and the internal peripheral bus. For connections from the main bus to the slave interfaces, see the slave interfaces in Table 15.1.

Arbitration between the ICode bus, the DCode bus, and the Memory bus 3 occurs in the slave interface of the ICode and the DCode bus areas. The arbitration method is selectable from fixed priority and round-robin. For more information, see section 15.3.2.

Arbitration between the system bus and DMA bus occurs in the slave interface of the system bus area. The arbitration method is selectable from fixed priority and round-robin. For more information, see section 15.3.2.

Different master and slave transfer combinations can proceed simultaneously.

15.2.3 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 from a peripheral bus at the same time.

Figure 15.2 shows an example of parallel operations. In this example, the CPU uses the instruction and operand buses for simultaneous access to the flash and SRAM, respectively. Additionally, the DMAC/DTC simultaneously use the DMA bus for access to a peripheral bus during access to the flash and SRAM by the CPU.

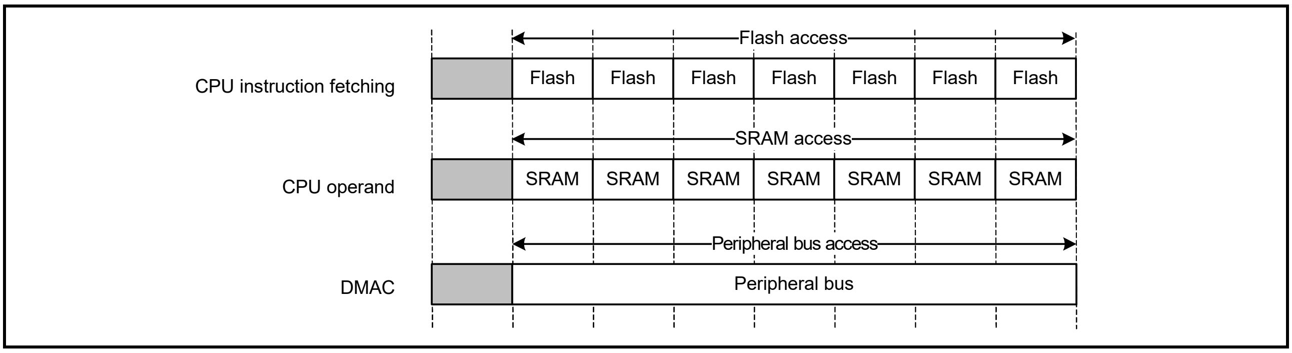

Flash access
CPU instruction fetching
Flash
Flash
Flash
Flash
Flash
Flash
Flash
SRAM access
CPU operand
SRAM
SRAM
SRAM
SRAM
SRAM
SRAM
SRAM
Peripheral bus access
DMAC
Peripheral bus

Figure 15.2 Example of parallel operations

15.2.4 Restriction on Endianness

Memory space must be little-endian to execute code on the Cortex®-M4 core.

15.3 Register Descriptions

15.3.1 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: A bus error is reported 1: A bus error is not reported.	R/W

Note: Changing reserved bits from the initial value of 0 is prohibited. Operation during the change is not guaranteed.

IERES bit (Ignore Error Responses)

The IERES bit specifies the enable or disable of an error response of the AHB-Lite protocol.

Table 15.3 lists the registers associated with each bus type.

Table 15.3 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	-	-
Internal peripheral bus 1, 3, 4, 5, 7	-	BUSSCNTPnB [n = 0, 2, 3, 4, 6]	-	-
Internal peripheral bus 9	-	BUSSCNTFBU	-	-

15.3.2 Slave Bus Control Register (BUSSCNT<slave>)

Address(es): BUS.BUSSCNTFLI 4000 4100h, BUS.BUSSCNTMBIU 4000 4108h, BUS.BUSSCNTRAM0 4000 410Ch, BUS.BUSSCNTP0B 4000 4114h, BUS.BUSSCNTP2B 4000 4118h, BUS.BUSSCNTP3B 4000 411Ch, BUS.BUSSCNTP4B 4000 4120h, BUS.BUSSCNTP6B 4000 4128h, BUS.BUSSCNTFBU 4000 4130h

	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	Specify the priority between groups: 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 reserved bits from the initial value of 0 is prohibited. Operation during the change is not guaranteed.

ARBMET[1:0] bits (Arbitration Method)

The ARBMET[1:0] bits specify the arbitration method, with priority defined for all bus masters. For fixed priority, see Table 15.4. For round-robin, see Table 15.5. For the associations between bus types and registers, see Table 15.3.

Table 15.4 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)
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)

Table 15.5 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)
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)

15.3.3 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		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 only cleared by resets other than MPU related resets. For more information, see section 6, Resets and section 16, Memory Protection Unit (MPU).

Table 15.3 lists the registers associated with each bus type.

BERAD[31:0] bits (Bus Error Address)

The BERAD[31:0] bits store the accessed address when a bus error occurred. For more information, see BUSnERRSTAT.ERRSTAT and section 15.4, Bus Error Monitoring Section.

The value of the BUSnERRADD.BERAD[31:0] bits (n = 1 to 4) is valid only when BUSnERRSTAT.ERRSTAT bit (n = 1 to 4) is set to 1.

15.3.4 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	R

Bit	Symbol	Bit name	Description	R/W
b7	ERRSTAT	Bus Error Status	0: No bus error occurred 1: Bus error occurred.	R

Note: This register is only cleared by resets other than MPU-related resets. For more information, see section 6, Resets and section 16, Memory Protection Unit (MPU).

Table 15.3 lists the registers associated with each bus type.

ACCSTAT bit (Error Access Status)

The ACCSTAT bit indicates the access status, write access or read access, when a bus error occurs. For more information, see BUSnERRSTAT.ERRSTAT and section 15.4, Bus Error Monitoring Section.

The value is valid only when the BUSnERRSTAT.ERRSTAT bit (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. The BUSnERRSTAT.ERRSTAT bit (n = 1 to 4) is set to 1.

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 MPU errors or bus slave MPU errors, and reset is selected in the OAD bit, if the bus access that caused the MPU error completes later than the internal reset signal being generated, which can occur with the wait setting, BUSnERRSTAT.ERRSTAT (n = 1 to 4) is not set to 1.

When detecting bus master MPU errors or bus slave MPU errors, and NMI is selected in the OAD bit, BUSnERRSTAT.ERRSTAT (n = 1 to 4) is set to 1 after the bus access that caused the MPU error completes.

For more information on bus errors, see section 15.4, Bus Error Monitoring Section, and section 16, Memory Protection Unit (MPU).

15.4 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.

15.4.1 Error Type that Occurs by Bus

Four types of errors can occur on each bus:

- Illegal address access
- Bus master MPU error
- Bus slave MPU error
- Timeout.

Table 15.6 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).

15.4.2 Operation when a Bus Error Occurs

When a bus error occurs, operation is not guaranteed and the error is returned to the requesting master IP. The bus error information occurred in each master is stored in the BUSnERRADD and BUSnERRSTAT registers. These registers must be cleared by reset only. For more information, see sections 15.3.3 and 15.3.4.

Note: The DMAC and DTC do not receive bus errors. If the DMAC or DTC accesses the bus, the transfer continues.

15.4.3 Conditions Leading to Illegal Address Access Errors

Table 15.6 lists the address spaces for each bus that trigger illegal address access errors.

Table 15.6 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 mirror area	*1	E
0280 0000h to 1FFF FFFFh	Reserved	E	E
2000 0000h to 2001 7FFFh	Memory bus 4	—	—
2001 8000h 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	Reserved	—	—
6800 0000h to 7FFF FFFFh	Reserved	E	E
8000 0000h to 97FF FFFFh	Reserved	—	—
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 path that 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 (depends 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 (depends on the switched address).
If the MMF is disabled and the CPU accesses 0200 0000h, the bus module can detect the error.

Note 2. The bus module detects an access error resulting from access to reserved area, for example if no area is assigned for the slave.
0280 0000h to 1FFF FFFFh: Access error detection.
0000 0000h to 01FF FFFFh: Memory bus 1 no access error detection.

15.4.4 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.

15.5 Notes on using Flash Cache

When using flash cache by access from the CPU, Arm® MPU should also be set to cacheable. See references 1. and 2. for more information.

15.6 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 specifications, and Table 16.2 shows the behavior on detection of each MPU error.

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) • 4 regions (code flash, SRAM, 2 secure functions).

Table 16.2 Behavior on MPU error detection

MPU type	Notification type	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. For information on error access for other MPUs, see 15.3.3 and 15.3.4 in section 15, Buses.

16.2 CPU Stack Pointer Monitor

The CPU stack pointer monitor detects underflows and overflows of the stack pointer. Because the Arm CPU has two stack pointers, a Main Stack Pointer (MSP) and a Process Stack Pointer (PSP), it supports two CPU stack pointer monitors. If a stack pointer underflow or overflow is detected, the CPU stack pointer monitor generates a reset or a non-maskable interrupt. The CPU stack pointer monitor is enabled by setting the Stack Pointer Monitor Enable 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 a block diagram, and Figure 16.2 shows the register setting flow.

Table 16.3 CPU stack pointer monitor specifications (1 of 2)

Parameter	Description
SRAM region	Region to be covered by memory protection

Table 16.3 CPU stack pointer monitor specifications (2 of 2)

Parameter	Description
Number of regions	2 regions: • Main Stack Pointer (MSP) • Process Stack Pointer (PSP).
Address specification for individual regions	Specifying start and end addresses for individual regions
Stack pointer monitor enable or disable setting for individual regions	Enabling or disabling stack pointer monitor for individual regions
Operation on error detection	Reset or non-maskable interrupts can be generated
Register protection	Protecting registers from illegal writes

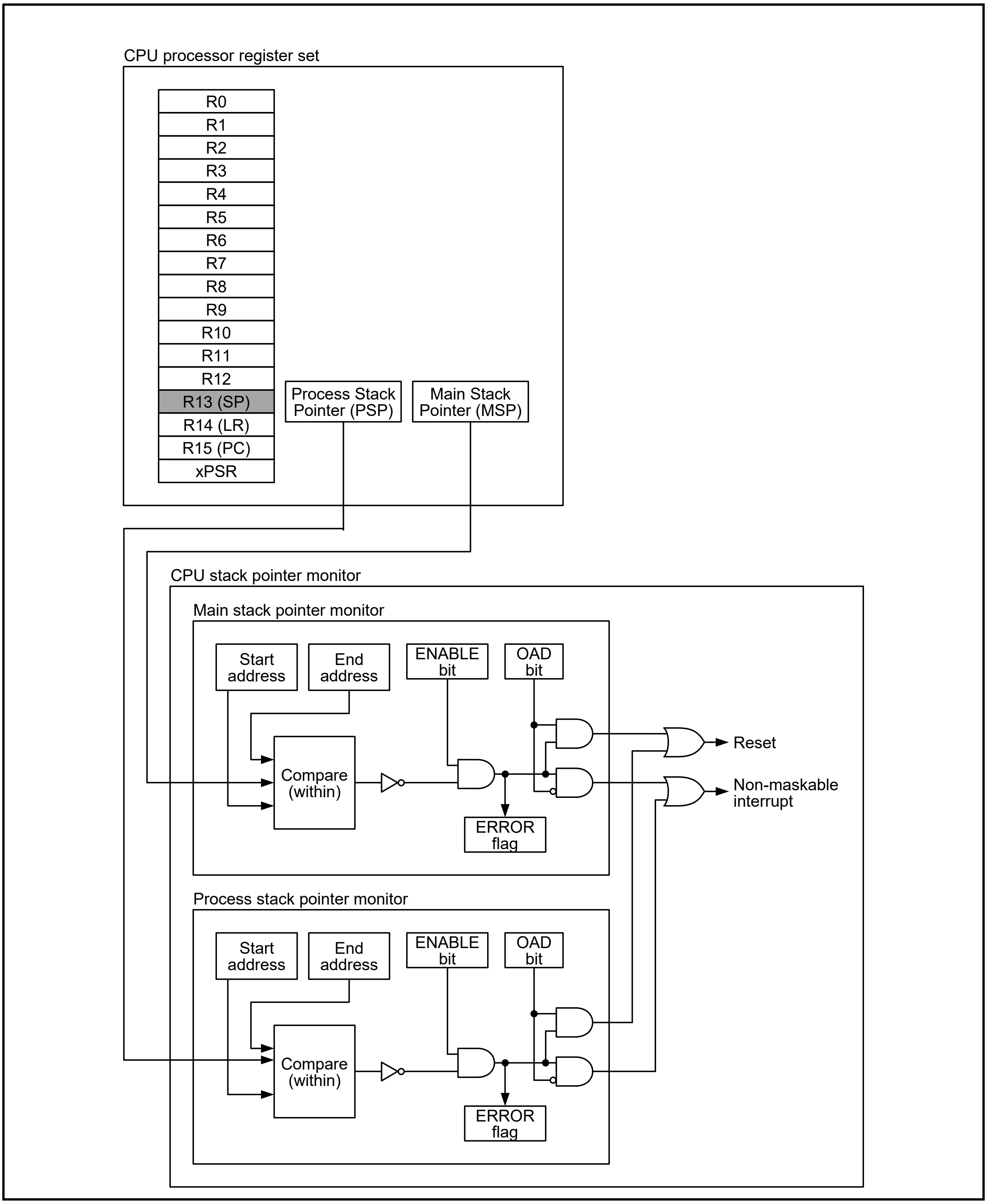

CPU processor register set
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13 (SP)
R14 (LR)
R15 (PC)
xPSR
Process Stack Pointer (PSP)
Main Stack Pointer (MSP)
CPU stack pointer monitor
Main stack pointer monitor
Start address
End address
ENABLE bit
OAD bit
Compare (within)
ERROR flag
Reset
Non-maskable interrupt
Process stack pointer monitor
Start address
End address
ENABLE bit
OAD bit
Compare (within)
ERROR flag

Figure 16.1 CPU stack pointer monitor block diagram

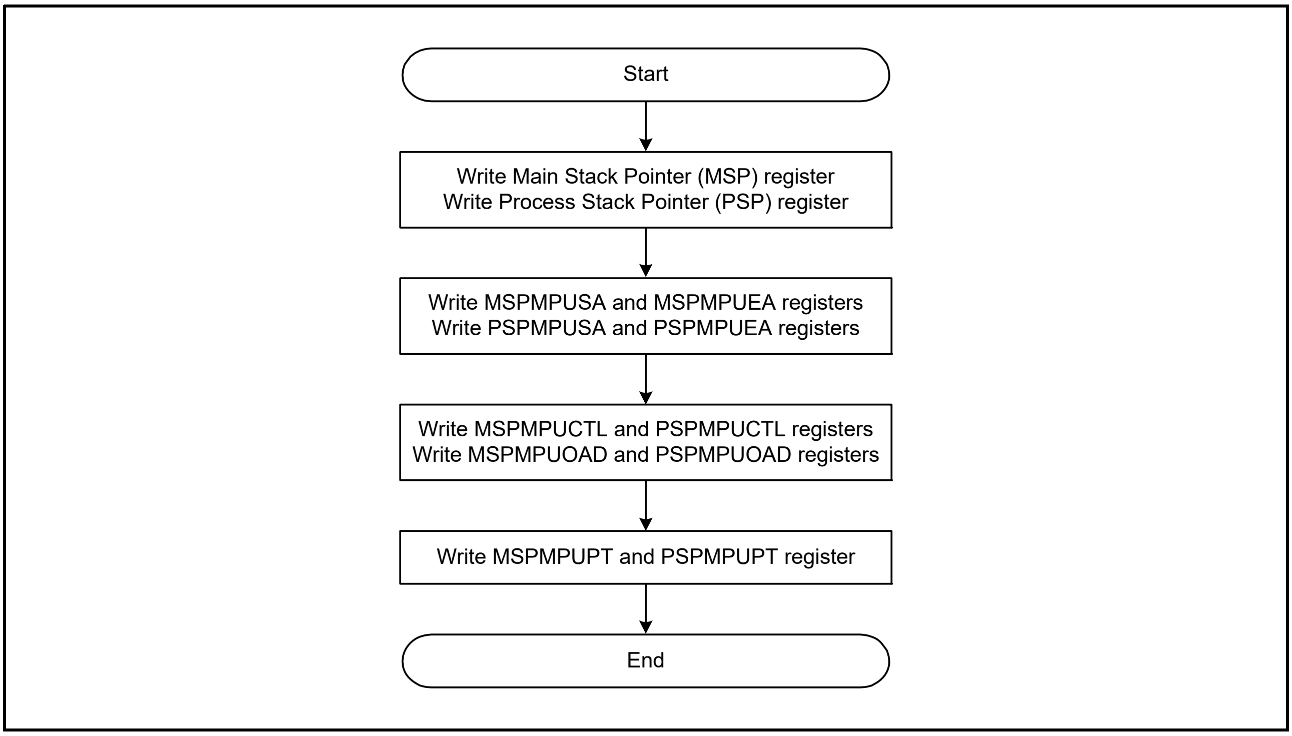

Start
Write Main Stack Pointer (MSP) register
Write Process Stack Pointer (PSP) register
Write MSPMPUSA and MSPMPUEA registers
Write PSPMPUSA and PSPMPUEA registers
Write MSPMPUCTL and PSPMPUCTL registers
Write MSPMPUOAD and PSPMPUOAD registers
Write MSPMPUPT and PSPMPUPT register
End

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. A memory protection error can choose between non-maskable interrupt or reset in the OAD bit setting.

The non-maskable interrupt status is indicated in ICU.NMISR.SPEST. For details, see section 14, Interrupt Controller Unit (ICU). The reset status is indicated in 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 bits 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 is generated continuously while the stack pointer overflows or underflows. To clear the non-maskable interrupt flag, set the stack pointer within the specified region and then clear the non-maskable interrupt flag by setting the ICU.NMICLR.SPECLR bit to 1. Then, write 0 to the ERROR bits 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 is from 2000 0000h to 200F FFFCh, excluding reserved areas.	R/W

The MSPMPUSA and MSPMPUEA registers specify the CPU stack region of SRAM (2000 0000h to 200F FFFFh, excluding 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, for use in region determination. The lower 2 bits should be 1. The value range must be 2000 0003h to 200F FFFFh, excluding 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, for use in region determination. The lower 2 bits should be 0. The value range must be 2000 0000h to 200F FFFCh, excluding reserved areas.	R/W

The PSPMPUSA and PSPMPUEA registers specify the CPU stack region of SRAM (2000 0000h to 200F FFFFh, excluding 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

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PSPMPUEA[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
	PSPMPUEA[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	PSPMPUEA[31:0]	Region End Address	Address where the region ends, for use in region determination. The lower 2 bits should be 1. The value range is from 2000 0003h to 200F FFFFh, excluding 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

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

Bit	Symbol	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 enable or disable writes to the OAD bit	R/(W)*1

Note 1. Write data is not saved.

OAD bit (Operation after Detection)

The OAD bit selects either a reset or a non-maskable interrupt to occur when a stack pointer underflow or overflow is detected by the CPU stack pointer monitor. The main and the process stack pointer monitors each uses an OAD bit to determine which signal is generated when a stack pointer underflow or overflow is detected. Write A5h in KEY[7:0] bits in halfword access simultaneously when setting the OAD bit.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the OAD bit. When writing to the OAD bit, simultaneously write 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 status of the stack pointer monitor. Each stack point monitor has an independent ERROR bit.

[Setting condition]

- Overflow or underflow of the 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. Reads are permitted.	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 enable or disable writes to the PROTECT bit	R/(W)*1

Note 1. Write data is not saved.

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)

The KEY[7:0] bits enable or disable writes 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 eight region MPUs and provides full support for:

- Protected regions
- Overlapping protected 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 2. in section 16.7, References.

16.4 Bus Master MPU

The bus master MPU monitors the addresses accessed by the bus master in the entire address space (0000 0000h to FFFF FFFFh). The access control information, consisting of read and write permissions, can be independently set for up to 16 regions. The bus master MPU monitors access to each region based on these settings. If access to a protected region is detected, the bus master MPU generates an internal reset or a non-maskable interrupt. For information on error access, see section 15.3.3 and section 15.3.4 in section 15, Buses.

Table 16.4 lists the specifications of the bus master MPU, and Figure 16.3 shows a block diagram.

Table 16.4 Bus master MPU specifications

Parameter	Description
Protected master groups	Bus master MPU group A: DMA bus
Protected region	0000 0000h to FFFF FFFFh
Number of regions	Bus master MPU group A: 16 regions
Address specification for individual regions	Specifying start and end address for individual regions
Enable or disable setting for memory protection in individual regions	Enabling or disabling setting for the associated region
Access-control settings for individual regions	Permission to read and write
Operation on error detection	Reset or non-maskable interrupts
Register protection	Register can be protected from illegal writes

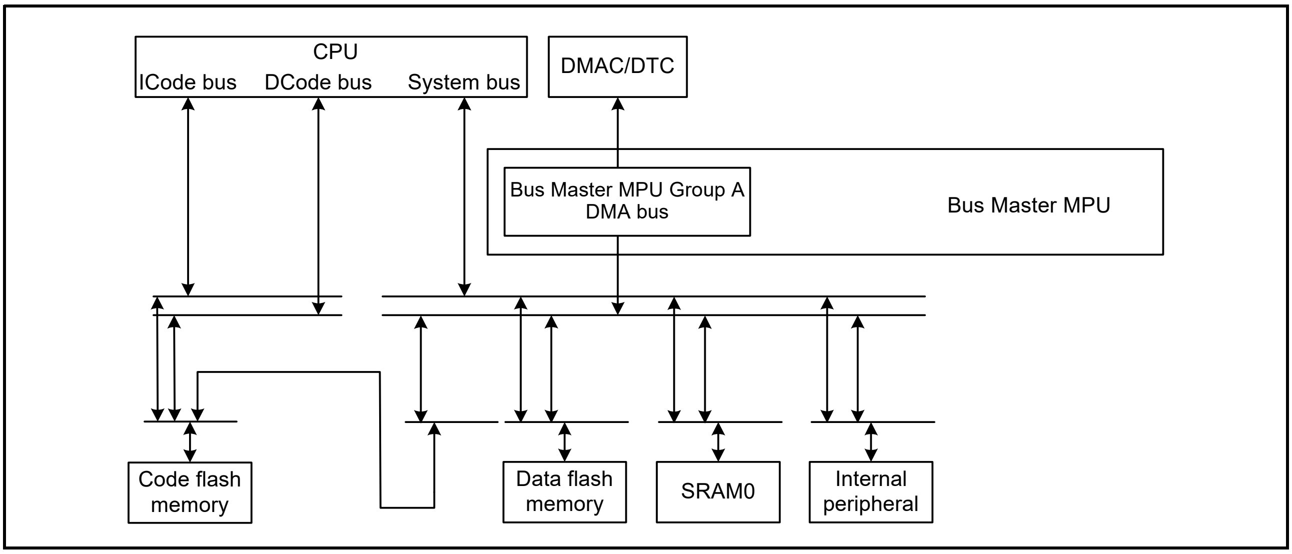

CPU
ICode bus
DCode bus
System bus
DMAC/DTC
Bus Master MPU Group A
DMA bus
Bus Master MPU
Code flash memory
Data flash memory
SRAM0
Internal peripheral

Figure 16.3 MPU bus master block diagram

Figure 16.4 shows the MPU bus master group A.

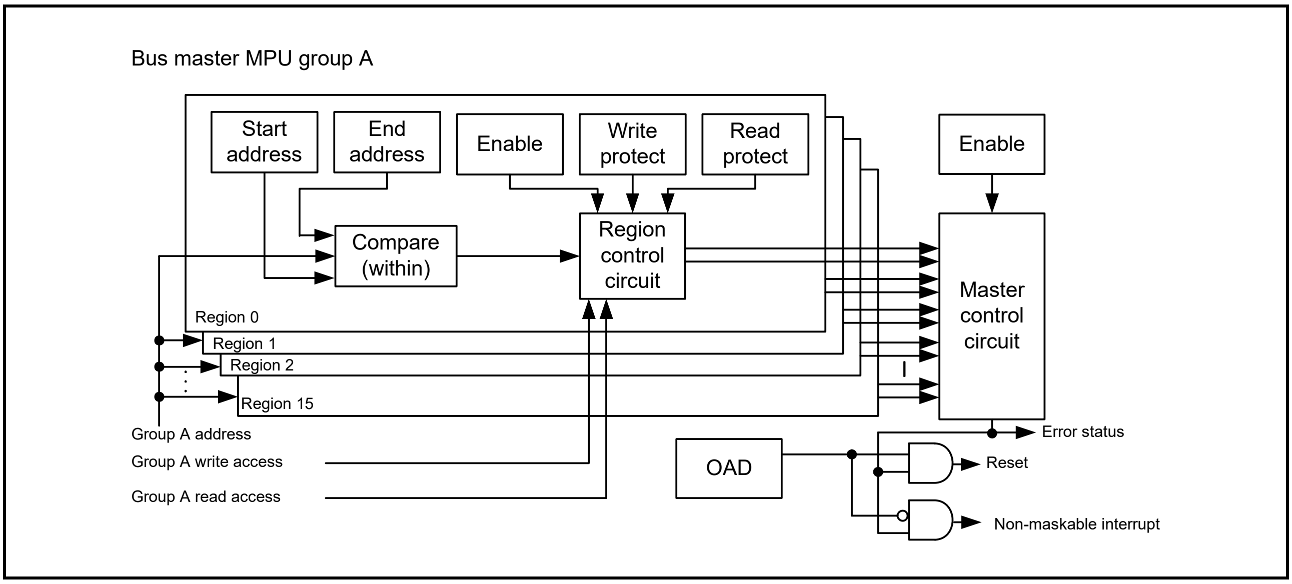

Bus master MPU group A
Start address
End address
Enable
Write protect
Read protect
Compare (within)
Region control circuit
Region 0
Region 1
Region 2
Region 15
Enable
Master control circuit
Group A address
Group A write access
Group A read access
OAD
Error status
Reset
Non-maskable interrupt

Figure 16.4 MPU bus master group A

16.4.1 Register Descriptions

Note: Bus access must be stopped before writing to MPU registers.

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

<table>
<tr><td></td><td>b31</td><td>b30</td><td>b29</td><td>b28</td><td>b27</td><td>b26</td><td>b25</td><td>b24</td><td>b23</td><td>b22</td><td>b21</td><td>b20</td><td>b19</td><td>b18</td><td>b17</td><td>b16</td></tr>
<tr><td></td><td colspan="16">MMPUSA[31:16]</td></tr>
<tr><td>Value after reset:</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr>
<tr><td></td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr>
<tr><td></td><td colspan="16">MMPUSA[15:0]</td></tr>
<tr><td>Value after reset:</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>0</td><td>0</td></tr>
</table>

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MMPUSA[31:0]	Region Start Address	Address where the region starts, for use in region determination. The lower 2 bits should be 0.	R/W

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

<table>
<tr><td></td><td>b31</td><td>b30</td><td>b29</td><td>b28</td><td>b27</td><td>b26</td><td>b25</td><td>b24</td><td>b23</td><td>b22</td><td>b21</td><td>b20</td><td>b19</td><td>b18</td><td>b17</td><td>b16</td></tr>
<tr><td></td><td colspan="16">MMPUEA[31:16]</td></tr>
<tr><td>Value after reset:</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr>
<tr><td></td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr>
<tr><td></td><td colspan="16">MMPUEA[15:0]</td></tr>
<tr><td>Value after reset:</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>1</td><td>1</td></tr>
</table>

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	MMPUEA[31:0]	Region End Address	Address where the region ends, for use in region determination. The lower 2 bits should be 1.	R/W

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

Bit	Symbol	Bit name	Description	R/W
b1	RP	Read Protection	0: Read access permitted 1: Read access protected.	R/W
b2	WP	Write Protection	0: Write access permitted 1: Write access protected.	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ENABLE, RP, and WP bits are individually configurable for each group A region n.

ENABLE bit (Region Enable)

The ENABLE bit enables or disables group A region n unit.

When the ENABLE bit is set to 1, the RP bit and the WP bit can be set to permit or protect access to the region that is set in MMPUSAn and MMPUEAn. When the ENABLE bit is set to 0, no region is specified for group A region n access.

RP bit (Read Protection)

The RP bit enables or disables read protection for group A region n. The RP bit is available when the ENABLE bit is set to 1.

WP bit (Write Protection)

The WP bit enables or disables write protection for group A region n. The WP bit is available when the ENABLE bit is set to 1.

Table 16.5 Function of region control circuit

MMPUACAn.ENABLE	MMPUACAn.RP	MMPUACAn.WP	Access	Region	Output of group A region n unit
0	—	—	Read	—	Outside of region
			Write		Outside of region
1	0	0	Read	Inside	Permitted region
				Outside	Outside of region
			Write	Inside	Permitted region
				Outside	Outside of region
	0	1	Read	Inside	Permitted region
				Outside	Outside of region
			Write	Inside	Protection region
				Outside	Outside of region
	1	0	Read	Inside	Protection region
				Outside	Outside of region
			Write	Inside	Permitted region
				Outside	Outside of region
	1	1	Read	Inside	Protection region
				Outside	Outside of region
			Write	Inside	Protection region
				Outside	Outside of 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	Protected region	Don't care	Don't care	Generate error
1	Don't care	Protected region	Don't care	Generate error
1	Don't care	Don't care	Protected region	Generate error
1	Outside of region	Outside of region	Outside of region	Generate error
Other case				No error

A master MPU error occurs on the following conditions:

- MMPUCTLA.ENABLE = 1, and output of one or more region n units is to a protected region
- MMPUCTLA.ENABLE = 1, and output of all region n units is outside of region.

Other cases are for 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 enable or disable writes to the OAD and ENABLE bits	R/(W)*1

Note 1. Write data is not saved.

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 to the KEY[7:0] bits simultaneously, using halfword access.

OAD bit (Operation After Detection)

The OAD bit generates either a reset or non-maskable interrupt when access to the protected region is detected by the bus master MPU. When the OAD bit is set, write A5h to the KEY[7:0] bits simultaneously using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the ENABLE and OAD bits. When writing to the ENABLE and OAD bits, write A5h to the KEY[7:0] bits 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 are 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 writes are permitted 1: All bus master MPU group A register writes are protected. Reads are 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 enable or disable writes to the PROTECT bit	R/(W)*1

Note 1. Write data is not saved.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

MMPUPTA.PROTECT controls the bus master MPU group A protection registers. The following registers are protected by MMPUPTA.PROTECT:

- MMPUSAn
- MMPUEAn
- MMPUACAn
- MMPUCTLA.

When the PROTECT bit is set, write A5h to the KEY[7:0] bits simultaneously using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the PROTECT bit. When writing to the PROTECT bit, write A5h to the KEY[7:0] bits 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 00h.

16.4.2 Operation

16.4.2.1 Memory protection

The bus master MPU monitors memory access using control settings made individually for the access control regions. If access to a protected region is detected, the bus master MPU generates a memory protection error.

The bus master MPU can be set for up to 16 protected regions. Protected regions include those with overlapping permitted and protected regions, and those with two overlapping permitted regions.

The bus master MPU has group A. The memory protection function checks the address of the bus for the master group and all master group accesses are protected. The bus master MPU sets the permission for all of the regions after reset. Setting MMPUCTLA.ENABLE to 1 protects all of the regions. Each region sets up a permitted region within the protected region. If access to the protected region is detected, the bus master MPU generates an error.

Figure 16.5 shows the use case of a bus master MPU.

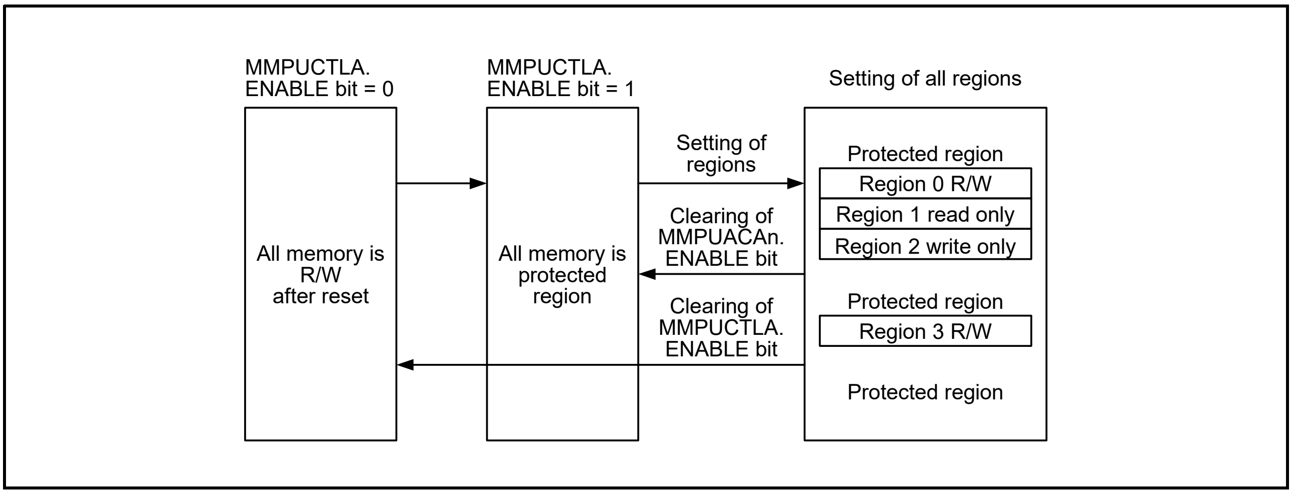

MMPUCTLA.
ENABLE bit = 0
MMPUCTLA.
ENABLE bit = 1
Setting of all regions
All memory is R/W after reset
All memory is protected region
Setting of regions
Clearing of MMPUACAn. ENABLE bit
Clearing of MMPUCTLA. ENABLE bit
Protected region
Region 0 R/W
Region 1 read only
Region 2 write only
Protected region
Region 3 R/W
Protected region

Figure 16.5 Use case of bus master MPU

Figure 16.6 shows the access permission or protection for the overlapping bus master MPU regions.

Access control for the overlapping regions is as follows:

- The region is handled as a protected region when output of one or more region units is a protected region
- The region is handled as a protected region when output of all region units is outside of the regions
- Other cases are handled as permitted regions.

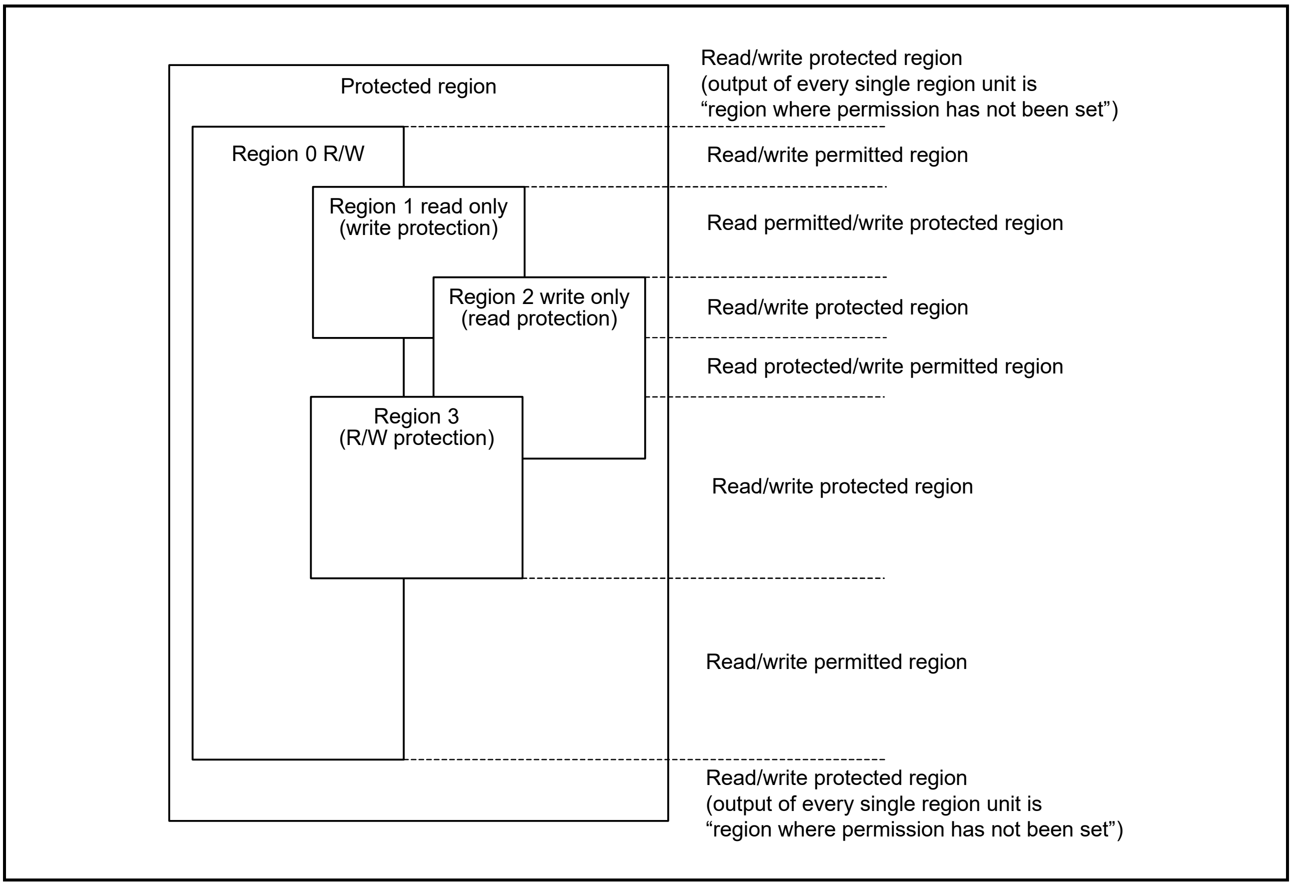

Protected region
Region 0 R/W
Region 1 read only (write protection)
Region 2 write only (read protection)
Region 3 (R/W protection)
Read/write protected region (output of every single region unit is "region where permission has not been set")
Read/write permitted region
Read permitted/write protected region
Read/write protected region
Read protected/write permitted region
Read/write protected region
Read/write permitted region
Read/write protected region (output of every single region unit is "region where permission has not been set")

Figure 16.6 Access permission or protection by overlap of the bus master MPU regions

Figure 16.7 shows the register setting flow after reset. During this register setting, stop all the masters except the CPU.

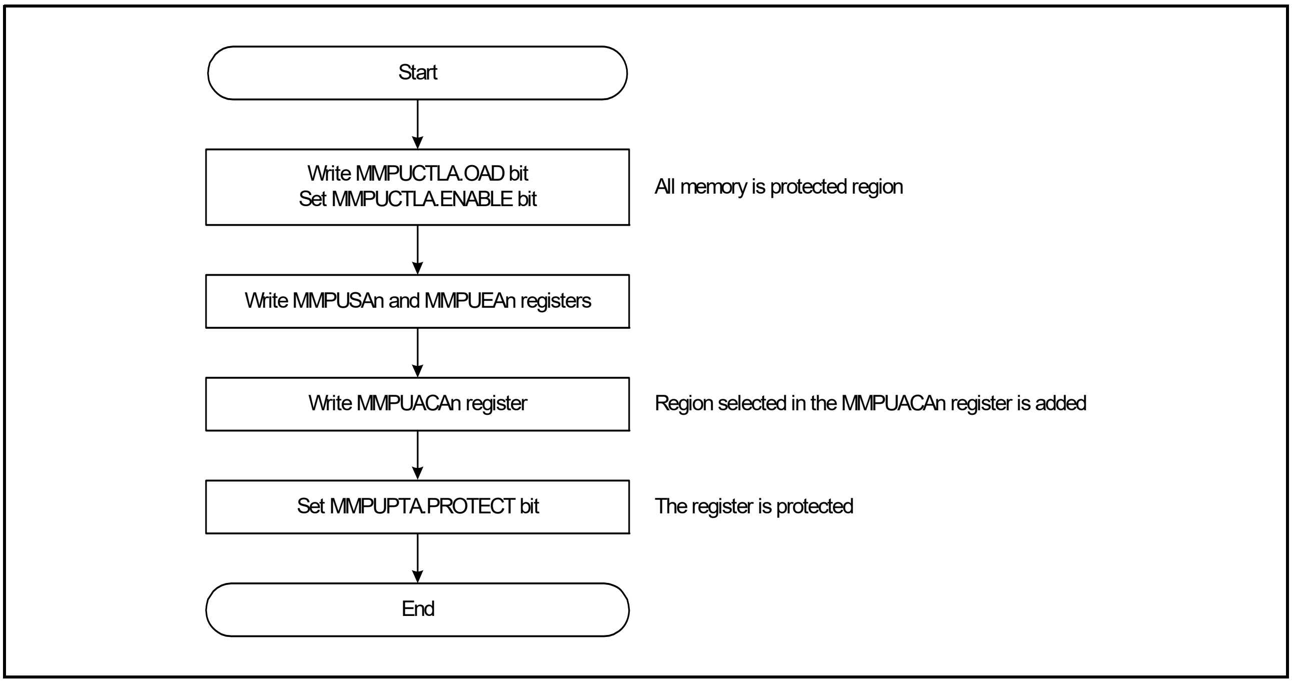

Start
Write MMPUCTLA.OAD bit
Set MMPUCTLA.ENABLE bit
All memory is protected region
Write MMPUSAn and MMPUEAn registers
Write MMPUACAn register
Region selected in the MMPUACAn register is added
Set MMPUPTA.PROTECT bit
The register is protected
End

Figure 16.7 Register setting flow after reset

Figure 16.8 shows the register setting flow for adding regions. During this register setting, stop all masters except the CPU.

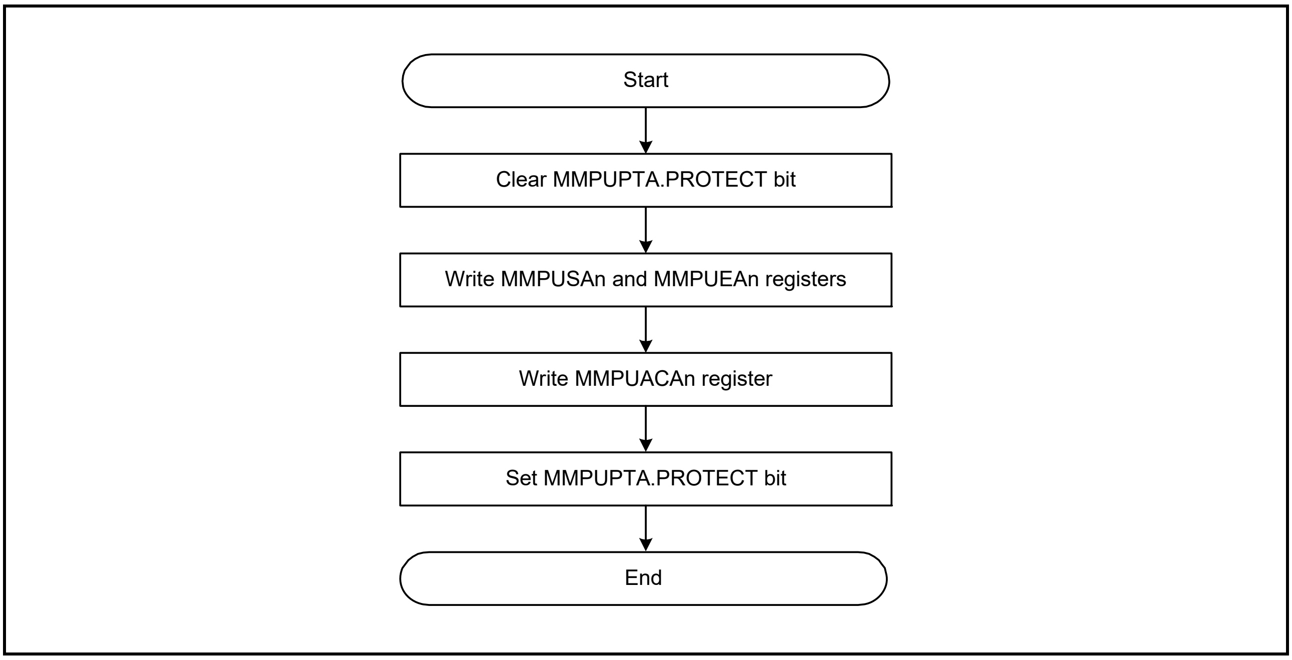

Start
Clear MMPUPTA.PROTECT bit
Write MMPUSAn and MMPUEAn registers
Write MMPUACAn register
Set MMPUPTA.PROTECT bit
End

Figure 16.8 Register setting flow for region addition

16.4.2.2 Protecting the registers

To protect the registers related to the bus master MPU, set the PROTECT bit in the MMPUPTA register.

16.4.2.3 Memory protection error

If access to the protected region is detected, the bus master MPU generates an error. Set the OAD bit to select whether the error is reported as a non-maskable interrupt or reset. The non-maskable interrupt status is indicated in ICU.NMISR.BUSMST. For details, see section 14, Interrupt Controller Unit (ICU). The reset status is indicated in SYSTEM.RSTSR1.BUSMRF. For details, see section 6, Resets.

16.5 Bus Slave MPU

The bus slave MPU monitors access to the bus slave functions, such as flash or SRAM. The bus slave function can be accessed from two bus masters, the CPU, and the bus master MPU group A. The bus slave MPU has a separate protection register for each of the two bus masters, with individual access protection control. If access to a protected region is detected, the bus slave MPU generates a reset or a non-maskable interrupt, and can store the bus error status, error access status, and bus error address in the I/O Registers. For details, see 15.3.3 and 15.3.4 in section 15, Buses. The supported access control information for the individual regions consists of permissions to read and write.

Table 16.7 lists the specifications of the bus slave MPU and Figure 16.9 shows a block diagram.

Table 16.7 Specifications of bus slave MPU

Specifications	Description
Protected bus master	Bus master MPU group A: DMA bus
Protected slave functions	Memory bus 3: Code flash memory Memory bus 4: SRAM0 Internal peripheral bus 1: Connected to peripheral modules related system control Internal peripheral bus 3: Connected to peripheral modules (CAC, ELC, I/O ports, POEG, RTC, WDT, IWDT, IIC, CAN, ADC14, DAC12, and DOC) Internal peripheral bus 4: Connected to peripheral modules (SCI, SPI, and CRC) Internal peripheral bus 5: Connected to peripheral modules (KINT, AGT, USBFS, DAC8, OPAMP, ACMPLP, and CTSU) Internal peripheral bus 7: Connected to SecureIP (SCE5) Internal peripheral bus 9: Flash memory (in P/E) and data flash memory
Access-control information settings for individual regions	Permission to read and write
Operation after detection	Reset, non-maskable interrupt, or exception
Protection of register	Register can be protected from illegal writes

The bus slave MPU is located on each bus slave side and controls the permission or protection of access from each bus master to each bus slave.

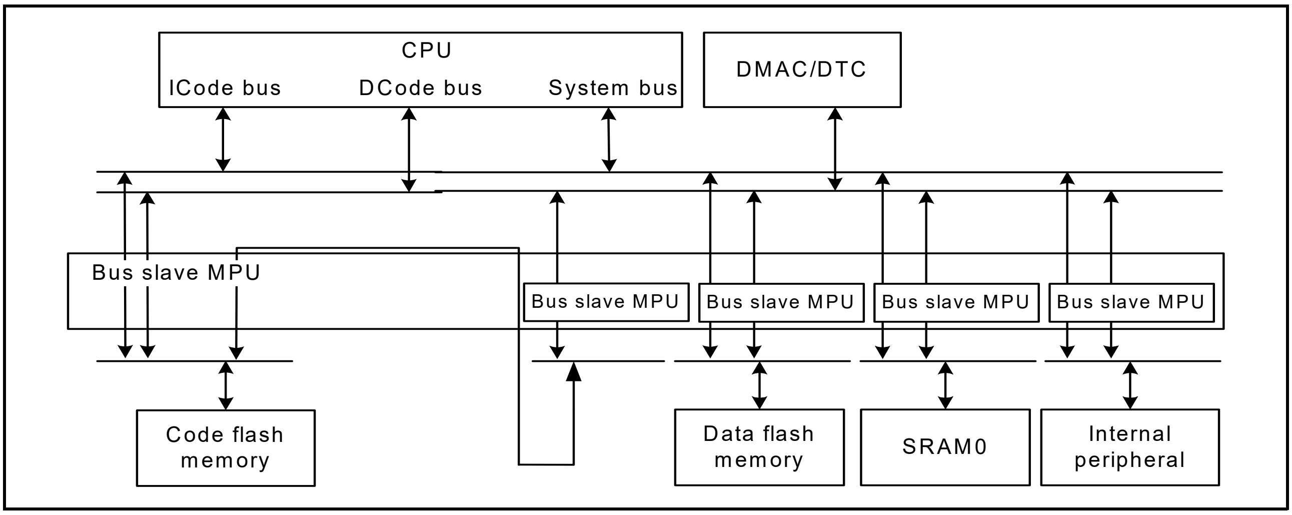


Figure 16.9 Bus slave MPU block diagram

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 memory protection disabled 1: Master group A read memory protection enabled.	R/W
b3	WPGRPA	Master Group A Write Protection	0: Master group A write memory protection disabled 1: Master group A write 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 reads on memory bus 3.

WPGRPA bit (Master Group A Write Protection)

The WPGRPA bit enables or disables memory protection for master group A writes on 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: Memory protection for CPU read disabled 1: Memory protection for CPU read enabled.	R/W
b1	WPCPU	CPU Write Protection	0: Memory protection for CPU write disabled 1: Memory protection for CPU write enabled.	R/W
b2	RPGRPA	Master Group A Read protection	0: Memory protection for master group A read disabled 1: Memory protection for master group A read enabled.	R/W
b3	WPGRPA	Master Group A Write protection	0: Memory protection for master group A write disabled 1: Memory protection for master group A write 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 reads on internal peripheral bus 9.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for CPU writes on internal peripheral bus 9.

RPGRPA bit (Master Group A Read protection)

The RPGRPA bit enables or disables memory protection for master group A reads on internal peripheral Bus 9.

WPGRPA bit (Master Group A Write protection)

The WPGRPA bit enables or disables memory protection for master group A writes on 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: Memory protection for CPU read disabled 1: Memory protection for CPU read enabled.	R/W
b1	WPCPU	CPU Write protection	0: Memory protection for CPU write disabled 1: Memory protection for CPU write enabled.	R/W
b2	RPGRPA	Master Group A Read protection	0: Memory protection for master group A read disabled 1: Memory protection for master group A read enabled.	R/W
b3	WPGRPA	Master Group A Write protection	0: Memory protection for master group A write disabled 1: Memory protection for master group A write 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 reads on memory bus 4.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for CPU writes on memory bus 4.

RPGRPA bit (Master Group A Read protection)

The RPGRPA bit enables or disables memory protection for master group A reads on memory bus 4.

WPGRPA bit (Master Group A Write protection)

The WPGRPA bit enables or disables memory protection for master group A writes on memory bus 4.

16.5.1.4 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: Memory protection for CPU read disabled 1: Memory protection for CPU read enabled.	R/W
b1	WPCPU	CPU Write protection	0: Memory protection for CPU write disabled 1: Memory protection for CPU write enabled.	R/W

Bit	Symbol	Bit name	Description	R/W
b2	RPGRPA	Master Group A Read protection	0: Memory protection for master group A read disabled 1: Memory protection for master group A read enabled.	R/W
b3	WPGRPA	Master Group A Write protection	0: Memory protection for master group A write disabled 1: Memory protection for master group A write 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 reads on internal peripheral bus 1.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for CPU writes on internal peripheral bus 1.

RPGRPA bit (Master Group A Read protection)

The RPGRPA bit enables or disables memory protection for master group A reads on internal peripheral bus 1.

WPGRPA bit (Master Group A Write protection)

The WPGRPA bit enables or disables memory protection for master group A writes on internal peripheral bus 1.

16.5.1.5 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: Memory protection for CPU read disabled 1: Memory protection for CPU read enabled.	R/W
b1	WPCPU	CPU Write protection	0: Memory protection for CPU write disabled 1: Memory protection for CPU write enabled.	R/W
b2	RPGRPA	Master Group A Read protection	0: Memory protection for master group A read disabled 1: Memory protection for master group A read enabled.	R/W
b3	WPGRPA	Master Group A Write protection	0: Memory protection for master group A write disabled 1: Memory protection for master group A write 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 reads on 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 writes on 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 reads on 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 writes on internal peripheral bus 3, internal peripheral bus 4, and internal peripheral bus 5.

16.5.1.6 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: Memory protection for CPU read disabled 1: Memory protection for CPU read enabled.	R/W
b1	WPCPU	CPU Write protection	0: Memory protection for CPU write disabled 1: Memory protection for CPU write enabled.	R/W
b2	RPGRPA	Master Group A Read protection	0: Memory protection for master group A read disabled 1: Memory protection for master group A read enabled.	R/W
b3	WPGRPA	Master Group A Write protection	0: Memory protection for master group A write disabled 1: Memory protection for master group A write 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 reads on internal peripheral bus 7.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for CPU writes on internal peripheral bus 7.

RPGRPA bit (Master Group A Read protection)

The RPGRPA bit enables or disables memory protection for master group A reads on internal peripheral bus 7.

WPGRPA bit (Master Group A Write protection)

The WPGRPA bit enables or disables memory protection for master group A writes on internal peripheral bus 7.

16.5.1.7 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 permitted 1: All bus slave register writes are protected. Reads are permitted.	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 writes to the OAD and PROTECT bits.	R/(W)*1

Note 1. Write data is not saved.

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 to the KEY[7:0] bits simultaneously using halfword access.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected. The following registers are protected by SMPUCTL.PROTECT:

- SMPUMBIU
- SMPUFBIU
- SMPUSRAM0
- SMPUP0BIU
- SMPUP2BIU
- SMPUP6BIU

When the PROTECT bit is set, write A5h to the KEY[7:0] bits simultaneously using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the OAD and PROTECT bits. When writing to the OAD and PROTECT bits, 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 are always read as 00h.

16.5.2 Functions

16.5.2.1 Memory protection

The bus slave MPU monitoring uses access control information that is set for the individual access control registers, whether or not access by the bus slaves violates the access control settings. If access to the protected region is detected, the bus slave MPU generates a memory protection error.

The bus slave MPU is enabled by writing 1 to the Write Protect (WPCPU or WPGRPA) bit or the Read Protect (RPCPU or RPGRPA) bit in the access control registers (SMPUMBIU, SMPUFBIU, SMPUSRAM0, SMPUP0BIU, SMPUP2BIU, and SMPUP6BIU).

16.5.2.2 Protecting the 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 a protected region is detected, the bus slave MPU generates a memory protection error. Set the OAD bit to select whether the error is reported as a non-maskable interrupt or reset.

The non-maskable interrupt status is indicated in ICU.NMISR.BUSSST. For details, see section 14, Interrupt Controller Unit (ICU). The reset status is indicated in SYSTEM.RSTSR1.BUSSRF. For details, see section 6, Resets.

16.6 Security MPU

The MCU incorporates a security MPU with four secure regions that include the code flash, the SRAM, and two security functions. The secure regions can be protected from non-secure program accesses. A non-secure program cannot access a protected region.

Table 16.8 lists the specifications of the security MPU and Figure 16.10 shows a block diagram.

Table 16.8 Security MPU specifications (1 of 2)

Specifications	Description
Secure regions	Code flash, SRAM, two security functions

Table 16.8 Security MPU specifications (2 of 2)

Specifications	Description
Protected regions	0000 0000h to 00FF FFFFh (Code flash memory) 1FF0 0000h to 200F FFFFh (SRAM) 400C 0000h to 400D FFFFh 4010 0000h to 407F FFFFh (secure data of security functions)
Number of regions	Program Counter = 2 regions Data Access = 4 regions
Address specification for individual regions	Setting the address where regions start and end
Enable or disable setting for memory protection in individual regions	Settings enabled or disabled for the associated region

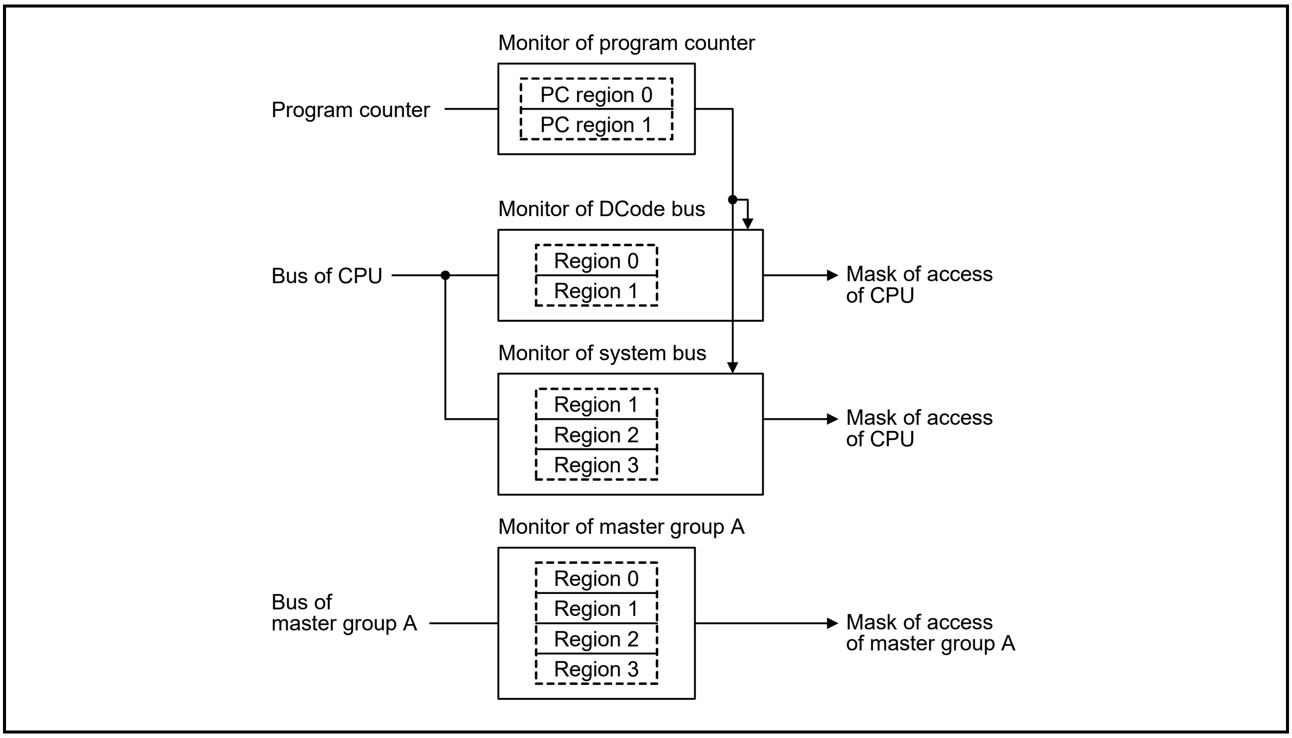

Monitor of program counter
Program counter
PC region 0
PC region 1
Monitor of DCode bus
Bus of CPU
Region 0
Region 1
Mask of access of CPU
Monitor of system bus
Region 1
Region 2
Region 3
Mask of access of CPU
Monitor of master group A
Bus of master group A
Region 0
Region 1
Region 2
Region 3
Mask of access of master group A

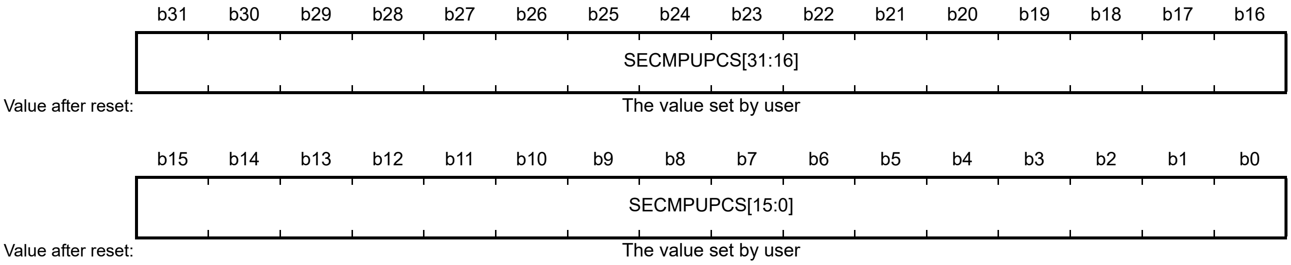
Figure 16.10 Security MPU block diagram

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


b31 b30 b29 b28 b27 b26 b25 b24 b23 b22 b21 b20 b19 b18 b17 b16
SECMPUPCS[31:16]
Value after reset:
The value set by user
b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
SECMPUPCS[15:0]
Value after reset:
The value set by user

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 00FF FFFCh and 1FF0 0000h to 200F FFFCh, excluding reserved areas. When setting this register value in the option-setting memory, the write value of the lower 2 bits should be 0.	R

The SECMPUPCSn and SECMPUPCEn registers specify the security fetch region for the code flash (0000 0000h to 00FF FFFFh, not including the reserved areas) or SRAM (1FF0 0000h to 200F FFFFh, not including the reserved areas). The secure program is executed in the memory space defined by the SECMPUPCSn and SECMPUPCEn registers and can access the secure data specified in the SECMPUSm and SECMPUEm registers (m = 0 to 3). The set up of memory mirror space (0200 0000h to 027F FFFFh) for MMF is not allowed.

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

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SECMPUPCE[31:16]															
Value after reset:	The value set by user															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SECMPUPCE[15:0]															
Value after reset:	The value set by user															

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 00FF FFFFh and 1FF0 0003h to 200F FFFFh, excluding reserved areas. When setting this register value in the option-setting memory, the write value of the lower 2 bits should be 1.	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, excluding reserved areas. When setting this register value in the option-setting memory, the write value of the lower 2 bits should be 0.	R
b31 to b24	—	Reserved	These bits are read as 0. When setting this register value in the option-setting memory, the write value of these bits should be 0.	R

The SECMPUS0 and SECMPUE0 registers specify the secure program and the flash data (0000 0000h to 00FF FFFFh, not including the reserved areas). The memory space defined in the SECMPUS0 and SECMPUE0 registers can only be accessed from the secure program set up in the SECMPUPCSn and SECMPUPCEn registers.

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, excluding reserved areas. When setting this register value in the option-setting memory, the write value of the lower 2 bits should be 1.	R
b31 to b24	—	Reserved	These bits are read as 0. When setting this register value in the option-setting memory, the write value of these bits should be 0.	R

16.6.1.5 Security MPU Region 1 Start Address Register (SECMPUS1)

Address(es): SECMPUS1 0000 0420h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SECMPUS1[31:16]															
Value after reset:	The value set by user															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SECMPUS1[15:0]															
Value after reset:	The value set by user														0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	SECMPUS1[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 1FF0 0000h to 200F FFFCh, excluding reserved areas. When setting this register value in the option-setting memory, the write value of the lower 2 bits should be 0, and the write value of bits [31:20] should be 1FFh or 200h.	R

The SECMPUS1 and SECMPUE1 registers specify the secure data of the SRAM (1FF0 0000h to 200F FFFFh, not including the reserved areas). The memory space defined in the SECMPUS1 and SECMPUE1 registers can only be accessed from the secure program set up in the SECMPUPCSn and SECMPUPCEn registers.

Setting of the stack area and the vector table are prohibited.

16.6.1.6 Security MPU Region 1 End Address Register (SECMPUE1)

Address(es): SECMPUE1 0000 0424h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SECMPUE1[31:16]															
Value after reset:	The value set by user															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SECMPUE1[15:0]															
Value after reset:	The value set by user														1	1

Bit	Symbol	Bit name	Description	R/W
b31 to b0	SECMPUE1[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 1FF0 0003h to 200F FFFFh, excluding reserved areas. When setting this register value in the option-setting memory, the write value of the lower 2 bits should be 1, and the write value of bits [31:20] should be 1FFh or 200h.	R

16.6.1.7 Security MPU Region 2 Start Address Register (SECMPUS2)

Address(es): SECMPUS2 0000 0428h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	SECMPUS2[22:16]						
Value after reset:	0	1	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
	SECMPUS2[15:0]															
Value after reset:	The value set by user														0	0

Bit	Symbol	Bit name	Description	R/W
b22 to b0	SECMPUS2[22:0]	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 400C 0000h to 400D FFFCh and 4010 0000h to 407F FFFCh. When setting this register value in the option-setting memory, the write value of the lower 2 bits should be 0.	R
b31 to b23	—	Reserved	These bits are read as 0100 0000 0b. When setting this register value in the option-setting memory, the write value of these bits should be 0100 0000 0b.	R

The SECMPUS2 and SECMPUE2 registers specify the secure data of security functions (400C 0000 to 400D FFFFh and 4010 0000 to 407F FFFFh). The memory space defined in the SECMPUS2 and SECMPUE2 registers can only be accessed from the secure program set up in the SECMPUPCSn and SECMPUPCEn registers.

16.6.1.8 Security MPU Region 2 End Address Register (SECMPUE2)

Address(es): SECMPUE2 0000 042Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	SECMPUE2[22:16]						
Value after reset:	0	1	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
	SECMPUE2[15:0]															
Value after reset:	The value set by user														1	1

Bit	Symbol	Bit name	Description	R/W
b22 to b0	SECMPUE2[22:0]	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 400C 0003h to 400D FFFFh and 4010 0003h to 407F FFFFh. When setting this register value in the option-setting memory, the write value of the lower 2 bits should be 1.	R
b31 to b23	—	Reserved	These bits are read as 0100 0000 0b. When setting this register value in the option-setting memory, the write value of these bits should be 0100 0000 0b.	R

16.6.1.9 Security MPU Region 3 Start Address Register (SECMPUS3)

Address(es): SECMPUS3 0000 0430h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	SECMPUS3[22:16]						
Value after reset:	0	1	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
	SECMPUS3[15:0]															
Value after reset:	The value set by user														0	0

Bit	Symbol	Bit name	Description	R/W
b22 to b0	SECMPUS3[22:0]	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 400C 0000h to 400D FFFCh and 4010 0000h to 407F FFFCh. When setting this register value in the option-setting memory, the write value of the lower 2 bits should be 0.	R
b31 to b23	—	Reserved	These bits are read as 0100 0000 0b. When setting this register value in the option-setting memory, the write value of these bits should be 0100 0000 0b.	R

The SECMPUS3 and SECMPUE3 registers specify the secure data of the security functions (400C 0000h to 400D FFFFh and 4010 0000h to 407F FFFFh). The memory space defined in the SECMPUS3 and SECMPUE3 registers can only be accessed from the secure program set up in the SECMPUPCSn and SECMPUPCEn registers.

16.6.1.10 Security MPU Region 3 End Address Register (SECMPUE3)

Address(es): SECMPUE3 0000 0434h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	SECMPUE3[22:16]						
Value after reset:	0	1	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
	SECMPUE3[15:0]															
Value after reset:	The value set by user														1	1

Bit	Symbol	Bit name	Description	R/W
b22 to b0	SECMPUE3[22:0]	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 400C 0003h to 400D FFFFh and 4010 0003h to 407F FFFFh. When setting this register value in the option-setting memory, the write value of the lower 2 bits should be 1.	R
b31 to b23	—	Reserved	These bits are read as 0100 0000 0b. When setting this register value in the option-setting memory, the write value of these bits should be 0100 0000 0b.	R

16.6.1.11 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	—	—	—	—	DIS3	DIS2	DIS1	DIS0
Value after reset:	1	1	1	1	1	1	The value set by user		1	1	1	1	The value set by user			

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
b1	DIS1	Region 1 Disable	0: Security MPU region 1 enabled 1: Security MPU region 1 disabled.	R
b2	DIS2	Region 2 Disable	0: Security MPU region 2 enabled 1: Security MPU region 2 disabled.	R
b3	DIS3	Region 3 Disable	0: Security MPU region 3 enabled 1: Security MPU region 3 disabled.	R
b7 to b4	—	Reserved	These bits are read as 1. When setting this register value in the option-setting memory, the write value of bits [7:4] should be 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. When setting this register value in the option-setting memory, the write value of bits [15:10] should be 1.	R

Note: When flash memory is erased, the security MPU is disabled.

Note: To enable or disable the security MPU, see section 16.6.2, Memory Protection.

DIS0 bit (Region 0 Disable)

The DIS0 bit enables or disables the security MPU region 0. If security MPU region 0 is enabled, the code flash region within the limits set up by the SECMPUS0 and SECMPUE0 is secure data.

DIS1 bit (Region 1 Disable)

The DIS1 bit enables or disables the security MPU region 1. If security MPU region 1 is enabled, the SRAM region within the limits set up by the SECMPUS1 and SECMPUE1 is secure data.

DIS2 bit (Region 2 Disable)

The DIS2 bit enables or disables the security MPU region 2. If security MPU region 2 is enabled, the secure data of the security function region within the limits set up by the SECMPUS2 and the SECMPUE2 is secure data.

DIS3 bit (Region 3 Disable)

The DIS3 bit enables or disables the security MPU region 3. If security MPU region 3 is enabled, the secure data of the security function region within the limits set up by the SECMPUS3 and the SECMPUE3 is secure data.

DISPC0 bit (PC Region 0 Disable)

The DISPC0 bit enables or disables the security MPU PC region 0. If security MPU PC region 0 is enabled, the code flash or the SRAM region within the limits set up by SECMPUPCS0 and SECMPUPCE0 contains a secure program.

DISPC1 bit (PC Region 1 Disable)

The DISPC1 bit enables or disables the security MPU PC region 1. If security MPU PC region 1 is enabled, the code flash or the SRAM region within the limits set up by SECMPUPCS1 and SECMPUPCE1 contains a secure program.

16.6.2 Memory Protection

The security MPU protects the secured regions (the code flash, the SRAM, two security functions) from being accessed by programs other than a secure program. If access to a protected region is detected, the access becomes invalid.

When the security MPU is enabled, DISPC0 or DISPC1 in the Security MPU Access Control Register (SECMPUAC) must be set to 0, and DIS0, DIS1, DIS2, or DIS3 in the Security MPU Access Control Register (SECMPUAC) must be set to 0.

When the security MPU is disabled, all bits in DISPC0, DISPC1, DIS0, DIS1, DIS2 and DIS3 in the Security MPU Access Control Register (SECMPUAC) must be set to 1.

Other settings in the Security MPU Access Control Register (SECMPUAC) are prohibited.

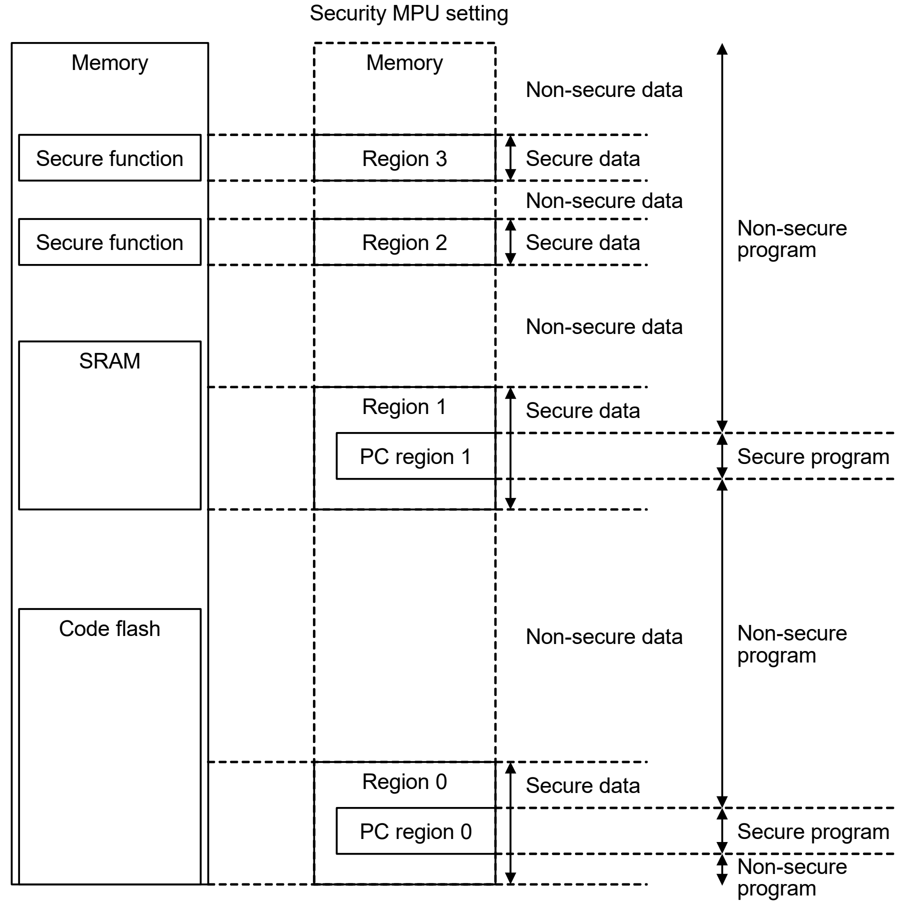
The security MPU provides access protection in the following conditions:

- Secure data is accessed from a non-secure program
- Secure data is accessed from other than the CPU (DMAC, DTC)
- Secure data is accessed from the debugger.

Secure data is accessible only from a secure program.

Note:

Secure program:	Code flash or SRAM regions within the limits set up by the SECMPUPCS0 and SECMPUPCE0, Code flash or SRAM regions within the limits set up by SECMPUPCS1 and SECMPUPCE1.
Non-secure program:	All regions outside the secure program.
Secure data:	Code flash region within the limits set up by SECMPUS0 and SECMPUE0. SRAM region within the limits set up by SECMPUS1 and SECMPUE1, Security function region within the limits set up by SECMPUS2 and SECMPUE2, Security function region within the limits set up by SECMPUS3 and SECMPUE3.


Security MPU setting
Memory
Secure function
Secure function
SRAM
Code flash
Memory
Region 3
Region 2
Region 1
PC region 1
Region 0
PC region 0
Non-secure data
Secure data
Non-secure data
Secure data
Non-secure data
Secure data
Non-secure data
Secure data
Non-secure program
Secure program
Non-secure program
Secure program
Non-secure program

Secure program in code flash (PC region 0) can access all data (secure data and non-secure data).
Secure program in SRAM (PC region 1) can access all data (secure data and non-secure data).
Non-secure program (not PC region 0 or PC region 1) cannot access secure data (region 0, region 1, region 2, region 3).
Non-secure program (not PC region 0 or PC region 1) can access non-secure data.

Figure 16.11 Use case of security MPU

16.6.3 Notes on Debug

The protected memory cannot be debugged if the security MPU is enabled. Disable the security MPU when debugging a secure program.

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 MCU includes a 4-channel DMA Controller (DMAC) that 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 a block diagram.

Table 17.1 DMAC specifications

Parameter		Description
Number of channels		4 channels (DMACm, 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: • 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		Module-stop state can be set to reduce power consumption

Note 1. For details on DMAC activation sources, see Table 14.3, Interrupt Vector Table in section 14, Interrupt Controller Unit (ICU).

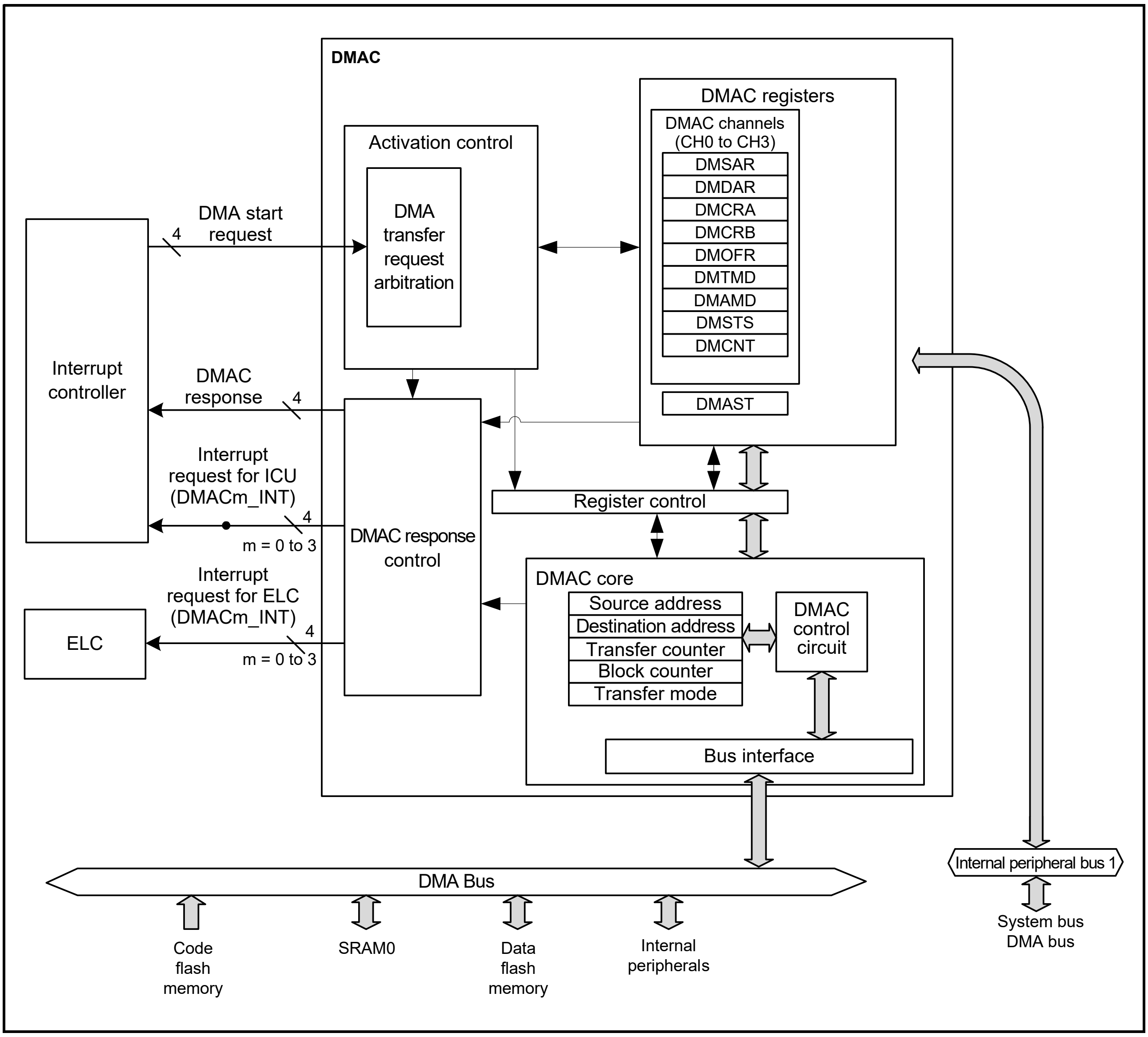

DMAC
Activation control
DMA transfer request arbitration
DMAC registers
DMAC channels (CH0 to CH3)
DMSAR
DMDAR
DMCRA
DMCRB
DMOFR
DMTMD
DMAMD
DMSTS
DMCNT
DMAST
DMA start request
4
Interrupt controller
DMAC response
4
Interrupt request for ICU (DMACm_INT)
4
m = 0 to 3
Interrupt request for ELC (DMACm_INT)
4
m = 0 to 3
ELC
DMAC response control
Register control
DMAC core
Source address
Destination address
Transfer counter
Block counter
Transfer mode
DMAC control circuit
Bus interface
DMA Bus
Internal peripheral bus 1
System bus
DMA bus
Code flash memory
SRAM0
Data flash memory
Internal peripherals

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

							DMCRAH									
	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

	DMCRAL															
	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

- Repeat transfer mode, block transfer mode

							DMCRAH									
	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

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

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 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: Disable 1: Enable.	R/W
b1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable	0: Disable 1: Enable.	R/W
b2	RPTIE	Repeat Size End Interrupt Enable	0: Disable 1: Enable.	R/W
b3	ESIE	Transfer Escape End Interrupt Enable	0: Disable 1: Enable.	R/W
b4	DTIE	Transfer End Interrupt Enable	0: Disable 1: Enable.	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 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 is set 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. When 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 is set 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 the RPTIE 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 is set 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).

When the RPTIE 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 is set 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)

The ESIE 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. To clear the transfer escape end interrupt, clear this bit or the ESIF flag in DMSTS to 0.

DTIE bit (Transfer End Interrupt Enable)

The DTIE 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 flag in DMSTS is set to 1. To clear the transfer end interrupt, clear this bit or the DTIF flag 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: Source address is incremented 1 1: Source address is decremented.	R/W

DARA[4:0] bits (Destination Address Extended Repeat Area)

The DARA[4:0] 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)

The DM[1:0] 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)

The SARA[4:0] 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)

The SM[1:0] 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 (1 of 2)

SARA[4:0] or DARA[4:0]	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

Table 17.2 SARA[4:0] or DARA[4:0] settings and corresponding repeat areas (2 of 2)

SARA[4:0] or DARA[4:0]	Extended repeat area
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, but 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: Disable 1: Enable.	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 the SWREQ bit generates a DMA transfer request. After the DMA transfer starts, SWREQ is set to 0 if the CLRS bit is set to 0. SWREQ does not clear if CLRS bit is 1. The DMA transfer request can be issued again 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 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, the CLRS bit specifies whether to clear the SWREQ bit to 0 after the DMA transfer starts. When the CLRS bit is set to 0, SWREQ is set to 0 after the DMA transfer starts. When the CLRS bit is set to 1, SWREQ does not clear to 0. The DMA transfer request can be issued again 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 occurred 1: Interrupt occurred.	R/W*1
b3 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R/W
b4	DTIF	Transfer End Interrupt Flag	0: No interrupt occurred 1: Interrupt occurred.	R/W*1
b6, b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R/W
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)

The ESIF 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 flag
- When 1 is written to the DTE bit in DMCNT.

DTIF flag (Transfer End Interrupt Flag)

The DTIF flag indicates that a transfer end interrupt occurred.

[Setting conditions]

- In normal transfer mode, when the specified number of unit transfers completes (DMCRAL becomes 0 on completion of transfer)
- In repeat transfer mode, when the specified number of repeat transfer operations completes (DMCRB becomes 0 on completion of transfer)
- In block transfer mode, when the specified number of blocks is transferred (DMCRB becomes 0 on completion of transfer).

[Clearing conditions]

- When 0 is written to this flag
- When 1 is written to the DTE bit in DMCNT.

ACT flag (DMA Active Flag)

The ACT 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 DMAC 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: Disable 1: Enable.	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 the DMST bit to 1 enables DMAC activation for all channels. When the DMST 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, the DMA transfer is suspended after the current data transfer associated with a single transfer request is complete. To resume the 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). A transfer end interrupt request can be generated after completion of the specified number of transfer operations, except when in free running mode. 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 (1 of 2)

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

Table 17.3 Register update operation in normal transfer mode (2 of 2)

Register	Function	Update operation after completion of a transfer for one transfer request
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.

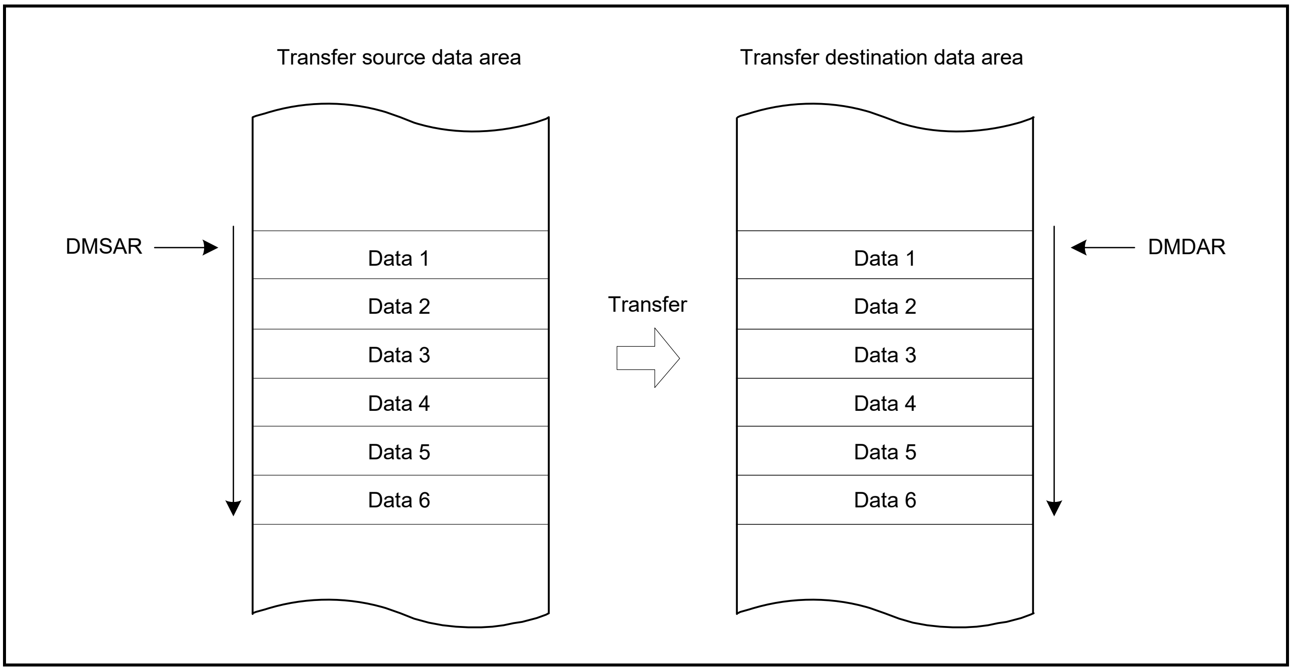


Figure 17.2 Operation in normal transfer mode

(2) Repeat transfer mode

In repeat transfer mode, one data unit is transferred for one transfer request.

The repeat transfer size, up to a maximum of 1K data units, is set in DMACm.DMCRA. The number of repeat transfers, up to a maximum number of 64K, is set in DMACm.DMCRB. The total data transfer size can be set to a maximum of 64M data units (1K data units × 64K repeat transfers).

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, the DMA transfer can be stopped and a repeat size end interrupt can be requested. 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, or 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, or 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

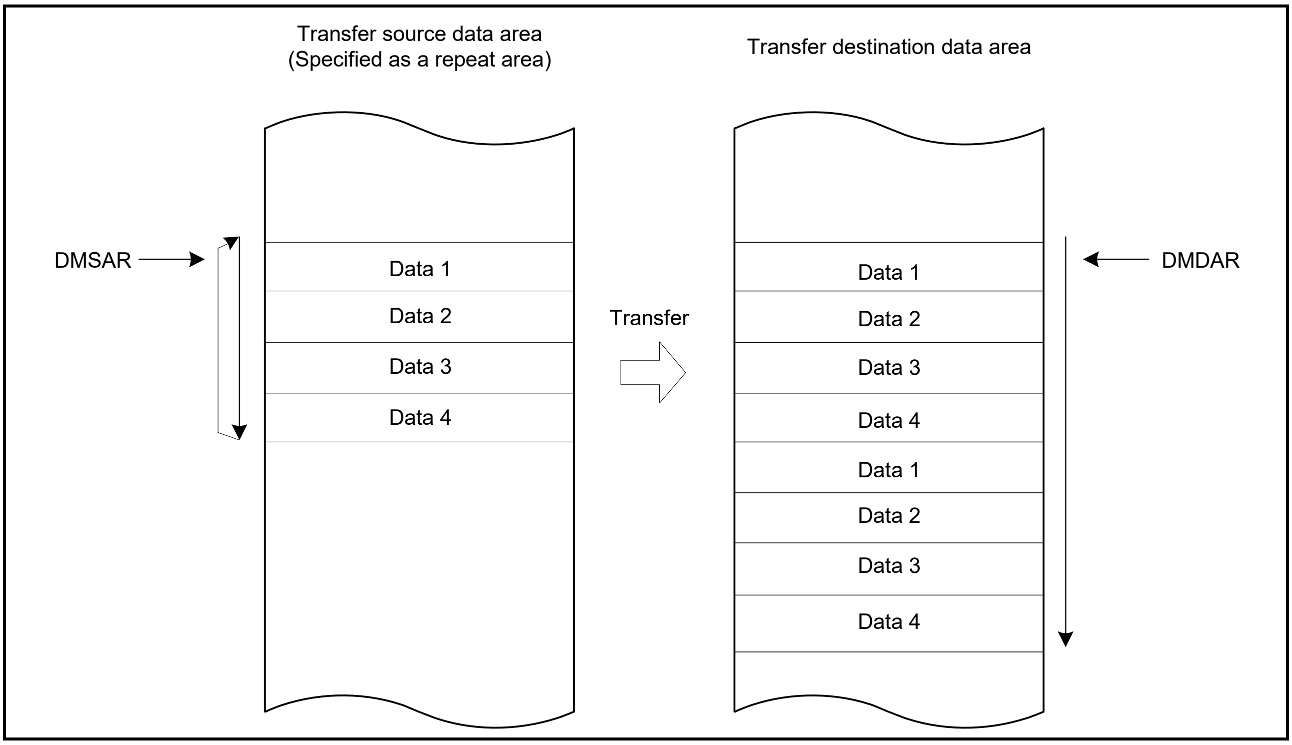


Figure 17.3 Operation in repeat transfer mode

(3) Block transfer mode

In block transfer mode, a single data block is transferred for one transfer request. The block transfer size, up to a maximum of 1K data units, is set in DMACm.DMCRA. The number of blocks transfers, up to a maximum of 64K, is set in DMACm.DMCRB. A total data transfer size up to a maximum of 64M data units (1K data units × 64K block transfers) can be set.

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

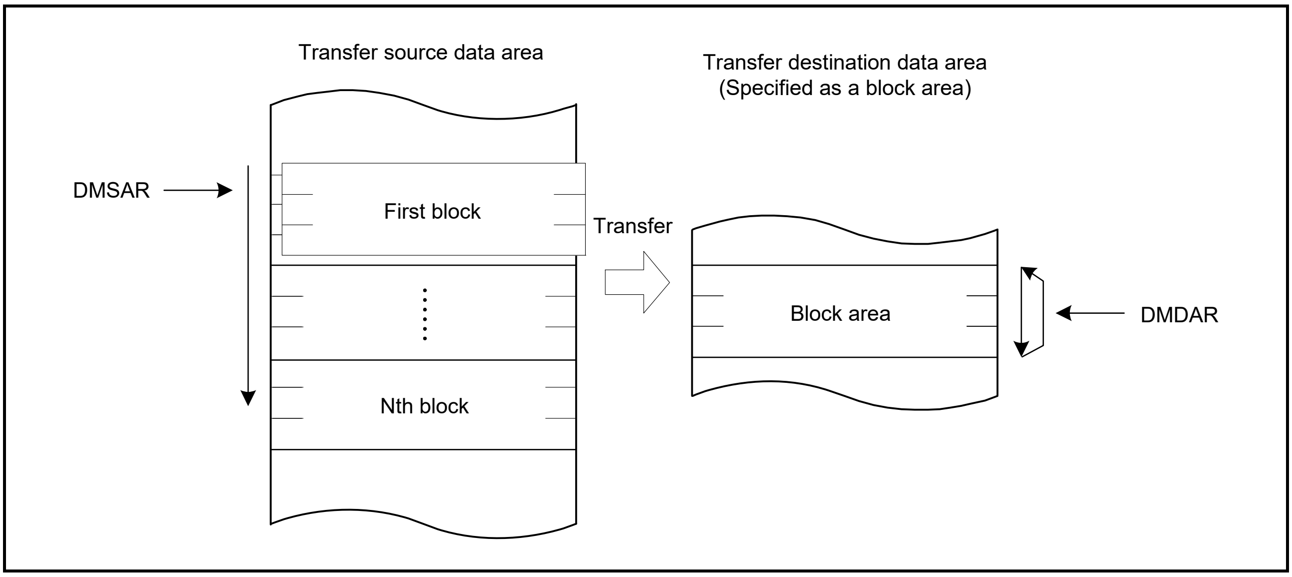

Transfer source data area
Transfer destination data area
(Specified as a block area)
DMSAR
First block
Transfer
Block area
DMDAR
Nth block

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 DMA Source Address Register (DMSAR) and DMA 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.

Example:
8 bytes are specified as an extended repeat area by the lower 3 bits of DMACm.DMSAR (SARA[4:0] in DMACm.DMAMD = 00011b)
The data size is 8 bits (SZ[1:0] bits in DMACm.DMTMD = 00b)

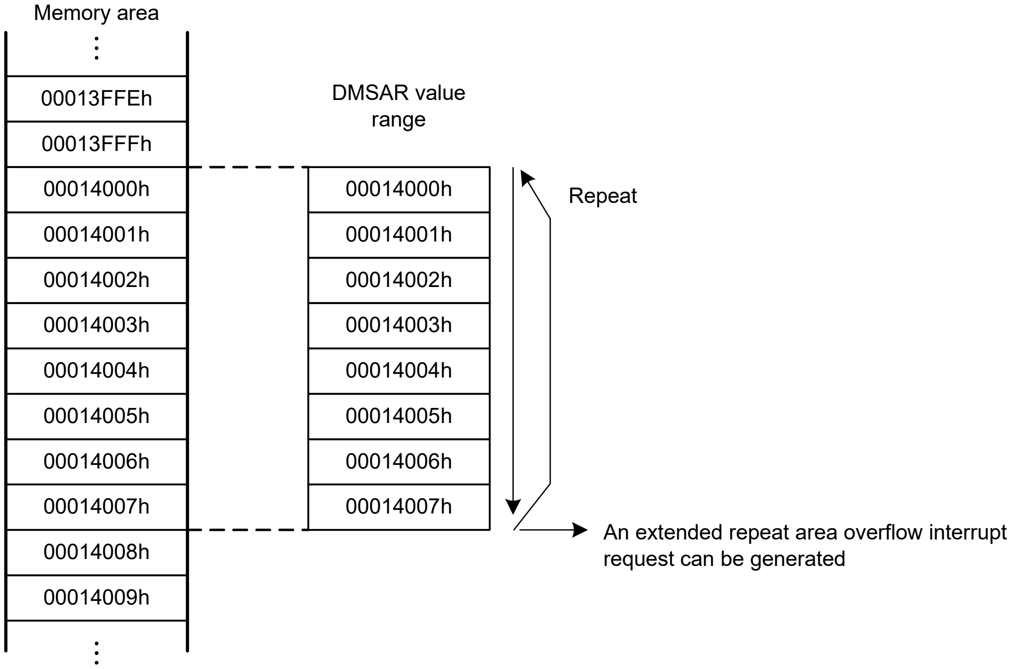

Memory area
⋮
00013FFEh
00013FFFh
00014000h
00014001h
00014002h
00014003h
00014004h
00014005h
00014006h
00014007h
00014008h
00014009h
⋮
DMSAR value range
00014000h
00014001h
00014002h
00014003h
00014004h
00014005h
00014006h
00014007h
Repeat
An extended repeat area overflow interrupt request can be generated

Figure 17.5 Example of extended repeat area operation

When using an extended repeat area overflow interrupt in block transfer mode, consider the following points:

- 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.

Example:
8 bytes are specified as an extended repeat area by the lower 3 bits of DMACm.DMSAR (SARA[4:0] bits in DMACm.DMAMD = 00011b), block transfer mode with block size 5 is set (DMACm.DMCRA = 00050005h), and transfer source address is not specified as a block area. Data size is 8 bits (SZ[1:0] bits in DMACm.DMTMD = 00b).

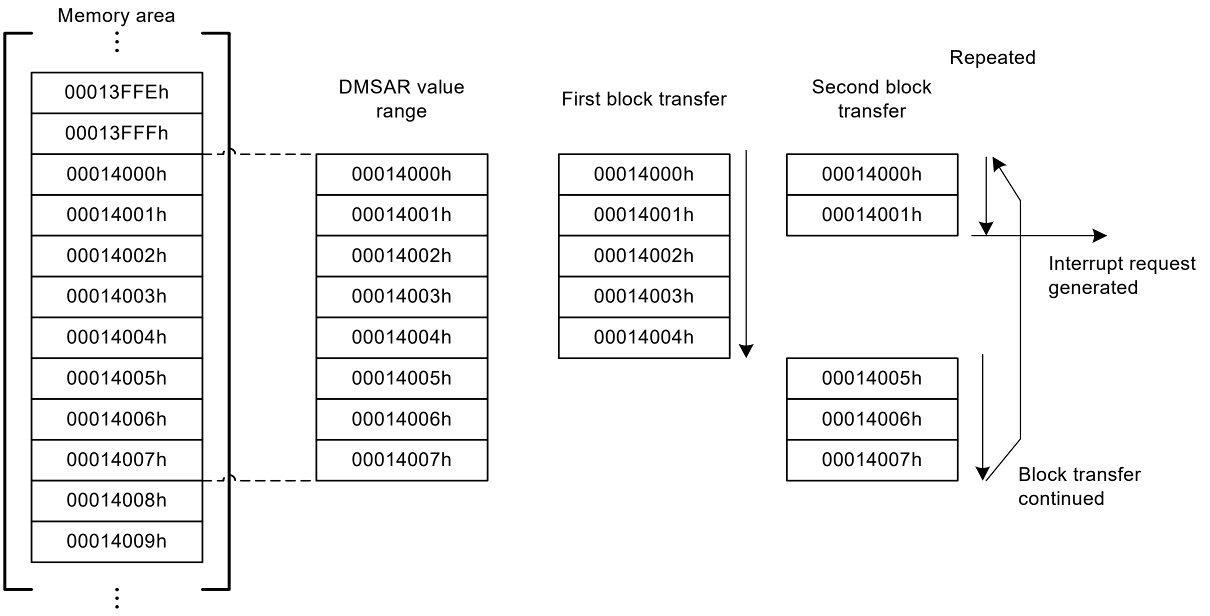

Memory area
00013FFEh
00013FFFh
00014000h
00014001h
00014002h
00014003h
00014004h
00014005h
00014006h
00014007h
00014008h
00014009h
DMSAR value range
00014000h
00014001h
00014002h
00014003h
00014004h
00014005h
00014006h
00014007h
First block transfer
00014000h
00014001h
00014002h
00014003h
00014004h
Second block transfer
00014000h
00014001h
00014005h
00014006h
00014007h
Repeated
Interrupt request generated
Block transfer continued

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 two’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 in two’s complement, obtained by the following formula:
two’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.

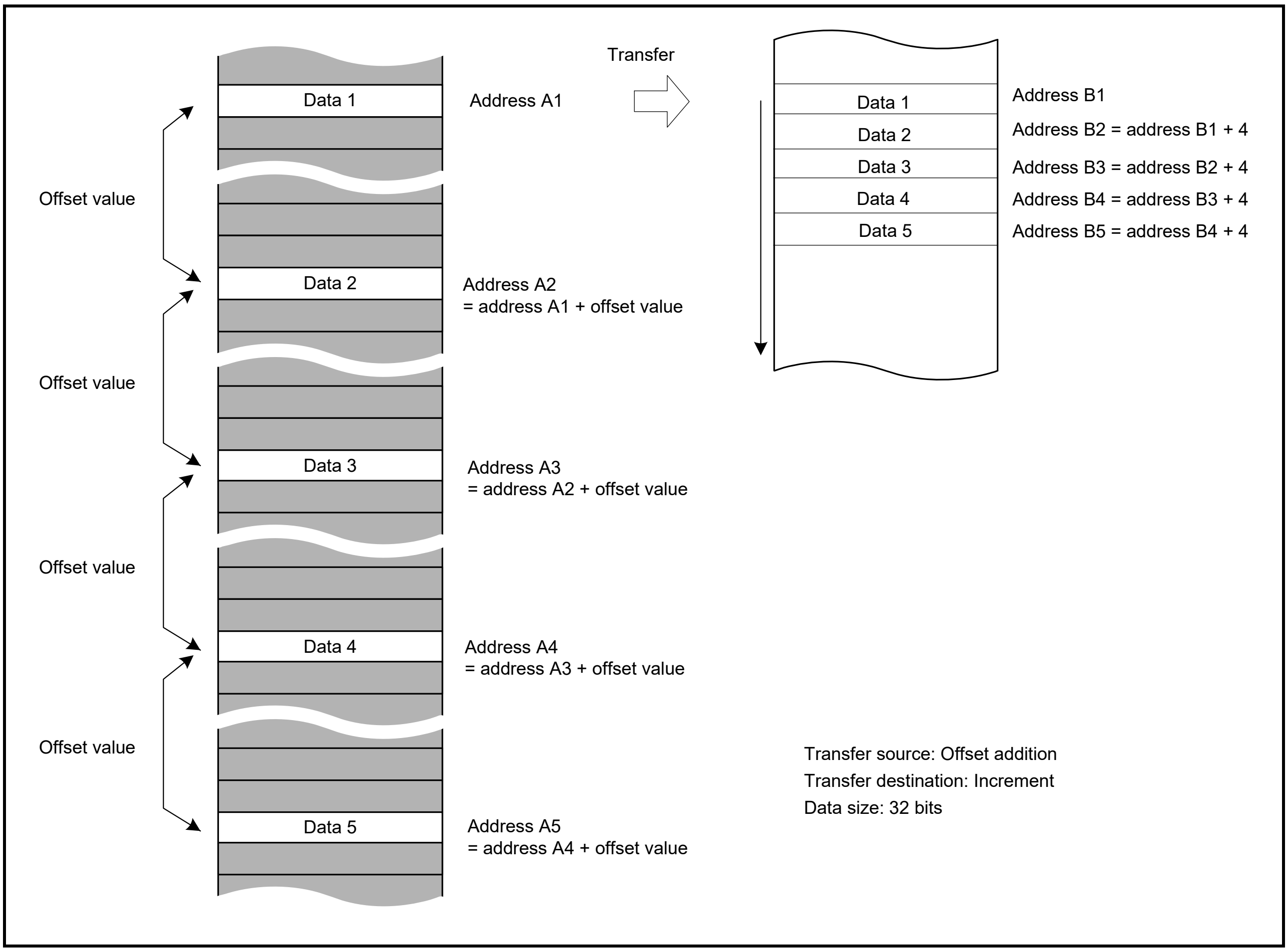

Transfer
Data 1
Address A1
Offset value
Data 2
Address A2
= address A1 + offset value
Offset value
Data 3
Address A3
= address A2 + offset value
Offset value
Data 4
Address A4
= address A3 + offset value
Offset value
Data 5
Address A5
= address A4 + offset value
Data 1
Data 2
Data 3
Data 4
Data 5
Address B1
Address B2 = address B1 + 4
Address B3 = address B2 + 4
Address B4 = address B3 + 4
Address B5 = address B4 + 4
Transfer source: Offset addition
Transfer destination: Increment
Data size: 32 bits

Figure 17.7 Example of address updating through 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.

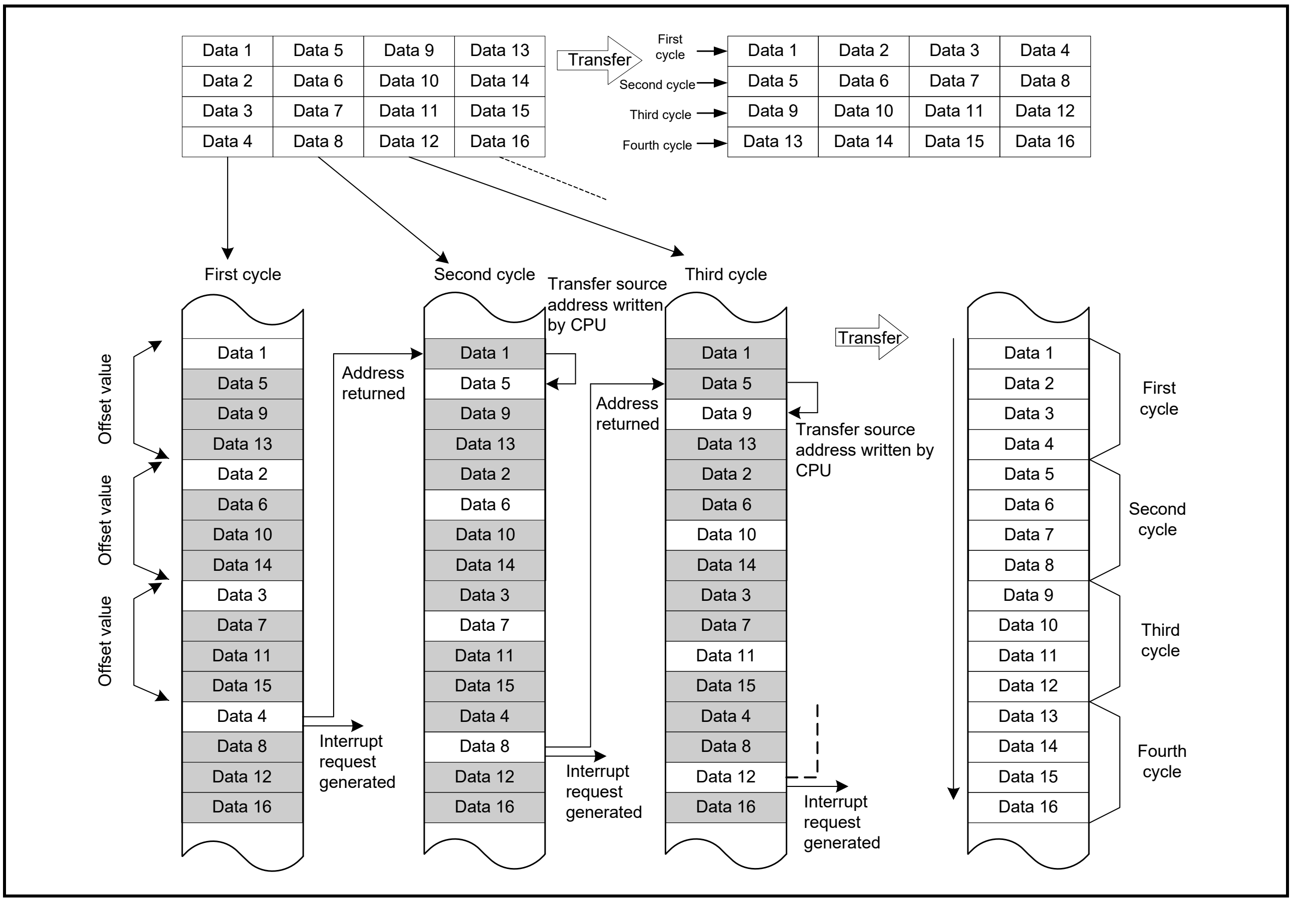

Data 1
Data 5
Data 9
Data 13
Data 2
Data 6
Data 10
Data 14
Data 3
Data 7
Data 11
Data 15
Data 4
Data 8
Data 12
Data 16
Transfer
First cycle
Second cycle
Third cycle
Fourth cycle
Offset value
Address returned
Interrupt request generated
Transfer source address written by CPU
Transfer
Second cycle
Third cycle
Fourth cycle

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 flow of the XY conversion.

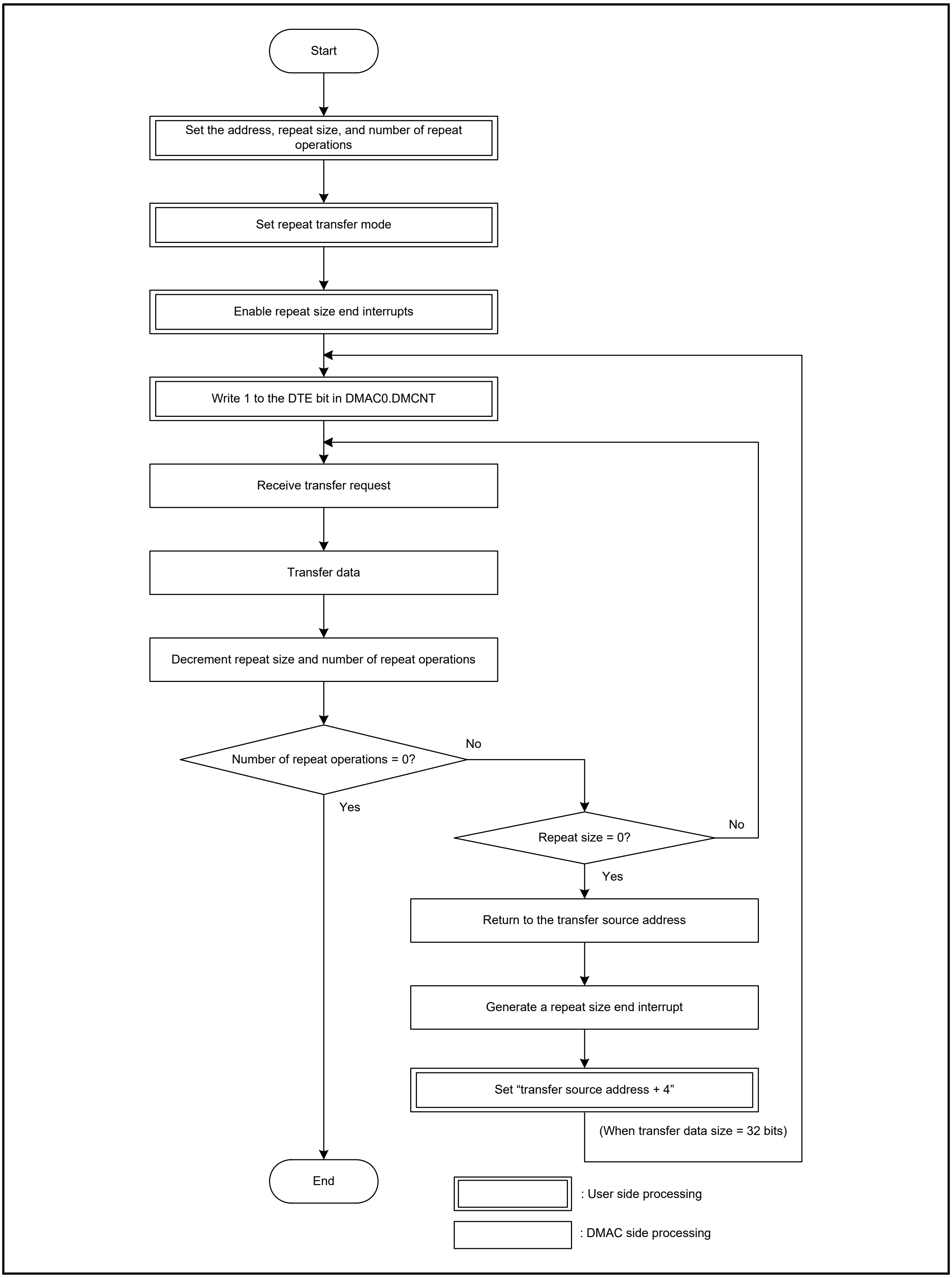

Start
Set the address, repeat size, and number of repeat operations
Set repeat transfer mode
Enable repeat size end interrupts
Write 1 to the DTE bit in DMAC0.DMCNT
Receive transfer request
Transfer data
Decrement repeat size and number of repeat operations
Number of repeat operations = 0?
No
Yes
Repeat size = 0?
No
Yes
Return to the transfer source address
Generate a repeat size end interrupt
Set "transfer source address + 4"
(When transfer data size = 32 bits)
End
: User side processing
: DMAC side processing

Figure 17.9 XY conversion flow 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 through 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 the 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 individually selected for each channel in ICU.DELSRn.DELS[7:0] (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 diagrams show the minimum number of execution cycles.

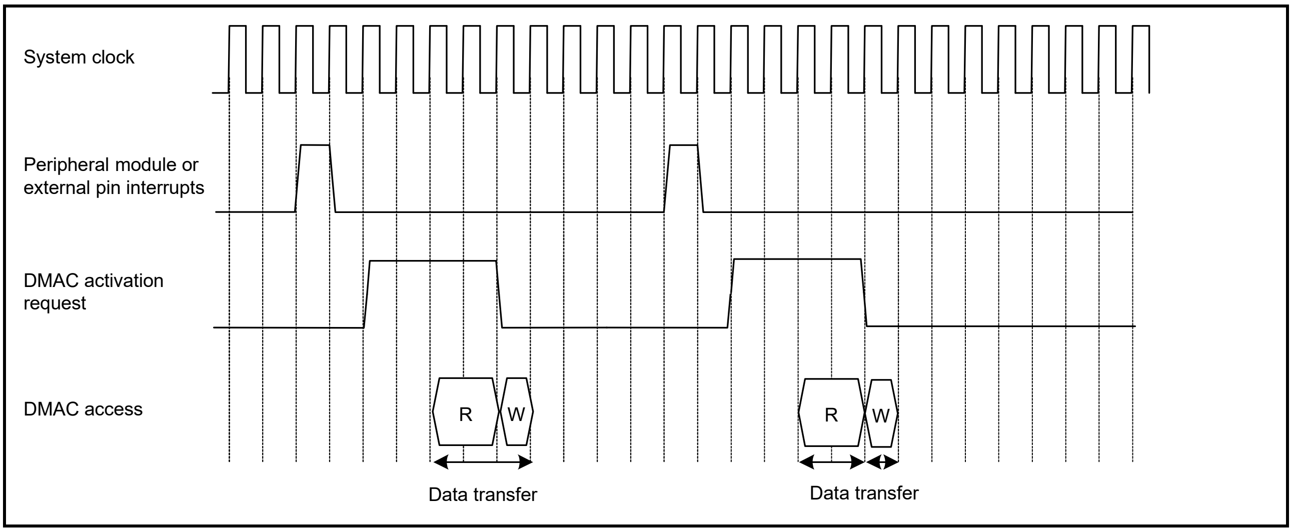


Figure 17.10 DMAC operation timing example 1 with DMA activation by interrupt from peripheral module/ external interrupt input pin, in normal transfer mode or repeat transfer mode

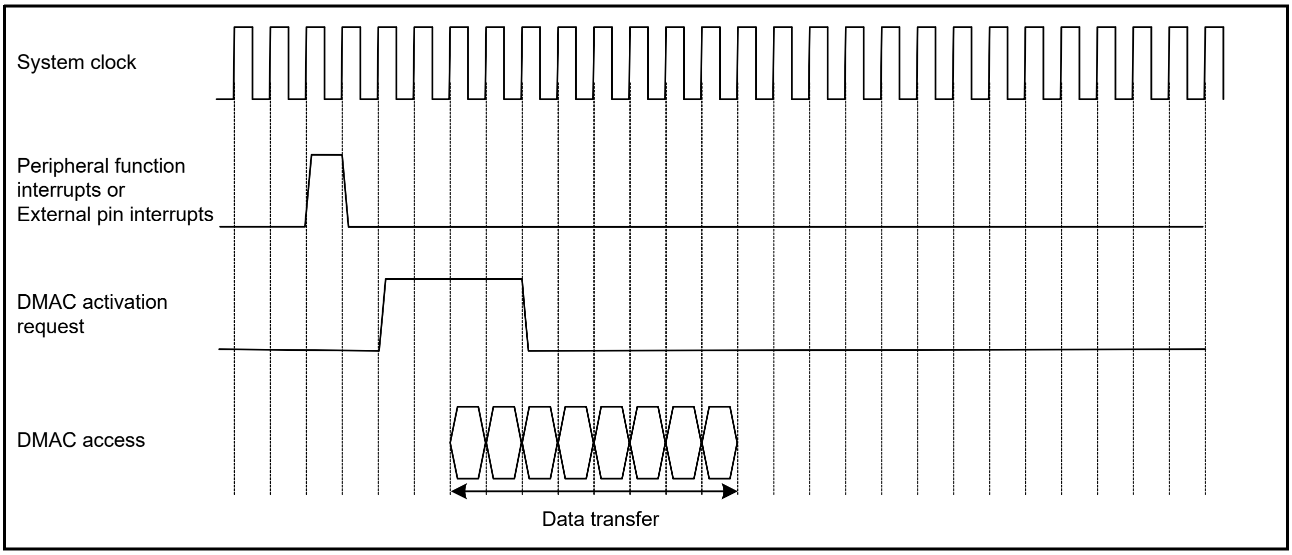


Figure 17.11 **DMAC operation timing example 2 with DMA activation by interrupt from peripheral module/ external interrupt input pin, in block transfer mode with block size = 4**

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: P = Block size (DMCRAH register setting).
Cr = Read destination access cycle.
Cw = Data write destination access cycle.

Note 1. This is the case when the block size is 2 or more. When the block size is 1, normal transfer cycle applies.

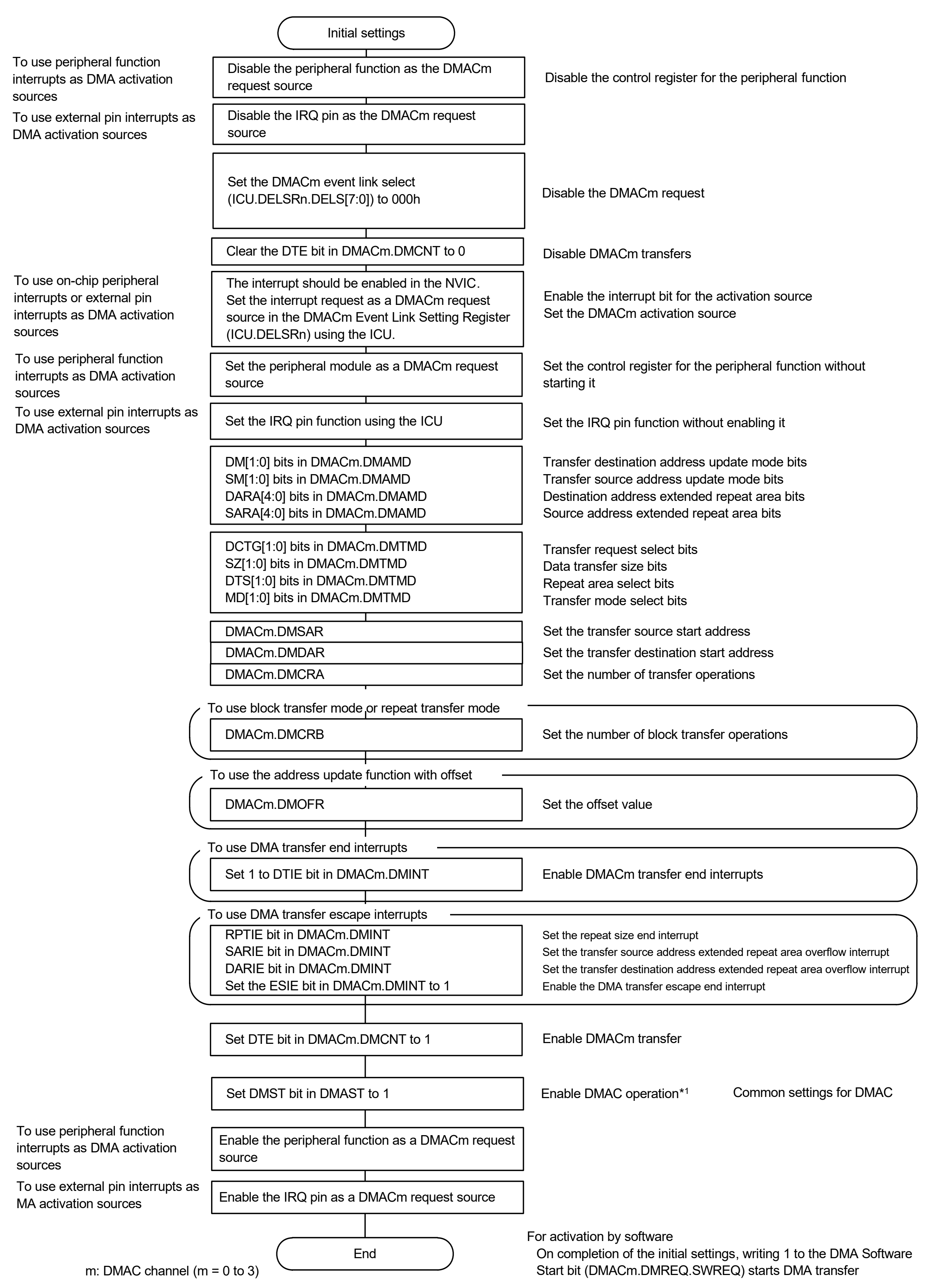
Cr and Cw depend on the access destination. For the number of cycles for each access destination, see section 42, SRAM, section 43, Flash Memory, and section 15, Buses. The frequency ratio of the system clock and the peripheral clock is also taken into consideration.

The unit for +1 in the data transfer (read) column is 1 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.


Initial settings
To use peripheral function interrupts as DMA activation sources
Disable the peripheral function as the DMACm request source
Disable the control register for the peripheral function
To use external pin interrupts as DMA activation sources
Disable the IRQ pin as the DMACm request source
Set the DMACm event link select (ICU.DELSRn.DELS[7:0]) to 000h
Disable the DMACm request
Clear the DTE bit in DMACm.DMCNT to 0
Disable DMACm transfers
To use on-chip peripheral interrupts or external pin interrupts as DMA activation sources
The interrupt should be enabled in the NVIC. Set the interrupt request as a DMACm request source in the DMACm Event Link Setting Register (ICU.DELSRn) using the ICU.
Enable the interrupt bit for the activation source
Set the DMACm activation source
To use peripheral function interrupts as DMA activation sources
Set the peripheral module as a DMACm request source
Set the control register for the peripheral function without starting it
To use external pin interrupts as DMA activation sources
Set the IRQ pin function using the ICU
Set the IRQ pin function without enabling it
DM[1:0] bits in DMACm.DMAMD
SM[1:0] bits in DMACm.DMAMD
DARA[4:0] bits in DMACm.DMAMD
SARA[4:0] bits in DMACm.DMAMD
Transfer destination address update mode bits
Transfer source address update mode bits
Destination address extended repeat area bits
Source address extended repeat area bits
DCTG[1:0] bits in DMACm.DMTMD
SZ[1:0] bits in DMACm.DMTMD
DTS[1:0] bits in DMACm.DMTMD
MD[1:0] bits in DMACm.DMTMD
Transfer request select bits
Data transfer size bits
Repeat area select bits
Transfer mode select bits
DMACm.DMSAR
Set the transfer source start address
DMACm.DMDAR
Set the transfer destination start address
DMACm.DMCRA
Set the number of transfer operations
To use block transfer mode or repeat transfer mode
DMACm.DMCRB
Set the number of block transfer operations
To use the address update function with offset
DMACm.DMOFR
Set the offset value
To use DMA transfer end interrupts
Set 1 to DTIE bit in DMACm.DMINT
Enable DMACm transfer end interrupts
To use DMA transfer escape interrupts
RPTIE bit in DMACm.DMINT
SARIE bit in DMACm.DMINT
DARIE bit in DMACm.DMINT
Set the ESIE bit in DMACm.DMINT to 1
Set the repeat size end interrupt
Set the transfer source address extended repeat area overflow interrupt
Set the transfer destination address extended repeat area overflow interrupt
Enable the DMA transfer escape end interrupt
Set DTE bit in DMACm.DMCNT to 1
Enable DMACm transfer
Set DMST bit in DMAST to 1
Enable DMAC operation*1
Common settings for DMAC
To use peripheral function interrupts as DMA activation sources
Enable the peripheral function as a DMACm request source
To use external pin interrupts as MA activation sources
Enable the IRQ pin as a DMACm request source
End
m: DMAC channel (m = 0 to 3)
For activation by software
On completion of the initial settings, writing 1 to the DMA Software Start bit (DMACm.DMREQ.SWREQ) starts DMA transfer

Note 1. The DMAST.DMST bit setting does not necessarily have to follow the settings for the individual activation sources.

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 flag 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 flag 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 is automatically 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 is automatically 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 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 flag 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 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 flag 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 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 flag 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 1-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 for 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.DELS[7:0] 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.

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.

Table 17.8 lists the interrupt sources and their associated status flags and enable bits. Figure 17.13 shows the schematic logic diagram of the interrupt outputs (DMAC0 to DMAC3). Figure 17.14 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		

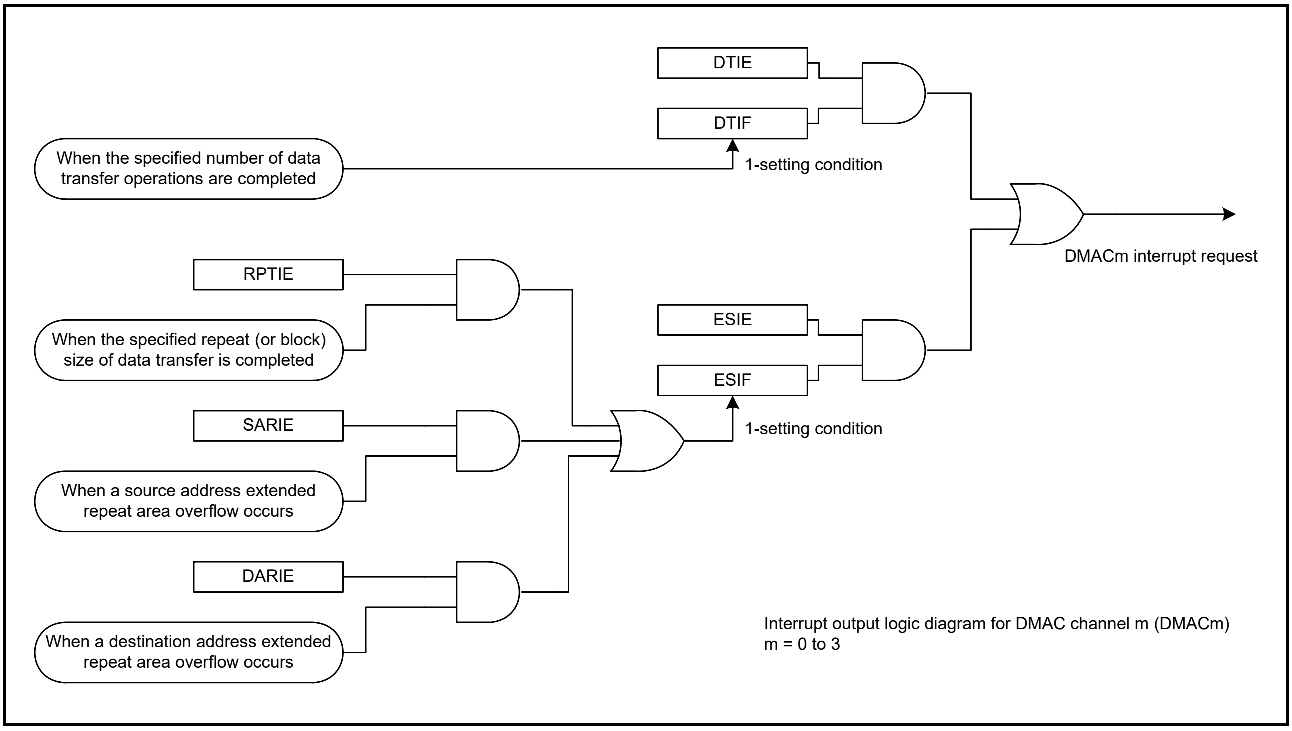

DTIE
DTIF
When the specified number of data transfer operations are completed
1-setting condition
DMACm interrupt request
RPTIE
When the specified repeat (or block) size of data transfer is completed
ESIE
ESIF
SARIE
1-setting condition
When a source address extended repeat area overflow occurs
DARIE
When a destination address extended repeat area overflow occurs
Interrupt output logic diagram for DMAC channel m (DMACm)
m = 0 to 3

Figure 17.13 Schematic logic diagram of interrupt outputs for 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 flag in DMACm.DMSTS to clear a transfer end interrupt, and to the ESIF flag 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 flag in DMSTS of DMACm automatically is set to 0 (interrupt source cleared), and the DMA transfer resumes.

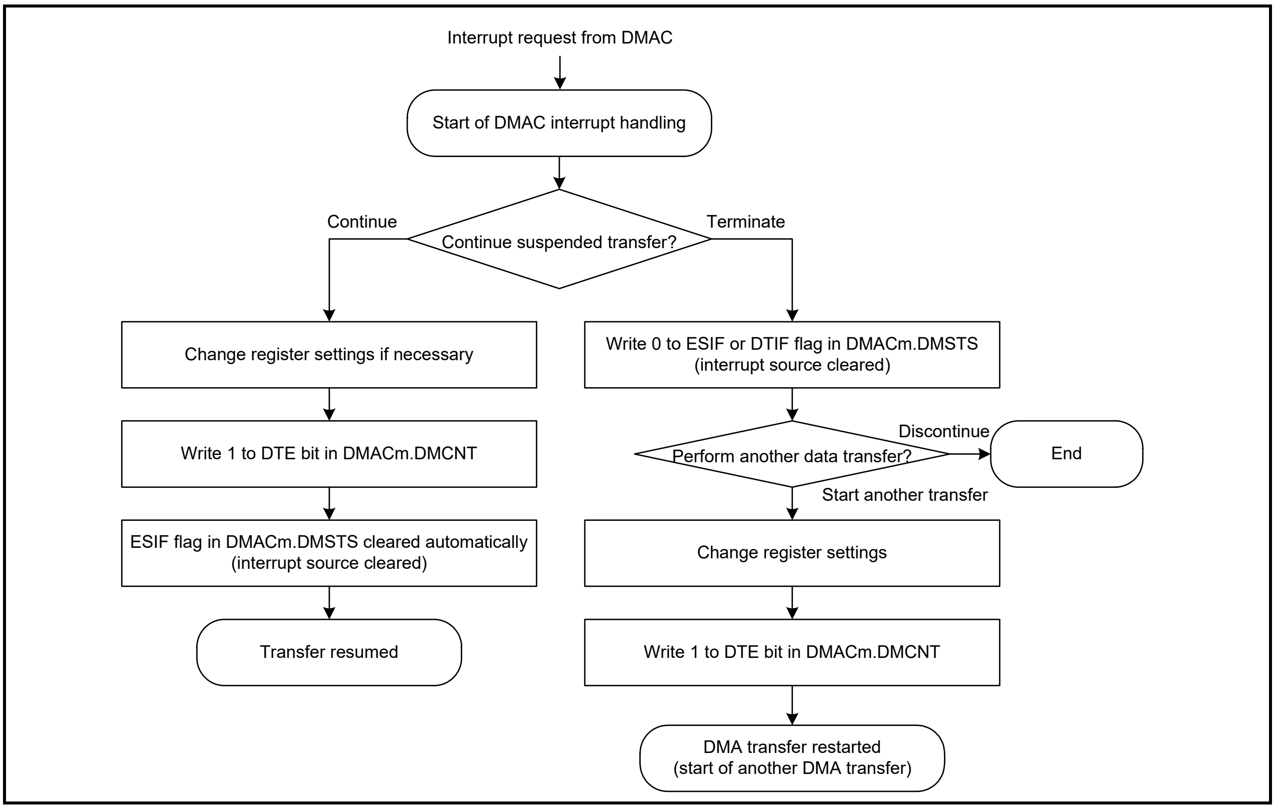

Interrupt request from DMAC
Start of DMAC interrupt handling
Continue
Continue suspended transfer?
Terminate
Change register settings if necessary
Write 0 to ESIF or DTIF flag in DMACm.DMSTS (interrupt source cleared)
Write 1 to DTE bit in DMACm.DMCNT
Perform another data transfer?
Discontinue
End
Start another transfer
ESIF flag in DMACm.DMSTS cleared automatically (interrupt source cleared)
Change register settings
Transfer resumed
Write 1 to DTE bit in DMACm.DMCNT
DMA transfer restarted (start of another DMA transfer)

Figure 17.14 DMAC interrupt handling routine to resume or 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. 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 settings in the sections that follow.

(1) Module-stop function

Writing 1 to the MSTPA22 bit in MSTPCRA enables the module-stop function of the DMAC. If a DMA transfer is in progress when 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) Notes 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 Access to Registers during DMA Transfer

Do not write to the following registers of DMACm while the ACT flag 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.

17.8.2 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.

17.8.3 Setting the 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 bit (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).

17.8.4 Suspending or Restarting DMA Activation

To suspend a DMA activation request, write 0 to the DMAC Event Link Select (ICU.DELSRn.DELS[7:0]) bit. To restart the DMA transfer, write the event number to the ICU.DELSRn.DELS[7:0] bit with the settings shown in section 17.3.7, Activating DMAC.

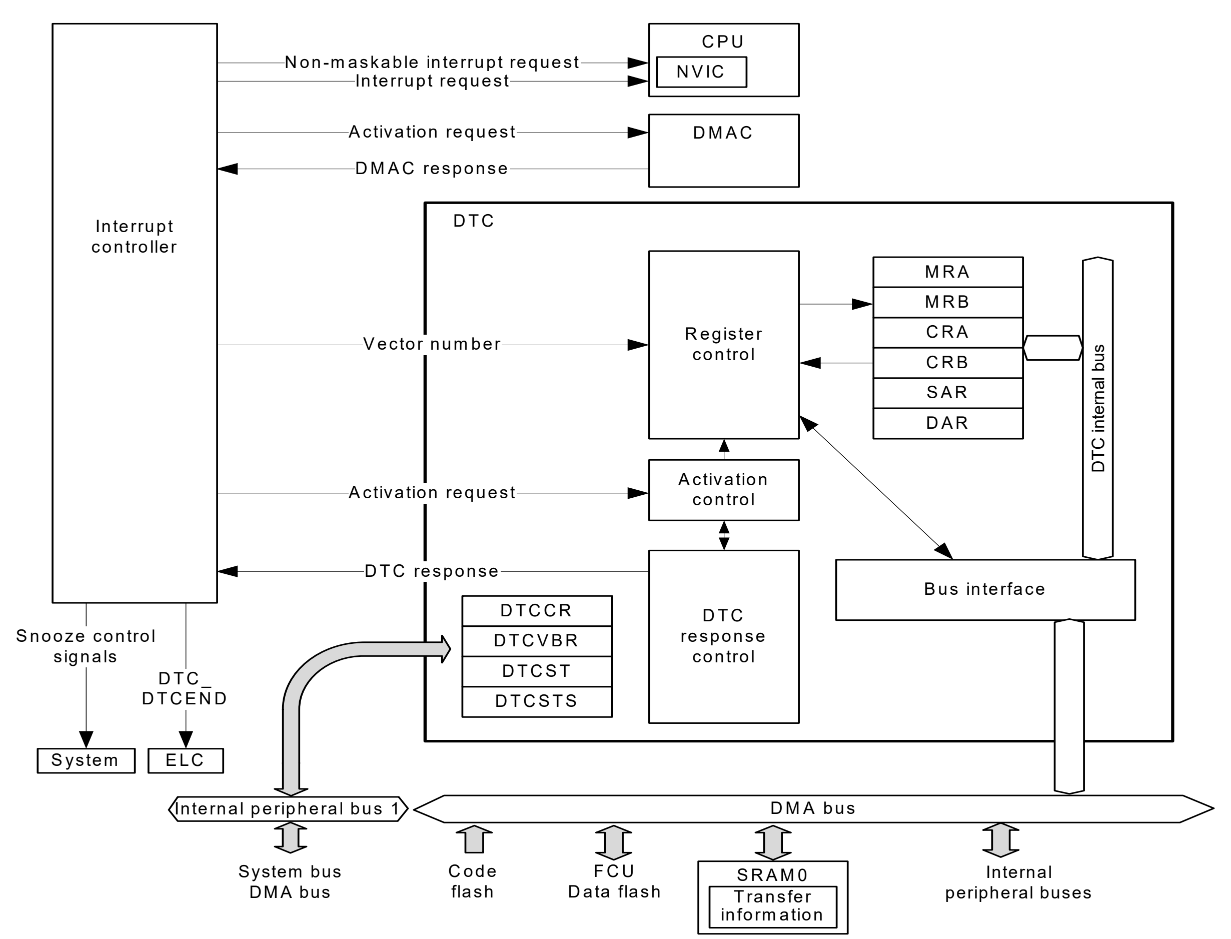
18. Data Transfer Controller (DTC)

18.1 Overview

The MCU includes a Data Transfer Controller (DTC) that performs data transfers when activated by an interrupt request. Table 18.1 lists the DTC specifications and Figure 18.1 shows a 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 start address after the number of data transfers reaches the specified repeat size. The maximum number of repeat transfers is 256 and the maximum data transfer size is 256 × 32 bits (1024 bytes). • Block transfer mode A single activation leads to a transfer of a single block. The maximum block size is 256 × 32 bits = 1024 bytes.
Transfer channel	• Channel transfer can be associated with the interrupt source (transferred by a DTC activation request from the ICU) • Multiple data units can be transferred on a single activation source (chain transfer) • Chain transfers are selectable to either execute when the counter is 0, or always execute.
Transfer space	• 4 GB area from 0000 0000h to FFFF FFFFh, not including reserved areas
Data transfer units	• Single data unit: 1 byte (8 bits), 1 halfword (16 bits), 1 word (32 bits) • Single block size: 1 to 256 data units
CPU interrupt source	• An interrupt request can be generated to the CPU on a DTC activation interrupt • An interrupt request can be generated to the CPU after a single data transfer • An interrupt request can be generated to the CPU after a data transfer of a specified volume.
Event link function	An event link request is generated after one data transfer (for block, after one block transfer)
Read skip	Read of transfer information can be skipped
Write-back skip	When the transfer source or destination address is specified as fixed, a write-back of transfer information can be skipped
Module-stop function	Module-stop state can be set to reduce power consumption


Interrupt controller
CPU
NVIC
Non-maskable interrupt request
Interrupt request
Activation request
DMAC response
DMAC
DTC
Vector number
Register control
MRA
MRB
CRA
CRB
SAR
DAR
DTC internal bus
Activation request
Activation control
DTC response
DTC response control
Bus interface
DTCCR
DTCVBR
DTCST
DTCSTS
Snooze control signals
DTC_DTCEND
System
ELC
Internal peripheral bus 1
DMA bus
System bus DMA bus
Code flash
FCU Data flash
SRAM0
Transfer information
Internal peripheral buses

MRA:	DTC Mode Register A	DTCCR:	DTC Control Register
MRB:	DTC Mode Register B	DTCVBR:	DTC Vector Base Register
CRA:	DTC Transfer Count Register A	DTCST:	DTC Module Start Register
CRB:	DTC Transfer Count Register B	DTCSTS:	DTC Status Register
SAR:	DTC Transfer Source Register		
DAR:	DTC Transfer Destination 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

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, however the CPU can access the SRAM area (transfer information (n) start address + 03h) and the DTC automatically transfers the MRA transfer information to and from the 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.	—

Bit	Symbol	Bit name	Description	R/W
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: Generate an interrupt request to the CPU when specified data transfer is complete 1: Generate an interrupt request to the CPU each time DTC data transfer is performed.	—
b6	CHNS	DTC Chain Transfer Select	0: Select continuous chain transfer 1: Select chain transfer to occur only when the transfer counter is changed from 1 to 0 or 1 to CRAH.	—
b7	CHNE	DTC Chain Transfer Enable	0: Chain transfer disabled 1: Chain transfer enabled.	—

The MRB register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 02h) and the DTC automatically transfers the MRB transfer information to and from 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 a 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 in the CHNS bit. For details, 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 and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 04h) and the DTC automatically transfers the transfer information to and from the SAR 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. Bits [1] and [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 register sets the transfer destination start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 08h) and the DTC automatically transfers the transfer information to and from 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. Bits [1] and [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

	CRA															
	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

- Repeat transfer mode/block transfer mode

	CRAH								CRAL							
	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

Symbol	Register name	Description	R/W
CRAL	Transfer Counter A Lower Register	Set the 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.

The CRA register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0Eh) and the DTC automatically transfers the transfer information to and from 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, 65,535, and 65,536 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 holds the transfer count and the CRAL register functions as an 8-bit transfer counter. The transfer count is 1, 255, and 256 when the set value is 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 register sets the block transfer count for block transfer mode. The transfer count is 1, 65,535, and 65,536 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 register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0Ch) and the DTC automatically transfers the transfer information to and from 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 bit enables skipping of transfer information reads when vector numbers match. The DTC vector number is compared with the vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, the DTC data transfer is performed without reading the transfer information. However, when the previous transfer is a chain transfer, the transfer information is read regardless of the RRS bit.

When the transfer counter (CRA register) becomes 0 during the previous normal transfer and when the transfer counter (CRB register) becomes 0 during the previous block transfer, the transfer information is read regardless of the RRS bit value.

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. The lower 10 bits should be 0.	R/W

The DTCVBR register 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 stopped 1: DTC module started.	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 the DTCST 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 transitioning to one of the following state or mode:

- Module-stop state
- Software Standby mode without Snooze mode transition.

For details on these transitions, see section 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 (ACT flag is 1).	R
b14 to b8	—	Reserved	These bits are read as 0.	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, the VECN[7:0] bits indicate the vector number associated with the activation source for the transfer. The value read from the VECN[7:0] bits is valid if the ACT flag is 1, indicating a DTC transfer is in progress, and invalid if the ACT flag is 0, indicating no current DTC transfer is in progress.

ACT flag (DTC Active Flag)

The ACT flag indicates the state of the DTC transfer operation.

[Setting condition]

- When the DTC is activated by a transfer request.

[Clearing condition]

- When transfer by the DTC, in response to a transfer request, is complete.

18.3 Activation Sources

The DTC is activated by an interrupt request. Setting the ICU.IELSRn.DTCE bit to 1 enables activation of the DTC by the associated interrupt. The selector output *n* number set in ICU.IELSRn is defined as the interrupt vector number, where n = 0 to 31. For an enabled interrupt, the specific DTC interrupt source associated with each interrupt vector number n is selected by ICU.IELSRn.IELS[7:0], where n = 0 to 31, as listed in 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 highest priority request is accepted on completion of the transfer. When multiple activation requests are generated while the DTC Module Start bit (DTCST.DTCST) is 0, the DTC accepts the highest priority request when DTCST.DTCST is subsequently set to 1. The smaller interrupt vector number has higher priority.

The DTC performs the following operations at the start of a single data transfer or for a chain transfer, after the last of the consecutive transfers:

- On completion of a specified round of data transfer, the ICU.IELSRn.DTCE bit is set to 0 and an interrupt request is sent to the CPU
- If the MRB.DISEL bit is 1, an interrupt request is sent to the CPU on completion of a data transfer
- For other transfers, the ICU.IELSRn.IR 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 relationship between the DTC vector table and transfer information. Figure 18.3 shows the allocation of transfer information in the SRAM area.

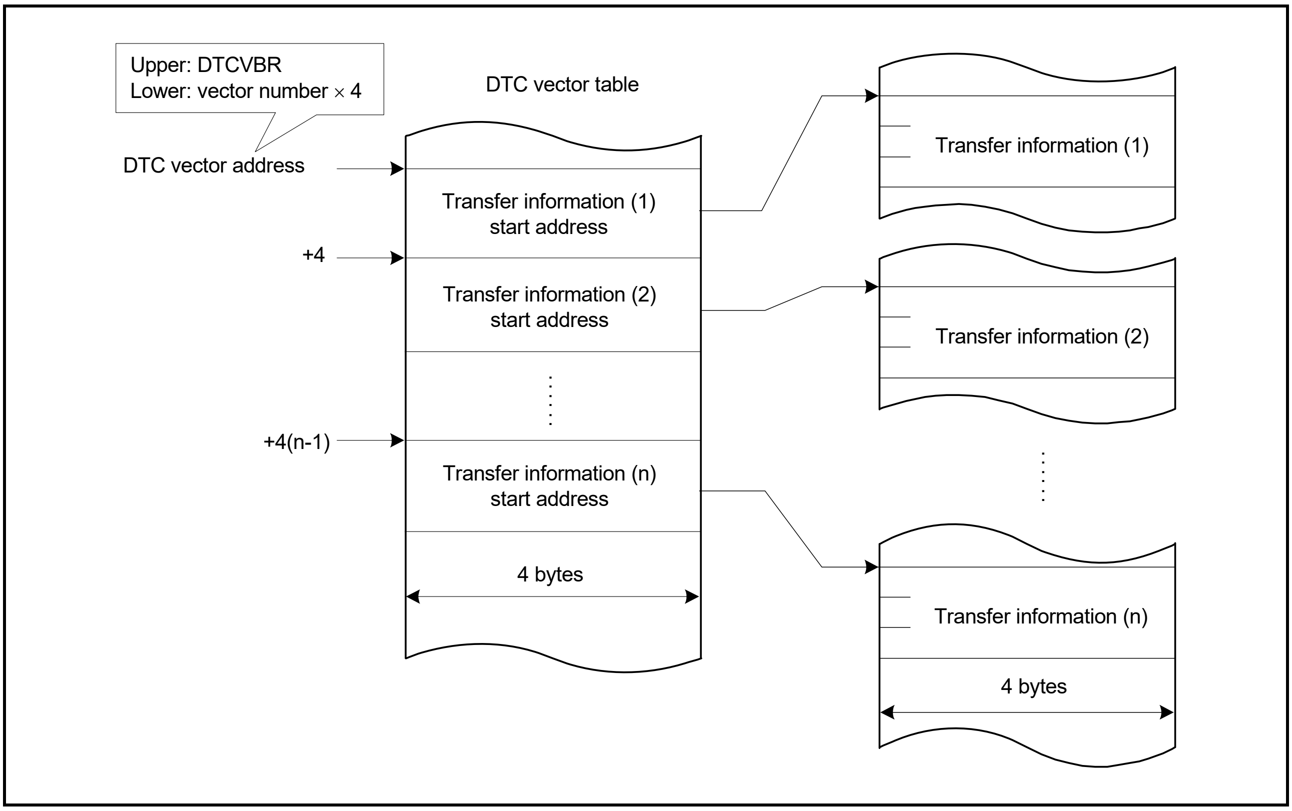

Upper: DTCVBR
Lower: vector number × 4
DTC vector address
DTC vector table
+4
+4(n-1)
Transfer information (1) start address
Transfer information (2) start address
Transfer information (n) start address
4 bytes
Transfer information (1)
Transfer information (2)
Transfer information (n)
4 bytes

Figure 18.2 DTC vector table and transfer information

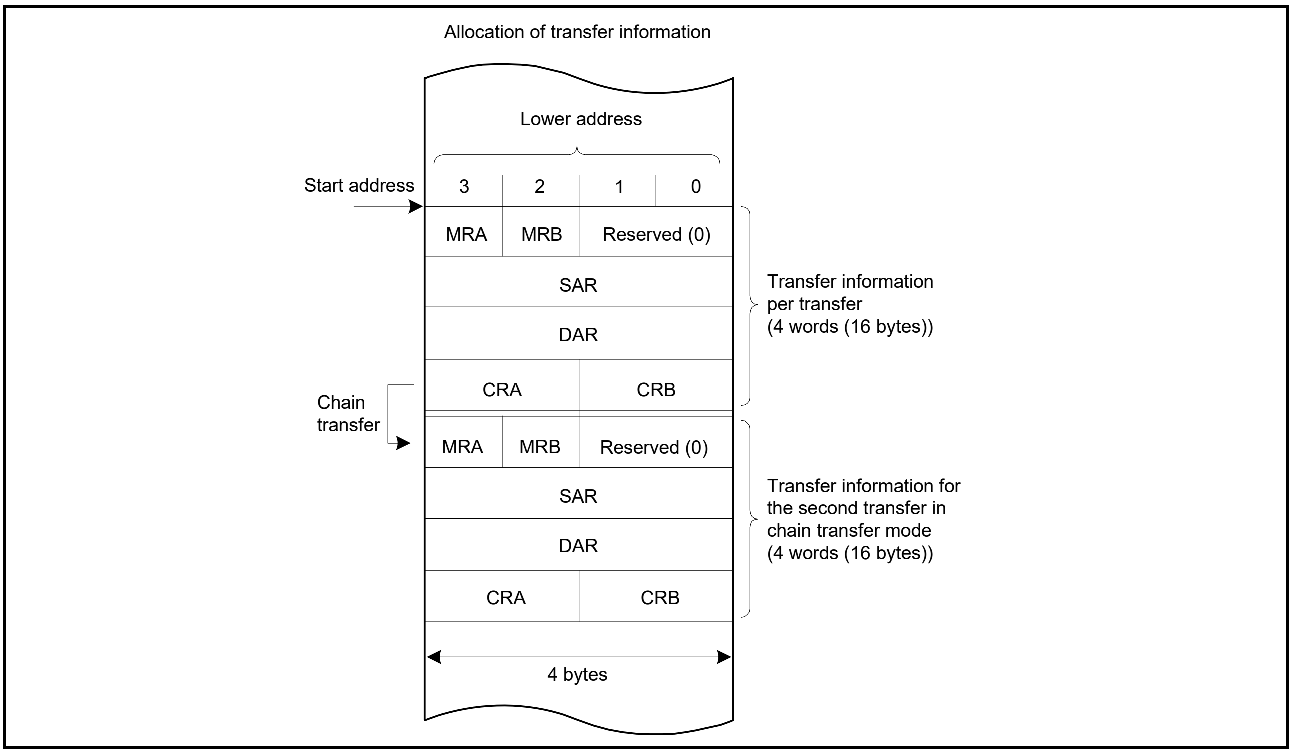

Allocation of transfer information
Lower address
Start address
3
2
1
0
MRA
MRB
Reserved (0)
SAR
DAR
CRA
CRB
Transfer information per transfer (4 words (16 bytes))
Chain transfer
MRA
MRB
Reserved (0)
SAR
DAR
CRA
CRB
Transfer information for the second transfer in chain transfer mode (4 words (16 bytes))
4 bytes

Figure 18.3 Allocation of transfer information in the SRAM area

18.4 Operation

The DTC transfers data according to the transfer information. Storage of the transfer information in the SRAM area is required before a DTC operation. When the DTC is activated, it reads the DTC vector associated with the vector number. The DTC then reads the transfer information from the transfer information store address referenced by the DTC vector and transfers the data. After the data transfer, the DTC writes back the transfer information. Storing the transfer information in the SRAM area allows data transfer of any number of channels.

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 independently after the data transfer.

Table 18.2 describes the DTC transfer modes.

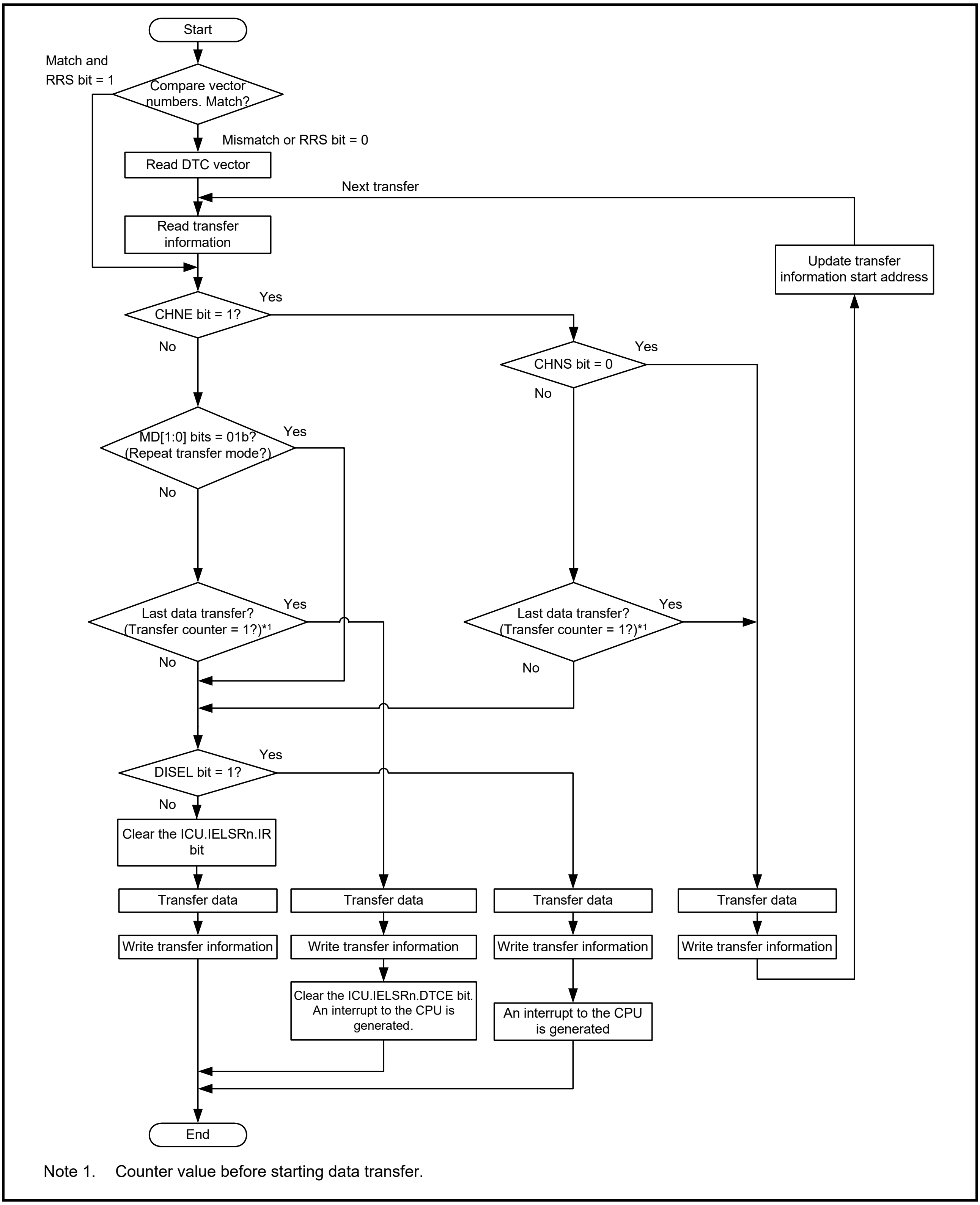
Table 18.2 DTC transfer modes

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), or 1 word (32 bit)	Incremented or decremented by 1, 2, or 4 or address fixed	1 to 65,536
Repeat transfer mode*1	1 byte (8 bit), 1 halfword (16 bit), or 1 word (32 bit)	Incremented or decremented by 1, 2, or 4 or address fixed	1 to 256*3
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), or 1 to 256 words (4 to 1024 bytes))	Incremented or decremented by 1, 2, or 4 or address fixed	1 to 65,536

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 flow 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.


Start
Match and RRS bit = 1
Compare vector numbers. Match?
Mismatch or RRS bit = 0
Read DTC vector
Next transfer
Read transfer information
Update transfer information start address
CHNE bit = 1?
Yes
No
CHNS bit = 0
Yes
No
MD[1:0] bits = 01b? (Repeat transfer mode?)
Yes
No
Last data transfer? (Transfer counter = 1?)*1
Yes
No
Last data transfer? (Transfer counter = 1?)*1
Yes
No
DISEL bit = 1?
Yes
No
Clear the ICU.IELSRn.IR bit
Transfer data
Transfer data
Transfer data
Transfer data
Write transfer information
Write transfer information
Write transfer information
Write transfer information
Clear the ICU.IELSRn.DTCE bit. An interrupt to the CPU is generated.
An interrupt to the CPU is generated
End

Note 1. Counter value before starting data transfer.

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 counter used depends on the transfer modes as follows:
Normal transfer mode — CRA register
Repeat transfer mode — CRAL register
Block transfer mode — CRB register

Note 2. On completion of a data transfer, the counters operate as follows:
1 → 0 in normal and block transfer modes
1 → CRAH in repeat transfer mode
(1 → *) in the table indicates both of these two operations, depending on the mode.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The conditions for the combination of the second transfer and CHNE bit = 1 is omitted.

18.4.1 Transfer Information Read Skip Function

Reading of vector addresses and transfer information can be skipped by setting the DTCCR.RRS bit. When a DTC activation request is generated, the current DTC vector number is compared 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 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, and 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 the 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

The normal transfer mode allows a 1-byte (8 bit), 1-halfword (16 bit), 1-word (32 bit) data transfer on a single activation source. The transfer count can be set from 1 to 65536. Transfer source and transfer destination addresses can be independently 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 of 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, or fixed*1
DAR	Transfer destination address	Increment, decrement, or 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.

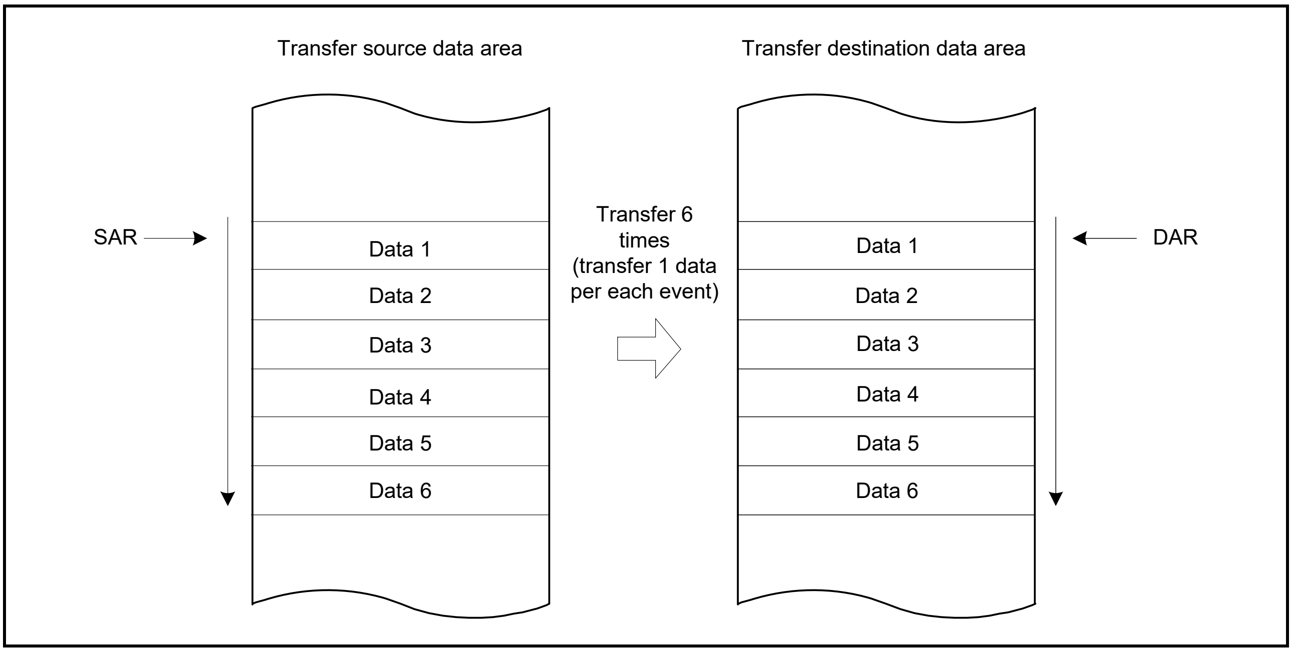

Transfer source data area
Transfer destination data area
SAR
DAR
Transfer 6 times (transfer 1 data per each event)
Data 1
Data 2
Data 3
Data 4
Data 5
Data 6
Data 1
Data 2
Data 3
Data 4
Data 5
Data 6

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

The repeat transfer mode allows a 1-byte (8-bit), 1-halfword (16-bit), or 1-word (32-bit) data transfer on a single activation source. Transfer source or transfer destination for the repeat area must be specified in the MRB.DTS bit. The transfer count can be set from 1 to 256. When the specified-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 of 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, or fixed*1	• When the MRB.DTS bit is 0 Increment, decrement, or fixed*1 • When the MRB.DTS bit is 1 SAR register initial value.
DAR	Transfer destination address	Increment, decrement, or fixed*1	• When the MRB.DTS bit is 0 DAR register initial value • When the MRB.DTS bit is 1 Increment, decrement, or fixed*1.
CRAH	Retains transfer counter	CRAH	CRAH
CRAL	Transfer counter A	CRAL - 1	CRAH
CRB	Transfer counter B	Not updated	Not updated

Note 1. Write-back is skipped in address-fixed mode.

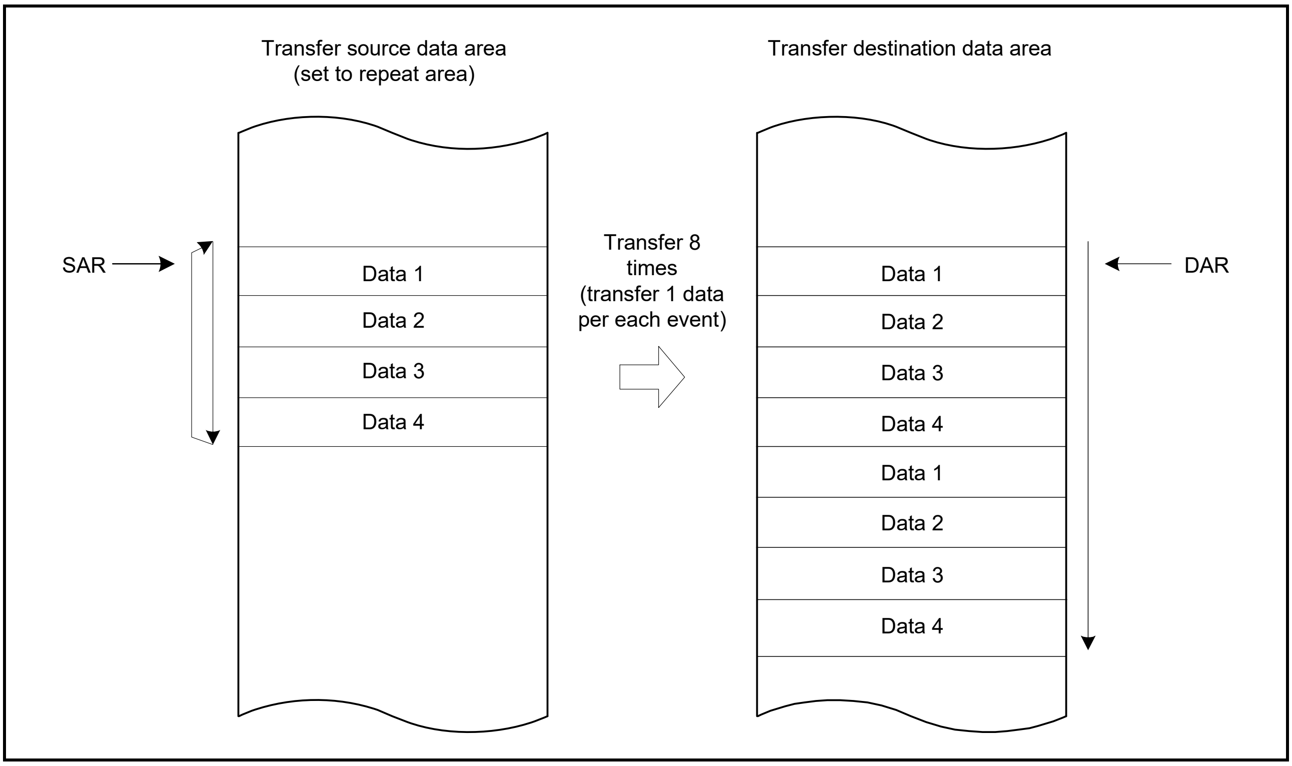

Transfer source data area
(set to repeat area)
Transfer destination data area
SAR
Data 1
Data 2
Data 3
Data 4
Transfer 8 times (transfer 1 data per each event)
Data 1
Data 2
Data 3
Data 4
Data 1
Data 2
Data 3
Data 4
DAR

Figure 18.6 Memory map of repeat transfer mode when transfer source is a repeat area (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRAH = 04h)

18.4.5 Block Transfer Mode

The block transfer mode allows single-block data transfer on a single activation source. Transfer source or transfer destination for the block area must be specified in the MRB.DTS bit. The block size can be set from 1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), or 1 to 256 words (4 to 1024 bytes). When transfer of the specified block completes, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS bit is 1 or the DAR register when the DTS bit is 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 65,536. 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, or fixed*1 • When MRB.DTS bit is 1 SAR register initial value.
DAR	Transfer destination address	• When MRB.DTS bit is 0 DAR register initial value • When MRB.DTS bit is 1 Increment, decrement, or fixed*1.
CRAH	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.

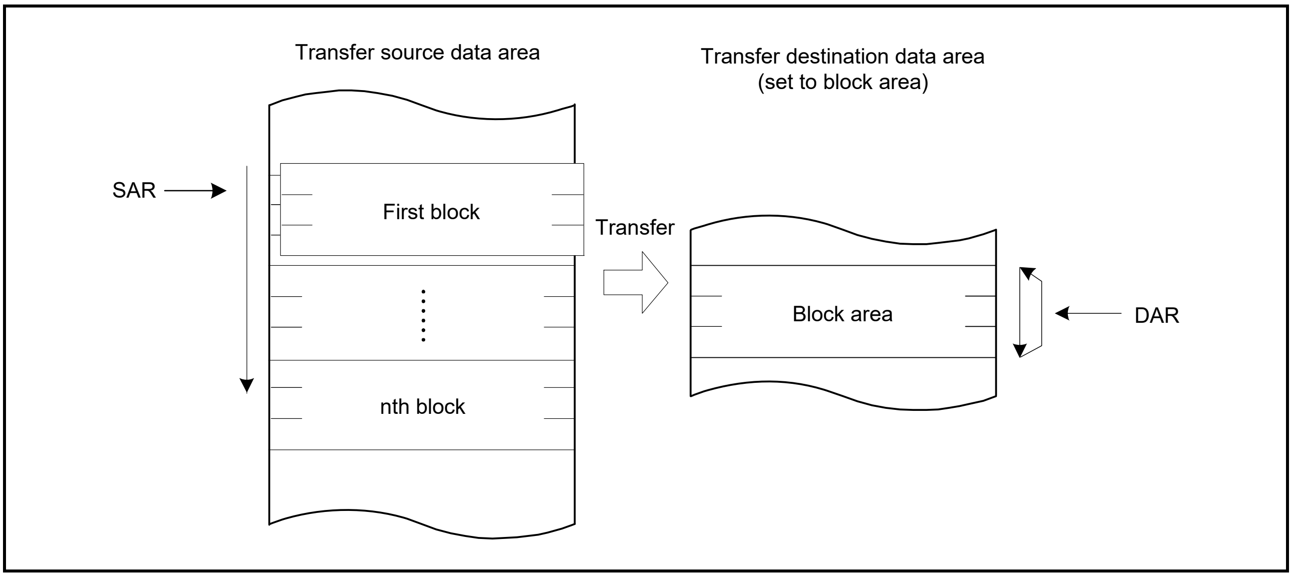

Transfer source data area
Transfer destination data area
(set to block area)
SAR
First block
Transfer
Block area
DAR
nth block

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.

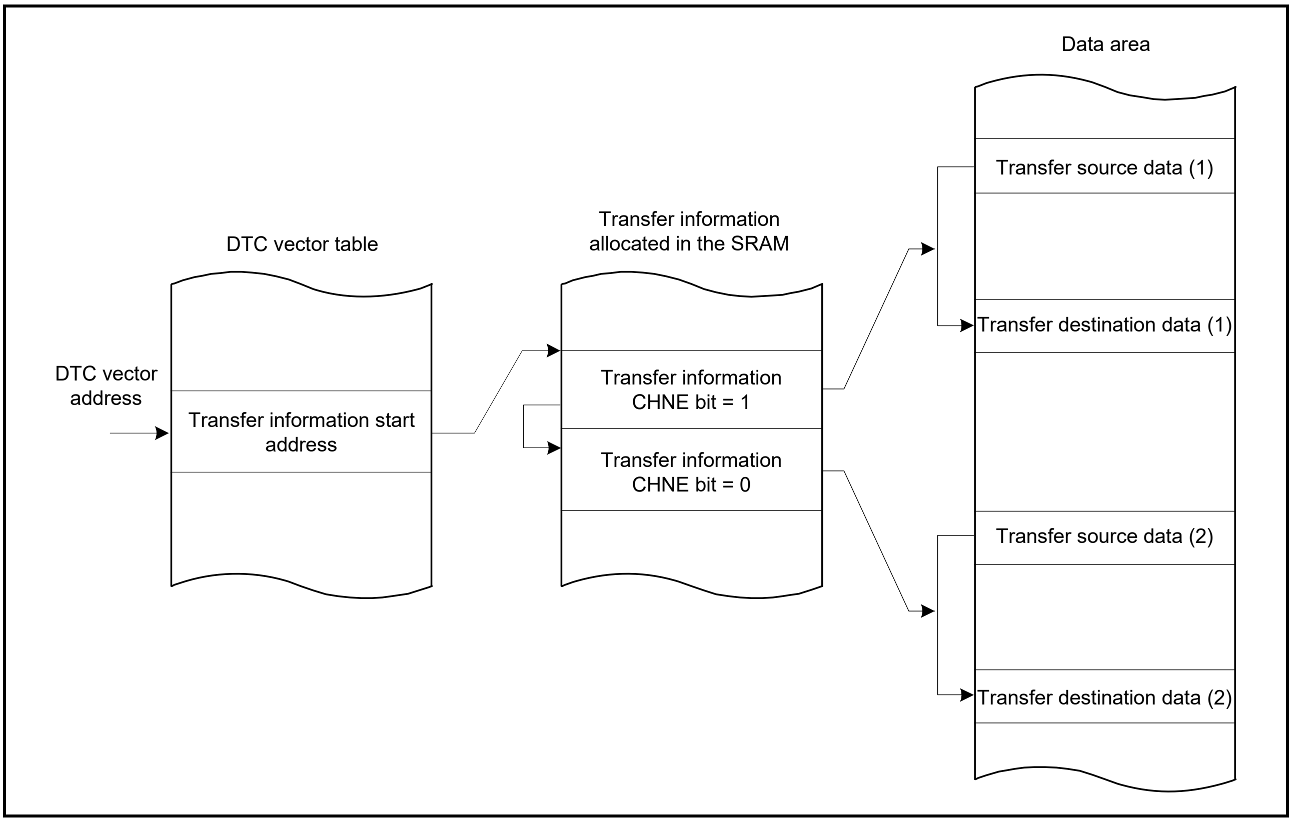

Data area
Transfer source data (1)
Transfer information
allocated in the SRAM
DTC vector table
Transfer destination data (1)
DTC vector
address
Transfer information
CHNE bit = 1
Transfer information start
address
Transfer information
CHNE bit = 0
Transfer source data (2)
Transfer destination data (2)

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.

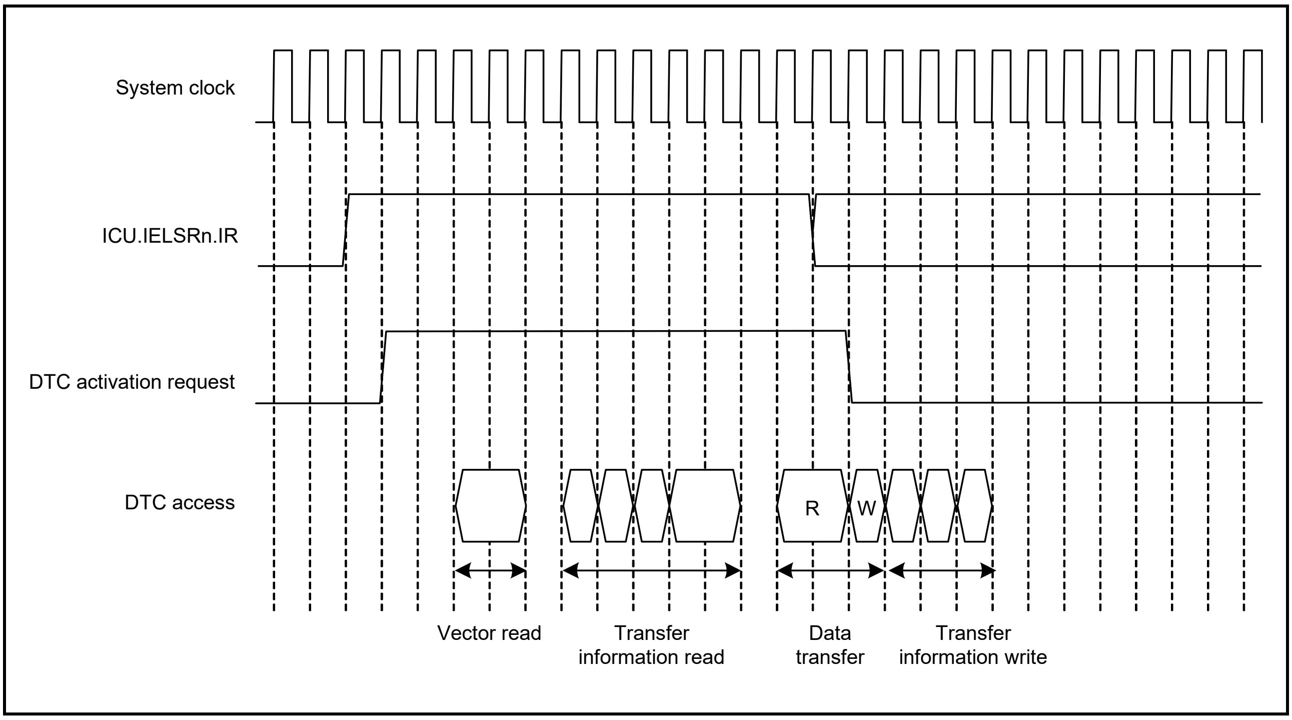

System clock
ICU.IELSRn.IR
DTC activation request
DTC access
R
W
Vector read
Transfer information read
Data transfer
Transfer information write

Figure 18.9 Example 1 of DTC operation timing in normal transfer mode and repeat transfer mode

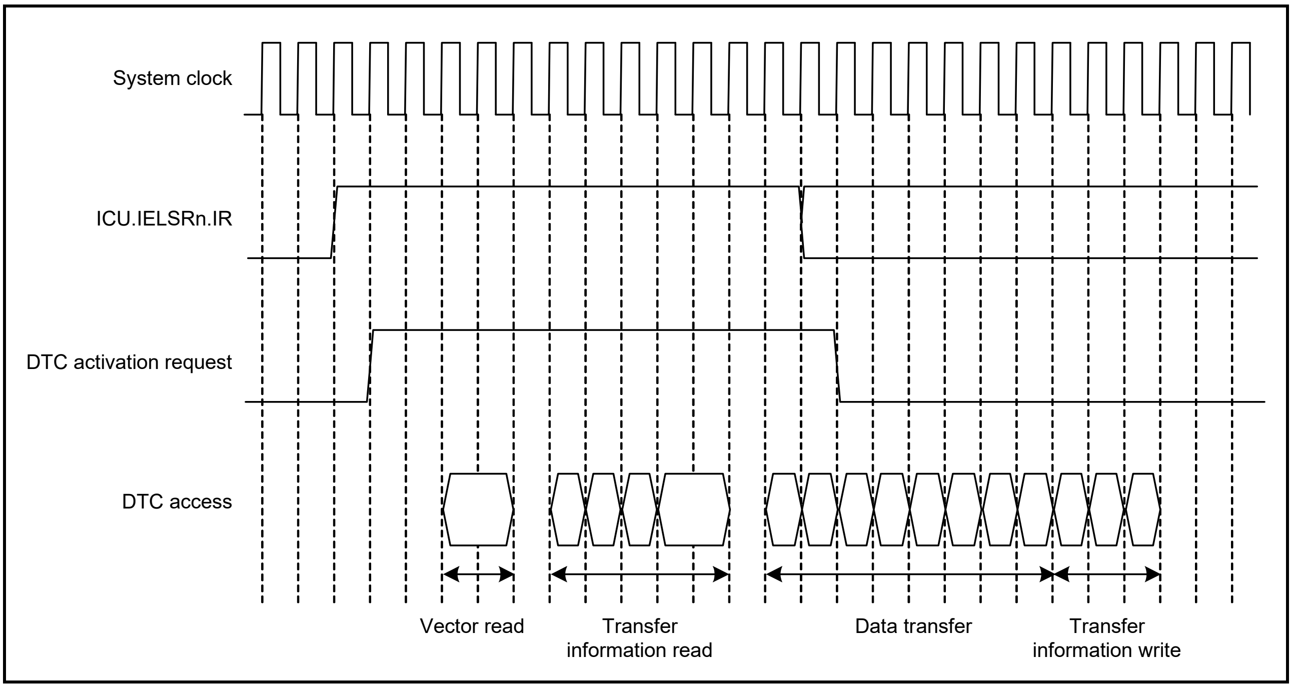

System clock
ICU.IELSRn.IR
DTC activation request
DTC access
Vector read
Transfer information read
Data transfer
Transfer information write

Figure 18.10 Example 2 of DTC operation timing in block transfer mode when block size = 4

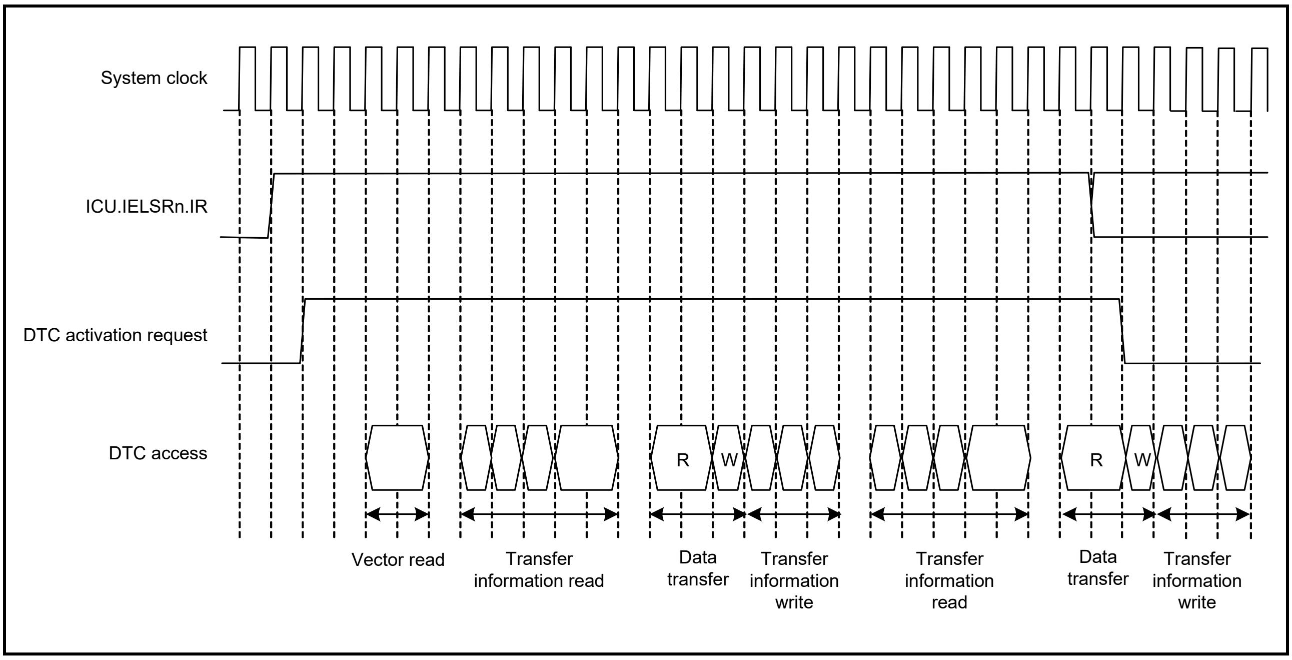
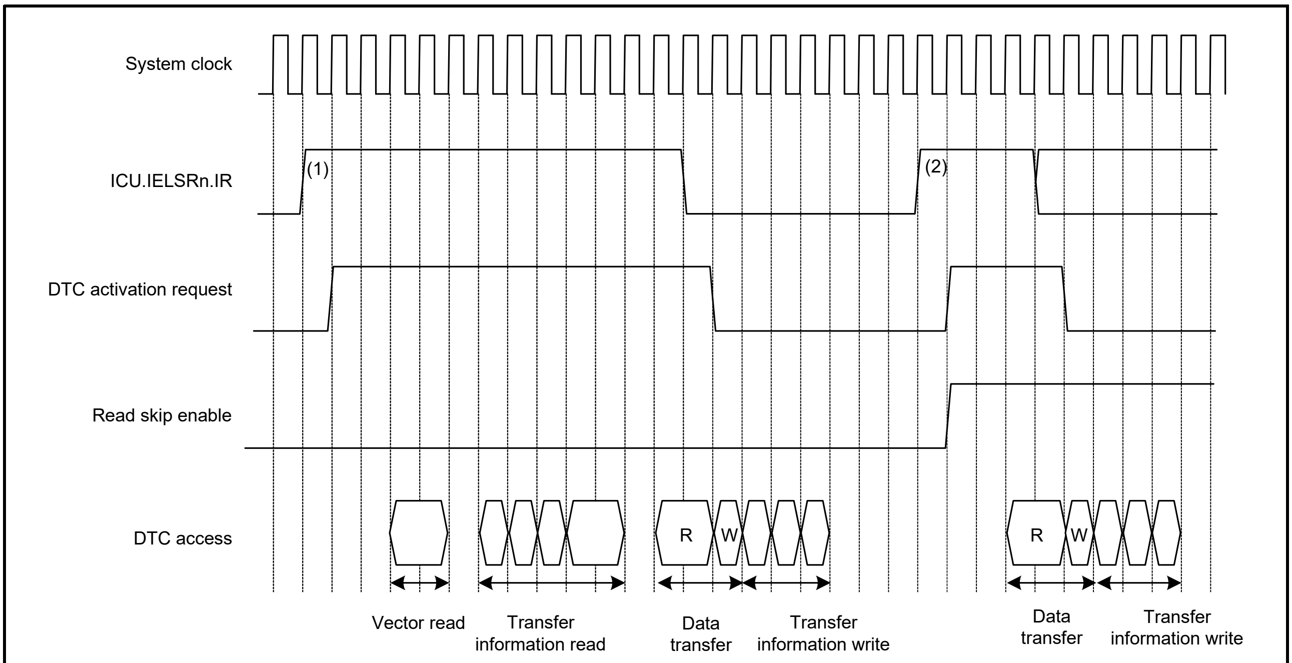

System clock
ICU.IELSRn.IR
DTC activation request
DTC access
R
W
R
W
Vector read
Transfer information read
Data transfer
Transfer information write
Transfer information read
Data transfer
Transfer information write

Figure 18.11 Example 3 of DTC operation timing for chain transfer


System clock
ICU.IELSRn.IR
(1)
(2)
DTC activation request
Read skip enable
DTC access
R
W
R
W
Vector read
Transfer information read
Data transfer
Transfer information write
Data transfer
Transfer information write

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 a transfer information read is skipped, with the vector, transfer information, transfer destination data on the SRAM, and the 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.

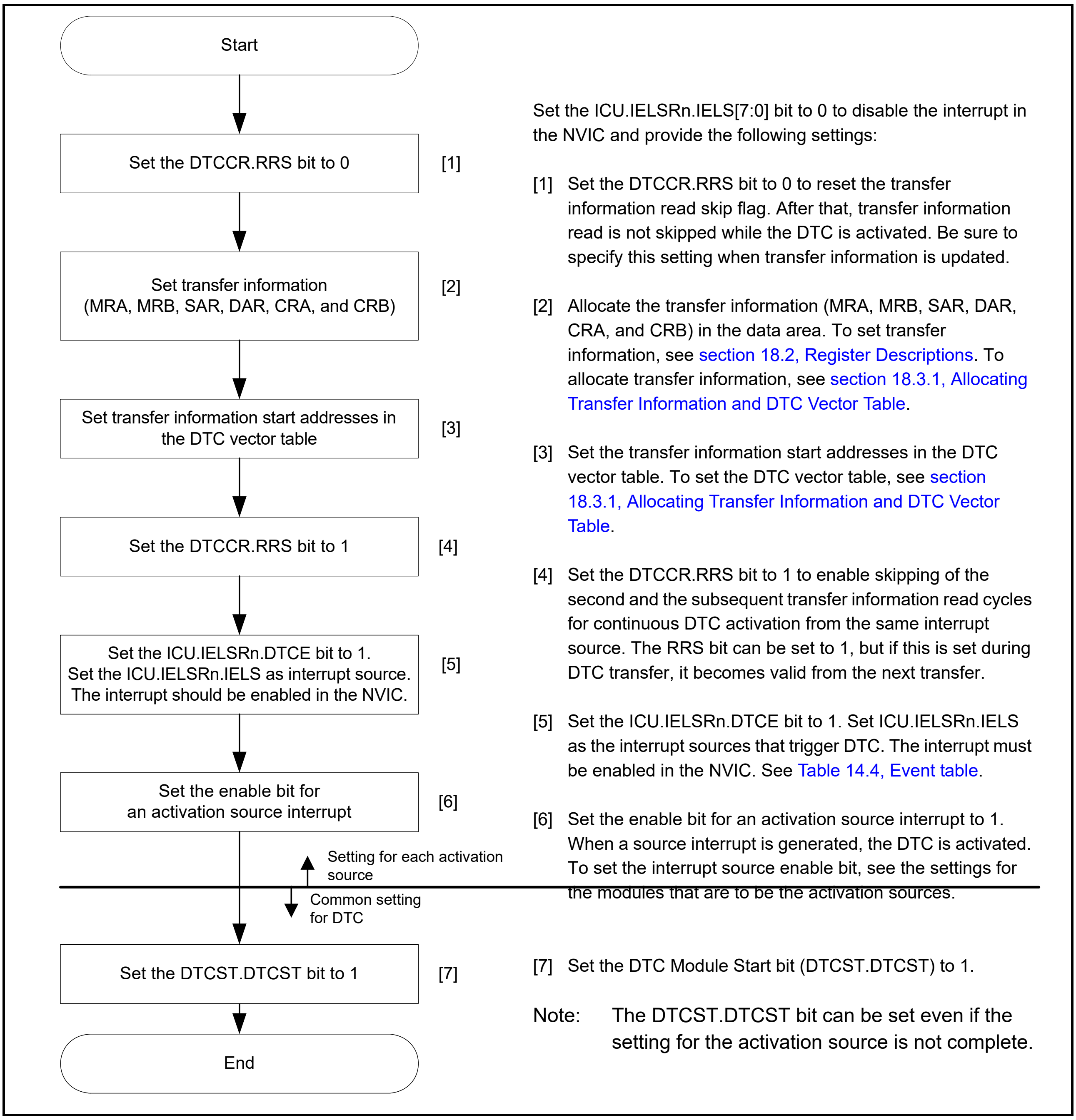
P: Block size (initial settings of CRAH and CRAL)
Cv: Cycles for access to vector transfer information storage destination
Ci: Cycles for access to transfer information storage destination address
Cr: Cycles for access to data read destination
Cw: Cycles for access to data write destination
The unit 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 42, SRAM, section 43, 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 for setting the DTC.


Start
Set the DTCCR.RRS bit to 0
[1]
Set transfer information
(MRA, MRB, SAR, DAR, CRA, and CRB)
[2]
Set transfer information start addresses in the DTC vector table
[3]
Set the DTCCR.RRS bit to 1
[4]
Set the ICU.IELSRn.DTCE bit to 1.
Set the ICU.IELSRn.IELS as interrupt source.
The interrupt should be enabled in the NVIC.
[5]
Set the enable bit for an activation source interrupt
[6]
Setting for each activation source
Common setting for DTC
Set the DTCST.DTCST bit to 1
[7]
End

Set the ICU.IELSRn.IELS[7:0] bit to 0 to disable the interrupt in the NVIC and provide the following settings:

[1] Set the DTCCR.RRS bit to 0 to reset the transfer information read skip flag. After that, transfer information read is not skipped while the DTC is activated. Be sure to specify this setting when transfer information is updated.

[2] Allocate the transfer information (MRA, MRB, SAR, DAR, CRA, and CRB) in the data area. To set transfer information, see section 18.2, Register Descriptions. To allocate transfer information, see section 18.3.1, Allocating Transfer Information and DTC Vector Table.

[3] Set the transfer information start addresses in the DTC vector table. To set the DTC vector table, see section 18.3.1, Allocating Transfer Information and DTC Vector Table.

[4] Set the DTCCR.RRS bit to 1 to enable skipping of the second and the subsequent transfer information read cycles for continuous DTC activation from the same interrupt source. The RRS bit can be set to 1, but if this is set during DTC transfer, it becomes valid from the next transfer.

[5] Set the ICU.IELSRn.DTCE bit to 1. Set ICU.IELSRn.IELS as the interrupt sources that trigger DTC. The interrupt must be enabled in the NVIC. See Table 14.4, Event table.

[6] Set the enable bit for an activation source interrupt to 1. When a source interrupt is generated, the DTC is activated. To set the interrupt source enable bit, see the settings for the modules that are to be the activation sources.

[7] Set the DTC Module Start bit (DTCST.DTCST) to 1.

Note: The DTCST.DTCST bit can be set even if the setting for the activation source is not complete.

Figure 18.13 DTC setting procedure

18.6 Examples of DTC Usage

18.6.1 Normal Transfer

This section provides an example of DTC usage and its application when receiving 128 bytes of data from an SCI.

(1) Transfer information settings

In the MRA register, select a fixed source address (MRA.SM[1:0] 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 settings

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

(3) ICU settings and DTC module activation

Set the ICU.IELSRn.DTCE bit to 1 and set ICU.IELSRn.IELS as the SCI interrupt. The interrupt must be enabled in the NVIC. Set the DTCST.DTCST bit to 1.

(4) SCI settings

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

Each 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 use in the output of pulses by the General PWM Timer (GPT). You can use chain transfers to transfer PWM timer compare data and change the period of the PWM timer for the GPT.

For the first of the chain transfers, normal transfer mode is specified for transfer to the GPTm.GTCCRC register (m = 320 to 323, 164 to 169). For the second 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 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 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 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 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[7:0] to 97 (61h) for the GPT320 counter overflow.
3. Set the DTCST.DTCST bit to 1.

(7) GPT settings

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

Each 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 transfer 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 (65536) 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 a 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 destination 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 is 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 destination 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 is 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.

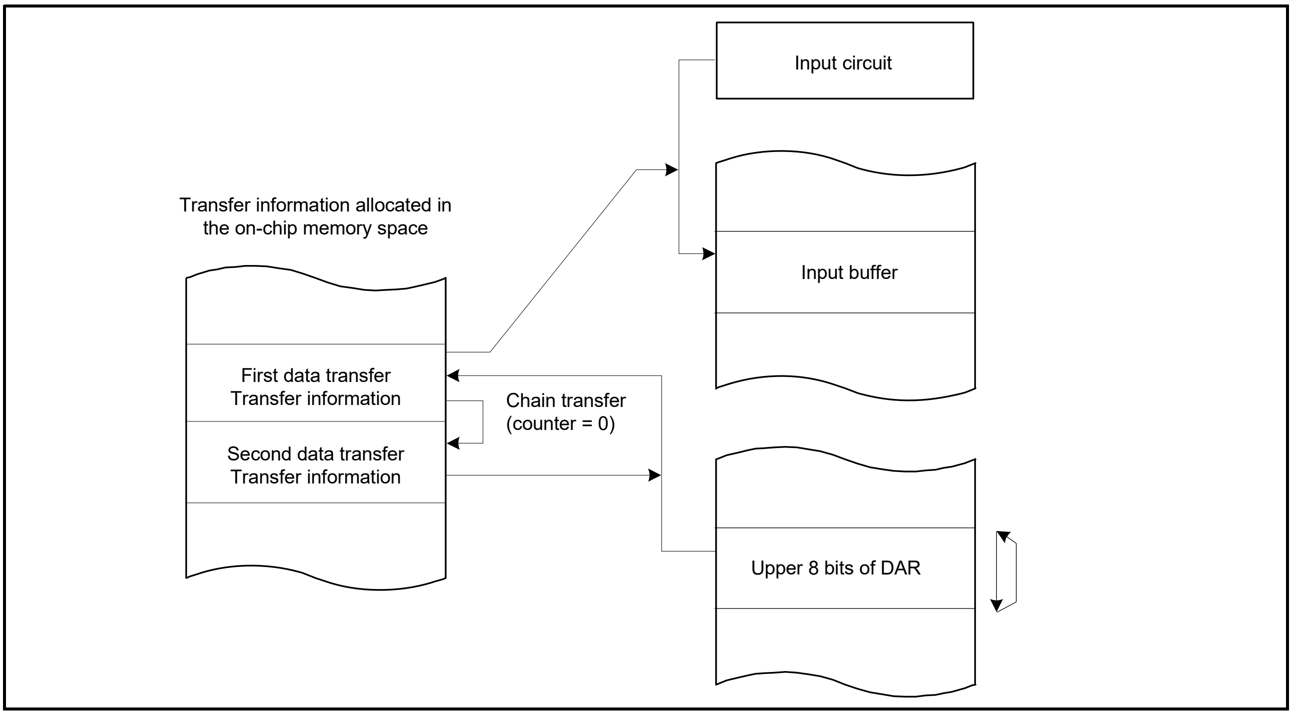


Figure 18.14 Chain transfer when counter = 0

18.7 Interrupt Source

When the DTC completes data transfer of the specified count or when data transfer with MRB.DISEL set to 1 is complete, a DTC activation source generates an interrupt to the CPU. Interrupts to the CPU are controlled according to the settings in the NVIC and ICU.IELSRn.IELS[7:0]. 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 can produce an event link request on completion of one transfer request.

18.9 Snooze Control Interface

To return to Software Standby mode from Snooze mode through the DTC, set 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, 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 are 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 operations described in the following sections. 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 when 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 transitioning 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 the 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 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, and 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)

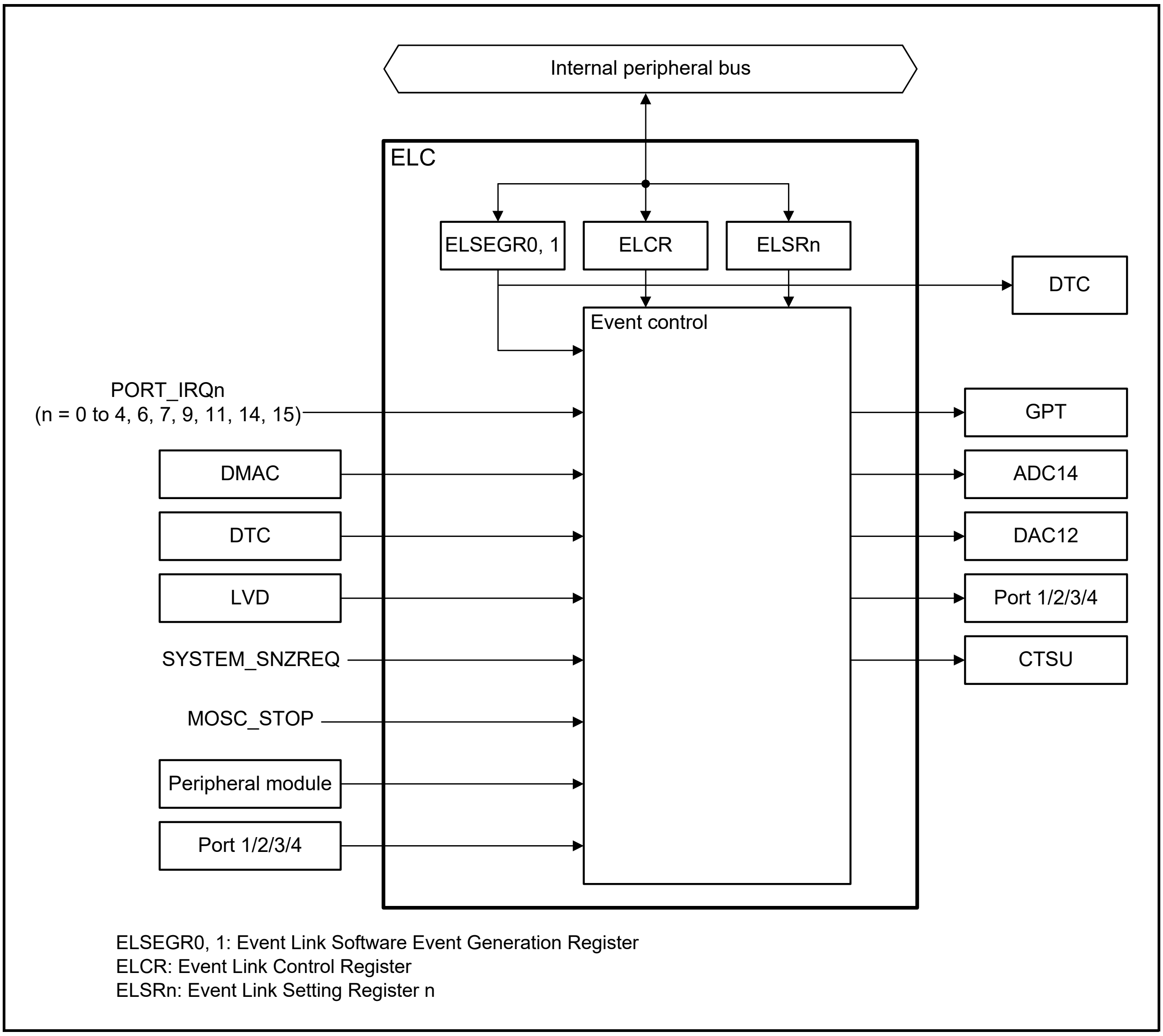
19.1 Overview

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 a block diagram.

Table 19.1 ELC specifications

Parameter	Description
Event link function	135 types of event signals can be directly connected to modules.The ELC can generate an ELC event signal, and events that activate the DTC.
Module-stop function	Module-stop state can be set


Internal peripheral bus
ELC
ELSEGR0, 1
ELCR
ELSRn
DTC
Event control
PORT_IRQn
(n = 0 to 4, 6, 7, 9, 11, 14, 15)
DMAC
DTC
LVD
SYSTEM_SNZREQ
MOSC_STOP
Peripheral module
Port 1/2/3/4
GPT
ADC14
DAC12
Port 1/2/3/4
CTSU

ELSEGR0, 1: Event Link Software Event Generation Register
ELCR: Event Link Control Register
ELSRn: Event Link Setting Register n

Figure 19.1 ELC block diagram (n = 0 to 9, 12, 14 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 the SEG bit while the WE bit is 1, a software event is generated. This bit is read as 0. Even when 1 is written to this bit, data is not stored. The WE bit must be set to 1 before writing to this bit.

A software event can trigger a linked DTC event.

WE bit (SEG Bit Write Enable)

The SEG bit can only be written to when the WE bit is 1. Clear the WI bit to 0 before writing to this bit.

[Setting condition]

- If 1 is written to this bit while the WI bit is 0, this bit becomes 1.

[Clearing condition]

- If 0 is written to this bit while the WI bit is 0, this bit becomes 0.

WI bit (ELSEGR Register Write Disable)

The ELSEGR register can only be written to when the write value to the WI bit is 0. This bit is read as 1. Before setting the WE or SEG bit, the WI bit must be set to 0.

19.2.3 Event Link Setting Register n (ELSRn) (n = 0 to 9, 12, 14 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.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[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	ELS[7:0]	Event Link Select	b7 b0 00000000:Event output disabled for the associated peripheral module 00000001 to 11010100: Number setting for the event signal to be linked. Other settings are prohibited.	R/W
b15 to b8	—	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 each peripheral module. Table 19.2 shows the association between the ELSRn registers and the peripheral modules. Table 19.3 shows the association between the event signal names set in the ELSRn registers and the signal numbers.

Table 19.2 Association between the ELSRn registers and peripheral functions

Register name	Peripheral function (module)	Event name
ELSR0	GPT (A)	ELC_GPTA
ELSR1	GPT (B)	ELC_GPTB
ELSR2	GPT (C)	ELC_GPTC
ELSR3	GPT (D)	ELC_GPTD
ELSR4	GPT (E)	ELC_GPTE
ELSR5	GPT (F)	ELC_GPTF
ELSR6	GPT (G)	ELC_GPTG
ELSR7	GPT (H)	ELC_GPTH
ELSR8	ADC14A	ELC_AD00
ELSR9	ADC14B	ELC_AD01
ELSR12	DAC12	ELC_DA0
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 4)

Event number	Interrupt request source	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
007h		PORT_IRQ6*1	External pin interrupt 6
008h		PORT_IRQ7*1	External pin interrupt 7
00Ah		PORT_IRQ9*1	External pin interrupt 9
00Ch		PORT_IRQ11*1	External pin interrupt 11
00Fh		PORT_IRQ14*1	External pin interrupt 14
010h		PORT_IRQ15*1	External pin interrupt 15
011h	DMAC0	DMAC0_INT	DMAC transfer end 0
012h	DMAC1	DMAC1_INT	DMAC transfer end 1
013h	DMAC2	DMAC2_INT	DMAC transfer end 2
014h	DMAC3	DMAC3_INT	DMAC transfer end 3
016h	DTC	DTC_DTCEND*3	DTC transfer end
019h	LVD	LVD_LVD1	Voltage monitor 1 interrupt
01Ch	MOSC	MOSC_STOP	Main clock oscillation stop
01Dh	Low power mode	SYSTEM_SNZREQ*2, *3	Snooze entry
01Eh	AGT0	AGT0_AGTI	AGT interrupt
01Fh		AGT0_AGTCMAI	Compare match A
020h		AGT0_AGTCMBI	Compare match B
021h	AGT1	AGT1_AGTI	AGT interrupt
022h		AGT1_AGTCMAI	Compare match A
023h		AGT1_AGTCMBI	Compare match B
024h	IWDT	IWDT_NMIUNDF	IWDT underflow
025h	WDT	WDT_NMIUNDF	WDT underflow
027h	RTC	RTC_PRD	Periodic interrupt
029h	ADC140	ADC140_ADI	A/D scan end interrupt
02Dh		ADC140_WCMPM*3	Compare match
02Eh		ADC140_WCMPUM*3	Compare mismatch
02Fh	ACMPLP	ACMP_LP0	Low-power analog comparator interrupt 0
030h		ACMP_LP1	Low-power analog comparator interrupt 1
035h	IIC0	IIC0_RXI	Receive data full
036h		IIC0_TXI	Transmit data empty
037h		IIC0_TEI	Transmit end
038h		IIC0_EEI	Transfer error
03Ah	IIC1	IIC1_RXI	Receive data full
03Bh		IIC1_TXI	Transmit data empty
03Ch		IIC1_TEI	Transmit end
03Dh		IIC1_EEI	Transfer error
04Ah	DOC	DOC_DOPCI*3	Data operation circuit interrupt

Table 19.3 Association between event signal names set in ELSRn.ELS bits and signal numbers (2 of 4)

Event number	Interrupt request source	Name	Description
053h	I/O port	IOPORT_GROUP1	Port 1 event
054h		IOPORT_GROUP2	Port 2 event
055h		IOPORT_GROUP3	Port 3 event
056h		IOPORT_GROUP4	Port 4 event
057h	ELC	ELC_SWEVT0	Software event 0
058h		ELC_SWEVT1	Software event 1
05Bh	GPT320	GPT0_CCMPA	Compare match A
05Ch		GPT0_CCMPB	Compare match B
05Dh		GPT0_CMPC	Compare match C
05Eh		GPT0_CMPD	Compare match D
05Fh		GPT0_CMPE	Compare match E
060h		GPT0_CMPF	Compare match F
061h		GPT0_OVF	Overflow
062h		GPT0_UDF	Underflow
063h	GPT321	GPT1_CCMPA	Compare match A
064h		GPT1_CCMPB	Compare match B
065h		GPT1_CMPC	Compare match C
066h		GPT1_CMPD	Compare match D
067h		GPT1_CMPE	Compare match E
068h		GPT1_CMPF	Compare match F
069h		GPT1_OVF	Overflow
06Ah		GPT1_UDF	Underflow
06Bh	GPT322	GPT2_CCMPA	Compare match A
06Ch		GPT2_CCMPB	Compare match B
06Dh		GPT2_CMPC	Compare match C
06Eh		GPT2_CMPD	Compare match D
06Fh		GPT2_CMPE	Compare match E
070h		GPT2_CMPF	Compare match F
071h		GPT2_OVF	Overflow
072h		GPT2_UDF	Underflow
073h	GPT323	GPT3_CCMPA	Compare match A
074h		GPT3_CCMPB	Compare match B
075h		GPT3_CMPC	Compare match C
076h		GPT3_CMPD	Compare match D
077h		GPT3_CMPE	Compare match E
078h		GPT3_CMPF	Compare match F
079h		GPT3_OVF	Overflow
07Ah		GPT3_UDF	Underflow

Table 19.3 Association between event signal names set in ELSRn.ELS bits and signal numbers (3 of 4)

Event number	Interrupt request source	Name	Description
07Bh	GPT164	GPT4_CCMPA	Compare match A
07Ch		GPT4_CCMPB	Compare match B
07Dh		GPT4_CMPC	Compare match C
07Eh		GPT4_CMPD	Compare match D
07Fh		GPT4_CMPE	Compare match E
080h		GPT4_CMPF	Compare match F
081h		GPT4_OVF	Overflow
082h		GPT4_UDF	Underflow
083h	GPT165	GPT5_CCMPA	Compare match A
084h		GPT5_CCMPB	Compare match B
085h		GPT5_CMPC	Compare match C
086h		GPT5_CMPD	Compare match D
087h		GPT5_CMPE	Compare match E
088h		GPT5_CMPF	Compare match F
089h		GPT5_OVF	Overflow
08Ah		GPT5_UDF	Underflow
09Bh	GPT168	GPT8_CCMPA	Compare match A
09Ch		GPT8_CCMPB	Compare match B
09Dh		GPT8_CMPC	Compare match C
09Eh		GPT8_CMPD	Compare match D
09Fh		GPT8_CMPE	Compare match E
0A0h		GPT8_CMPF	Compare match F
0A1h		GPT8_OVF	Overflow
0A2h		GPT8_UDF	Underflow
0ABh	GPT	GPT_UVWEDGE	UVW edge event
0ACh	SCI0	SCI0_RXI*4	Receive data full
0ADh		SCI0_TXI*4	Transmit data empty
0AEh		SCI0_TEI	Transmit end
0AFh		SCI0_ERI*4	Receive error
0B0h		SCI0_AM	Address match event
0B2h	SCI1	SCI1_RXI*4	Receive data full
0B3h		SCI1_TXI*4	Transmit data empty
0B4h		SCI1_TEI	Transmit end
0B5h		SCI1_ERI*4	Receive error
0B6h		SCI1_AM	Address match event
0C1h	SCI4	SCI4_RXI*4	Receive data full
0C2h		SCI4_TXI*4	Transmit data empty
0C3h		SCI4_TEI	Transmit end
0C4h		SCI4_ERI*4	Receive error
0C5h		SCI4_AM	Address match event
0C6h	SCI9	SCI9_RXI*4	Receive data full
0C7h		SCI9_TXI*4	Transmit data empty
0C8h		SCI9_TEI	Transmit end
0C9h		SCI9_ERI*4	Receive error
0CAh		SCI9_AM	Address match event

Table 19.3 Association between event signal names set in ELSRn.ELS bits and signal numbers (4 of 4)

Event number	Interrupt request source	Name	Description
0CBh	SPI0	SPI0_SPRI	Receive buffer full
0CCh		SPI0_SPTI	Transmit buffer empty
0CDh		SPI0_SPII	Idle
0CEh		SPI0_SPEI	Error
0CFh		SPI0_SPTEND	Transmission completed event
0D0h	SPI1	SPI1_SPRI	Receive buffer full
0D1h		SPI1_SPTI	Transmit buffer empty
0D2h		SPI1_SPII	Idle
0D3h		SPI1_SPEI	Error
0D4h		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 Linking Events

When an event occurs and that event is already set as a trigger in the Event Link Setting Register (ELSRn), the associated module is activated. The operation of the module must be set up in advance. Table 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[7:0] 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 or 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. The ELC is initially stopped after reset. Releasing the module-stop state enables access to the registers. The ELCON bit must be set to 0 before disabling ELC operation using the Module Stop Control Register. For more information, see section 11, Low Power Modes.

19.4.4 ELC Delay Time

In Figure 19.2, module A accesses module B through the ELC. There is a delay time in the ELC between module A and module B. 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, the ELC module has some delays. The time delay is defined by the slower clock frequency between module A and module B clocks.

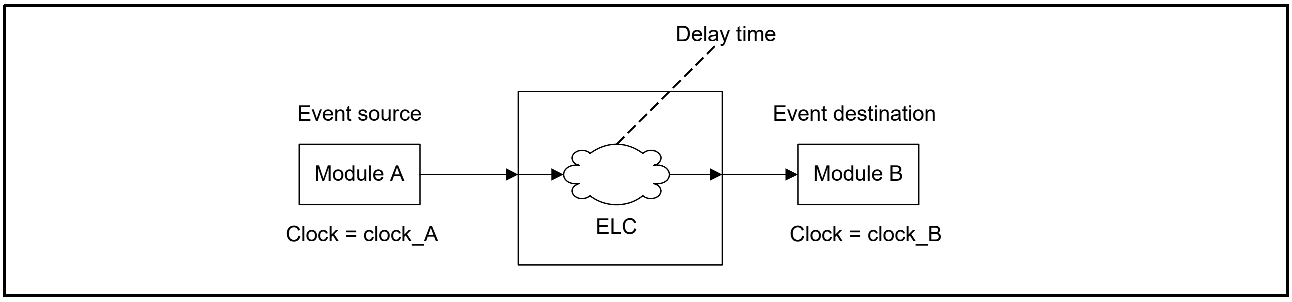


Figure 19.2 ELC delay time

Table 19.5 ELC delay time

Clock domain	Clock frequency	ELC delay time
clock_A = clock_B	clock_A = clock_B	0 cycle
clock_A ≠ clock_B	clock_A = clock_B	1 cycle to 2 cycles
	clock_A > clock_B	1 cycle to 2 cycles of B
	clock_A < clock_B	1 cycle to 2 cycles of A

20. I/O Ports

20.1 Overview

The I/O port pins operate as general I/O port pins, I/O pins for peripheral modules, interrupt input pins, analog I/O, port group function for ELC, or bus control pins. All pins operate as input pins immediately after a reset, and pin functions are switched by register settings. The I/O ports and peripheral modules for each pin are specified in the associated registers.

Figure 20.1 shows a connection diagram for the I/O port registers. The configuration of the I/O ports differs depending on the package. Table 20.1 shows the I/O port specifications, and Table 20.2 lists the port functions.

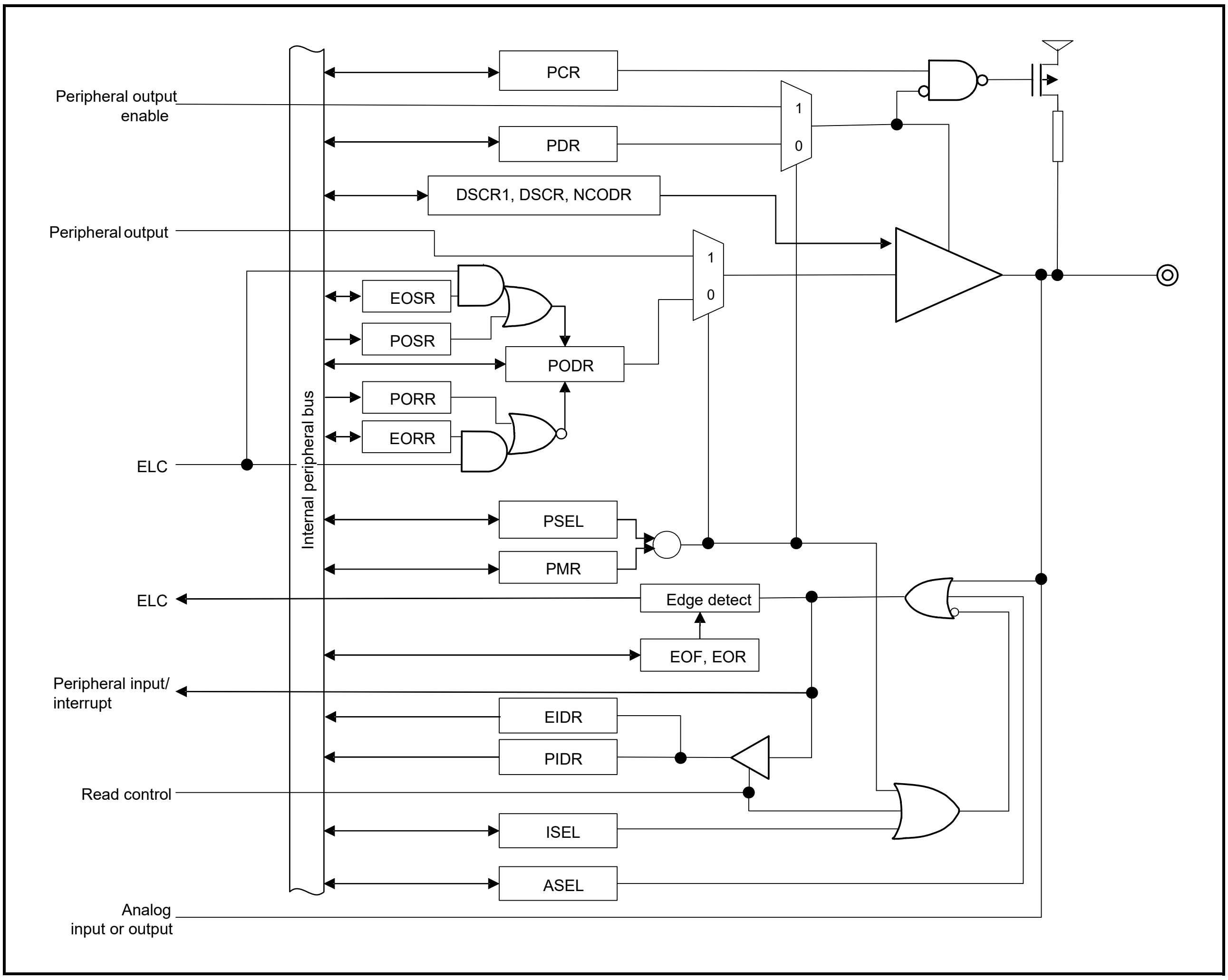

Peripheral output enable
Peripheral output
ELC
ELC
Peripheral input/ interrupt
Read control
Analog input or output
Internal peripheral bus
PCR
PDR
DSCR1, DSCR, NCODR
EOSR
POSR
PODR
PORR
EORR
PSEL
PMR
Edge detect
EOF, EOR
EIDR
PIDR
ISEL
ASEL
1
0
1
0

Figure 20.1 Connection diagram for I/O port registers

Note: Figure 20.1 shows a basic port configuration. The configuration differs depending on the ports.

Table 20.1 I/O port specifications (1 of 2)

Port	Port name	Number of pins
PORT0	P004, P010, P011, P014, P015	5
PORT1	P100 to P111	12
PORT2	P200, P201, P204 to P206, P212 to P215	9
PORT3	P300	1
PORT4	P402, P404, P407, P409, P414	5

Table 20.1 I/O port specifications (2 of 2)

Port	Port name	Number of pins
PORT5	P501	1
PORT9	P914, P915	2

✓: available

Table 20.2 I/O port functions

Port	Port name	Input pull-up	Open-drain output	Drive capacity switch	5 V tolerant
PORT0	P004, P010, P011, P014, P015	✓	-	Low, middle	-
PORT1	P100 to P111	✓	✓	Low, middle	-
PORT2	P200, P214, P215	-	-	-	-
	P201, P204	✓	✓	Low, middle	-
	P205, P206	✓	✓	Low, middle	✓
	P212, P213	✓	✓	-	-
PORT3	P300	✓	✓	Low, middle	-
PORT4	P402, P407	✓	✓	Low, middle	✓
	P404, P409, P414	✓	✓	Low, middle	-
PORT5	P501	✓	✓	Low, middle	-
PORT9	P914, P915	-	-	-	-

✓: 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, 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, 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, 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 5, 9
n = 00 to 15

The Port Control Register 1 (PCNTR1/PODR/PDR) is a 32-bit and 16-bit read/write register that controls the port direction and port output data.

The PCNTR1 specifies the port direction and output data, and is accessed in 32-bit units. The PDRn (bits [15:0] in PCNTR1) and PODRn (bits [31:16] in PCNTR1) respectively, are accessed in 16-bit units.

The PDRn bits select the input or output direction for individual pins on the associated port when the pins are configured as general I/O pins. Each pin on port m is associated with a PORTm.PCNTR1.PDRn bit. The I/O direction can be specified in 1-bit units. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0. P200, 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 serve the same function as the PDR bit in the PFS.PmnPFS register.

The PODRn bits hold data to be output from the general I/O pins. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0. P200, P214, P215 are 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, 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, 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 5, 9

n = 00 to 15

x = 1 to 4

Note 1. Supported for PORT1 to PORT4.

The Port Control Register 2 (PCNTR2/EIDR/PIDR) allows read access to the Pmn state and the port event input data using 32-bit or 16-bit access.

The PCNTR2 specifies the Pmn state and the port event input data, and is accessed in 32-bit units. The PIDRn (bits [15:0] in PCNTR2) and EIDRn (bits [31:16] in PCNTR2) respectively, are accessed in 16-bit units. Bits associated with non-existent pins are reserved. Reserved bit is read as undefined.

The PIDRn bits reflect the individual pin states of the port, regardless of the values set in PmnPFS.PMR and PmnPFS.PDR. 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)
- USB 2.0 Full-Speed Module (USBFS).

The EIDRn bits latch a pin state when an ELC_PORTx signal occurs. Pin states can only be input to EIDRn when PmnPFS.PMR and PmnPFS.PDR are 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, 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, 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, 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 effect on output 1: High output.	W
b31 to b16	PORRn	Pmn Output Reset	0: No effect on output 1: Low output.	W

m = 0 to 5, 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.

The Port Control Register 3 (PCNTR3/PORR/POSR) is a 32- and 16-bit write register that controls the setting or resetting of the port output data.

The PCNTR3 controls the setting or resetting of the port output data, and is accessed in 32-bit units. The POSRn (bits [15:0] in PCNTR3) and PORRn (bits [31:16] in PCNTR3) respectively, are accessed in 16-bit units.

POSR changes PODR when set by a software write. For example, for P100, when PORT1.PCNTR3.POSR00 is 1, PORT1.PCNTR1.PODR00 outputs 1. 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.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.PCNTR3.PORR00 is 1, PORT1.PCNTR1.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 an ELC_PORTx occurs: 0: No effect on output 1: High output.	R/W
b31 to b16	EORRn	Pmn Event Output Reset	When an ELC_PORTx occurs: 0: No effect on 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.

The Port Control Register 4 is a 32-bit and 16-bit read/write register that controls the setting or resetting of the port output data by an event input from the ELC.

The PCNTR4 controls the setting or resetting of the port output data by an event input from the ELC, and is accessed in 32-bit units. The EOSRn (bits [15:0] in PCNTR4) and EORRn (bits [31:16] in PCNTR4) respectively, are accessed in 16-bit units.

EOSR changes PODR when set because an ELC_PORTx signal occurs. For example, for P100, if PORT1.PCNTR4.EOSR00 is set to 1 when the ELC_PORTx occurs, PORT1.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. P200, 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.PCNTR4.EORR00 is set to 1 when the ELC_PORTx occurs, PORT1.PCNTR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. P200, 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 5, 9; n = 00 to 15)

Address(es): PFS.P004PFS 4004 0810h, PFS.P010PFS 4004 0828h, PFS.P011PFS 4004 082Ch, PFS.P014PFS 4004 0838h, PFS.P015PFS 4004 083Ch, PFS.P100PFS 4004 0840h to PFS.P111PFS 4004 086Ch, PFS.P200PFS 4004 0880h, PFS.P201PFS 4004 0884h, PFS.P204PFS 4004 0890h to PFS.P206PFS 4004 0898h, PFS.P212PFS 4004 08B0h to PFS.P215PFS 4004 08BCh, PFS.P300PFS 4004 08C0h, PFS.P402PFS 4004 0908h, PFS.P404PFS 4004 0910h, PFS.P407PFS 4004 091Ch, PFS.P409PFS 4004 0924h, PFS.P414PFS 4004 0938h, PFS.P501PFS 4004 0944h, PFS.P914PFS 4004 0A78h, PFS.P915PFS 4004 0A7Ch,

PFS.P004PFS_HA 4004 0812h, PFS.P010PFS_HA 4004 082Ah, PFS.P011PFS_HA 4004 082Eh, PFS.P014PFS 4004 083Ah, PFS.P015PFS_HA 4004 083Eh, PFS.P100PFS_HA 4004 0842h to PFS.P111PFS_HA 4004 086Eh, PFS.P200PFS_HA 4004 0882h, PFS.P201PFS_HA 4004 0886h, PFS.P204PFS_HA 4004 0892h to PFS.P206PFS_HA 4004 089Ah, PFS.P212PFS_HA 4004 08B2h to PFS.P215PFS_HA 4004 08BEh, PFS.P300PFS_HA 4004 08C2h, PFS.P402PFS_HA 4004 090Ah, PFS.P404PFS_HA 4004 0912h, PFS.P407PFS_HA 4004 091Eh, PFS.P409PFS_HA 4004 0926h, PFS.P414PFS_HA 4004 093Ah, PFS.P501PFS_HA 4004 0946h, PFS.P914PFS_HA 4004 0A7Ah, PFS.P915PFS_HA 4004 0A7Eh

PFS.P004PFS_BY 4004 0813h, PFS.P010PFS_BY 4004 082Bh, PFS.P011PFS_BY 4004 082Fh, PFS.P014PFS_BY 4004 083Bh, PFS.P015PFS_BY 4004 083Fh, PFS.P100PFS_BY 4004 0843h to PFS.P111PFS_BY 4004 086Fh, PFS.P200PFS_BY 4004 0883h, PFS.P201PFS_BY 4004 0887h, PFS.P204PFS_BY 4004 0893h to PFS.P206PFS_BY 4004 089Bh, PFS.P212PFS_BY 4004 08B3h to PFS.P215PFS_BY 4004 08BFh, PFS.P300PFS_BY 4004 08C3h, PFS.P402PFS_BY 4004 090Bh, PFS.P404PFS_BY 4004 0913h, PFS.P407PFS_BY 4004 091Fh, PFS.P409PFS_BY 4004 0927h, PFS.P414PFS_BY 4004 093Bh, PFS.P501PFS_BY 4004 0947h, PFS.P914PFS_BY 4004 0A7Bh, 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	EOF	EOR	DSCR1	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: Disable an input pull-up 1: Enable 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	EOF/EOR	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 edges.	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

Bit	Symbol	Bit name	Description	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
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, P300, P914 and P915 is not 0000 0000h.
P108 is 0001 0010h, P109 is 0001 0000h, P110 is 0001 0010h, P201 is 0000 0010h, P300 is 0001 0010h,P914 is 0001 0000h, and P915 is 0001 0000h.
The Port mn Pin Function Select register (PmnPFS) selects the pin function.

The Port mn Pin Function Select register (PmnPFS/PmnPFS_HA/PmnPFS_BY) is a 32-bit, 16-bit, and 8-bit read/write control register that selects the port mn pin function, and is accessed in 32-bit units. PmnPFS_HA (bits [15:0] in PmnPFS) is accessed in 16-bit units. 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 DSCR1 and DSCR bits switch the drive capacity of the port. If the drive capacity of a pin is fixed, the associated bit is read/write, 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 as an analog pin by this bit:

1. Specify it as a general I/O port with the Port Mode Control bit (PmnPFS.PMR).
2. Disable the pull-up resistor with the Pull-up Control bit (PmnPFS.PCR).
3. Specify the input with 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, P300, P914, and P915 operate as general I/O ports after reset. General I/O ports are organized as 16 bits per port and can be accessed by port with the Port Control Registers (PCNTRn, where n = 1 to 4), or by individual pins with the Pin Function Select Registers. For details on these registers, see section 20.2, Register Descriptions.

Each port has the following bits:

- Port Direction bit (PDRn), which selects input or output direction
- Port Output Data bit (PODRn), which holds data for output
- Port Input Data bit (PIDRn), which indicates the pin state
- Event Input Data bit (EIDRn), which indicates the pin state when an ELC_PORT1, 2, 3, or 4 signal occurs
- Port Output Set bit (POSRn), which indicates the output value when a software write occurs
- Port Output Reset bit (PORRn), which indicates the output value when a software write occurs
- Event Output Set bit (EOSRn), which indicates the output value when an ELC_PORT1, 2, 3, or 4 signal occurs
- Event Output Reset bit (EORRn), which indicates the output value when an ELC_PORT1, 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 direction, output data setting, and read input data
- Alternate functions: 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
- DSCR1, DSCR: Drive capacity control bit that selects 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 control bit to specify the pin function of each port
- PSEL[4:0]: Port function select bits to select the associated peripheral function.

These configurations can be made by a single-register access to the Pin Function Select Register. For details, see section 20, Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) (m = 0 to 5, 9; n = 00 to 15).

20.3.3 Port Group Function for the ELC

In the MCU, PORT1 to PORT4 are assigned for the port group function.

20.3.3.1 Behavior when ELC_PORT1, 2, 3, or 4 is input from the ELC

The MCU supports the two functions described in this section when an ELC_PORT1, 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_PORT1, 2, 3, or 4 signal comes from the ELC, the input enable of the I/O cell is asserted, and data from the external pins are read into the EIDR bit.

For the GPO function (PDR = 1) or the peripheral mode (PMR = 1), 0 is input into the EIDR bit from the external pins.

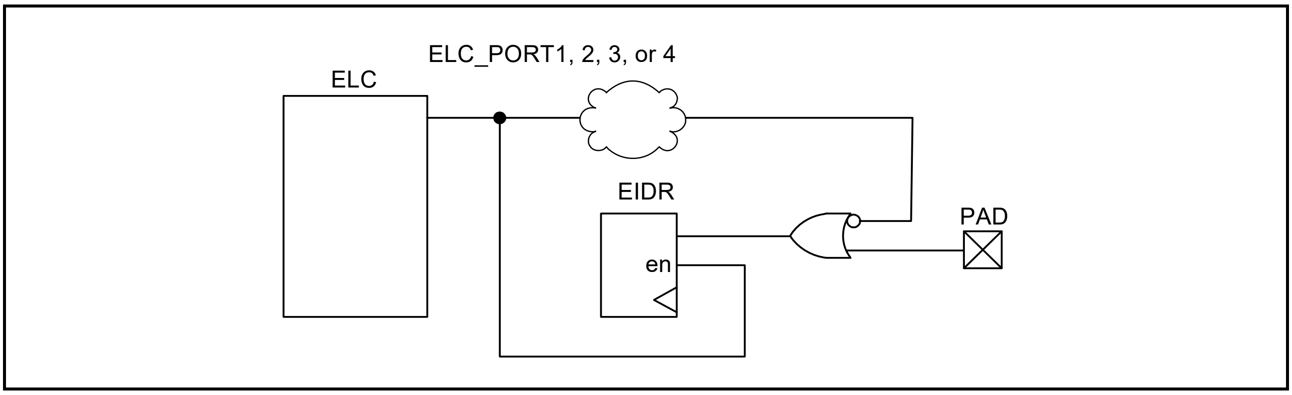


Figure 20.2 Event ports input data

(2) Output from PODR by EOSR/EORR

When an ELC_PORT1, 2, 3, or 4 signal occurs, the data is output from the PODR to the external pin based on the EOSR/EORR bit settings as follows:

- If EOSR is set to 1, when an ELC_PORT1, 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 EORR is set to 1, when an ELC_PORT1, 2, 3, or 4 signal occurs, the PODR register outputs 0 to the external pin. Otherwise, when EORR = 0, the PODR value is kept.

See Figure 20.3.

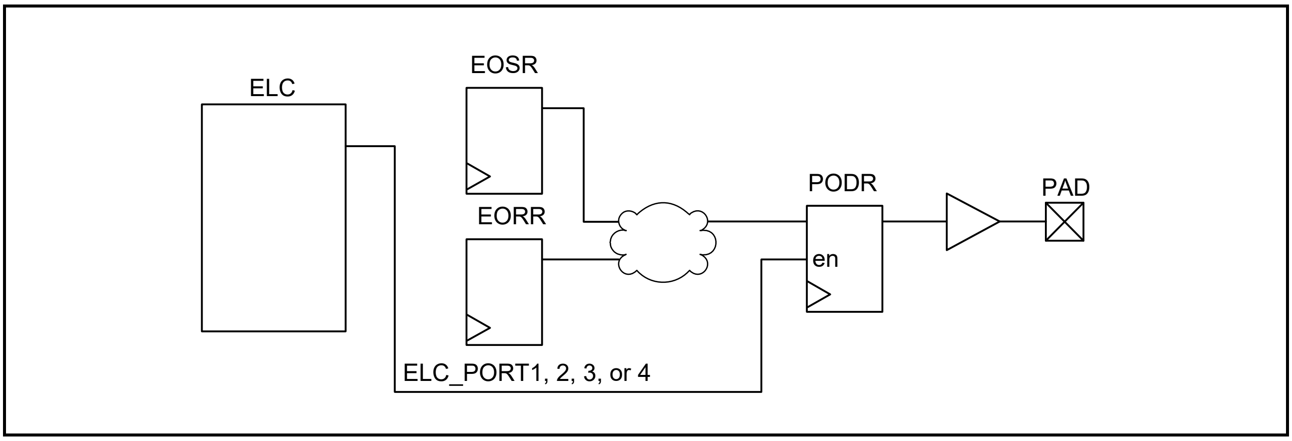

ELC
EOSR
EORR
PODR
en
PAD
ELC_PORT1, 2, 3, or 4

Figure 20.3 Event ports output data

20.3.3.2 Behavior when an event pulse is output to the ELC

To output the event pulse from the external pins to the ELC, set the EOR/EOF bits 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 5, 9; n = 00 to 15). When the EOR/EOF bits are set, the input enable of the I/O cell is asserted.

Data from the external pin is the input. For example, for PORT1, when the data is input from P100 to P111, the data of those 12 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. See Figure 20.4.

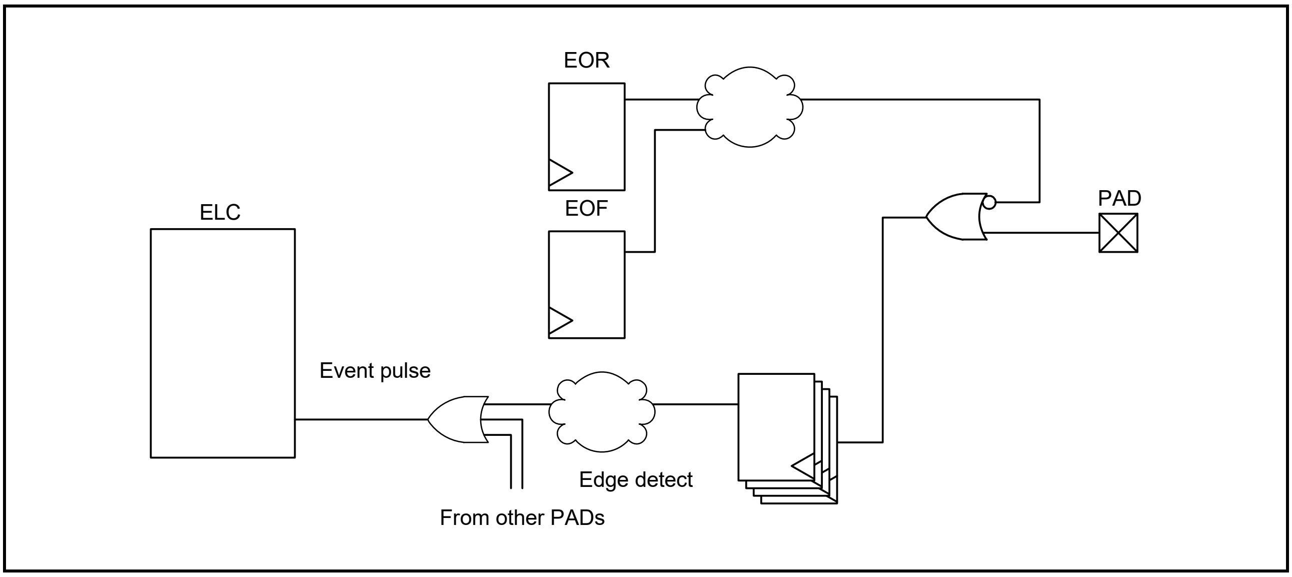

EOR
EOF
ELC
PAD
Event pulse
Edge detect
From other PADs

Figure 20.4 Generation of event pulse

20.4 Handling of Unused Pins

Table 20.3 shows how to handle unused pins.

Table 20.3 Handling of unused pins

Pin name	Description
P201/MD	Use as a mode pin
RES	Connect to VCC through a resistor (pulling up)
USB_DP, USB_DM	When both P914PFS.PMR and P915PFS.PMR bits are set to 1, keep these pins open. When P914PFS.PMR or P915PFS.PMR bit is set to 0, configure it in the same way as ports 1 to 9.
P200/NMI	Connect to VCC through a resistor (pulling up)
P212/EXTAL	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P212). When this pin is not used as port P212, configure it in the same way as ports 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, configure it in the same way as ports 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, configure it in the same way as ports 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, configure it in the same way as ports 1 to 9.
P004, P010, P011, P014, 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), release the pin.*1,*3

Note 1. Clear the PmnPFS.PMR, PmnPFS.ISEL, PmnPFS.PCR, and PmnPFS.ASEL bits to 0.
Note 2. P108, P110, and P300 are recommended for pull up VCC (pulled up) through a resistor, because these pins are input pull-up enabled from the initial value (PmnPFS.PCR=1).
Note 3. P109 is recommended to be set as an output (PCNTR1.PDRn = 1), because this pin is output from the initial value.

20.5 Usage Notes

20.5.1 Procedure for Specifying the Pin Functions

To specify the I/O pin functions:

1. Clear the B0WI bit in the PWPR register. This enables writing to the PFSWE bit in the PWPR register.
2. Set 1 to the PFSWE bit in the PWPR register. This enables writing to the PmnPFS register.
3. Clear the Port Mode Control bit in the PMR 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 required 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[7:0] bits to 0000 0000b to ignore unexpected pulses. 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 of 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.EORRn is set to 1 when an ELC_PORT1, 2, 3, or 4 signal occurs.
2. Output 1 if the PCNTR4.EOSRn is set to 1 when an ELC_PORT1, 2, 3, or 4 signal occurs.
3. Output 0 if PCNTR3.PORRn is set to 1.
4. Output 1 if PCNTR3.POSRn is set to 1.
5. Output 0 or 1 because PCNTR1.PODRn is set.
6. Output 0 or 1 because PmnPFS.PODRn is set.

Numbers in this list correspond to the priority for writing to the PODR. For example, if 1. and 3. from the list occur at the same time, the higher priority 1. is executed.

20.5.4 Notes on Using Analog Functions

To use an analog function, set 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 (ASEL) in the Port mn Pin Function Select register (PmnPFS.ASEL) to 1.

20.5.5 I/O Buffer Specification

When the P402 pin is configured as output or input 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. 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 pin can be used as the RTC input pin RTCIC0. When this input pin is enabled by the VBTICTLR register, the output function of this pin 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.

20.5.6 Selecting the USB_DP and USB_DM Pins

The USB_DP pin is shared with pin P914. The USB_DM pin is shared with pin P915. USB_DP and P914 pins can be set with the PFS.P914PFS.PMR bit, and USB_DM and P915 pins can be set with the PFS.P915PFS.PMR bit. Table 20.4 shows the setting values of bits PFS.P914PFS.PMR and PFS.P915PFS.PMR with each selected pin.

Table 20.4 Selecting the USB/PORT pins

PMR bit settings		Pins selected	
P914PFS.PMR bit	**P915PFS.PMR bit**	**P914/USB_DP pin**	**P915/USB_DM pin**
0	0	P914	P915
0	1	P914	P915
1	0	P914	P915
1	1	USB_DP	USB_DM

Note: When using P914/USB_DP and P915/USB_DM as GPIO pins (P914 and P915), use the USB registers with their initial values.

Note: When using P914/USB_DP and P915/USB_DM as USB pins (USB_DP and USB_DM), use the GPIO registers for P914 and P915 with their initial values.

Note: When using P914/USB_DP and P915/USB_DM as GPIO pins or USB pins, set these pins only once after a reset.

20.5.7 Pull-up/Pull-down Setting for P914 and P915 using USBFS/GPIO Function

When P914 and P915 are used as GPIO pins, their operation is affected by the pull-up/pull-down function of the USBFS registers.

Therefore, before using the GPIO function, disable the pull-up and pull-down control of the USBFS registers using the SYSCFG.DMRPU, SYSCFG.DPRPU, and SYSCFG.DRPD bits.

20.6 Peripheral Select Settings for each Product

This section describes the pin function select configuration by the PmnPFS register. Assigning the same function to two or more pins simultaneously is prohibited.

Table 20.5 Register settings for I/O pin functions (PORT0)

PSEL[4:0] bit settings	Function	Pin P004	P010	P011	P014	P015
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z				
01100b	CTSU	—	TS30	TS31	—	TS28
ASEL bit		AN004/ AMP2O	AN005/ VREFH0/ AMP2-	AN006/ VREFL0/ AMP2+	AN009/ DA0	AN010
ISEL bit		IRQ3	IRQ14	IRQ15	—	IRQ7
PCR bit		✓	✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M	L/M

✓: Available
—: Setting prohibited

Table 20.6 Register settings for I/O pin functions (PORT1) (1)

PSEL[4:0] bit settings	Function	Pin P100	P101	P102	P103	P104	P105	P106	P107
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
00001b	AGT	AGTIO0	AGTEE0	AGTO0	—	—	—	—	—
00010b	GPT	GTETRGA	GTETRGB	GTOWLO	GTOWUP	GTETRGB	GTETRGA	—	—
00011b	GPT	GTIOC5B	GTIOC5A	GTIOC2B	GTIOC2A	GTIOC1B	GTIOC1A	GTIOC8B	GTIOC8A
00100b	SCI	RXD0/ MISO0/ SCL0	TXD0/ MOSI0/ SDA0	SCK0	CTS0_RTS0/ SS0	RXD0/ MISO0/ SCL0	—	—	—
00101b	SCI	SCK1	CTS1_RTS1/ SS1	TXD2/ MOSI2/ SDA2	—	—	—	—	—
00110b	SPI	MISOA	MOSIA	RSPCKA	SSLA0	SSLA1	SSLA2	SSLA3	—
00111b	IIC	SCL1	SDA1	—	—	—	—	—	—
01000b	KINT	KR00	KR01	KR02	KR03	KR04	KR05	KR06	KR07
01001b	CLKOUT/ ACMPLP/ RTC	—	—	—	—	—	—	—	—
01010b	CAC/ADC14	—	—	ADTRG0	—	—	—	—	—
01100b	CTSU	—	—	—	—	TS13	TS34	—	—
01101b	SLCDC	VL1	VL2	—	VL4	COM0	COM1	COM2	COM3
10000h	CAN	—	—	CRX0	CTX0	—	—	—	—
ASEL bit		CMPIN0	CMPREF0	AN020/ CMPIN1	AN019/ 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

✓: Available
—: Setting prohibited

Table 20.7 Register settings for I/O pin functions (PORT1) (2)

PSEL[4:0] bit settings	Function	Pin P108	P109	P110	P111
00000b (value after reset)	Hi-Z/JTAG/SWD	TMS/ SWDIO	TDO/ TRACESWO	TDI	Hi-Z
00001b	AGT	—	—	—	—
00010b	GPT	GTOULO	GTOVUP	GTOVLO	—
00011b	GPT	GTIOC0B	GTIOC1A	GTIOC1B	GTIOC3A
00100b	SCI	—	SCK1	CTS2_RTS2/ SS2	SCK2
00101b	SCI	CTS9_RTS9/ SS9	TXD9/ MOSI9/ SDA9	RXD9/ MISO9/ SCL9	SCK9
00110b	SPI	SSLB0	MOSIB	MISOB	RSPCKB
00111b	IIC	—	—	—	—
01000b	KINT	—	—	—	—
01001b	CLKOUT/ ACMPLP/ RTC	—	CLKOUT	VCOUT	—
01010b	CAC/ADC14	—	—	—	—
01100b	CTSU	—	TS10	—	TS12
01101b	SLCDC	—	SEG52	SEG53	—
10000b	CAN	—	CTX0	CRX0	—
ASEL bit		—	—	—	—
ISEL bit		—	—	IRQ3	IRQ4
NCODR bit		✓	✓	✓	✓
PCR bit		✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M

✓: Available
—: Setting prohibited

Table 20.8 Register settings for I/O pin functions (PORT2) (1)

PSEL[4:0] bit settings	Function	Pin				
		P200	P201	P204	P205	P206
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z				
00001b	AGT	—	—	AGTIO1	AGTO1	—
00010b	GPT	—	—	GTIW	GTIV	GTIU
00011b	GPT	—	—	GTIOC4B	GTIOC4A	—
00100b	SCI	—	—	SCK4	TXD4/ MOSI4/ SDA4	RXD4/ MISO4/ SCL4
00101b	SCI	—	—	SCK9	CTS9_RTS9/ SS9	—
00110b	SPI	—	—	RSPCKB	SSLB0	SSLB1
00111b	IIC	—	—	SCL0	SCL1	SDA1
01001b	CLKOUT/ ACMPLP/ RTC	—	—	—	CLKOUT	—
01010b	CAC/ADC14	—	—	CACREF	—	—
01100b	CTSU	—	—	TS00	TSCAP	TS01
01101b	SLCDC	—	—	SEG23	SEG20	SEG12
10011b	USBFS	—	—	USB_OVRCURB	USB_OVRCURA	USB_VBUSEN
ISEL bit		NMI	—	—	IRQ1	IRQ0
NCODR bit		—	✓	✓	✓	✓
PCR bit		—	✓	✓	✓	✓
DSCR bit		—	L/M	L/M	L/M	L/M

✓: Available
—: Setting prohibited

Table 20.9 Register settings for I/O pin functions (PORT2) (2)

PSEL[4:0] bit settings	Function	Pin			
		P212	P213	P214	P215
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z			
00001b	AGT	AGTEE1	—	—	—
00010b	GPT	GTETRGB	GTETRGA	—	—
00011b	GPT	GTIOC0B	GTIOC0A	—	—
00100b	SCI	—	—	—	—
00101b	SCI	RXD1/ MISO1/ SCL1	TXD1/ MOSI1/ SDA1	—	—
00110b	SPI	—	—	—	—
00111b	IIC	—	—	—	—
01001b	CLKOUT/ ACMPLP/ RTC	—	—	—	—
01010b	CAC/ADC14	—	—	—	—
01100b	CTSU	—	—	—	—
01101b	SLCDC	—	—	—	—
10011b	USBFS	—	—	—	—
ISEL bit		IRQ3	IRQ2	—	—
NCODR bit		✓	✓	—	—
PCR bit		✓	✓	—	—
DSCR bit		—	—	—	—

✓: Available
—: Setting prohibited

Table 20.10 Register settings for I/O pin functions (PORT3)

PSEL[4:0] bit settings	Function	Pin P300
00000b (value after reset)	Hi-Z/JTAG/SWD	TCK/ SWCLK
00001b	AGT	—
00010b	GPT	GTOUUP
00011b	GPT	GTIOC0A
00110b	SPI	SSLB1
ISEL bit		—
NCODR bit		✓
PCR bit		✓
DSCR bit		L/M

✓: Available
—: Setting prohibited

Table 20.11 Register settings for I/O pin functions (PORT4) (1)

PSEL[4:0] bit settings	Function	Pin P402	P404	P407
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z		
00001b	AGT	AGTIO0*2/ AGTIO1*2	—	AGTIO0
00010b	GPT	—	—	—
00011b	GPT	—	GTIOC3B	—
00100b	SCI	—	—	CTS4_RTS4/ SS4
00101b	SCI	RXD1/ MISO1/ SCL1	—	—
00110b	SPI	—	—	SSLB3
00111b	IIC	—	—	SDA0
01001b	CLKOUT/ ACMPLP/ RTC	—	—	RTCOUT
01010b	CAC/ADC14	—	—	ADTRG0
01100b	CTSU	TS18	—	TS03
01101b	SLCDC	SEG06	—	SEG11
10000b	CAN	CRX0	—	—
10011b	USBFS	—	—	USB_VBUS
Don’t care		RTCIC0*1	—	—
ISEL bit		IRQ4	—	—
NCODR bit		✓	✓	✓
PCR bit		✓	✓	✓
DSCR bit		L/M	L/M	L/M

✓: 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).

Note 2. To use this pin function, set the PmnPFS.PSEL[4:0] bits and the AGTIOSEL.SEL[1:0] bits (described in section 24, AGT Pin Select Register (AGTIOSEL)).

Table 20.12 Register settings for I/O pin functions (PORT4) (2)

PSEL[4:0] bit settings	Function	Pin P409	P414
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z	
00001b	AGT	—	—
00010b	GPT	GTOWUP	—
00011b	GPT	GTIOC5A	GTIOC0B
00100b	SCI	—	—
00101b	SCI	—	—
00110b	SPI	—	SSLA1
00111b	IIC	—	—
01001b	CLKOUT/ ACMPLP/ RTC	—	—
01010b	CAC/ADC14	—	—
01100b	CTSU	—	—
01101b	SLCDC	SEG09	—
10000b	CAN	—	—
10011b	USBFS	—	—
Don’t care		—	—
ISEL bit		IRQ6	IRQ9
NCODR bit		✓	✓
PCR bit		✓	✓
DSCR bit		L/M	L/M

✓: Available
—: Setting prohibited

Note: To use this pin function, set the corresponding pin as general input (set the PmnPFS.PDR and PmnPFS.PMR bits to 0).

Note: To use this pin function, set the PmnPFS.PSEL[4:0] bits and the AGTIOSEL.SEL[1:0] bits (described in section 24, AGT Pin Select Register (AGTIOSEL)).

Table 20.13 Register settings for I/O pin functions (PORT5)

PSEL[4:0] bit settings	Function	Pin P501
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z
00001b	AGT	AGTOB0
00010b	GPT	GTIV
00011b	GPT	GTIOC2B
01101b	SLCDC	SEG49
10011b	USBFS	USB_OVRCURA
ASEL bit		AN017/ CMPIN1
ISEL bit		IRQ11
NCODR bit		✓
PCR bit		✓
DSCR bit		L/M

✓: Available
—: Setting prohibited

Table 20.14 Register settings for I/O pin functions (PORT9)

PSEL[4:0] bit settings	Function	Pin	
		P914	P915
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z	
00001b	AGT	—	—
00100b	SCI	—	—
01011b	BUS	—	—
Don’t care		(USB_DP)	(USB_DM)
NCODR bit		—	—
PCR bit		—	—
DSCR bit		—	—

✓: 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 a block diagram.

Table 21.1 Assignment of key interrupt detection pins

Flag	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

Parameter	Configuration
Input	KR00 to KR07
Control registers	Key Return Control register (KRCTL) Key Return Mode register (KRM) Key Return Flag register (KRF)

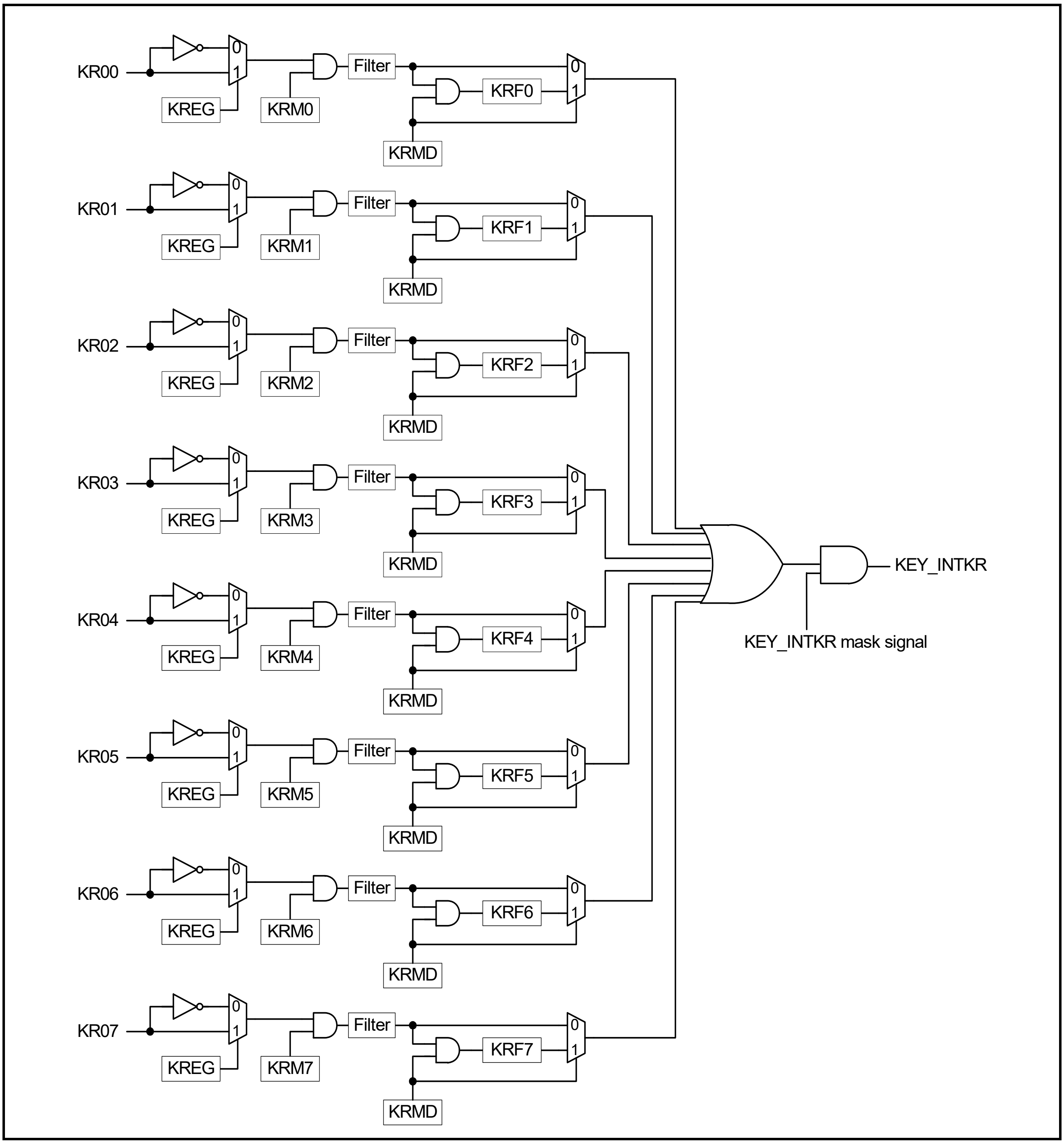

KR00
KREG
KRM0
Filter
KRF0
KRMD
KR01
KREG
KRM1
Filter
KRF1
KRMD
KR02
KREG
KRM2
Filter
KRF2
KRMD
KR03
KREG
KRM3
Filter
KRF3
KRMD
KR04
KREG
KRM4
Filter
KRF4
KRMD
KR05
KREG
KRM5
Filter
KRF5
KRMD
KR06
KREG
KRM6
Filter
KRF6
KRMD
KR07
KREG
KRM7
Filter
KRF7
KRMD
0
1
KEY_INTKR
KEY_INTKR mask signal

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 the AND gate to mask the merged key return factor by the KEY_INTKR mask signal. When using KRFn (KRMD = 1), the 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: Use key interrupt flags.	R/W

The KRCTL register 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 0 to the bit, then write 1 to the other bits.

The KRF register 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: No key interrupt signal detected 1: Key interrupt signal detected.	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 in the PmnPFS.PSEL bits. For more information, see section 20, I/O Ports.
An interrupt is generated when the target bit in the KRM register 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 register after disabling the interrupt handling.

The KRM register sets the key interrupt mode.

21.3 Operation

21.3.1 Operation 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 based on the input level of the key interrupt input pin, KR00 to KR07.

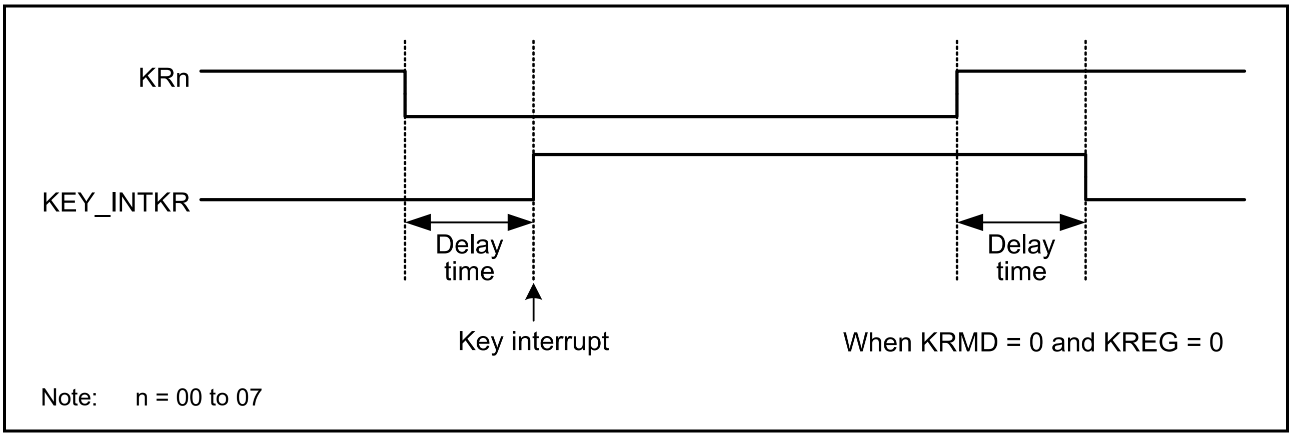


Figure 21.2 Operation of KEY_INTKR signal when a key interrupt is input to a single channel

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.

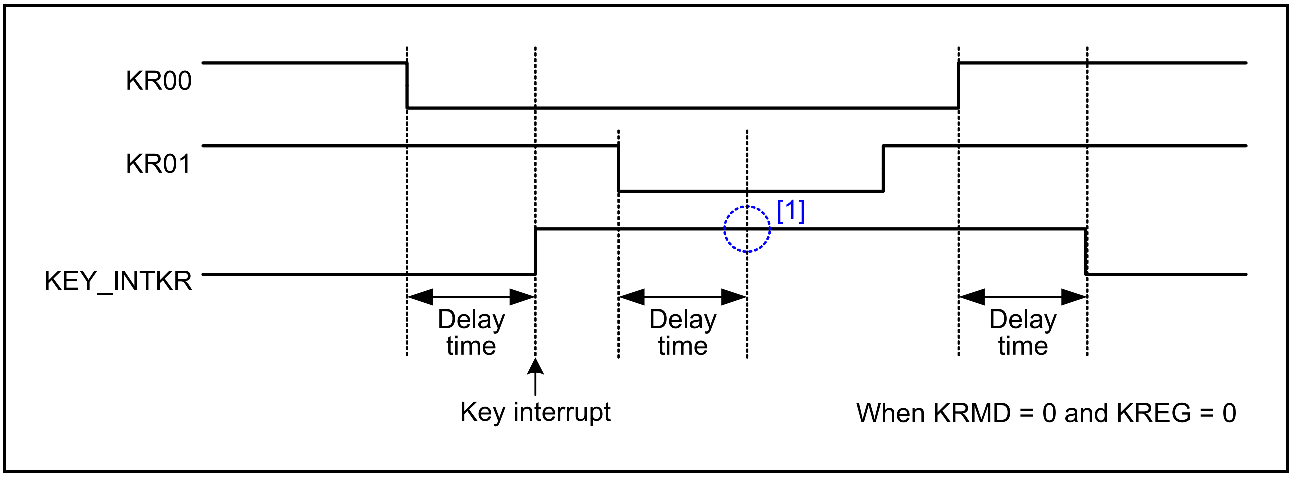


Figure 21.3 Operation of KEY_INTKR signal when key interrupts are input to multiple channels

21.3.2 Operation 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 register.

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.

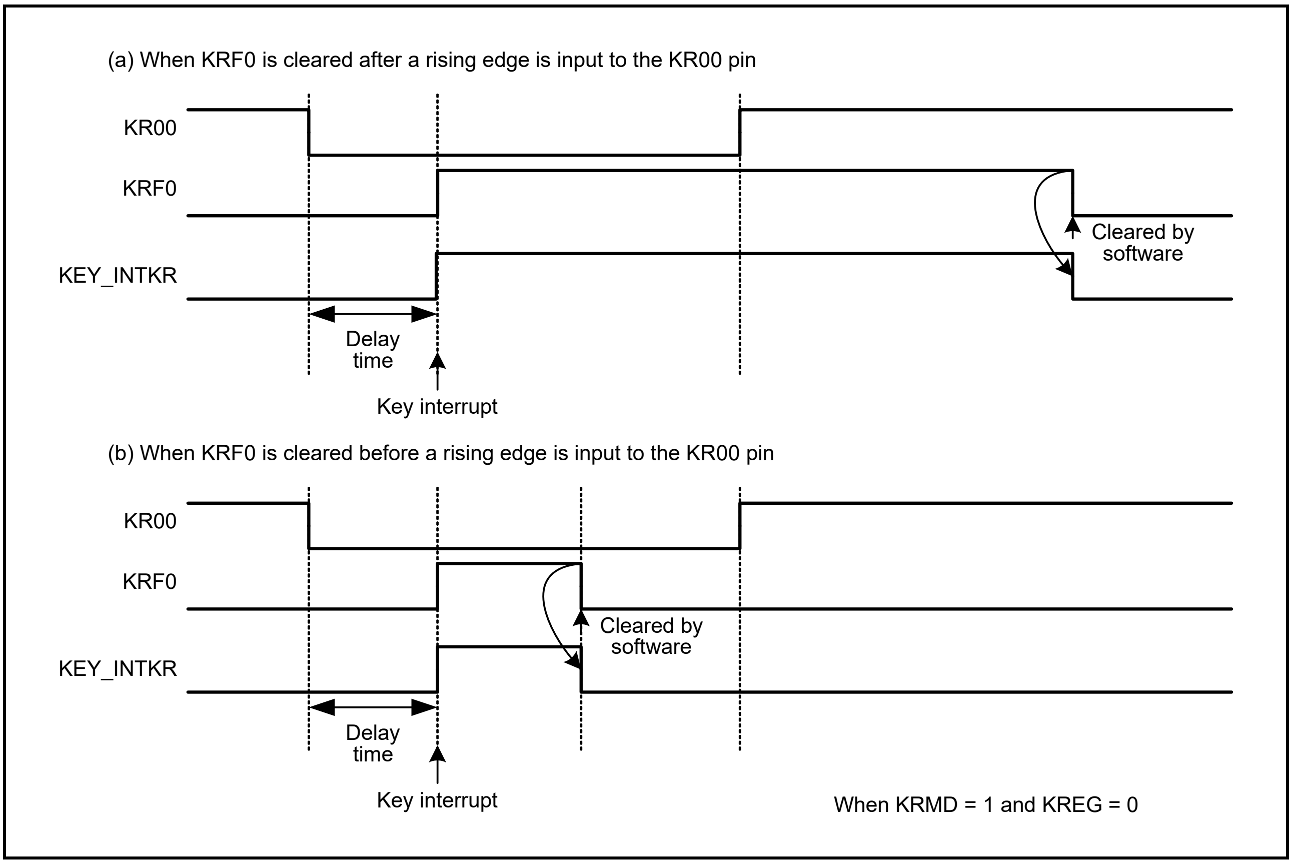

(a) When KRF0 is cleared after a rising edge is input to the KR00 pin
KR00
KRF0
KEY_INTKR
Cleared by software
Delay time
Key interrupt
(b) When KRF0 is cleared before a rising edge is input to the KR00 pin
KR00
KRF0
KEY_INTKR
Cleared by software
Delay time
Key interrupt
When KRMD = 1 and KREG = 0

Figure 21.4 Basic operation of KEY_INTKR signal when key interrupt flag is used

Figure 21.5 shows the operation when a valid edge is input to multiple key interrupt input pins. 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 1 PCLKB clock cycle, 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 1 PCLKB clock cycle, 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.

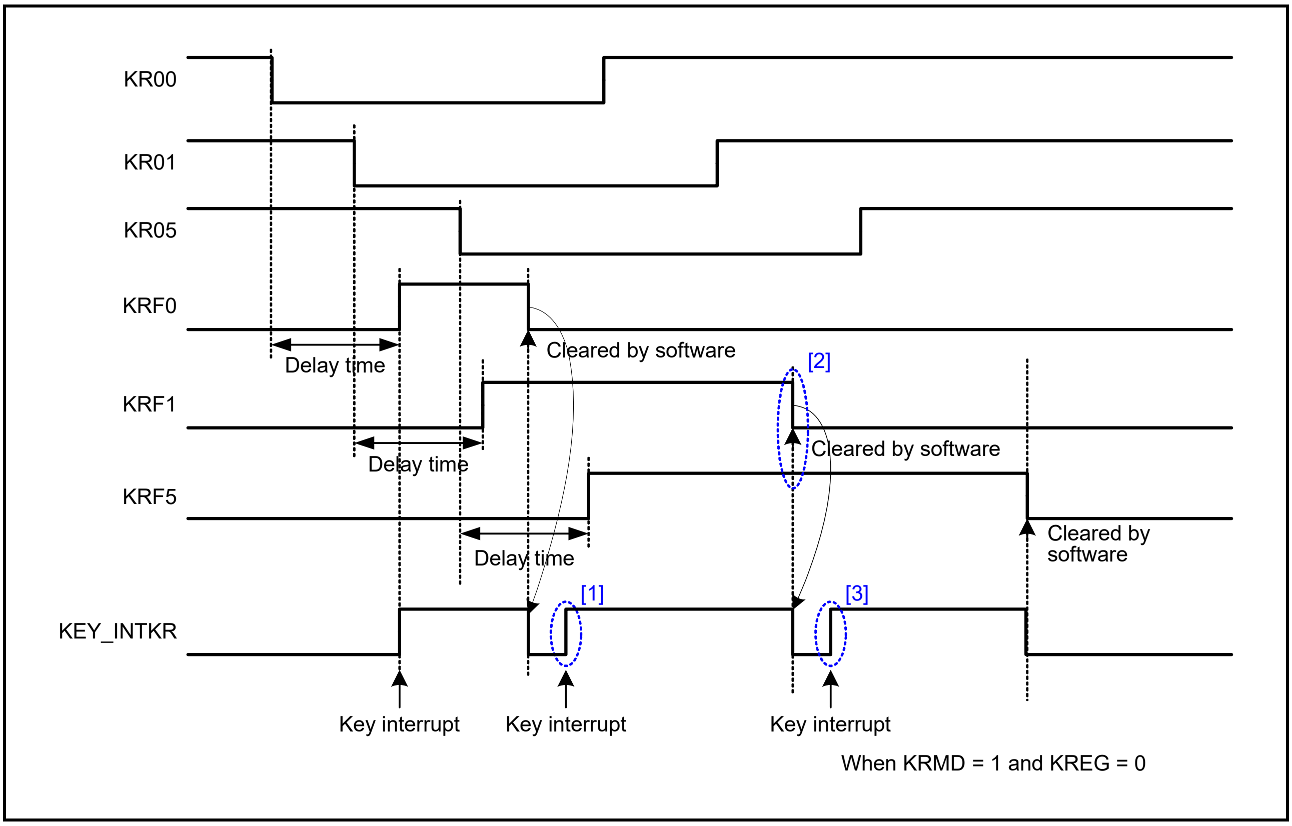

KR00
KR01
KR05
KRF0
KRF1
KRF5
KEY_INTKR
Delay time
Delay time
Delay time
Cleared by software
Cleared by software
Cleared by software
[1]
[2]
[3]
Key interrupt
Key interrupt
Key interrupt
When KRMD = 1 and KREG = 0

Figure 21.5 Operation of KEY_INTKR signal when key interrupts are input to multiple channels

21.4 Usage Notes

- If KEY_INTKR is used as the snooze request, the KRMD bit must be set to 0.
- If KEY_INTKR is used as the interrupt source for returning to Normal mode from Snooze and Software Standby modes, the KRMD bit must be set to 1.
- When the key interrupt function is assigned to a pin, this pin input is always enabled in Software Standby mode, and if this pin level changes, the associated KRFn can be set. Therefore, a key interrupt might occur on canceling Software Standby mode.

To ignore changes to the key interrupt pin during a software standby, clear the associated KRM bit before entering Software Standby mode. After canceling Software Standby mode, clear KRFn before the associated KRM bit is set.

22. Port Output Enable for GPT (POEG)

22.1 Overview

The Port Output Enable (POEG) can place the General PWM Timer (GPT) output pins in the output-disable state in one of the following ways:

- Input level detection of the GTETRGn (n = A, B) 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) pins can also be used as GPT external trigger input pins.

Table 22.1 lists the POEG specifications, Figure 22.1 shows a block diagram, and Table 22.2 lists the input pins.

Table 22.1 POEG specifications

Parameter	Description
Output-disable control through input level detection	The GPT output pins can be disabled when a GTETRGn rising edge or high level is sampled after polarity and filter selection
Output-disable request from the 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 output-disabled.
Output-disable control through oscillation stop detection	The GPT output pins can be disabled when oscillation of 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 requests from the GPT.
External trigger output function to GPT (count start, count stop, count clear, up-count, down-count, or 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 each input pin GTETRGn • Positive or negative polarity can be selected for each input pin, GTETRGn • The signal state after polarity and filter selection can be monitored.

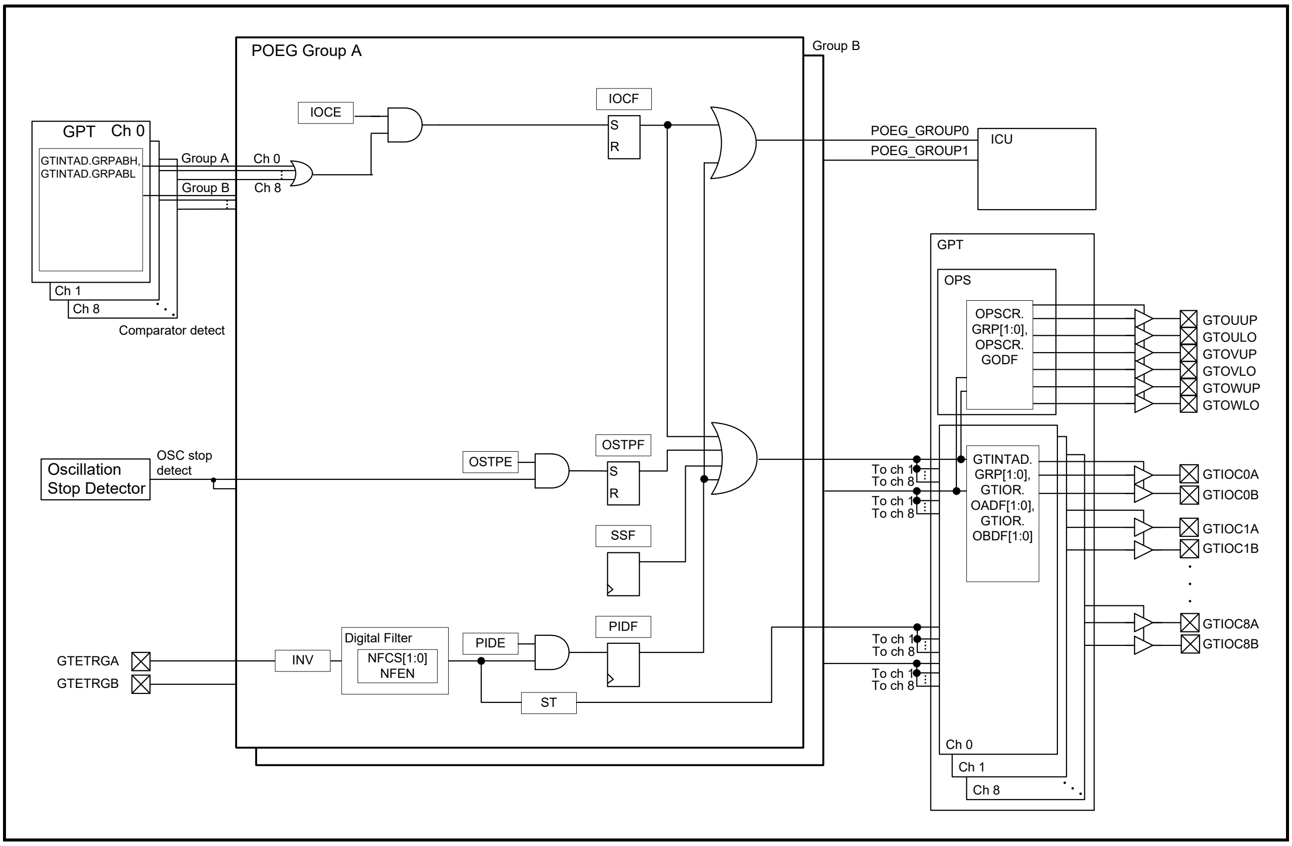

POEG Group A
Group B
IOCE
IOCF
S
R
GPT Ch 0
GTINTAD.GRPABH,
GTINTAD.GRPABL
Group A
Group B
Ch 0
Ch 8
Ch 1
Ch 8
Comparator detect
POEG_GROUP0
POEG_GROUP1
ICU
GPT
OPS
OPSCR.
GRP[1:0],
OPSCR.
GODF
GTOUUP
GTOULO
GTOVUP
GTOVLO
GTOWUP
GTOWLO
Oscillation Stop Detector
OSC stop detect
OSTPE
OSTPF
SSF
To ch 1
To ch 8
GTINTAD.
GRP[1:0],
GTIOR.
OADF[1:0],
GTIOR.
OBDF[1:0]
GTIOC0A
GTIOC0B
GTIOC1A
GTIOC1B
GTIOC8A
GTIOC8B
PIDF
GTETRGA
GTETRGB
INV
Digital Filter
NFCS[1:0]
NFEN
PIDE
ST
Ch 0
Ch 1
Ch 8

Figure 22.1 POEG block diagram

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

22.2 Register Descriptions

22.2.1 POEG Group n Setting Register (POEGGn) (n = A, B)

Address(es): POEG.POEGGA 4004 2000h, POEG.POEGGB 4004 2100h

	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
	—	—	—	—	—	—	—	—	—	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 occurred 1: Output-disable request from the GTETRGn pin occurred.	R(/W)*1

Bit	Symbol	Bit name	Description	R/W
b1	IOCF	Output-disable Request Detection Flag from GPT	0: No output-disable request from the GPT disable request occurred 1: Output-disable request from the GPT disable request occurred.	R(/W)*1
b2	OSTPF	Oscillation Stop Detection Flag	0: No output-disable request from oscillation stop detection 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 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	Output-disable Request Enable from GPT	0: Output-disable request from the GPT disable request disabled 1: Output-disable request from the GPT disable request enabled.	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
b15 to b7	—	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 as-is 1: GTETRGn input reversed.	R/W
b29	NFEN	Noise Filter Enable	0: Noise filtering disabled 1: Noise filtering 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, 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, the GTIOCxA, GTIOCxB, and the 3-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 applies to the group selected in the GPT registers GTINTAD.GRP[1:0] and OPSCR.GRP[1:0].
- 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 PWM output is disabled.

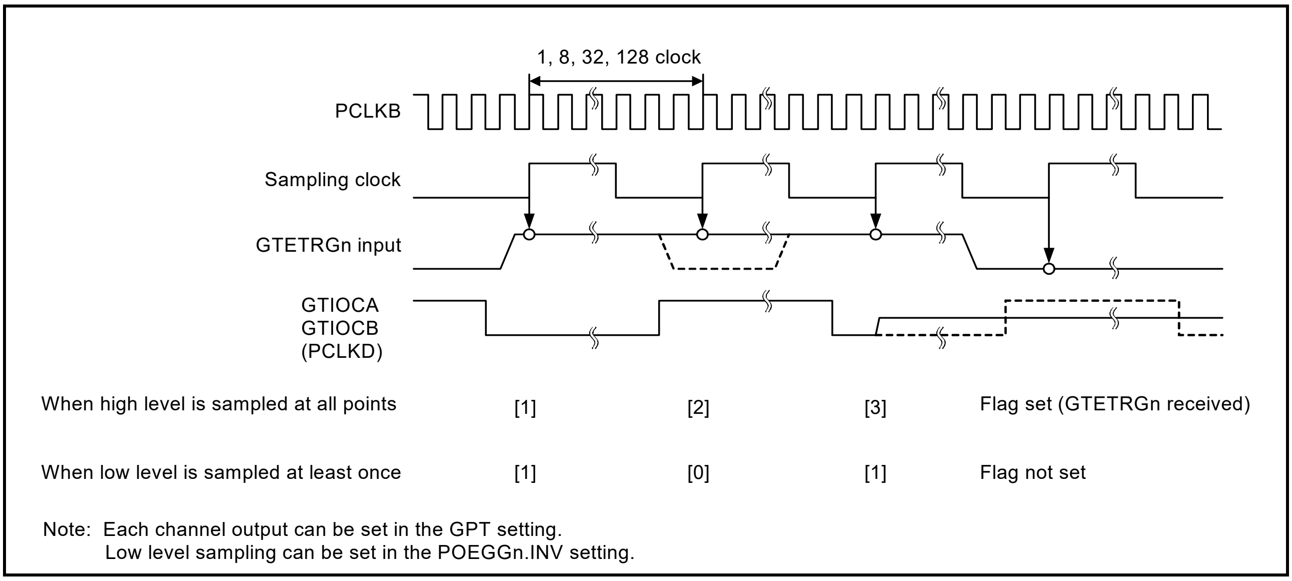
The output-disable state is controlled in the GPT. The output disable of the GTIOCxA and GTIOCxB pins is set in the GTINTAD.GRP[1:0], GTIOR.OADF[1:0], and GTIOR.OBDF[1:0] bits in the GPT. The output disable of the 3-phase PWM output for the BLDC motor control pins is set in the OPSCR.GRP[1:0] and OPSCR.GODF 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 output-disabled.

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 output-disabled. 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.


1, 8, 32, 128 clock
PCLKB
Sampling clock
GTETRGn input
GTIOCA
GTIOCB
(PCLKD)
When high level is sampled at all points
[1]
[2]
[3]
Flag set (GTETRGn received)
When low level is sampled at least once
[1]
[0]
[1]
Flag not set

Note: Each channel output can be set in the GPT setting.
Low level sampling can be set in the POEGGn.INV setting.

Figure 22.2 Example of digital filter operation

22.3.2 Output-Disable Request from GPT

For details on the operation, see GTIOC Pin Output Negate Control in section 23, General PWM Timer (GPT).

22.3.3 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 output-disabled for each group.

22.3.4 Output-Disable Control Using Registers

The GPT output pins can be directly controlled by writing to the Software Stop Flag, POEGGn.SSF.

22.3.5 Release from Output Disable

To release the GPT output pins in the output-disable state, either return them to their initial state with a reset or clear all of the following flags:

- POEGGn.PIDF
- POEGGn.IOCF
- POEGGn.OSTPF
- POEGGn.SSF.

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.

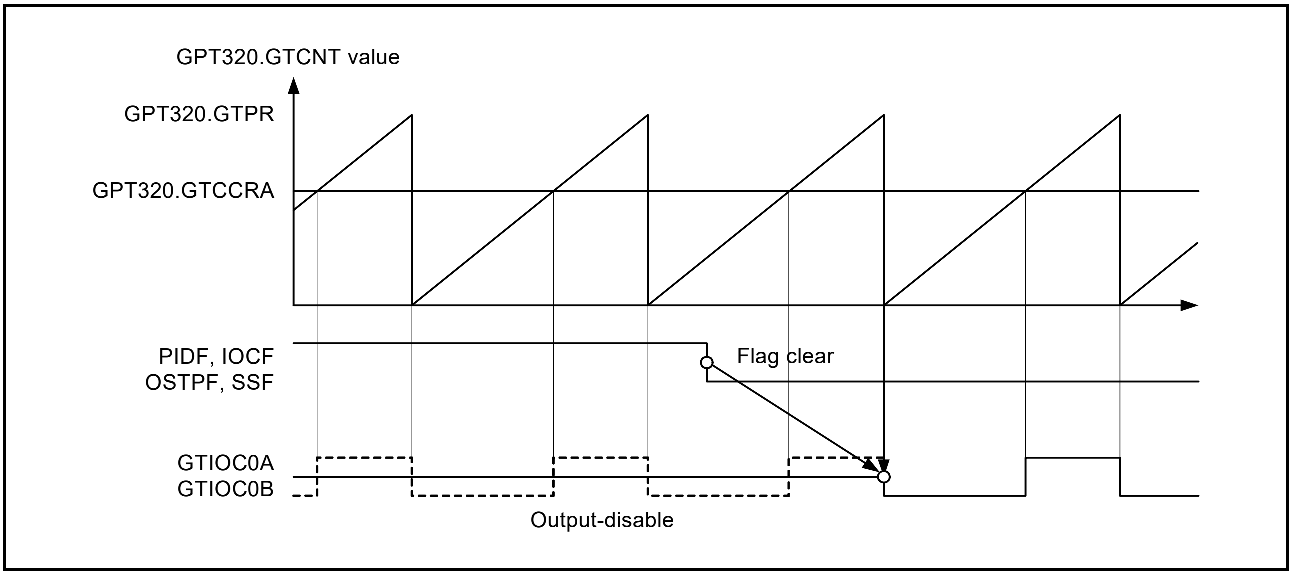


Figure 22.3 Output-disable release timing for GPT pin outputs

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 the GPT.

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
		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
		POEGGB.PIDF	An output-disable request from the GTETRGB pin occurred

22.5 External Trigger Output to GPT

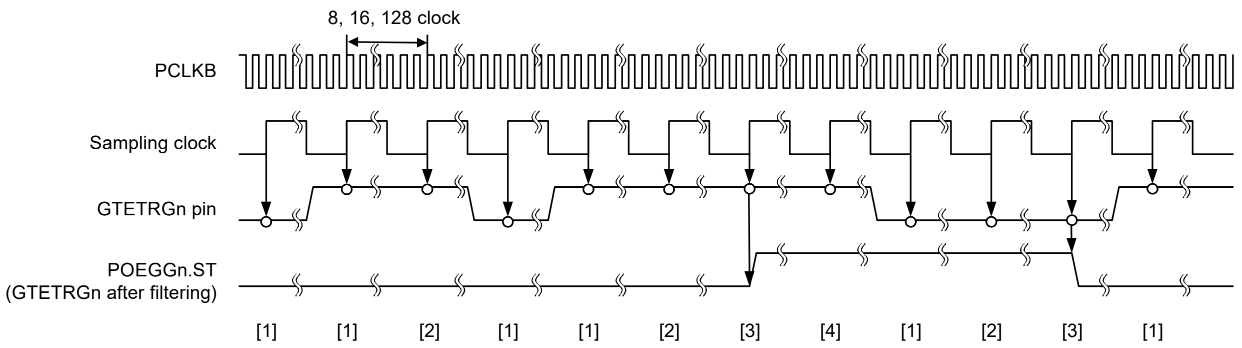
The POEG outputs the GTETRGn signals as the GPT operation trigger signal for the following:

- 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 bits, 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.



Note 1. Each channel output can be set in the GPT setting.
Polarity can be reversed in the POEGGn.INV setting.

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 Software Standby mode. In this mode, the POEG stops and therefore output disable of the pins cannot be controlled.

22.6.2 Specifying Pins Associated with the 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 General PWM Timer (GPT) is a 32-bit timer with four GPT32 channels, and a 16-bit timer with three GPT16 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.

Table 23.1 lists the GPT specifications, Table 23.2 shows the GPT functions, Figure 23.1 shows a block diagram, Figure 23.2 shows the correspondence between the GPT channels and module names, and Table 23.3 lists the I/O pins.

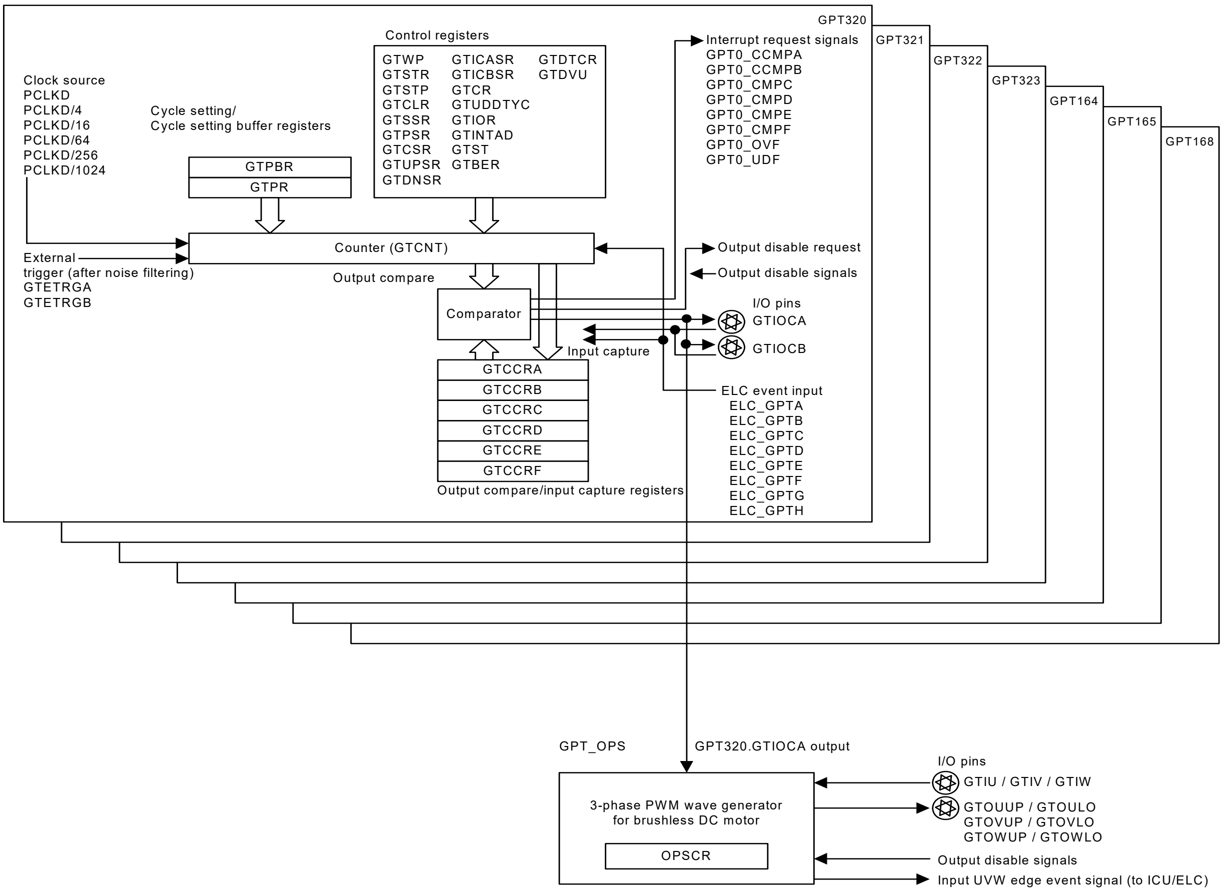
Table 23.1 GPT specifications

Parameter	Description
Functions	• 32 bits × 4 channels • 16 bits × 3 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: PCLKA • Core clock: PCLKD • Frequency ratio: PCLKA:PCLKD = 1:N (N = 1/2/4/8/16/32/64).

Table 23.2 GPT functions

Parameter		GPT32, GPT16
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
Counter clear sources		GTPR register compare match, input capture, input pin status, ELC event input, and GTETRGn (n = A, B) 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 5, 8
Event linking (ELC) function		Available
Noise filtering function		Available

Note 1. GTRETRGn connects to GPT through the POEG module. Therefore, to use the GPT function, supply the POEG clock by clearing the MSTPD14 bit.


GPT320
GPT321
GPT322
GPT323
GPT164
GPT165
GPT168
Control registers
GTWP
GTSTR
GTSTP
GTCLR
GTSSR
GTPSR
GTCSR
GTUPSR
GTDNSR
GTICASR
GTICBSR
GTCR
GTUDDTYC
GTIOR
GTINTAD
GTST
GTBER
GTDTCR
GTDVU
Interrupt request signals
GPT0_CCMPA
GPT0_CCMPB
GPT0_CMPC
GPT0_CMPD
GPT0_CMPE
GPT0_CMPF
GPT0_OVF
GPT0_UDF
Clock source
PCLKD
PCLKD/4
PCLKD/16
PCLKD/64
PCLKD/256
PCLKD/1024
Cycle setting/
Cycle setting buffer registers
GTPBR
GTPR
Counter (GTCNT)
External trigger (after noise filtering)
GTETRGA
GTETRGB
Output compare
Comparator
Input capture
Output disable request
Output disable signals
I/O pins
GTIOCA
GTIOCB
GTCCRA
GTCCRB
GTCCRC
GTCCRD
GTCCRE
GTCCRF
Output compare/input capture registers
ELC event input
ELC_GPTA
ELC_GPTB
ELC_GPTC
ELC_GPTD
ELC_GPTE
ELC_GPTF
ELC_GPTG
ELC_GPTH
GPT_OPS
GPT320.GTIOCA output
3-phase PWM wave generator for brushless DC motor
OPSCR
I/O pins
GTIU / GTIV / GTIW
GTOUUP / GTOULO
GTOVUP / GTOVLO
GTOWUP / GTOWLO
Output disable signals
Input UVW edge event signal (to ICU/ELC)

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.

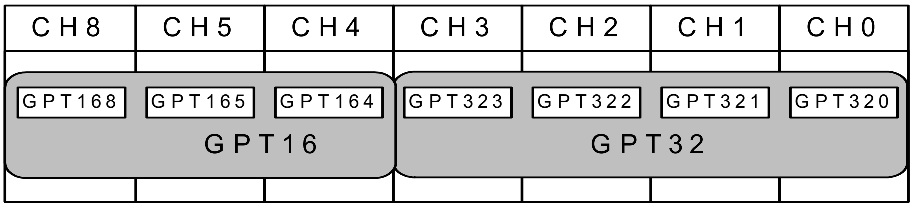

CH8
CH5
CH4
CH3
CH2
CH1
CH0
GPT168
GPT165
GPT164
GPT323
GPT322
GPT321
GPT320
GPT16
GPT32

Figure 23.2 Association between GPT channels and module names

Table 23.3 lists the I/O pins used in the GPT.

Table 23.3 GPT I/O pins

Channel	Pin name	I/O	Function
Shared	GTETRGA	Input	External trigger input pin A (after noise filtering)
	GTETRGB	Input	External trigger input pin B (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
GPT164	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
GPT165	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
GPT168	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
GPT_OPS	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U-phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U-phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V-phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V-phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W-phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W-phase)

23.2 Register Descriptions

Table 23.4 lists the registers in the GPT.

Table 23.4 GPT registers

Module symbol	Register name	Register symbol	Reset value	Address	Access size
GPT32m[*1] GPT16m[*2]	General PWM Timer Write Protection Register	GTWP	00000000h	4007 8000h + 0100h × m	32
	General PWM Timer Software Start Register	GTSTR	00000000h	4007 8004h + 0100h × m	32
	General PWM Timer Software Stop Register	GTSTP	FFFFFFFFh	4007 8008h + 0100h × m	32
	General PWM Timer Software Clear Register	GTCLR	00000000h	4007 800Ch + 0100h × m	32
	General PWM Timer Start Source Select Register	GTSSR	00000000h	4007 8010h + 0100h × m	32
	General PWM Timer Stop Source Select Register	GTPSR	00000000h	4007 8014h + 0100h × m	32
	General PWM Timer Clear Source Select Register	GTCSR	00000000h	4007 8018h + 0100h × m	32
	General PWM Timer Up Count Source Select Register	GTUPSR	00000000h	4007 801Ch + 0100h × m	32
	General PWM Timer Down Count Source Select Register	GTDNSR	00000000h	4007 8020h + 0100h × m	32
	General PWM Timer Input Capture Source Select Register A	GTICASR	00000000h	4007 8024h + 0100h × m	32
	General PWM Timer Input Capture Source Select Register B	GTICBSR	00000000h	4007 8028h + 0100h × m	32
	General PWM Timer Control Register	GTCR	00000000h	4007 802Ch + 0100h × m	32
	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	00000001h	4007 8030h + 0100h × m	32
	General PWM Timer I/O Control Register	GTIOR	00000000h	4007 8034h + 0100h × m	32
	General PWM Timer Interrupt Output Setting Register	GTINTAD	00000000h	4007 8038h + 0100h × m	32
	General PWM Timer Status Register	GTST	00008000h	4007 803Ch + 0100h × m	32
	General PWM Timer Buffer Enable Register	GTBER	00000000h	4007 8040h + 0100h × m	32
	General PWM Timer Counter	GTCNT	00000000h	4007 8048h + 0100h × m	32
	General PWM Timer Compare Capture Register A	GTCCRA	FFFFFFFFh[*3]	4007 804Ch + 0100h × m	32
	General PWM Timer Compare Capture Register B	GTCCRB	FFFFFFFFh[*3]	4007 8050h + 0100h × m	32
	General PWM Timer Compare Capture Register C	GTCCRC	FFFFFFFFh[*3]	4007 8054h + 0100h × m	32
	General PWM Timer Compare Capture Register E	GTCCRE	FFFFFFFFh[*3]	4007 8058h + 0100h × m	32
	General PWM Timer Compare Capture Register D	GTCCRD	FFFFFFFFh[*3]	4007 805Ch + 0100h × m	32
	General PWM Timer Compare Capture Register F	GTCCRF	FFFFFFFFh[*3]	4007 8060h + 0100h × m	32
	General PWM Timer Cycle Setting Register	GTPR	FFFFFFFFh[*3]	4007 8064h + 0100h × m	32
	General PWM Timer Cycle Setting Buffer Register	GTPBR	FFFFFFFFh[*3]	4007 8068h + 0100h × m	32
	General PWM Timer Dead Time Control Register	GTDTCR	00000000h	4007 8088h + 0100h × m	32
	General PWM Timer Dead Time Value Register U	GTDVU	FFFFFFFFh[*3]	4007 808Ch + 0100h × m	32
GPT_OPS	Output Phase Switching Control Register	OPSCR	00000000h	4007 8FF0h	32

Note 1. GPT32m (m = 0 to 3)
Note 2. GPT16m (m = 4, 5, 8)
Note 3. The reset value of GPT16m is 0000FFFFh.

23.2.1 General PWM Timer Write-Protection Register (GTWP)

Address(es): GPT32m.GTWP 4007 8000h + 0100h × m (m = 0 to 3),
GPT16m.GTWP 4007 8000h + 0100h × m (m = 4, 5, 8)

	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, writing to the WP bits 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

To prevent accidental modification, the GTWP enables or disables writing to registers. The following is a list of write enabled or disabled registers:

GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, 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 3),
GPT16m.GTSTR 4007 8004h + 0100h × m (m = 4, 5, 8)

	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
	—	—	—	—	—	—	—	CSTRT 8	—	—	CSTRT 5	CSTRT 4	CSTRT 3	CSTRT 2	CSTRT 1	CSTRT 0
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, where n = 0 to 5, 8.

The GTSTR bit number represents the channel number. The GTSTR register is shared by all of the channels. The GTCNT counter starts for the channel associated with the GTSTR bit number where 1 is written. Writing 0 has no effect on the status of GTCNT counter and the value of GTSTR register.

For the association between GTSTR bit number and a channel number, see Figure 23.2.

CSTRT[9:0] bits (Channel n GTCNT Count Start) (n = 0 to 5, 8)

The CSTRT[9:0] bits start channel n of the GTCNT counter operation. Writing to the GTSTR.CSTRTn bit (n = 0 to 5, 8) has no effect unless GPTm.GTSSR.CSTRTn bit is set to 1 (m = 320 to 323, 164, 165, 168).

The read data shows the counter status of each channel (GTCR.CST bit). Zero means the counter is stopped 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 3),
GPT16m.GTSTP 4007 8008h + 0100h × m (m = 4, 5, 8)

	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 8	—	—	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, where n = 0 to 5, 8.

The GTSTP bit number represents the channel number. Each channel of the GTSTP register is shared by all the channels. The GTCNT counter stops for the channel associated with the GTSTP bit number where 1 is written. Writing 0 has no effect on the status of the GTCNT counter and the value of GTSTP register.

For the association between the GTSTP bit number and a channel number, see Figure 23.2.

CSTOP[9:0] bits (Channel n GTCNT Count Stop) (n = 0 to 5, 8)

The CSTOP[9:0] bits stop channel n of the GTCNT counter operation. Writing to the GTSTP.CSTOPn bit (n = 0 to 5, 8) has no effect unless the GPT32m.GTPSR.CSTOPn bit is set to 1 (m = 320 to 323, 164, 165, 168). The 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 3),
GPT16m.GTCLR 4007 800Ch + 0100h × m (m = 4, 5, 8)

	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 8	—	—	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

The GTCLR is a write-only register that clears the GTCNT counter operation for each channel n, where n = 0 to 5, 8.

The GTCLR bit number represents the channel number. Each channel of the GTCLR register is shared by all the channels. The GTCNT counter is cleared for the channel associated with the GTCLR bit number where 1 is written. Writing 0 has no effect on the status of the GTCNT counter.

For the association between the GTCLR bit number and a channel number, see Figure 23.2.

CCLR[9:0] bits (Channel n GTCNT Count Clear) (n = 0 to 5, 8)

Channel n of the GTCNT counter value is cleared on writing 1 to the CCLR[9:0] 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 3),
GPT16m.GTSSR 4007 8010h + 0100h × m (m = 4, 5, 8)

	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	—	—	—	—	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 on the falling edge of GTETRGA input 1: Counter start enabled on 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 on the falling edge of GTETRGB input 1: Counter start enabled on the falling edge of GTETRGB input.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	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 on the falling edge of GTIOCA input when GTIOCB input is 0 1: Counter start enabled on the falling edge of GTIOCA input when GTIOCB input is 0.	R/W
b11	SSCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Start Enable	0: Counter start disabled on the falling edge of GTIOCA input when GTIOCB input is 1 1: Counter start enabled on 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

Bit	Symbol	Bit name	Description	R/W
b14	SSCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Start Enable	0: Counter start disabled on the falling edge of GTIOCB input when GTIOCA input is 0 1: Counter start enabled on 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 on the falling edge of GTIOCB input when GTIOCA input is 1 1: Counter start enabled on 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 input 1: Counter start enabled at the ELC_GPTA input.	R/W
b17	SSELCB	ELC_GPTB Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTB input 1: Counter start enabled at the ELC_GPTB input.	R/W
b18	SSELCC	ELC_GPTC Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTC input 1: Counter start enabled at the ELC_GPTC input.	R/W
b19	SSELCD	ELC_GPTD Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTD input 1: Counter start enabled at the ELC_GPTD input.	R/W
b20	SSELCE	ELC_GPTE Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTE input 1: Counter start enabled at the ELC_GPTE input.	R/W
b21	SSELCF	ELC_GPTF Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTF input 1: Counter start enabled at the ELC_GPTF input.	R/W
b22	SSELCG	ELC_GPTG Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTG input 1: Counter start enabled at the ELC_GPTG input.	R/W
b23	SSELCH	ELC_GPTH Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTH input 1: Counter start enabled at the ELC_GPTH input.	R/W
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

The GTSSR sets the source to start the GTCNT counter.

SSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Start Enable)

The SSGTRGAR bit enables or disables the GTCNT counter start on the rising edge of GTETRGA pin input.

SSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Start Enable)

The SSGTRGAF bit enables or disables the GTCNT counter start on the falling edge of GTETRGA pin input.

SSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Start Enable)

The SSGTRGBR bit enables or disables the GTCNT counter start on the rising edge of GTETRGB pin input.

SSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Start Enable)

The SSGTRGBF bit enables or disables the GTCNT counter start on the falling edge of GTETRGB pin input.

SSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Start Enable)

The SSCARBL bit enables or disables the 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)

The SSCARBH bit enables or disables the 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)

The SSCAFBL bit enables or disables the GTCNT counter start on 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)

The SSCAFBH bit enables or disables the GTCNT counter start on 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)

The SSCBRAL bit enables or disables the 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)

The SSCBRAH bit enables or disables the 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)

The SSCBFAL bit enables or disables the GTCNT counter start on 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)

The SSCBFAH bit enables or disables the GTCNT counter start on 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)

The SSELCm bit enables or disables the GTCNT counter start at the ELC_GPTm event input.

CSTRT bit (Software Source Counter Start Enable)

The CSTRT bit enables or disables the GTCNT counter start by GTSTR register.

23.2.6 General PWM Timer Stop Source Select Register (GTPSR)

Address(es): GPT32m.GTPSR 4007 8014h + 0100h × m (m = 0 to 3),
GPT16m.GTPSR 4007 8014h + 0100h × m (m = 4, 5, 8)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CSTOP	—	—	—	—	—	—	—	PSELC H	PSELC G	PSELC F	PSELC E	PSELC D	PSELC C	PSELC B	PSELC A
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
	PSCBF AH	PSCBF AL	PSCBR AH	PSCBR AL	PSCAF BH	PSCAF BL	PSCAR BH	PSCAR BL	—	—	—	—	PSGTR GBF	PSGTR GBR	PSGTR GAF	PSGTR GAR
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 on the falling edge of GTETRGA input 1: Counter stop enabled on 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

Bit	Symbol	Bit name	Description	R/W
b3	PSGTRGBF	GTETRGB Pin Falling Input Source Counter Stop Enable	0: Counter stop disabled on the falling edge of GTETRGB input 1: Counter stop enabled on the falling edge of GTETRGB input.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	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 on the falling edge of GTIOCA input when GTIOCB input is 0 1: Counter stop enabled on the falling edge of GTIOCA input when GTIOCB input is 0.	R/W
b11	PSCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Stop Enable	0: Counter stop disabled on the falling edge of GTIOCA input when GTIOCB input is 1 1: Counter stop enabled on 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 on the falling edge of GTIOCB input when GTIOCA input is 0 1: Counter stop enabled on 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 on the falling edge of GTIOCB input when GTIOCA input is 1 1: Counter stop enabled on 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 input 1: Counter stop enabled at the ELC_GPTA input.	R/W
b17	PSELCB	ELC_GPTB Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTB input 1: Counter stop enabled at the ELC_GPTB input.	R/W
b18	PSELCC	ELC_GPTC Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTC input 1: Counter stop enabled at the ELC_GPTC input.	R/W
b19	PSELCD	ELC_GPTD Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTD input 1: Counter stop enabled at the ELC_GPTD input.	R/W
b20	PSELCE	ELC_GPTE Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTE input 1: Counter stop enabled at the ELC_GPTE input.	R/W
b21	PSELCF	ELC_GPTF Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTF input 1: Counter stop enabled at the ELC_GPTF input.	R/W
b22	PSELCG	ELC_GPTG Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTG input 1: Counter stop enabled at the ELC_GPTG input.	R/W
b23	PSELCH	ELC_GPTH Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTH input 1: Counter stop enabled at the ELC_GPTH input.	R/W
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

The GTPSR sets the source to stop the GTCNT counter.

PSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Stop Enable)

The PSGTRGAR bit enables or disables the GTCNT counter stop on the rising edge of GTETRGA pin input.

PSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Stop Enable)

The PSGTRGAF bit enables or disables the GTCNT counter stop on the falling edge of GTETRGA pin input.

PSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Stop Enable)

The PSGTRGBR bit enables or disables the GTCNT counter stop on the rising edge of GTETRGB pin input.

PSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Stop Enable)

The PSGTRGBF bit enables or disables the GTCNT counter stop on the falling edge of GTETRGB pin input.

PSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Stop Enable)

The PSCARBL bit enables or disables the 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)

The PSCARBH bit enables or disables the 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)

The PSCAFBL bit enables or disables the GTCNT counter stop on 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)

The PSCAFBH bit enables or disables the GTCNT counter stop on 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)

The PSCBRAL bit enables or disables the 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)

The PSCBRAH bit enables or disables the 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)

The PSCBFAL bit enables or disables the GTCNT counter stop on 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)

The PSCBFAH bit enables or disables the GTCNT counter stop on 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)

The PSELCm bit enables or disables the GTCNT counter stop at the ELC_GPTm event input.

CSTOP bit (Software Source Counter Stop Enable)

The CSTOP bit enables or disables the GTCNT counter stop by GTSTP register.

23.2.7 General PWM Timer Clear Source Select Register (GTCSR)

Address(es): GPT32m.GTCSR 4007 8018h + 0100h × m (m = 0 to 3),
GPT16m.GTCSR 4007 8018h + 0100h × m (m = 4, 5, 8)

	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	—	—	—	—	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 on the falling edge of GTETRGA input 1: Counter clear enabled on 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 on the falling edge of GTETRGB input 1: Counter clear enabled on the falling edge of GTETRGB input.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	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 on the falling edge of GTIOCA input when GTIOCB input is 0 1: Counter clear enabled on the falling edge of GTIOCA input when GTIOCB input is 0.	R/W
b11	CSCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Clear Enable	0: Counter clear disabled on the falling edge of GTIOCA input when GTIOCB input is 1 1: Counter clear enabled on 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

Bit	Symbol	Bit name	Description	R/W
b14	CSCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Clear Enable	0: Counter clear disabled on the falling edge of GTIOCB input when GTIOCA input is 0 1: Counter clear enabled on 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 on the falling edge of GTIOCB input when GTIOCA input is 1 1: Counter clear enabled on 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 input 1: Counter clear enabled at the ELC_GPTA input.	R/W
b17	CSELCB	ELC_GPTB Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTB input 1: Counter clear enabled at the ELC_GPTB input.	R/W
b18	CSELCC	ELC_GPTC Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTC input 1: Counter clear enabled at the ELC_GPTC input.	R/W
b19	CSELCD	ELC_GPTD Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTD input 1: Counter clear enabled at the ELC_GPTD input.	R/W
b20	CSELCE	ELC_GPTE Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTE input 1: Counter clear enabled at the ELC_GPTE input.	R/W
b21	CSELCF	ELC_GPTF Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTF input 1: Counter clear enabled at the ELC_GPTF input.	R/W
b22	CSELCG	ELC_GPTG Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTG input 1: Counter clear enabled at the ELC_GPTG input.	R/W
b23	CSELCH	ELC_GPTH Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTH input 1: Counter clear enabled at the ELC_GPTH input.	R/W
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

The GTCSR sets the source to clear the GTCNT counter.

CSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Clear Enable)

The CSGTRGAR bit enables or disables the GTCNT counter clear on the rising edge of GTETRGA pin input.

CSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Clear Enable)

The CSGTRGAF bit enables or disables the GTCNT counter clear on the falling edge of GTETRGA pin input.

CSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Clear Enable)

The CSGTRGBR bit enables or disables the GTCNT counter clear on the rising edge of GTETRGB pin input.

CSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Clear Enable)

The CSGTRGBF bit enables or disables the GTCNT counter clear on the falling edge of GTETRGB pin input.

CSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Clear Enable)

The CSCARBL bit enables or disables the 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)

The CSCARBH bit enables or disables the 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)

The CSCAFBL bit enables or disables the GTCNT counter clear on 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)

The CSCAFBH bit enables or disables GTCNT counter clear on 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)

The CSCBRAL 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)

The CSCBRAH 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)

The CSCBFAL bit enables or disables GTCNT counter clear on 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)

The CSCBFAH bit enables or disables GTCNT counter clear on 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)

The CSELCm bit enables or disables GTCNT counter clear at the ELC_GPTm event input.

CCLR bit (Software Source Counter Clear Enable)

The CCLR 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 3),
GPT16m.GTUPSR 4007 801Ch + 0100h × m (m = 4, 5, 8)

	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	—	—	—	—	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 on the falling edge of GTETRGA input 1: Counter count up enabled on 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

Bit	Symbol	Bit name	Description	R/W
b3	USGTRGBF	GTETRGB Pin Falling Input Source Counter Count Up Enable	0: Counter count up disabled on the falling edge of GTETRGB input 1: Counter count up enabled on the falling edge of GTETRGB input.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	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 on the falling edge of GTIOCA input when GTIOCB input is 0 1: Counter count up enabled on the falling edge of GTIOCA input when GTIOCB input is 0.	R/W
b11	USCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Count Up Enable	0: Counter count up disabled on the falling edge of GTIOCA input when GTIOCB input is 1 1: Counter count up enabled on 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 on the falling edge of GTIOCB input when GTIOCA input is 0 1: Counter count up enabled on 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 on the falling edge of GTIOCB input when GTIOCA input is 1 1: Counter count up enabled on 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 input 1: Counter count up enabled at the ELC_GPTA input.	R/W
b17	USELCB	ELC_GPTB Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTB input 1: Counter count up enabled at the ELC_GPTB input.	R/W
b18	USELCC	ELC_GPTC Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTC input 1: Counter count up enabled at the ELC_GPTC input.	R/W
b19	USELCD	ELC_GPTD Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTD input 1: Counter count up enabled at the ELC_GPTD input.	R/W
b20	USELCE	ELC_GPTE Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTE input 1: Counter count up enabled at the ELC_GPTE input.	R/W
b21	USELCF	ELC_GPTF Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTF input 1: Counter count up is enabled at the ELC_GPTF input	R/W
b22	USELCG	ELC_GPTG Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTG input 1: Counter count up is enabled at the ELC_GPTG input	R/W
b23	USELCH	ELC_GPTH Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTH input 1: Counter count up enabled at the ELC_GPTH input	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTUPSR sets the source to count up the GTCNT counter.

When at least one bit in the GTUPSR register is set to 1, the GTCNT counter is counted up by the source that is set to 1 in this register. In this case, GTCR.TPCS has no effect.

USGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Up Enable)

The USGTRGAR bit enables or disables the GTCNT counter count up on the rising edge of GTETRGA pin input.

USGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Up Enable)

The USGTRGAF bit enables or disables the GTCNT counter count up on the falling edge of GTETRGA pin input.

USGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Up Enable)

The USGTRGBR bit enables or disables the GTCNT counter count up on the rising edge of GTETRGB pin input.

USGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Up Enable)

The USGTRGBF bit enables or disables the GTCNT counter count up on the falling edge of GTETRGB pin input.

USCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Count Up Enable)

The USCARBL bit enables or disables the 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)

The USCARBH bit enables or disables the 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)

The USCAFBL bit enables or disables the GTCNT counter count up on 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)

The USCAFBH bit enables or disables the GTCNT counter count up on 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)

The USCBRAL bit enables or disables the 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)

The USCBRAH bit enables or disables the 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)

The USCBFAL bit enables or disables the GTCNT counter count up on 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)

The USCBFAH bit enables or disables the GTCNT counter count up on 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)

The USELCm bit enables or disables the 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 3),
GPT16m.GTDNSR 4007 8020h + 0100h × m (m = 4, 5, 8)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DSELC H	DSELC G	DSELC F	DSELC E	DSELC D	DSELC C	DSELC B	DSELC A
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
	DSCBF AH	DSCBF AL	DSCBR AH	DSCBR AL	DSCAF BH	DSCAF BL	DSCAR BH	DSCAR BL	—	—	—	—	DSGTR GBF	DSGTR GBR	DSGTR GAF	DSGTR GAR
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 on the falling edge of GTETRGA input 1: Counter count down enabled on 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 on the falling edge of GTETRGB input 1: Counter count down enabled on the falling edge of GTETRGB input.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	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 on the falling edge of GTIOCA input when GTIOCB input is 0 1: Counter count down enabled on the falling edge of GTIOCA input when GTIOCB input is 0.	R/W
b11	DSCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Count Down Enable	0: Counter count down disabled on the falling edge of GTIOCA input when GTIOCB input is 1 1: Counter count down enabled on 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

Bit	Symbol	Bit name	Description	R/W
b14	DSCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Count Down Enable	0: Counter count down disabled on the falling edge of GTIOCB input when GTIOCA input is 0 1: Counter count down enabled on 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 on the falling edge of GTIOCB input when GTIOCA input is 1 1: Counter count down enabled on 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 input 1: Counter count down enabled at the ELC_GPTA input.	R/W
b17	DSELCB	ELC_GPTB Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTB input 1: Counter count down enabled at the ELC_GPTB input.	R/W
b18	DSELCC	ELC_GPTC Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTC input 1: Counter count down enabled at the ELC_GPTC input.	R/W
b19	DSELCD	ELC_GPTD Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTD input 1: Counter count down enabled at the ELC_GPTD input.	R/W
b20	DSELCE	ELC_GPTE Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTE input 1: Counter count down enabled at the ELC_GPTE input.	R/W
b21	DSELCF	ELC_GPTF Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTF input 1: Counter count down enabled at the ELC_GPTF input.	R/W
b22	DSELCG	ELC_GPTG Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTG input 1: Counter count down enabled at the ELC_GPTG input.	R/W
b23	DSELCH	ELC_GPTH Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTH input 1: Counter count down enabled at the ELC_GPTH input.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTDNSR sets the source to count down the GTCNT counter.

When at least one bit in the GTDNSR register is set to 1, the GTCNT counter is counted down by the source that is set to 1 in this register, but the GTCNT counter set by GTCR.TPCS does not perform the count.

DSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Down Enable)

The DSGTRGAR bit enables or disables the GTCNT counter count down on the rising edge of GTETRGA pin input.

DSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Down Enable)

The DSGTRGAF bit enables or disables the GTCNT counter count down on the falling edge of GTETRGA pin input.

DSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Down Enable)

The DSGTRGBR bit enables or disables the GTCNT counter count down on the rising edge of GTETRGB pin input.

DSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Down Enable)

The DSGTRGBF bit enables or disables the GTCNT counter count down on the falling edge of GTETRGB pin input.

DSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Count Down Enable)

The DSCARBL bit enables or disables the 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)

The DSCARBH bit enables or disables the 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)

The DSCAFBL bit enables or disables the GTCNT counter count down on 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)

The DSCAFBH bit enables or disables the GTCNT counter count down on 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)

The DSCBRAL bit enables or disables the 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)

The DSCBRAH bit enables or disables the 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)

The DSCBFAL bit enables or disables the GTCNT counter count down on 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)

The DSCBFAH bit enables or disables the GTCNT counter count down on 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)

The DSELCm bit enables or disables the 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 3),
GPT16m.GTICASR 4007 8024h + 0100h × m (m = 4, 5, 8)

	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	—	—	—	—	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 on the falling edge of GTETRGA input 1: GTCCRA input capture enabled on the falling edge of GTETRGA input.	R/W

Bit	Symbol	Bit name	Description	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 on the falling edge of GTETRGB input 1: GTCCRA input capture enabled on the falling edge of GTETRGB input.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	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 on the falling edge of GTIOCA input when GTIOCB input is 0 1: GTCCRA input capture enabled on the falling edge of GTIOCA input when GTIOCB input is 0.	R/W
b11	ASCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled on the falling edge of GTIOCA input when GTIOCB input is 1 1: GTCCRA input capture enabled on 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 on the falling edge of GTIOCB input when GTIOCA input is 0 1: GTCCRA input capture enabled on 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 on the falling edge of GTIOCB input when GTIOCA input is 1 1: GTCCRA input capture enabled on 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 input 1: GTCCRA input capture enabled at the ELC_GPTA input.	R/W
b17	ASELCB	ELC_GPTB Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTB input 1: GTCCRA input capture enabled at the ELC_GPTB input.	R/W
b18	ASELCC	ELC_GPTC Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTC input 1: GTCCRA input capture enabled at the ELC_GPTC input.	R/W
b19	ASELCD	ELC_GPTD Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTD input 1: GTCCRA input capture enabled at the ELC_GPTD input.	R/W
b20	ASELCE	ELC_GPTE Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTE input 1: GTCCRA input capture enabled at the ELC_GPTE input.	R/W
b21	ASELCF	ELC_GPTF Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTF input 1: GTCCRA input capture enabled at the ELC_GPTF input.	R/W
b22	ASELCG	ELC_GPTG Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTG input 1: GTCCRA input capture enabled at the ELC_GPTG input.	R/W
b23	ASELCH	ELC_GPTH Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTH input 1: GTCCRA input capture enabled at the ELC_GPTH input.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTICASR sets the source of input capture for GTCCRA.

ASGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGAR 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)

The ASGTRGAF bit enables or disables input capture for GTCCRA on the falling edge of GTETRGA pin input.

ASGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGBR 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)

The ASGTRGBF bit enables or disables input capture for GTCCRA on the falling edge of GTETRGB pin input.

ASCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source GTCCRA Input Capture Enable)

The ASCARBL 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)

The ASCARBH 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)

The ASCAFBL bit enables or disables input capture for GTCCRA on 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)

The ASCAFBH bit enables or disables input capture for GTCCRA on 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)

The ASCBRAL 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)

The ASCBRAH 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)

The ASCBFAL bit enables or disables input capture for GTCCRA on 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)

The ASCBFAH bit enables or disables input capture for GTCCRA on 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)

The ASELCm 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 3),
GPT16m.GTICBSR 4007 8028h + 0100h × m (m = 4, 5, 8)

	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	—	—	—	—	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 on the falling edge of GTETRGA input 1: GTCCRB input capture enabled on 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 on the falling edge of GTETRGB input 1: GTCCRB input capture enabled on the falling edge of GTETRGB input.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	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 on the falling edge of GTIOCA input when GTIOCB input is 0 1: GTCCRB input capture enabled on the falling edge of GTIOCA input when GTIOCB input is 0.	R/W
b11	BSCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled on the falling edge of GTIOCA input when GTIOCB input is 1 1: GTCCRB input capture enabled on 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

Bit	Symbol	Bit name	Description	R/W
b14	BSCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled on the falling edge of GTIOCB input when GTIOCA input is 0 1: GTCCRB input capture enabled on 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 on the falling edge of GTIOCB input when GTIOCA input is 1 1: GTCCRB input capture enabled on 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 input 1: GTCCRB input capture enabled at the ELC_GPTA input.	R/W
b17	BSELCB	ELC_GPTB Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTB input 1: GTCCRB input capture enabled at the ELC_GPTB input.	R/W
b18	BSELCC	ELC_GPTC Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTC input 1: GTCCRB input capture enabled at the ELC_GPTC input.	R/W
b19	BSELCD	ELC_GPTD Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTD input 1: GTCCRB input capture enabled at the ELC_GPTD input.	R/W
b20	BSELCE	ELC_GPTE Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTE input 1: GTCCRB input capture enabled at the ELC_GPTE input.	R/W
b21	BSELCF	ELC_GPTF Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTF input 1: GTCCRB input capture enabled at the ELC_GPTF input.	R/W
b22	BSELCG	ELC_GPTG Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTG input 1: GTCCRB input capture enabled at the ELC_GPTG input.	R/W
b23	BSELCH	ELC_GPTH Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTH input 1: GTCCRB input capture enabled at the ELC_GPTH input.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTICBSR sets the source of input capture for GTCCRB.

BSGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGAR 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)

The BSGTRGAF bit enables or disables input capture for GTCCRB on the falling edge of GTETRGA pin input.

BSGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGBR 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)

The BSGTRGBF bit enables or disables input capture for GTCCRB on the falling edge of GTETRGB pin input.

BSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source GTCCRB Input Capture Enable)

The BSCARBL 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)

The BSCARBH 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)

The BSCAFBL bit enables or disables input capture for GTCCRB on 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)

The BSCAFBH bit enables or disables input capture for GTCCRB on 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)

The BSCBRAL 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)

The BSCBRAH 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)

The BSCBFAL bit enables or disables input capture for GTCCRB on 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)

The BSCBFAH bit enables or disables input capture for GTCCRB on 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)

The BSELCm 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 3),
GPT16m.GTCR 4007 802Ch + 0100h × m (m = 4, 5, 8)

	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

Bit	Symbol	Bit name	Description	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

The GTCR controls the GTCNT.

CST bit (Count Start)

The CST bit controls the GTCNT counter start and stop.

[Setting conditions]

- The GTSTR value where the channel number associated with the bit number is set to 1 with the GTSSR.CSTRT bit at 1
- The ELC event input or the GTIOCA/GTIOCB/GTETRGn port input enabled by GTSSR as the counter start source, occurs
- 1 is written by software directly.

[Clearing conditions]

- The GTSTP value where the channel number associated with the bit number is set to 1 with the GTSSR.CSTOP bit at 1
- The ELC event input or the GTIOCA/GTIOCB/GTETRGn port input enabled by GTSSR as the counter stop source, occurs
- 0 is written by software directly.

MD[2:0] bits (Mode Select)

The MD[2:0] 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)

The TPCS[2:0] bits select the clock for the GTCNT. A clock prescaler can be selected independently for each channel. The TPCS[2:0] bits must be set while the 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 3),
GPT16m.GTUDDTYC 4007 8030h + 0100h × m (m = 4, 5, 8)

	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% or 100% Duty Setting	0: Apply output value set in 0% or 100% duty to GTIOA[3:2] function after releasing 0% or 100% duty setting 1: Apply masked compare match output value to GTIOA[3:2] function after releasing 0% or 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% or 100% Duty Setting	0: Apply output value set in 0% or 100% duty to GTIOB[3:2] function after releasing 0% or 100% duty setting 1: Apply masked compare match output value to GTIOB[3:2] function after releasing 0% or 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

The GTUDDTYC sets the direction in which GTCNT counts (up-counting or down-counting), and sets the duty of the 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 the GTCNT value becomes the GTPR value). When the UD value is set to 1 during down-counting, the count direction changes at an underflow (the timing synchronous with count clock after 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 changes while the UDF bit is 0 and counting stops, the change is not reflected in the count direction when counting starts.

When the UDF bit is set to 1 while counting is stopped, the UD value is reflected in the count direction when counting starts.

UD bit (Count Direction Setting)

The UD bit sets the count direction (up-counting or down-counting) for GTCNT.

UDF bit (Forcible Count Direction Setting)

The UDF bit forcibly sets the count direction when GTCNT starts operation as the UD value. Only 0 should be written to this bit during counter operation. When 1 is written to this bit while counting stops, this bit must 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. 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)

The OmDTY[1:0] bits set the output duty (0%, 100% or compare match control) of the GTIOCm pin.

OmDTYF bit (ForcibleGTIOCm Output Duty Setting) (m = A, B)

The OmDTYF bit forcibly sets the output duty cycle to the OmDTY setting. This bit must be set to 0 during counter operation. When this bit is set to 1 while counting stops, this bit must be returned to 0 until the first period ends after the counter starts.

OmDTYR bit (GTIOCm Output Value Selecting after Releasing 0% or 100% Duty Setting) (m = A, B)

The OmDTYR bit selects the value that is the object of output retained or toggled at cycle end, when the control changes from 0% or 100% duty setting to compare match for GTIOCm pin and GTIOR. The GTIOm[3:2] bits are set to 00b (output retained at cycle end) or the GTIOR.GTIOm[3:2] bits are set to 11b (output toggled at cycle end).

While the duty 0%/100% setting operation is running, compare match operation is continued inside the GPT32. When the OmDTYR bit is set to 1, the GTIOCm pin is in the output state selected by the GTIOR.GTIOm [3:2] bit at the end of the cycle in this compare match operation.

23.2.14 General PWM Timer I/O Control Register (GTIOR)

Address(es): GPT32m.GTIOR 4007 8034h + 0100h × m (m = 0 to 3),
GPT16m.GTIOR 4007 8034h + 0100h × m (m = 4, 5, 8)

	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 or stop of counting depends on the register setting 1: The GTIOCA pin output level is retained at the start or 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 on output disable 1 0: GTIOCA pin is set to 0 on output disable 1 1: GTIOCA pin is set to 1 on output disable.	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 on output disable 1 0: GTIOCB pin is set to 0 on output disable 1 1: GTIOCB pin is set to 1 on output disable.	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

Bit	Symbol	Bit name	Description	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

The GTIOR sets the functions of the GTIOCA and GTIOCB pins.

GTIOA[4:0] bits (GTIOCA Pin Function Select)

The GTIOA[4:0] bits select the GTIOCA pin function. For details, see Table 23.5.

OADFLT bit (GTIOCA Pin Output Value Setting at the Count Stop)

The OADFLT bit sets whether the GTIOCA pin outputs high or low when counting stops.

OAHLD bit (GTIOCA Pin Output Setting at the Start/Stop Count)

The OAHLD 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 in bit [4] of the GTIOA[4:0] bits is output when counting starts
- The value specified in the OADFLT bit is output when counting stops
- If the OADFLT bit is modified while counting stops, the new value is immediately reflected in the output.

When the OAHLD bit is set to 1:

- The output is retained when counting starts or stops.

OAE bit (GTIOCA Pin Output Enable)

The OAE 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 regardless of the OAE bit value.

OADF[1:0] bits (GTIOCA Pin Disable Value Setting)

The OADF[1:0] bits select the output value of GTIOCA pin when an output disable request occurs.

NFAEN bit (Noise Filter A Enable)

The NFAEN 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)

The NFCSA[1:0] 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)

The GTIOB[4:0] bits select the GTIOCB pin function. For details, see Table 23.5.

OBDFLT bit (GTIOCB Pin Output Value Setting at the Count Stop)

The OBDFLT bit sets whether the GTIOCB pin outputs high or low when counting stops.

OBHLD bit (GTIOCB Pin Output Setting at the Start/Stop Count)

The OBHLD 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 in bit [4] of the GTIOB[4:0] bits is output when counting starts
- The value specified in the OBDFLT bit is output when counting stops
- If the OBDFLT bit is modified while counting stops, the new value is immediately reflected in the output.

When the OBHLD bit is set to 1:

- The output is retained when counting starts or stops.

OBE bit (GTIOCB Pin Output Enable)

The OBE 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)

The OBDF[1:0] bits select the output value of GTIOCB pin when an output disable request occurs.

NFBEN bit (Noise Filter B Enable)

The NFBEN 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)

The NFCSB[1:0] 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*1,*2,*3	b1, b0*2
0	0	0	0	0	Initial output is low	Output retained at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	0	1	0			High output at GTCCRA/GTCCRB compare match
0	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	0	1	0	0		Low output at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	1	1	0			High output at GTCCRA/GTCCRB compare match
0	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	0	0	0		High output at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	0	1	0			High output at GTCCRA/GTCCRB compare match
0	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	1	0	0		Output toggled at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	1	1	0			High output at GTCCRA/GTCCRB compare match
0	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Table 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*1,*2,*3**	**b1, b0*2**
1	0	0	0	0	Initial output is high	Output retained at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	0	1	0			High output at GTCCRA/GTCCRB compare match
1	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	1	0	0		Low output at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	1	1	0			High output at GTCCRA/GTCCRB compare match
1	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	0	0	0		High output at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	0	1	0			High output at GTCCRA/GTCCRB compare match
1	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	1	0	0		Output toggled at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	1	1	0			High output at GTCCRA/GTCCRB compare match
1	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Note 1. The cycle end means an overflow (GTCNT changes from GTPR to 0 in up-counting) or underflow (GTCNT changes 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 one 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 3),
GPT16m.GTINTAD 4007 8038h + 0100h × m (m = 4, 5, 8)

	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 x: Setting prohibited.	R/W
b28 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	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

The 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 by GRP[1:0] bits when same time output level high or same time output level low occurs based on the output disable request enable bit.

GTST.ODF shows the request of the output disable source group that is selected with the GRP[1:0] bits. GRP[1:0] bits should be set when both GTIOR.OAE and GTIOR.OBE bits are 0.

GRPABH bit (Same Time Output Level High Disable Request Enable)

The GRPABH bit enables or disables the output disable request when the GTIOCA and GTIOCB pins output 1 at the same time.

GRPABL bit (Same Time Output Level Low Disable Request Enable)

The GRPABL bit enables or disables the output disable request when the GTIOCA and GTIOCB pins 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 3),
GPT16m.GTST 4007 803Ch + 0100h × m (m = 4, 5, 8)

	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

x: Undefined

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

Bit	Symbol	Bit name	Description	R/W
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) occurred 1: An overflow (crest) occurred.	R/(W)*1
b7	TCFPU	Underflow Flag	0: No underflow (trough) occurred 1: An underflow (trough) 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: GTCNT counter counts downward 1: 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 and GTIOCB pins do not output 1 at the same time 1: GTIOCA and GTIOCB pins output 1 at the same time.	R
b30	OABLF	Same Time Output Level Low Flag	0: GTIOCA and GTIOCB pins do not output 0 at the same time 1: GTIOCA and GTIOCB pins 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.

The GTST indicates the status of the GPT.

TCFA flag (Input Capture/Compare Match Flag A)

TCFA 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 flag.

TCFB flag (Input Capture/Compare Match Flag B)

TCFB 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 flag.

TCFC flag (Input Compare Match Flag C)

TCFC is the status flag for the compare match of GTCCRC.

[Setting condition]

- GTCNT = GTCCRC.

[Clearing condition]

- 0 is written to this flag.

[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)

TCFD is the status flag for the compare match of GTCCRD.

[Setting condition]

- GTCNT = GTCCRD.

[Clearing condition]

- 0 is written to this flag.

[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)

The TCFE is the status flag for the compare match of GTCCRE.

[Setting condition]

- GTCNT = GTCCRE.

[Clearing condition]

- 0 is written to this flag.

[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)

TCFF is the status flag for the compare match of GTCCRF.

[Setting condition]

- GTCNT = GTCCRF.

[Clearing condition]

- 0 is written to this flag.

[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)

The TCFPO flag 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-counting) has occurred
- In triangle-wave mode, a crest (GTCNT changes from GTPR to GTPR-1) has occurred

- In counting by hardware sources, an overflow (GTCNT changes from GTPR to 0 in up-counting) has occurred.

[Clearing condition]

- 0 is written to this flag.

TCFPU flag (Underflow Flag)

The TCFPU flag indicates when an underflow or 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 trough (GTCNT changes from 0 to 1) has occurred
- In counting by hardware sources, an underflow (GTCNT changes from 0 to GTPR in down-counting) has occurred.

[Clearing condition]

- 0 is written to this flag.

TUCF flag (Count Direction Flag)

The TUCF flag indicates the count direction of GTCNT. In event count operation, this flag is set to 1 in up-counting and is set to 0 in down-counting.

ODF flag (Output Disable Flag)

The ODF flag shows the request of the output disable source group that is selected in the GRP[1:0] bits.

When output is disabled, an output disable control is not released within the same cycle in which an output disable request is negated. It is released in the next cycle.

OABHF flag (Same Time Output Level High Flag)

The OABHF flag indicates that the GTIOCA and GTIOCB pins output 1 at the same time.

When GTIOCA or GTIOCB pin outputs 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 the POEG as the output disable request.

[Setting condition]

- The GTIOCA and GTIOCB pins output 1 at the same time when both OAE and OBE bits are set to 1.

[Clearing conditions]

- The GTIOCA pin output value is different from the GTIOCB pin output value when both OAE and OBE bits are set to 1
- The GTIOCA and GTIOCB pins output 0 at the same time when both OAE and OBE bits are set to 1
- Either the OAE bit or OBE bit is set to 0.

OABLF flag (Same Time Output Level Low Flag)

The OABLF flag indicates that the GTIOCA and GTIOCB pins output 0 at the same time.

When the GTIOCA pin or GTIOCB pin outputs 1, this flag is returned to 0. This flag is read only. Writing 0 to clear the flag is prohibited. When an interrupt by the OABLF flag is enabled (GTINTAD.GRPABL = 1), the OABLF flag is output to the POEG as the output disable request.

[Setting condition]

- The GTIOCA and GTIOCB pins output 0 at the same time when both the OAE and OBE bits are set to 1.

[Clearing conditions]

- The GTIOCA pin output value is different from the GTIOCB pin output value when both OAE and OBE bits are set to 1

- The GTIOCA and GTIOCB pins output 1 at the same time when both OAE and OBE bits are set to 1
- Either the OAE bit or the OBE bit is set to 0.

The compare-target signals to generate the OABHF/OABLF flag are the compare match outputs (PWM outputs) signals before they are masked by the output disable function. When the output disable state is active, a compare match is 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 3),
GPT16m.GTBER 4007 8040h + 0100h × m (m = 4, 5, 8)

	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: No buffer operation 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: No buffer operation 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: No buffer operation 0 1: Single buffer operation (GTPBR → GTPR) 1 x: Setting prohibited.	R/W
b22	CCRSWT	GTCCRA and GTCCRB Forcible Buffer Operation	Writing 1 to this bit forces a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after 1 is written. This bit is read as 0.	R/W
b31 to b23	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTBER register provides settings for the buffer operation and must be set while the GTCNT operation stops.

BD[0] bit (GTCCR Buffer Operation Disable)

The BD[0] bit disables 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)

The BD[1] bit disables buffer operation using GTPR and GTPBR combined.

CCRA[1:0] bits (GTCCRA Buffer Operation)

The CCRA[1:0] 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)

The CCRB[1:0] 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)

The PR[1:0] bits set buffer operation using GTPR and GTPBR combined.

CCRSWT bit (GTCCRA and GTCCRB Forcible Buffer Operation)

Writing 1 to the CCRSWT bit forces a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after 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 3),
GPT16m.GTCNT 4007 8048h + 0100h × m (m = 4, 5, 8)

	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 for GPT32m (m = 0 to 3). For GPT16m (m = 4, 5, 8), GTCNT is a 16-bit register. GTCNT can only be written to after counting stops. GTCNT must be accessed in 32-bit units. Access in 8-bit/16-bit units is prohibited.

For GPT16m (m = 4, 5, 8) the upper 16 bits for access in a 32-bit unit are always read as 0000h, and writing to these bits is ignored.

GTCNT must be set within the range of $0 \leq \text{GTCNT} \leq \text{GTPR}$.

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 3),
GPT16m.GTCCRA 4007 804Ch + 0100h × m (m = 4, 5, 8),
GPT32m.GTCCRB 4007 8050h + 0100h × m (m = 0 to 3),
GPT16m.GTCCRB 4007 8050h + 0100h × m (m = 4, 5, 8),
GPT32m.GTCCRC 4007 8054h + 0100h × m (m = 0 to 3),
GPT16m.GTCCRC 4007 8054h + 0100h × m (m = 4, 5, 8),
GPT32m.GTCCRD 4007 805Ch + 0100h × m (m = 0 to 3),
GPT16m.GTCCRD 4007 805Ch + 0100h × m (m = 4, 5, 8),
GPT32m.GTCCRE 4007 8058h + 0100h × m (m = 0 to 3),
GPT16m.GTCCRE 4007 8058h + 0100h × m (m = 4, 5, 8),
GPT32m.GTCCRF 4007 8060h + 0100h × m (m = 0 to 3),
GPT16m.GTCCRF 4007 8060h + 0100h × m (m = 4, 5, 8)

	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	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

Note 1. For GPT16m (m = 4, 5, 8), the value of the upper 16 bits after reset is 0000h.

GTCCRn registers are read/write registers. The effective size of GTCCRn is the same as GTCNT (16- or 32-bit). If the effective size of GTCCRn is 16-bit, the upper 16 bits for access in a 32-bit unit are always read as 0000h, and writing to these bits is ignored.

GTCCRA and GTCCRB are registers used for both output compare and input capture. GTCCRC and GTCCRE are comparison 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 3),
GPT16m.GTPR 4007 8064h + 0100h × m (m = 4, 5, 8)

	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	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

Note 1. For GPT16m (m = 4, 5, 8), value of the upper 16 bits after reset is 0000h.

GTPR is a read/write register that sets the maximum count value of GTCNT. The effective size of GTPR is the same as GTCNT (16- or 32-bit). If the effective size of GTPR is 16-bit, the upper 16 bits for access in a 32-bit unit are always read as 0000h, and writing to these bits is ignored.

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 3),
GPT16m.GTPBR 4007 8068h + 0100h × m (m = 4, 5, 8)

	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]	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

Note 1. For GPT16m (m = 4, 5, 8), the value of the upper 16 bits after reset is 0000h.

GTPBR is a read/write register that functions as a buffer register for GTPR. The effective size of GTPBR is the same as GTCNT (16- or 32-bit). If the effective size of GTPBR is 16-bit, the upper 16 bits for access in a 32-bit unit are always read as 0000h, and writing to these bits is ignored.

23.2.22 General PWM Timer Dead Time Control Register (GTDTCR)

Address(es): GPT32m.GTDTCR 4007 8088h + 0100h × m (m = 0 to 3),
GPT16m.GTDTCR 4007 8088h + 0100h × m (m = 4, 5, 8)

	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)

The TDE 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 3),
GPT16m.GTDVU 4007 808Ch + 0100h x m (m = 4, 5, 8)

	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]	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

Note 1. For GPT16m (m = 4, 5, 8), the value of the upper 16 bits after reset is 0000h.

GTDVU is a read/write register that sets the dead time for generating PWM waveforms with dead time. The effective size of GTDVU is the same as GTCNT (16- or 32-bit). If the effective size of GTDVU is 16-bit, the upper 16 bits for access in a 32-bit unit are always read as 0000h, and writing to these bits is ignored.

Setting a dead time value that exceeds the cycle value is prohibited. 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 GPT is running, changing the GTDVU values is prohibited. To change GTDVU to a new value, stop the GPT with the CST bit in the GTCR register. GTDVU must be accessed in 32-bit units. Access in 8-bit/16-bit units is prohibited.

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 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 that are synchronized by PCLKD are monitored by these bits 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: Do 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

Bit	Symbol	Bit name	Description	R/W
b16	FB	External Feedback Signal Enable	This bit selects the input phase from 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
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 aligned to PCLKD 1: Input phase aligned to 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 x: Setting Prohibited.	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 on the external input 1: Use a noise filter on 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[1:0] 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)

The UF, VF, WF bits set the input phase from the software settings. When OPSCR.FB bit is 1, these bits are valid. The set value of the UF/VF/WF takes the place of the U/V/W external input.

U, V, W bits (Input Phase Monitor)

When the OPSCR.FB bit is 0, external inputs that are synchronized by PCLKD are monitored by these bits. When the OPSCR.FB bit is 1, the OPSCR.U, OPSCR.V, and OPSCR.W bits can read the OPSCR.UF, OPSCR.VF, and OPSCR.WF bits.

EN bit (Enable-Phase Output Control)

The EN bit controls the output enable signal output phase (positive phase/reverse phase).

When the OPSCR.EN bit is 1, the signal waveform is output.

When the OPSCR.EN bit is 0, first set OPSCR.FB, OPSCR.UF/VF/WF (software setting is selected), OPSCR.P/N, OPSCR.INV, OPSCR.ALIGN, OPSCR.RV, OPSCR.GRP[1:0], OPSCR.GODF, OPSCR.NFEN, OPSCR.NFCS[1:0]. Then, set the EN bit to 1. Also when OPSCR.GODF is 1 and the signal value selected by the OPSCR.GRP[1:0] bit is high, the OPSCR.EN bit is set to 0.

FB bit (External Feedback Signal Enable)

The FB bit selects the input phase from the software settings (OPSCR.UF, VF, WF) and external input such as a Hall element.

P bit (Positive-Phase Output (P) Control)

The P bit selects one of the level signal output (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)

The N 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)

The INV 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)

The ALIGN 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 is 0, input phase is aligned to PCLKD.

Note: When PWM output is selected (OPSCR.P/N is 1) and the PCLKD input phase is aligned, the PWM pulse can be short-pulsed.

Note: When OPSCR.ALIGN bit is 1, input phase is aligned with PWM output.

GRP[1:0] bits (Output Disabled Source Selection)

The GRP[1:0] bits select the output disable source (A, B).

GODF bit (Group Output Disable Function)

When the OPSCR.GODF is 1 and the signal value selected by the OPSCR.GRP[1:0] bit is high, the OPSCR.EN bit is set to 0. When the OPSCR.GODF bit is 0, this bit is ignored.

NFEN bit (External Input Noise Filter Enable)

The NFEN bit selects the noise filter for external input. When OPSCR.NFEN bit is 0, a noise filter for the external input is not used.

Note: When this bit is switched because of an unintentional internal edge, set the OPSCR.EN bit to 0.

NFCS[1:0] bits (External Input Noise Filter Clock Selection)

The NFCS[1:0] bits select the clock for the external input noise filter. When the OPSCR.NFEN bit is 1, noise filter sampling clock setting of the external input is enabled.

1. Set the NFCS[1:0].
2. Wait for 2 cycles.
3. Set the OPSCR.EN bit to 1.

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 associated 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 and stop

The counter of each channel starts the count operation when GTCR.CST is set to 1. The GTCR.CST bit value is changed by the following sources:

- Writing to GTCR register
- Writing 1 to the bit in GTSTR associated with the GPT channel number when the GTSSR.CSTRT bit is set to 1
- Writing 1 to the bit in GTSTP associated with the GPT channel number when the GTPSR.CSTOP bit is 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 0000 0000h. 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 0000 0000h.

Figure 23.3 shows an example of a periodic count operation in up-counting.

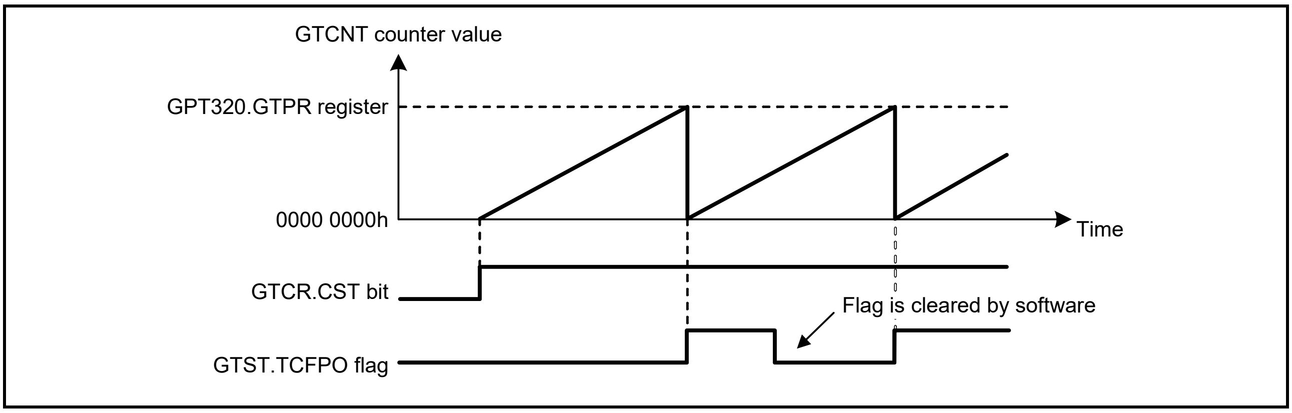
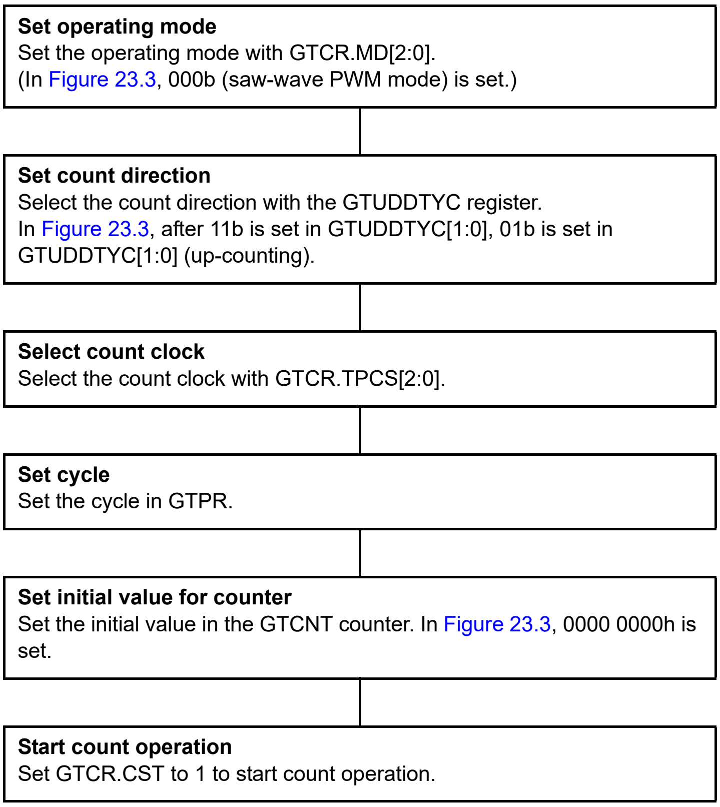

GTCNT counter value
GPT320.GTPR register
0000 0000h
Time
GTCR.CST bit
Flag is cleared by software
GTST.TCFPO flag

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.



Set operating mode
Set the operating mode with GTCR.MD[2:0].
(In Figure 23.3, 000b (saw-wave PWM mode) is set.)

Set count direction
Select the count direction with the GTUDDTYC register.
In Figure 23.3, 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 23.3, 0000 0000h is set.

Start count operation
Set GTCR.CST to 1 to start count operation.

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 0000 0000h. 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.

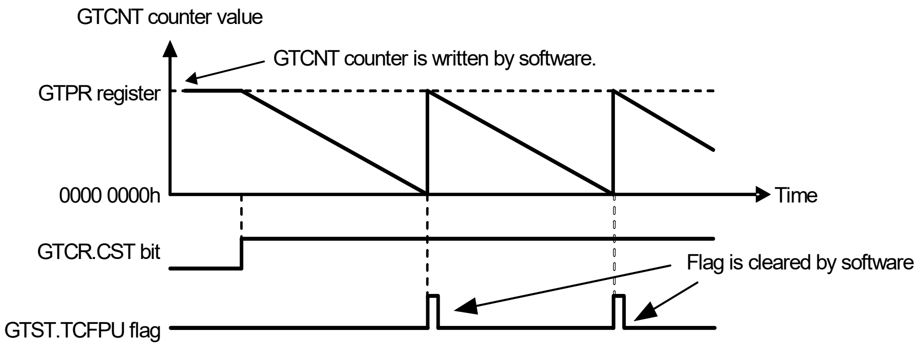
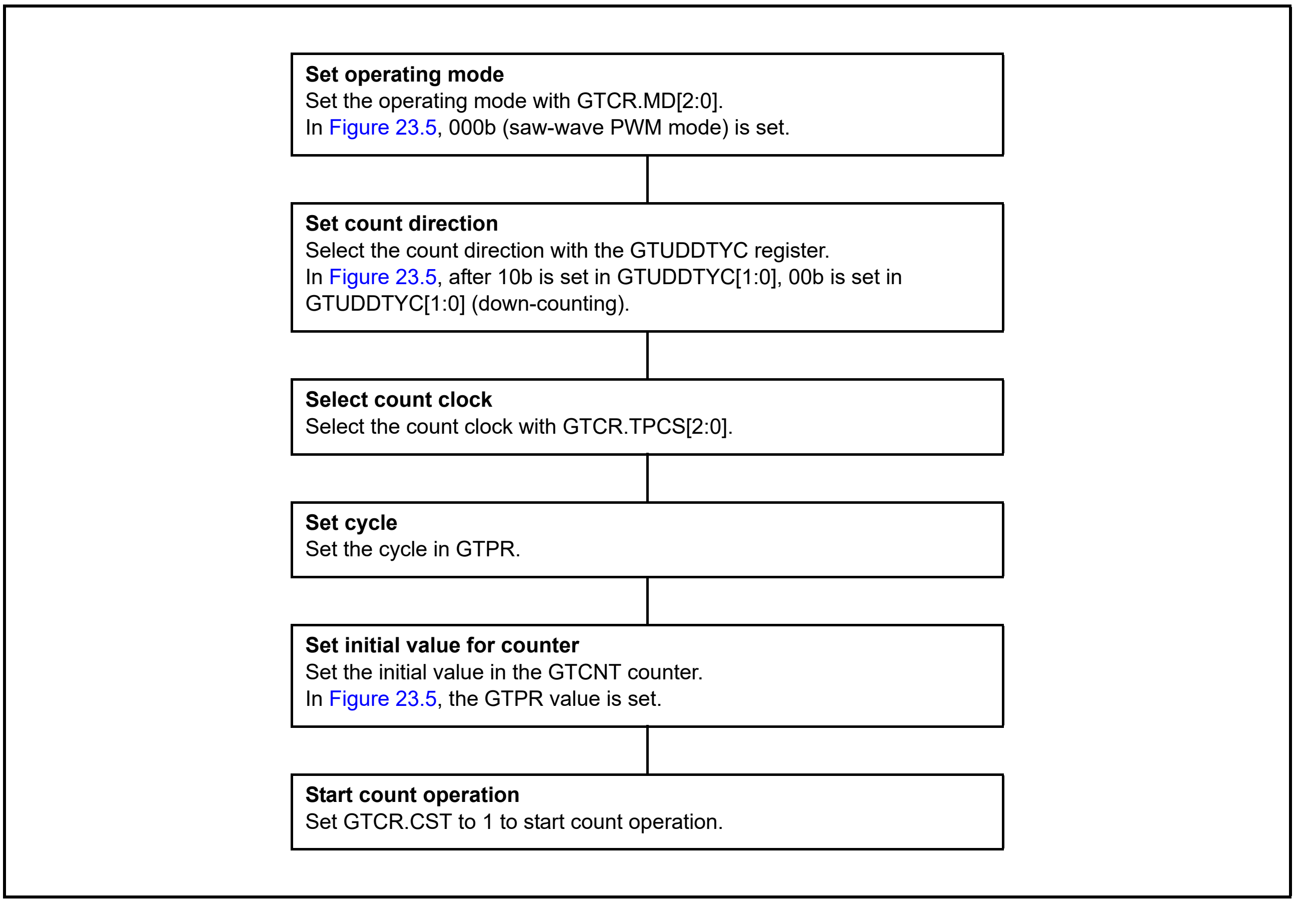


Figure 23.5 Example of periodic count operation in down-counting by the count clock

Figure 23.6 shows an example for setting periodic count operation in down-counting by the count clock.



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.5, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction with the GTUDDTYC register.
In Figure 23.5, after 10b is set in GTUDDTYC[1:0], 00b is set in GTUDDTYC[1:0] (down-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 23.5, the GTPR value is set.

Start count operation
Set GTCR.CST to 1 to start count operation.

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 in 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 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).

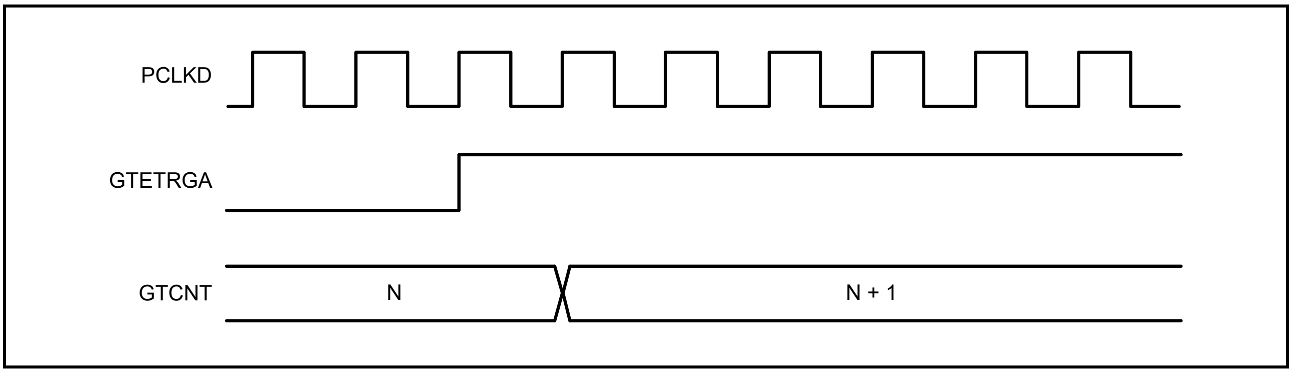
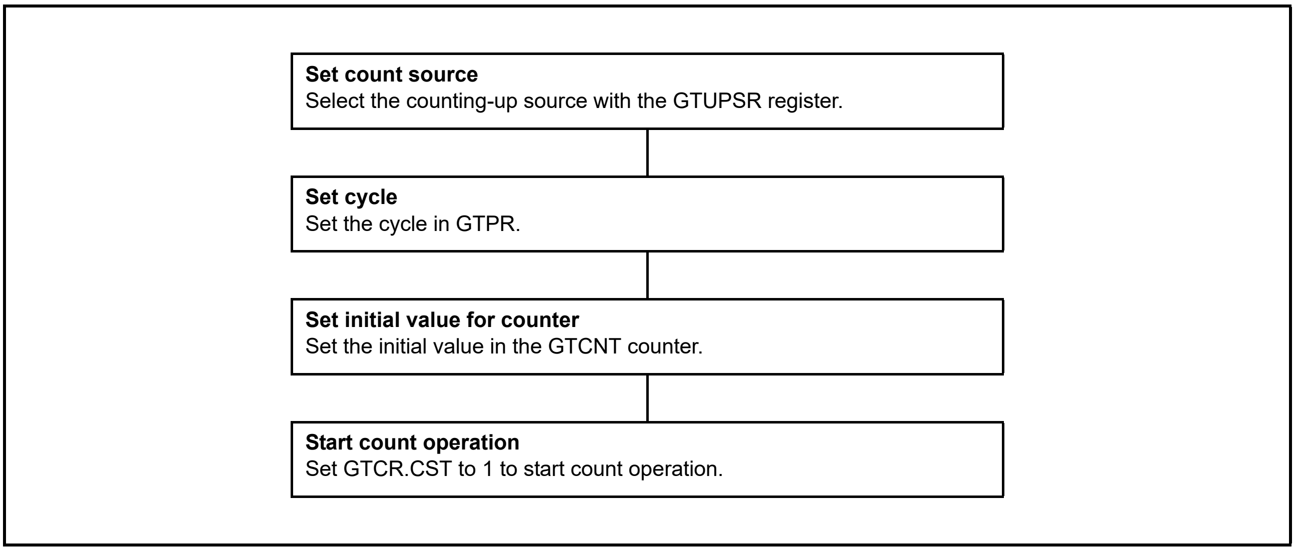


Figure 23.7 Example of periodic count operation in up-counting using hardware sources

Figure 23.8 shows an example for setting periodic count operation in down-counting by the count clock.



Set count source
Select the counting-up source with the GTUPSR register.

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Start count operation
Set GTCR.CST to 1 to start count operation.

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 in 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).

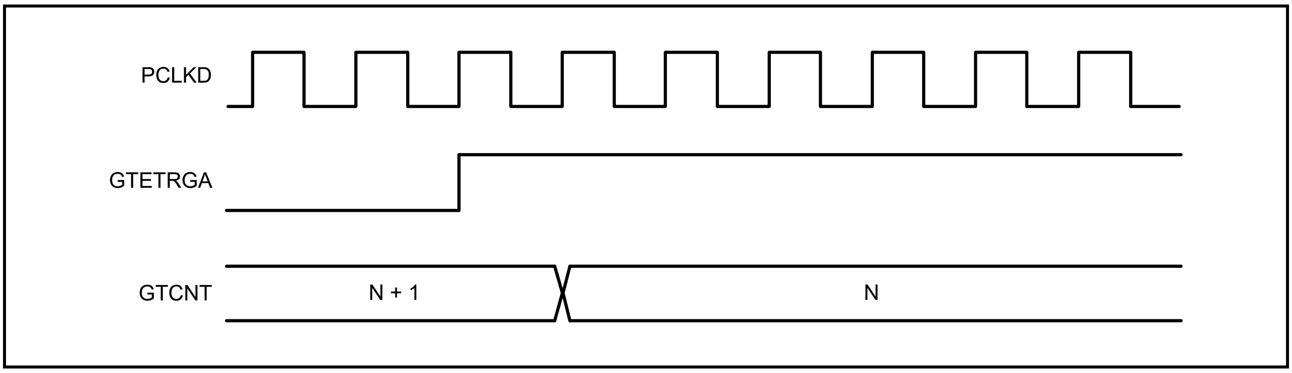
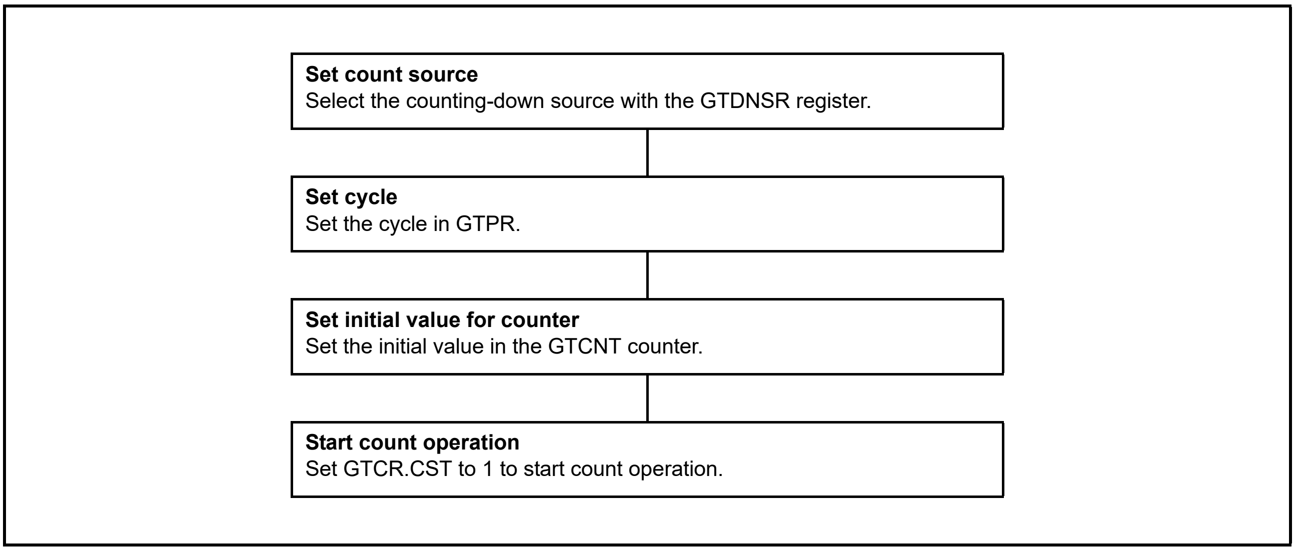


Figure 23.9 Example of event count operation in down-counting using hardware sources

Figure 23.10 shows an example for setting a periodic count operation in down-counting using a hardware resource.



Set count source
Select the counting-down source with the GTDNSR register.

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Start count operation
Set GTCR.CST to 1 to start count operation.

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 the 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 is 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 is 0), the GTCNT register is set to the value of the GTPR register when writing 1 to the GTCLR register or when clearing by hardware sources are performed. When not in saw-wave 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 the 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 in 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 associated 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.

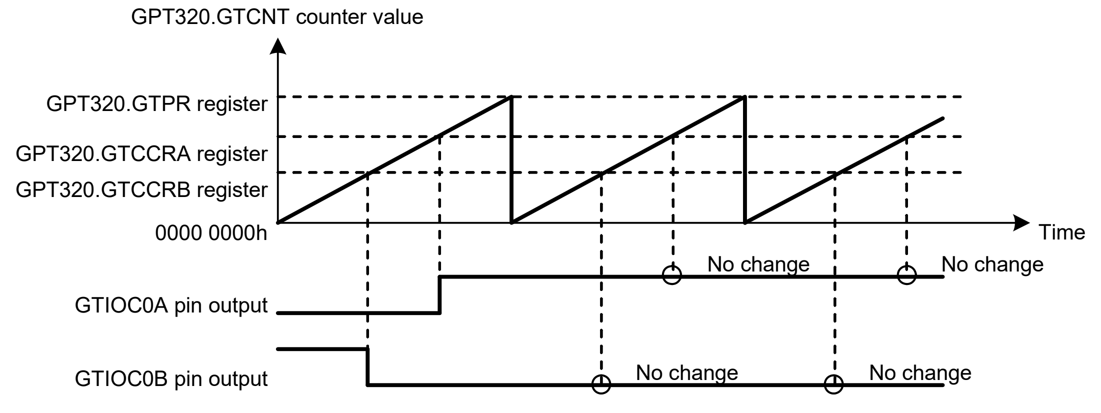
The cycle end is:

- 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.


GPT320.GTCNT counter value
GPT320.GTPR register
GPT320.GTCCRA register
GPT320.GTCCRB register
0000 0000h
Time
No change
No change
GTIOC0A pin output
GTIOC0B pin output
No change
No change

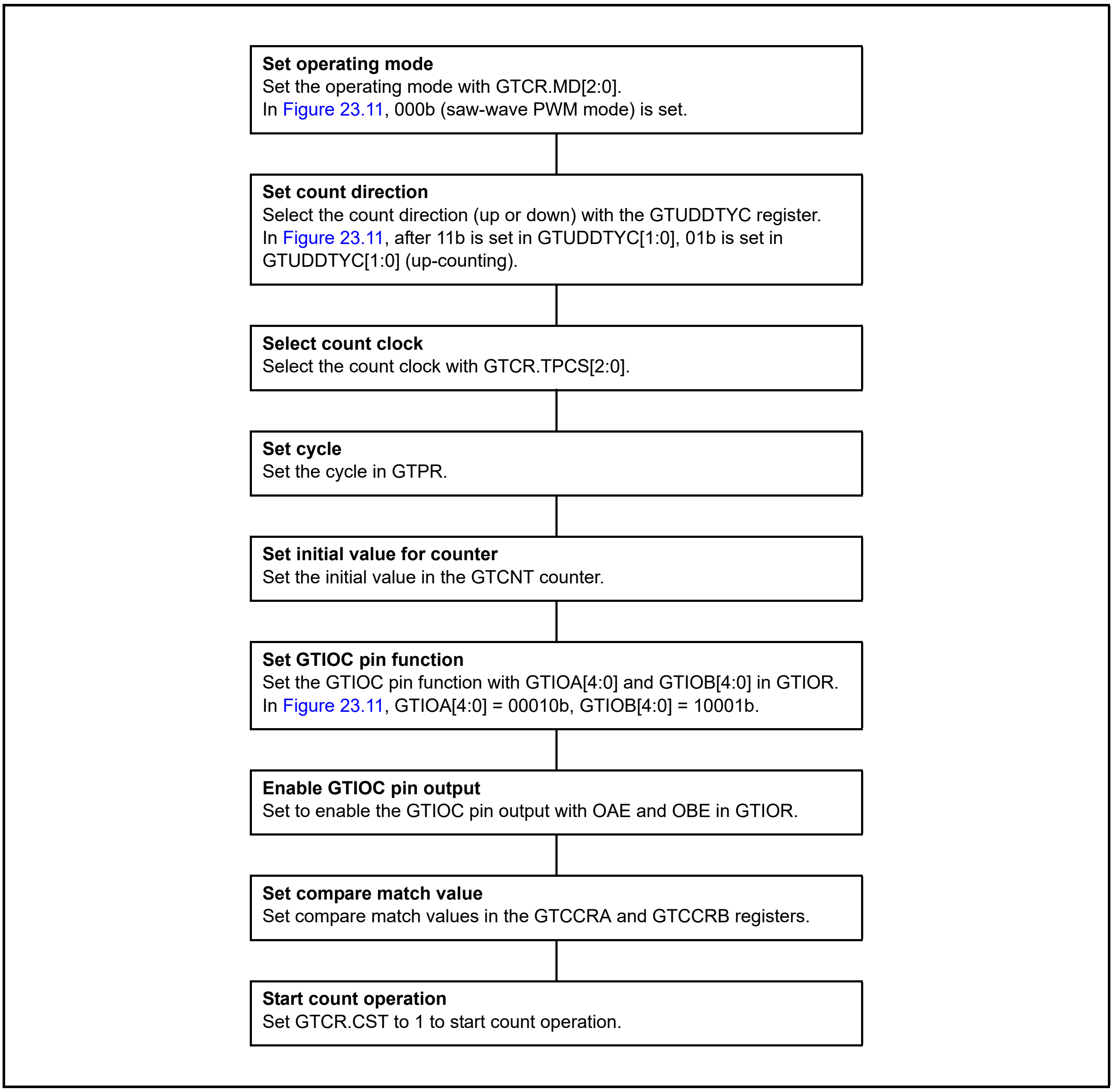
[Setting examples]

GPT320.GTIOR.GTIOA[4:0] bits: Initial output is low, high output at compare match, output retained at cycle end

GPT320.GTIOR.GTIOB[4:0] bits: Initial output is high, low output at compare match, output retained at cycle end

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.



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.11, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.11, 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.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.11, GTIOA[4:0] = 00010b, GTIOB[4:0] = 10001b.

Enable GTIOC pin output
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set compare match value
Set compare match values in the GTCCRA and GTCCRB registers.

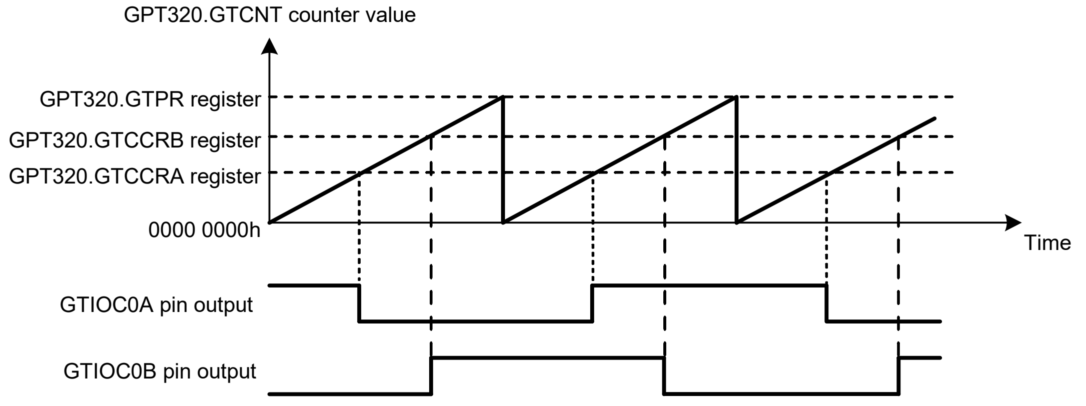
Start count operation
Set GTCR.CST to 1 to start count operation.

Figure 23.12 Example setting for low output and high output operation

(2) Toggled output

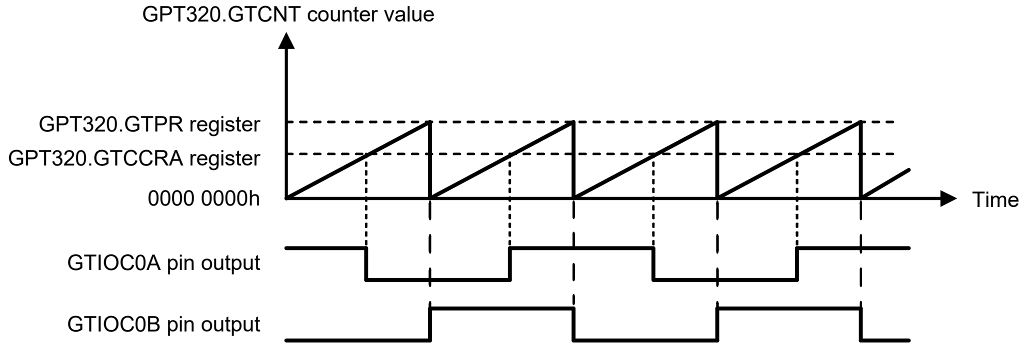
Figure 23.13 and Figure 23.14 show examples of toggled output operation by compare matches of GTCCRA and GTCCRB. In Figure 23.13, the GPT320.GTCNT counter performs up-counting, and settings are made so that the GTIOC0A pin output by a GPT320.GTCCRA compare match and GTIOC0B pin output by a GPT320.GTCCRB 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.


GPT320.GTCNT counter value
GPT320.GTPR register
GPT320.GTCCRB register
GPT320.GTCCRA register
0000 0000h
Time
GTIOC0A pin output
GTIOC0B pin output

[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

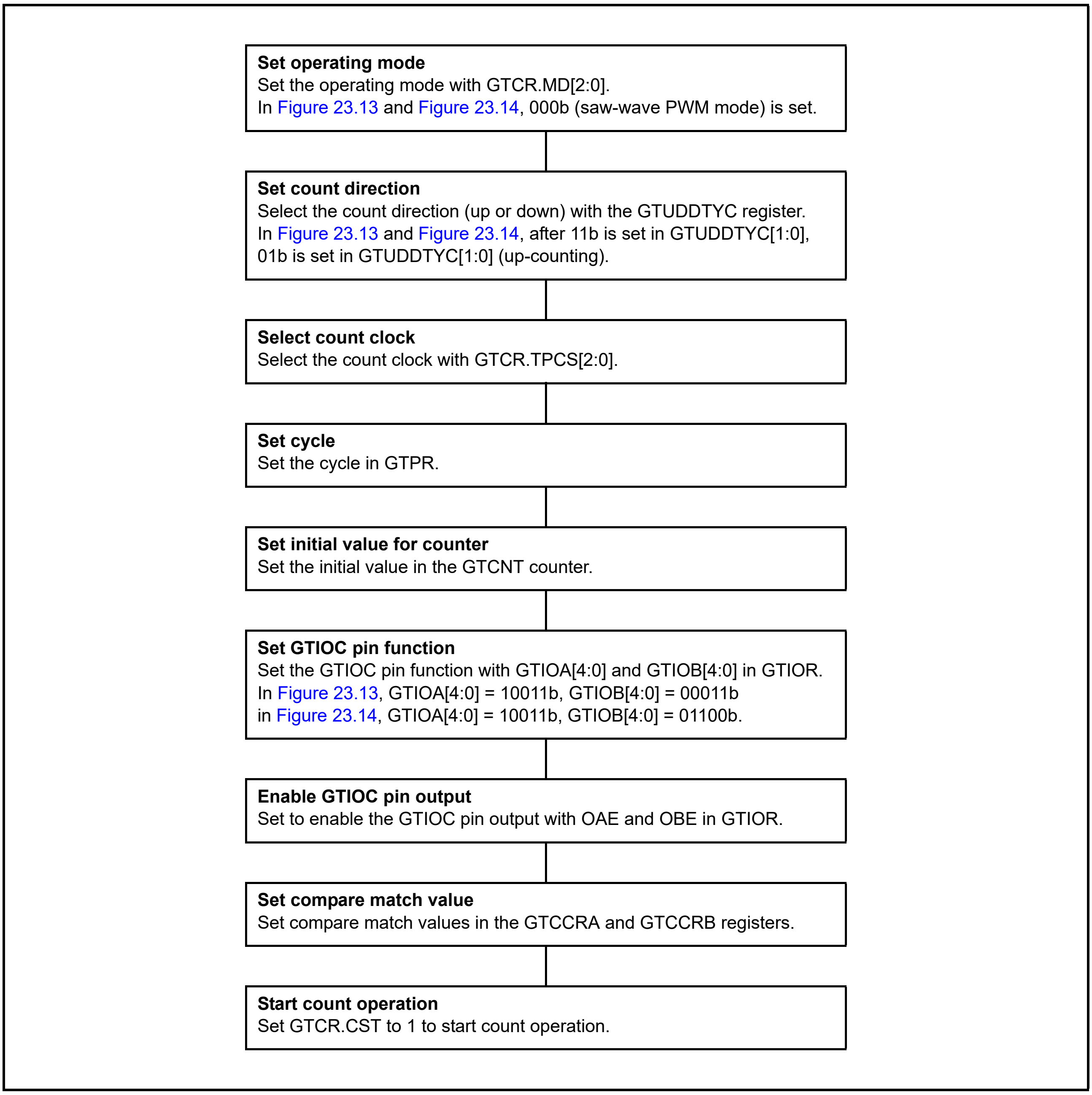
Figure 23.13 Example of toggled output operation (1)


GPT320.GTCNT counter value
GPT320.GTPR register
GPT320.GTCCRA register
0000 0000h
Time
GTIOC0A pin output
GTIOC0B pin output

[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 setting for toggled output operation.



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.13 and Figure 23.14, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.13 and Figure 23.14, 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.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.13, GTIOA[4:0] = 10011b, GTIOB[4:0] = 00011b
in Figure 23.14, GTIOA[4:0] = 10011b, GTIOB[4:0] = 01100b.

Enable GTIOC pin output
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set compare match value
Set compare match values in the GTCCRA and GTCCRB registers.

Start count operation
Set GTCR.CST to 1 to start count operation.

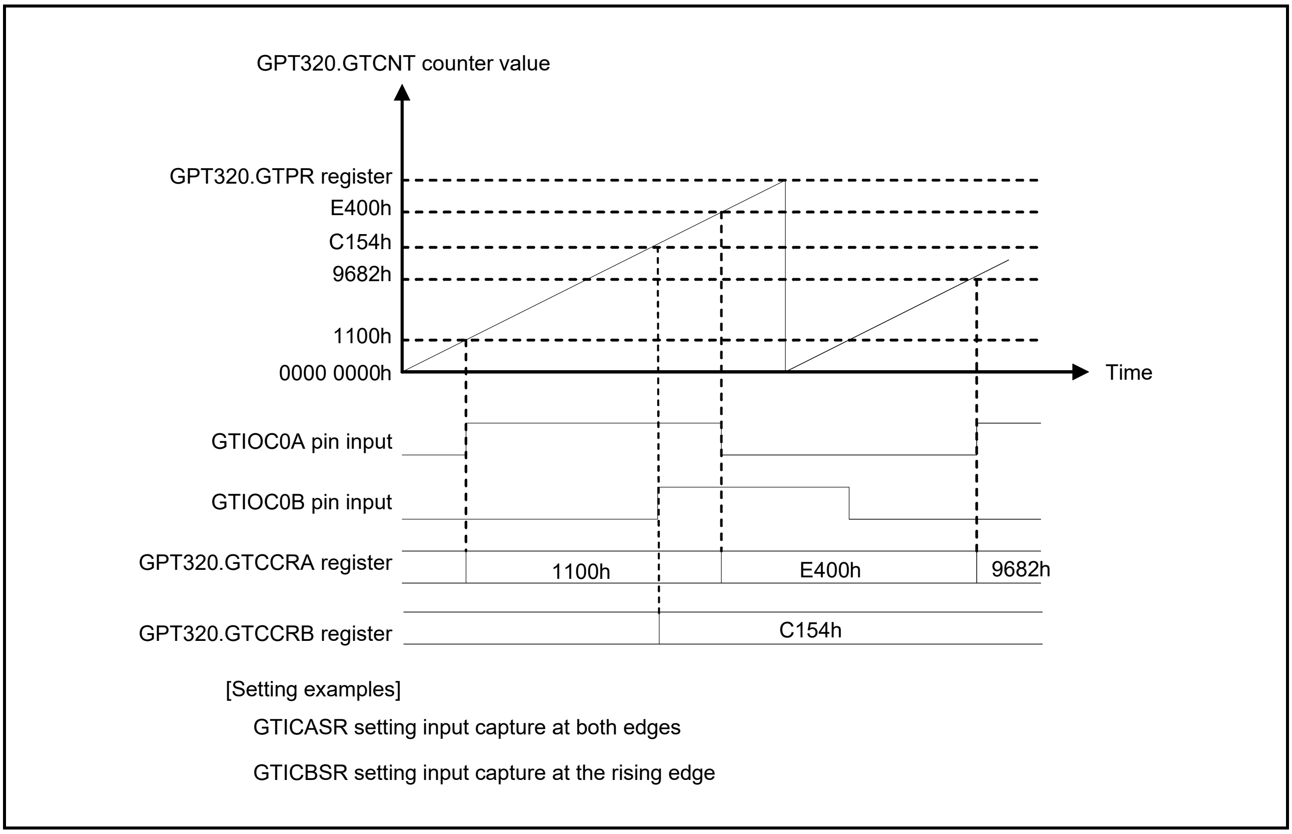
Figure 23.15 Example for setting toggled output operation

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.


GPT320.GTCNT counter value
GPT320.GTPR register
E400h
C154h
9682h
1100h
0000 0000h
Time
GTIOC0A pin input
GTIOC0B pin input
GPT320.GTCCRA register
1100h
E400h
9682h
GPT320.GTCCRB register
C154h

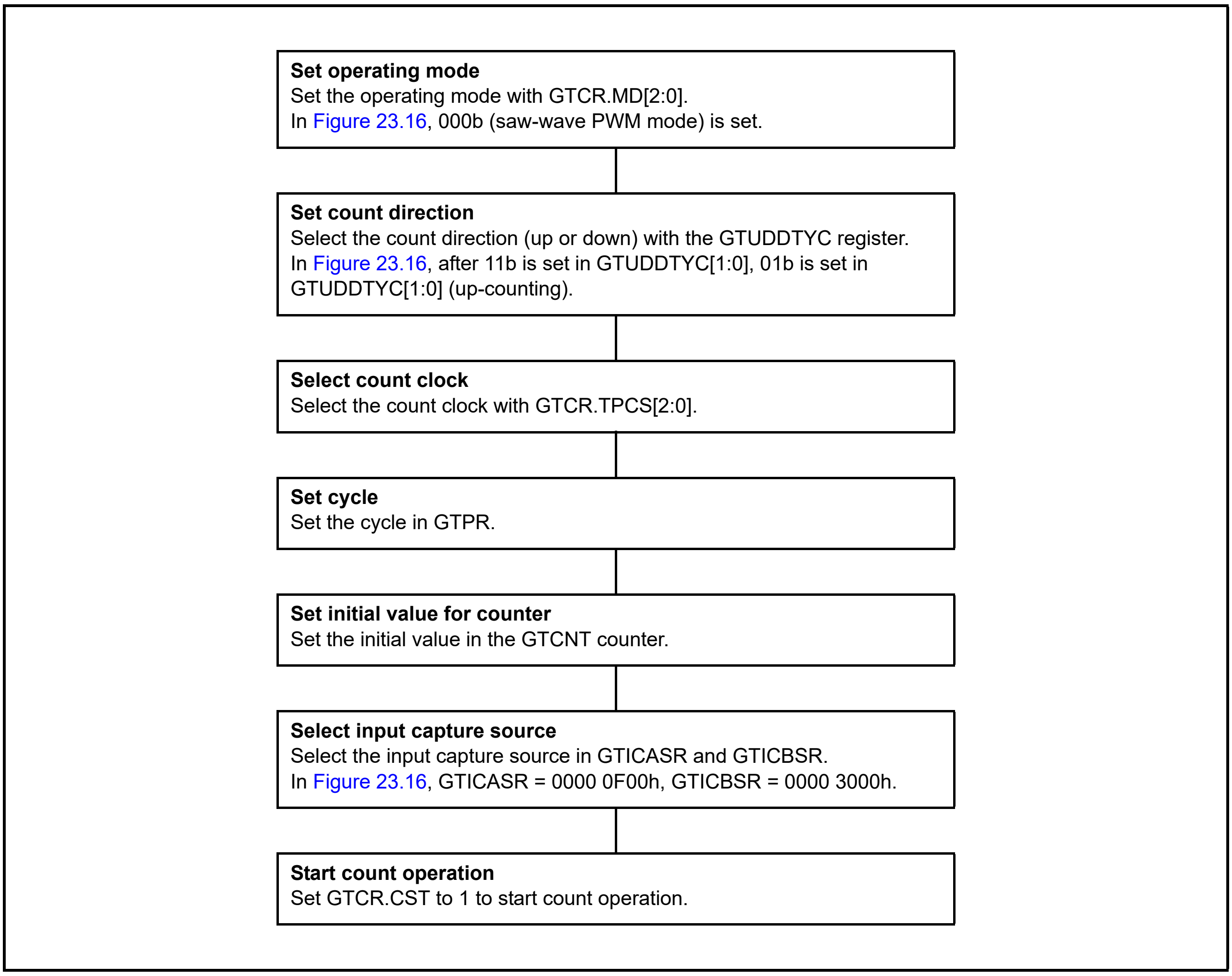
[Setting examples]

GTICASR setting input capture at both edges

GTICBSR setting input capture at the rising edge

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.



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.16, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.16, 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.

Select input capture source
Select the input capture source in GTICASR and GTICBSR.
In Figure 23.16, GTICASR = 0000 0F00h, GTICBSR = 0000 3000h.

Start count operation
Set GTCR.CST to 1 to start count operation.

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 setting for GTPR buffer operation.

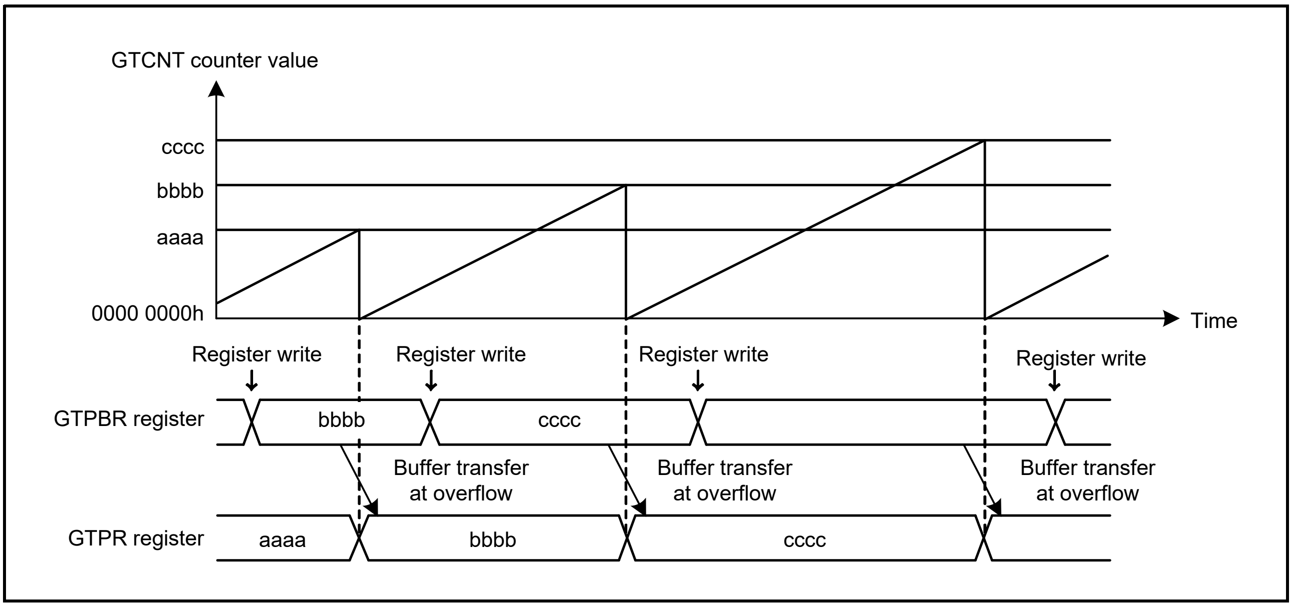

GTCNT counter value
cccc
bbbb
aaaa
0000 0000h
Time
Register write
Register write
Register write
Register write
GTPBR register
bbbb
cccc
Buffer transfer at overflow
Buffer transfer at overflow
Buffer transfer at overflow
GTPR register
aaaa
bbbb
cccc

Figure 23.18 Example of GTPR buffer operation with saw waves in up-counting

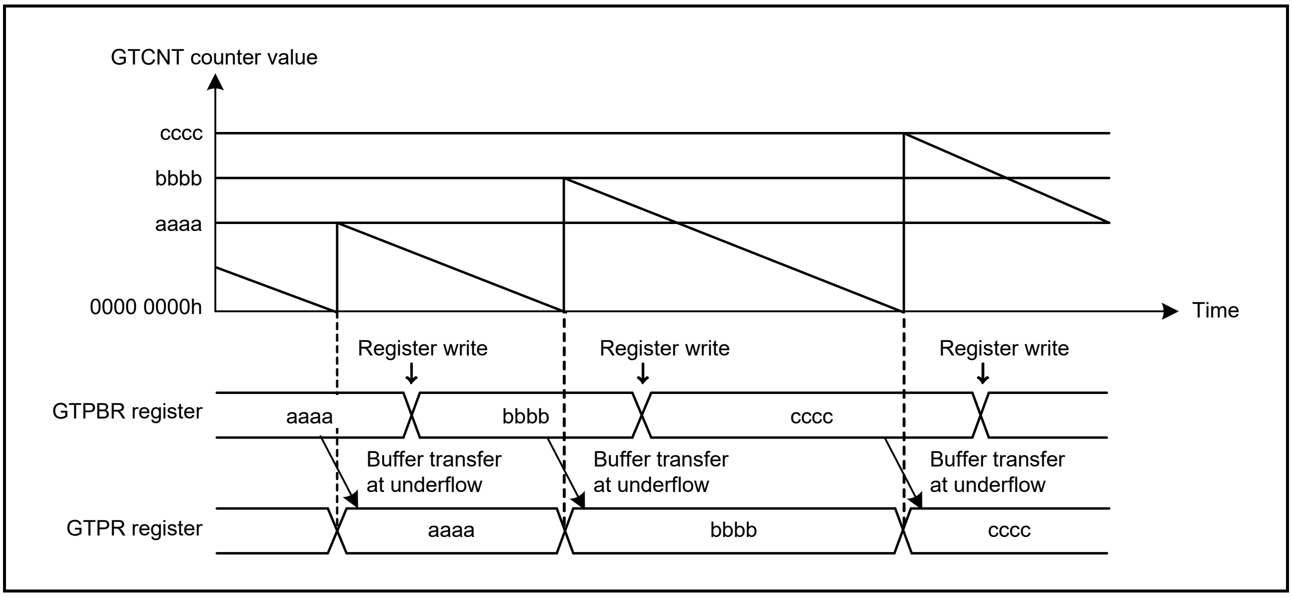

GTCNT counter value
cccc
bbbb
aaaa
0000 0000h
Time
Register write
Register write
Register write
GTPBR register
aaaa
bbbb
cccc
Buffer transfer at underflow
Buffer transfer at underflow
Buffer transfer at underflow
GTPR register
aaaa
bbbb
cccc

Figure 23.19 Example of GTPR buffer operation with saw waves in down-counting

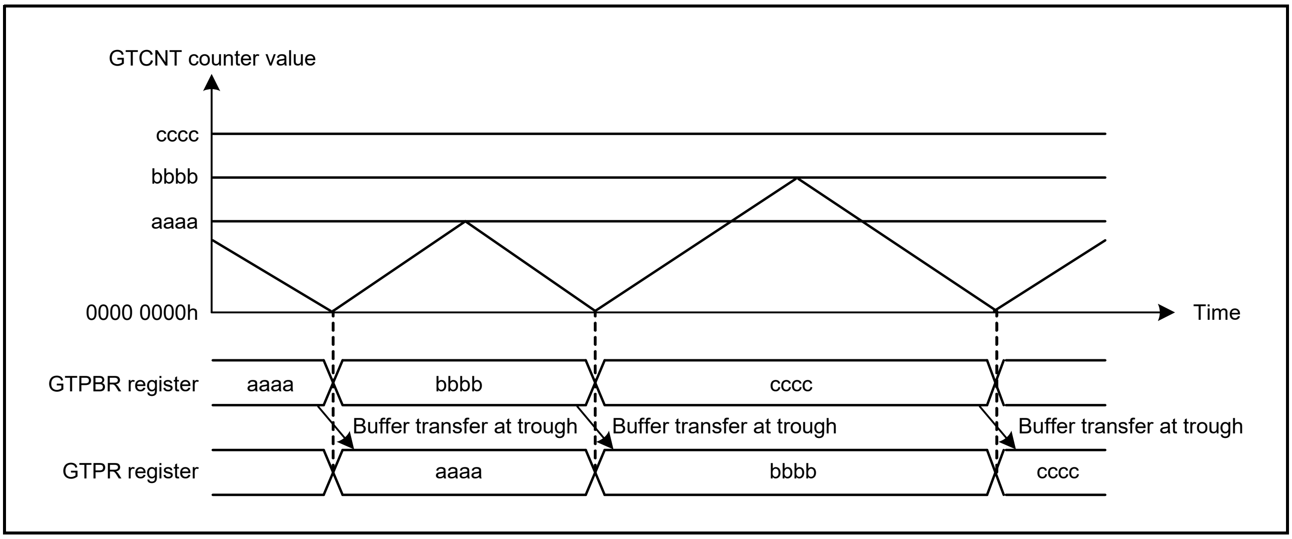
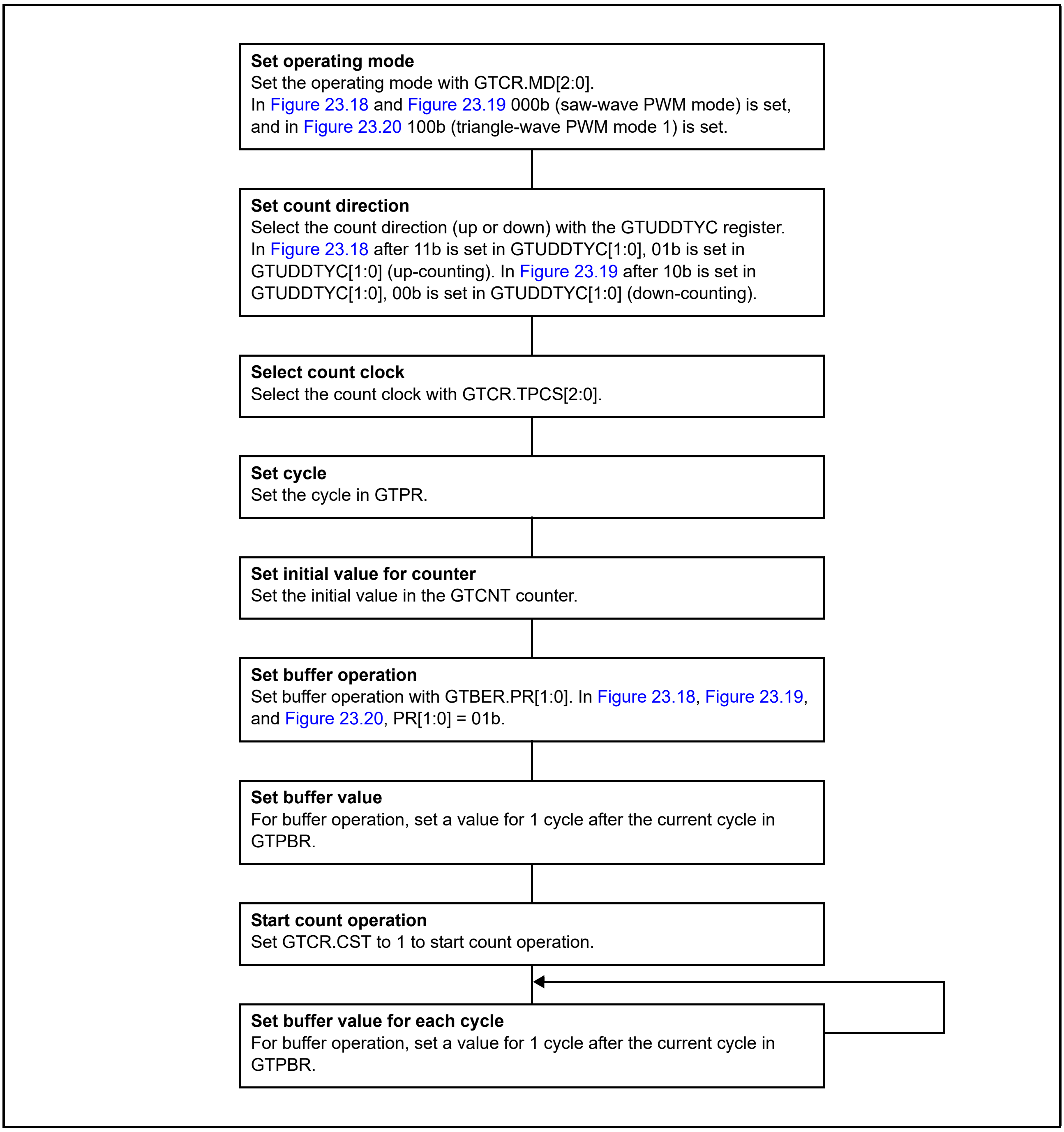

GTCNT counter value
cccc
bbbb
aaaa
0000 0000h
Time
GTPBR register
aaaa
bbbb
cccc
Buffer transfer at trough
Buffer transfer at trough
Buffer transfer at trough
GTPR register
aaaa
bbbb
cccc

Figure 23.20 Example of GTPR buffer operation with triangle waves



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.18 and Figure 23.19 000b (saw-wave PWM mode) is set, and in Figure 23.20 100b (triangle-wave PWM mode 1) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.18 after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting). In Figure 23.19 after 10b is set in GTUDDTYC[1:0], 00b is set in GTUDDTYC[1:0] (down-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.

Set buffer operation
Set buffer operation with GTBER.PR[1:0]. In Figure 23.18, Figure 23.19, and Figure 23.20, PR[1:0] = 01b.

Set buffer value
For buffer operation, set a value for 1 cycle after the current cycle in GTPBR.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
For buffer operation, set a value for 1 cycle after the current cycle in GTPBR.

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 a 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 an output compare register

Buffer transfer occurs in the following situations:

- Buffer transfer by overflow or underflow
  Buffer transfer is performed at an overflow (during up-counting) or an underflow (during down-counting) in saw-wave mode or in event count operation. In triangle-wave mode, buffer transfer is performed at a trough (triangle-wave PWM mode 1) or a crest and trough (triangle-wave PWM mode 2).
- Buffer transfer by counter clear
  In saw-wave mode or in event count operation, during counting, buffer transfer (which is the same as an overflow during up-counting or an underflow during down-counting) is performed by the counter clear sources 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 is stopped, the GTCCRA and the GTCCRB register buffer transfers are performed forcibly in saw-wave mode, in event count operation and in triangle-wave mode. Additionally, buffer transfer from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B are performed in saw-wave 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.

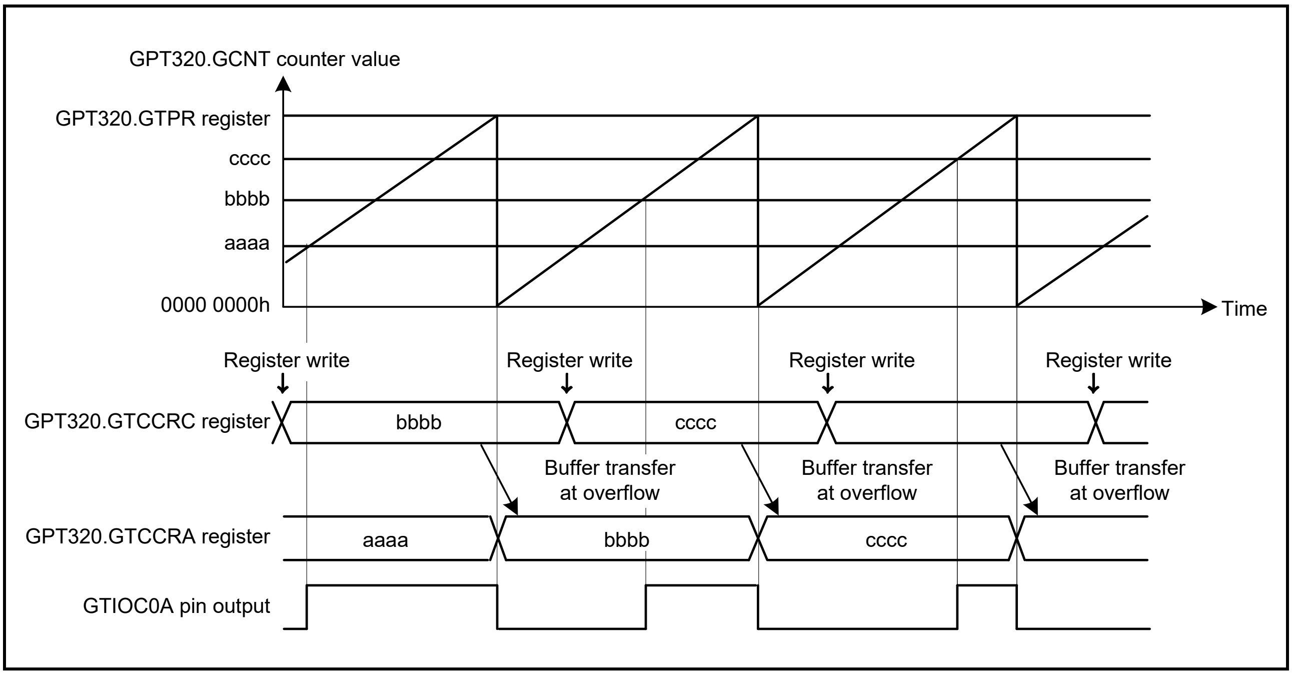

GPT320.GCNT counter value
GPT320.GTPR register
cccc
bbbb
aaaa
0000 0000h
Time
Register write
Register write
Register write
Register write
GPT320.GTCCRC register
bbbb
cccc
Buffer transfer at overflow
Buffer transfer at overflow
Buffer transfer at overflow
GPT320.GTCCRA register
aaaa
bbbb
cccc
GTIOC0A pin output

Figure 23.22 Example of GTCCRA and GTCCRB buffer operation with output compare, saw waves in up-counting, high output at GTCCRA compare match, low output at cycle end

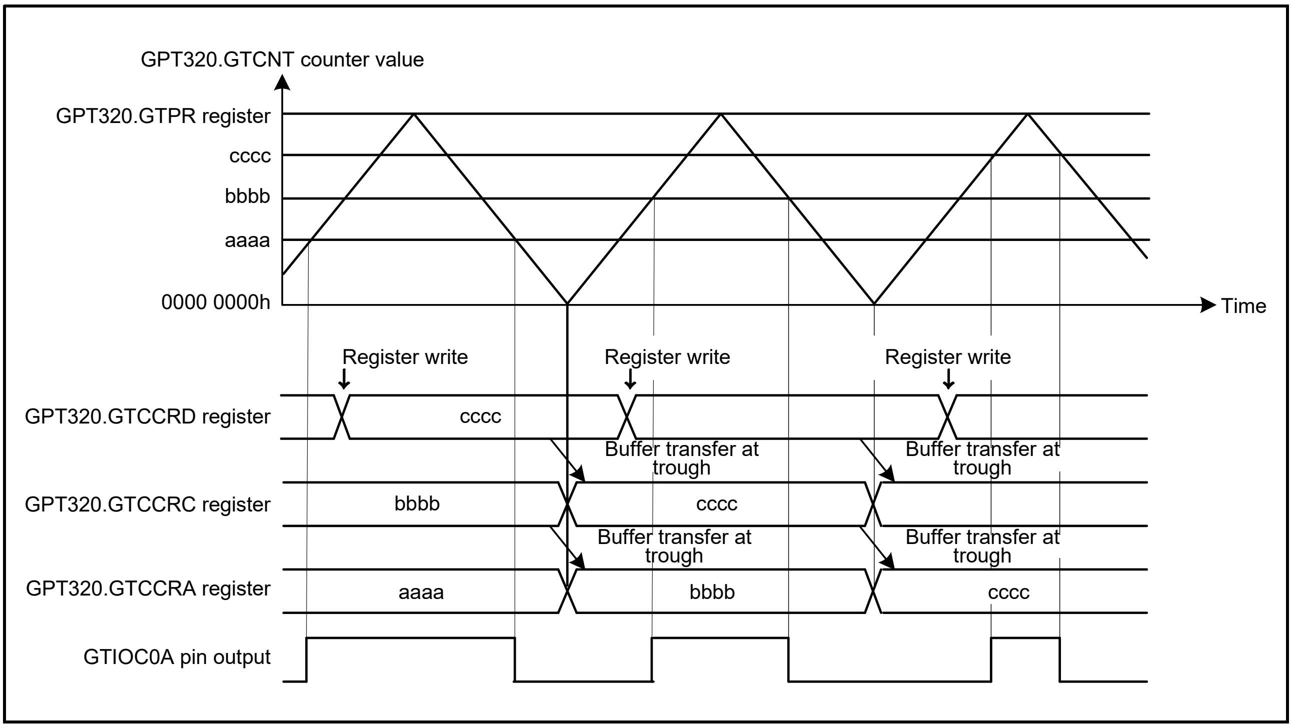

GPT320.GTCNT counter value
GPT320.GTPR register
cccc
bbbb
aaaa
0000 0000h
Time
Register write
Register write
Register write
GPT320.GTCCRD register
cccc
Buffer transfer at trough
Buffer transfer at trough
GPT320.GTCCRC register
bbbb
cccc
Buffer transfer at trough
Buffer transfer at trough
GPT320.GTCCRA register
aaaa
bbbb
cccc
GTIOC0A pin output

Figure 23.23 Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at trough, output toggled at GTCCRA compare match, and output retained at cycle end

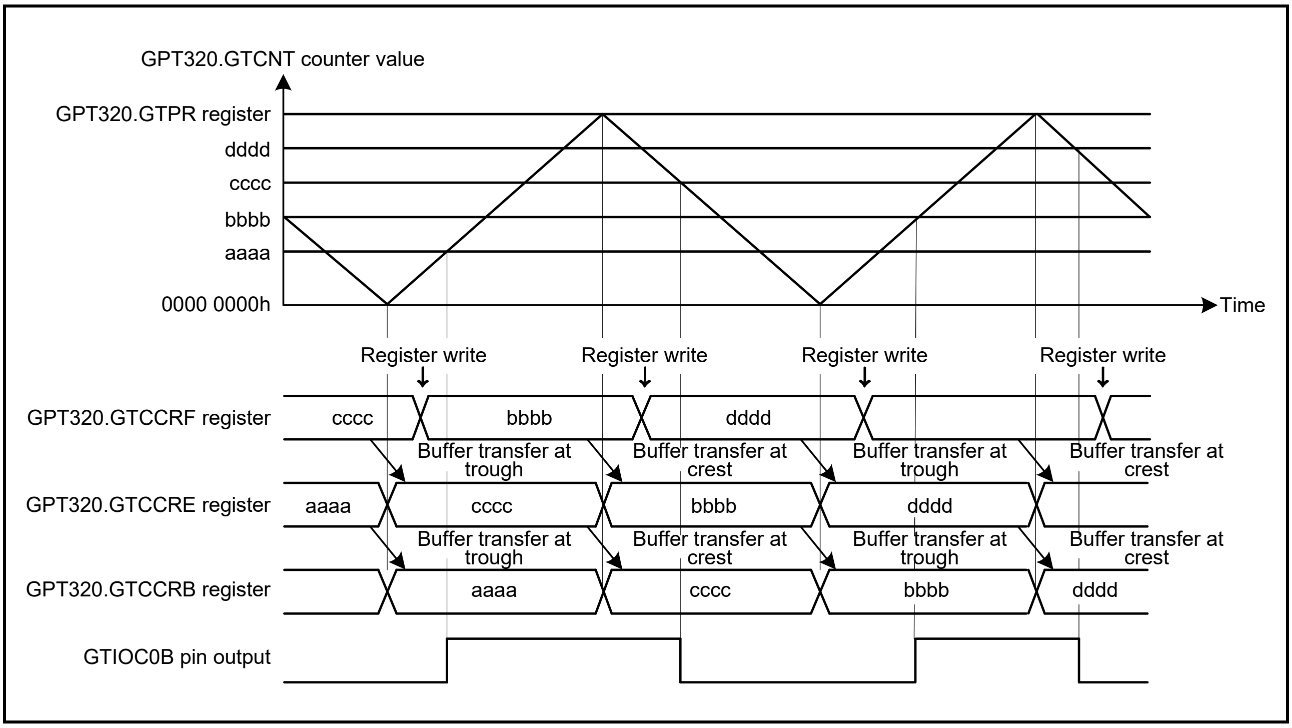
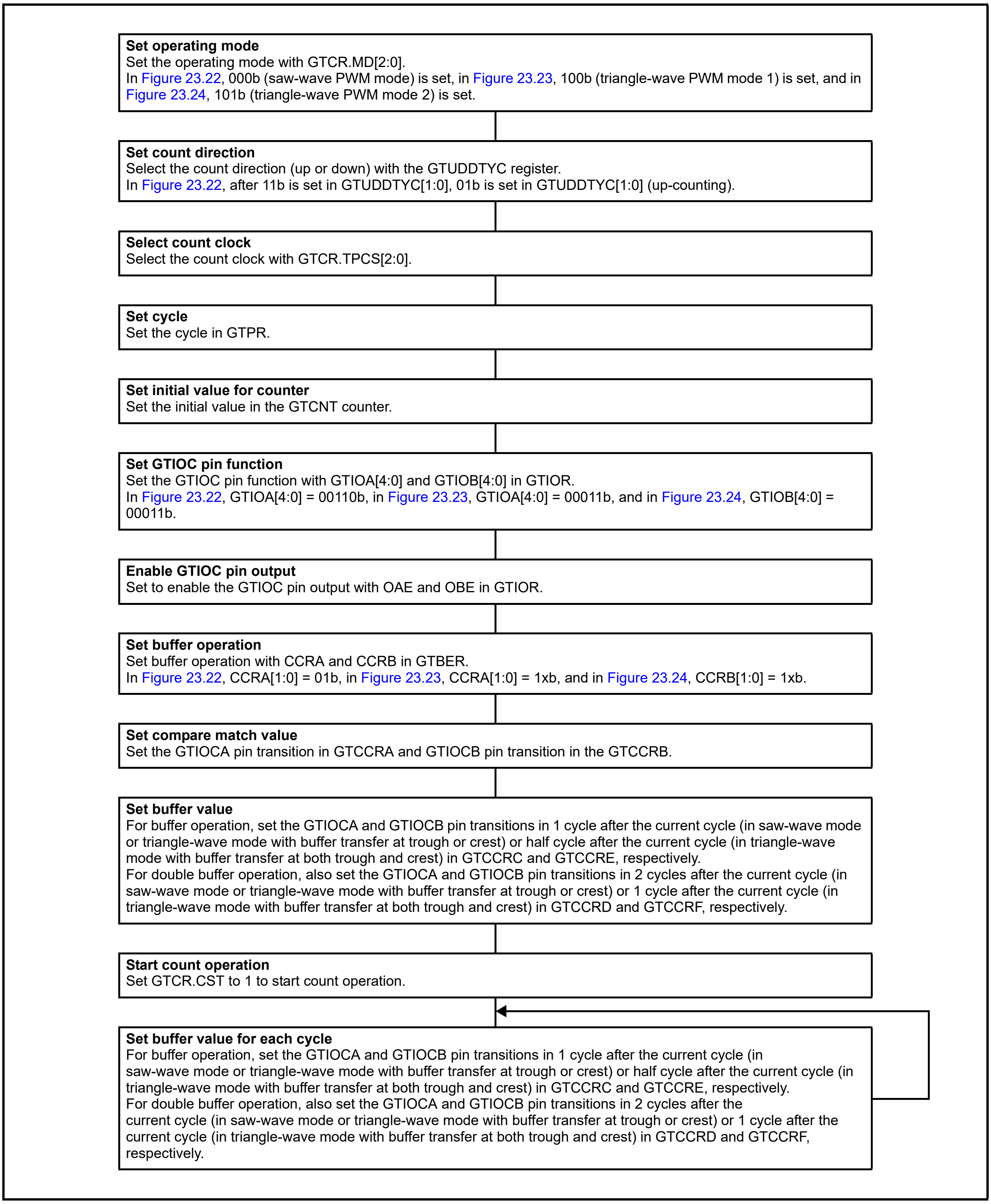

GPT320.GTCNT counter value
GPT320.GTPR register
dddd
cccc
bbbb
aaaa
0000 0000h
Time
Register write
Register write
Register write
Register write
GPT320.GTCCRF register
cccc
bbbb
dddd
Buffer transfer at trough
Buffer transfer at crest
Buffer transfer at trough
Buffer transfer at crest
GPT320.GTCCRE register
aaaa
cccc
bbbb
dddd
Buffer transfer at trough
Buffer transfer at crest
Buffer transfer at trough
Buffer transfer at crest
GPT320.GTCCRB register
aaaa
cccc
bbbb
dddd
GTIOC0B pin output

Figure 23.24 Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at both troughs and crests, output toggled at GTCCRB compare match, and output retained at cycle end



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.22, 000b (saw-wave PWM mode) is set, in Figure 23.23, 100b (triangle-wave PWM mode 1) is set, and in Figure 23.24, 101b (triangle-wave PWM mode 2) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.22, 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.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.22, GTIOA[4:0] = 00110b, in Figure 23.23, GTIOA[4:0] = 00011b, and in Figure 23.24, GTIOB[4:0] = 00011b.

Enable GTIOC pin output
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set buffer operation
Set buffer operation with CCRA and CCRB in GTBER.
In Figure 23.22, CCRA[1:0] = 01b, in Figure 23.23, CCRA[1:0] = 1xb, and in Figure 23.24, CCRB[1:0] = 1xb.

Set compare match value
Set the GTIOCA pin transition in GTCCRA and GTIOCB pin transition in the GTCCRB.

Set buffer value
For buffer operation, set the GTIOCA and GTIOCB pin transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA and GTIOCB pin transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTCCRD and GTCCRF, respectively.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
For buffer operation, set the GTIOCA and GTIOCB pin transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA and GTIOCB pin transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTCCRD and GTCCRF, respectively.

Figure 23.25 Example for setting GTCCRA and GTCCRB buffer operation for output compare

(2) When GTCCRA or GTCCRB functions as an 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 the GTCCRA and GTCCRB buffer operation.

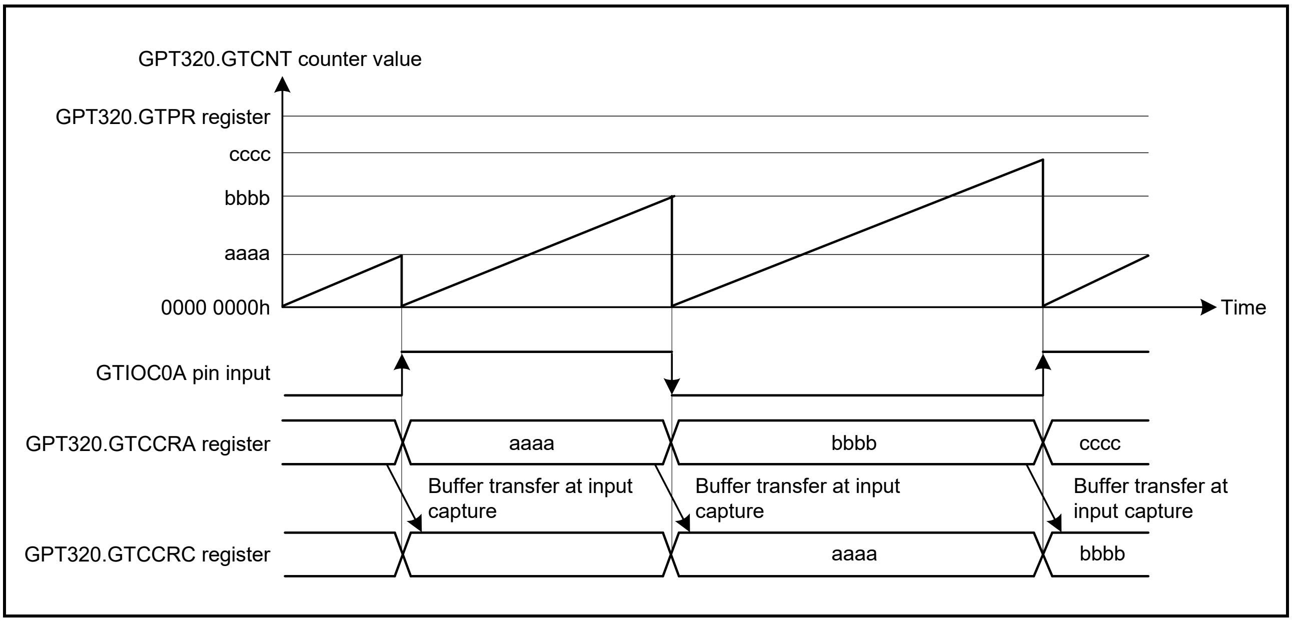

GPT320.GTCNT counter value
GPT320.GTPR register
cccc
bbbb
aaaa
0000 0000h
Time
GTIOC0A pin input
GPT320.GTCCRA register
aaaa
bbbb
cccc
Buffer transfer at input capture
Buffer transfer at input capture
Buffer transfer at input capture
GPT320.GTCCRC register
aaaa
bbbb

Figure 23.26 Example of GTCCRA and GTCCRB buffer operation with input capture at both edges of GTIOC0A input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOC0A input

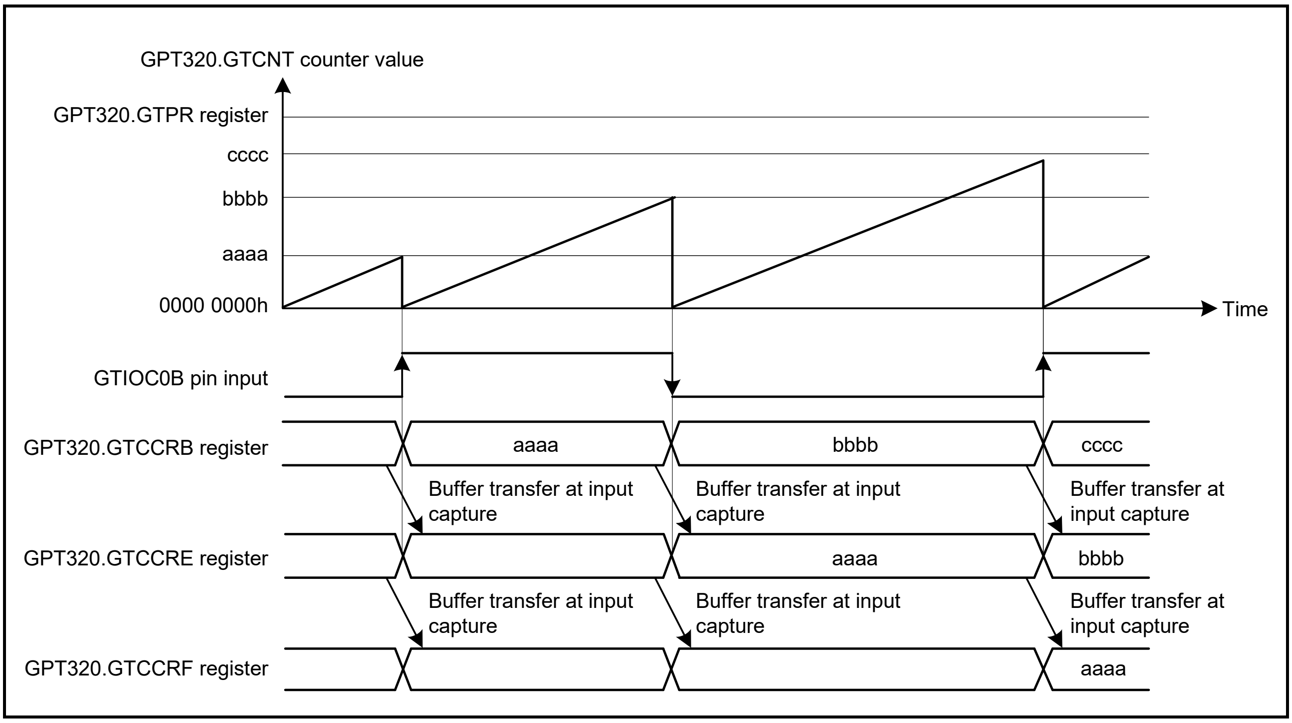
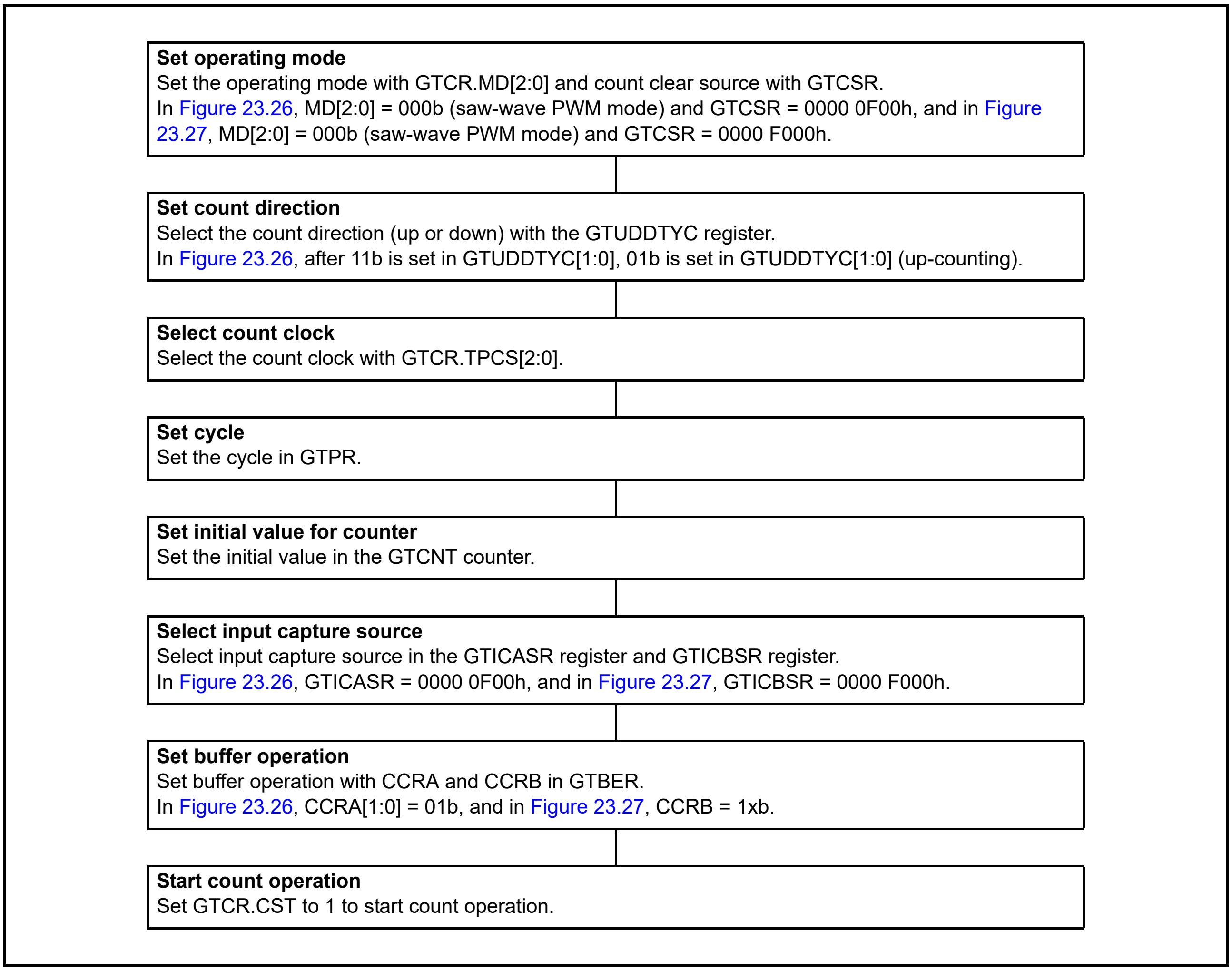

GPT320.GTCNT counter value
GPT320.GTPR register
cccc
bbbb
aaaa
0000 0000h
Time
GTIOC0B pin input
GPT320.GTCCRB register
aaaa
bbbb
cccc
Buffer transfer at input capture
Buffer transfer at input capture
Buffer transfer at input capture
GPT320.GTCCRE register
aaaa
bbbb
Buffer transfer at input capture
Buffer transfer at input capture
Buffer transfer at input capture
GPT320.GTCCRF register
aaaa

Figure 23.27 Example of GTCCRA and GTCCRB double buffer operation with input capture at both edges of GTIOC0B input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOC0B input



Set operating mode
Set the operating mode with GTCR.MD[2:0] and count clear source with GTCSR.
In Figure 23.26, MD[2:0] = 000b (saw-wave PWM mode) and GTCSR = 0000 0F00h, and in Figure 23.27, MD[2:0] = 000b (saw-wave PWM mode) and GTCSR = 0000 F000h.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.26, 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.

Select input capture source
Select input capture source in the GTICASR register and GTICBSR register.
In Figure 23.26, GTICASR = 0000 0F00h, and in Figure 23.27, GTICBSR = 0000 F000h.

Set buffer operation
Set buffer operation with CCRA and CCRB in GTBER.
In Figure 23.26, CCRA[1:0] = 01b, and in Figure 23.27, CCRB = 1xb.

Start count operation
Set GTCR.CST to 1 to start count operation.

Figure 23.28 Example setting for GTCCRA and GTCCRB buffer operation with 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 setting for saw-wave PWM mode.

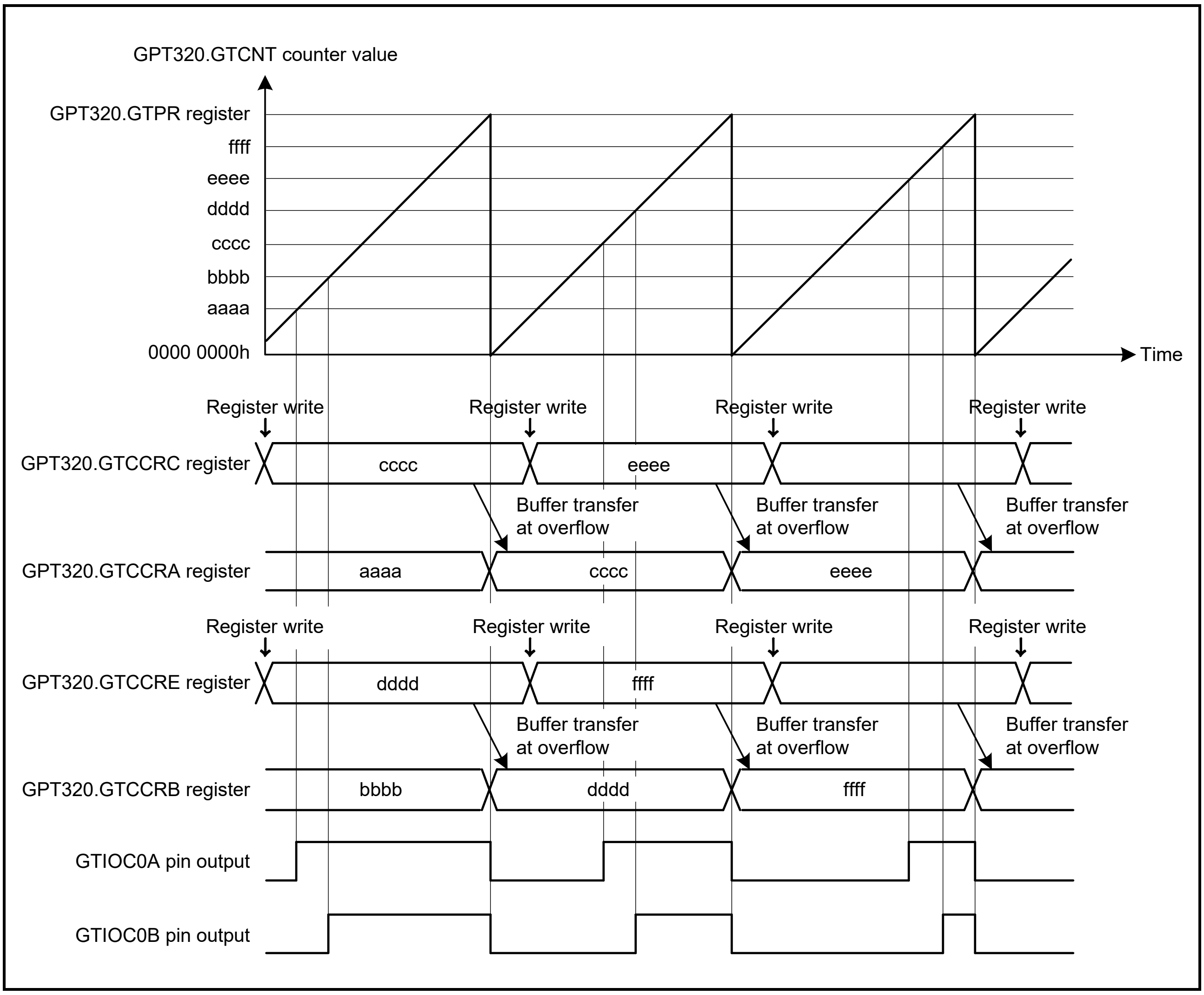
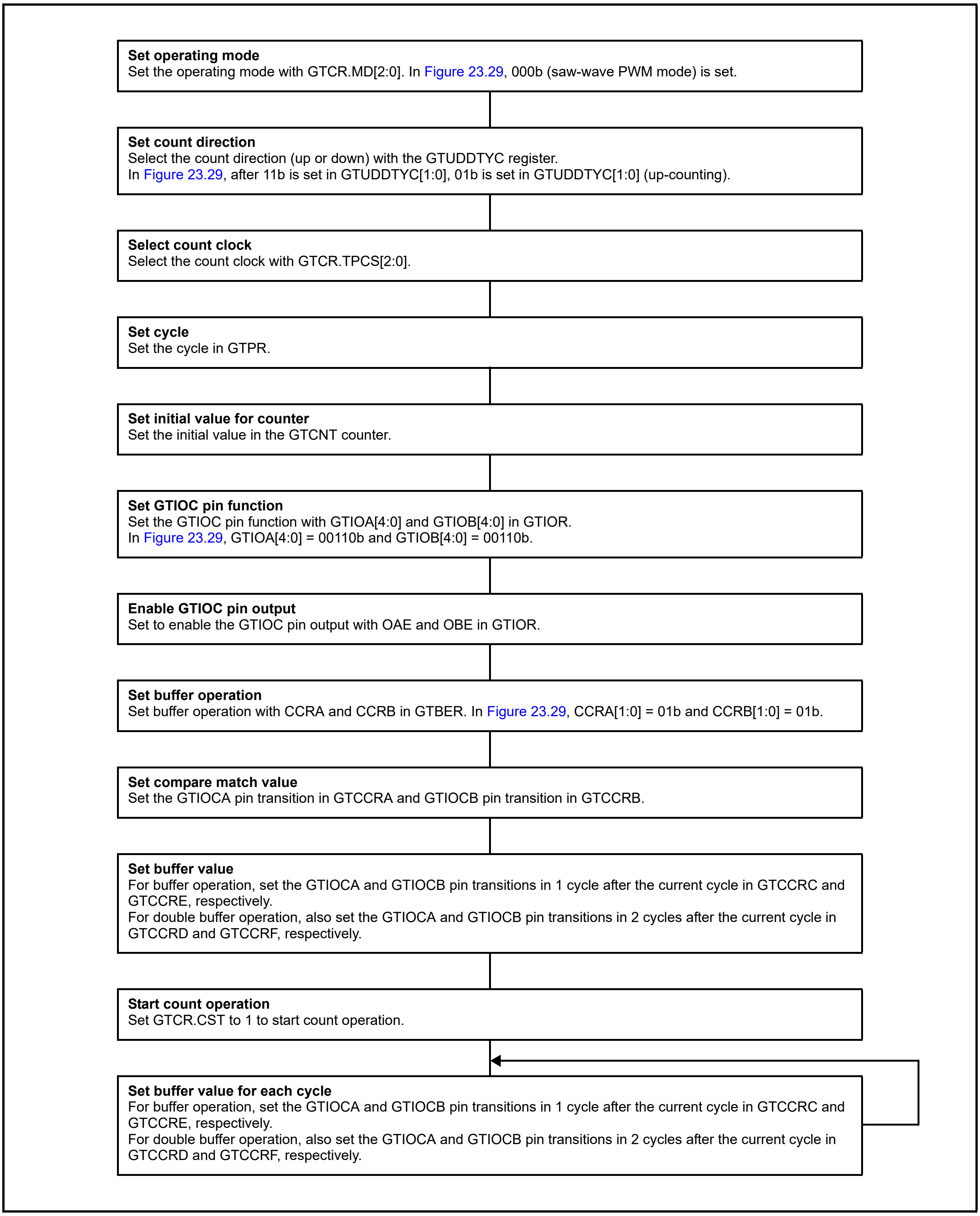

GPT320.GTCNT counter value
GPT320.GTPR register
ffff
eeee
dddd
cccc
bbbb
aaaa
0000 0000h
Time
Register write
Register write
Register write
Register write
GPT320.GTCCRC register
cccc
eeee
Buffer transfer at overflow
Buffer transfer at overflow
Buffer transfer at overflow
GPT320.GTCCRA register
aaaa
cccc
eeee
Register write
Register write
Register write
Register write
GPT320.GTCCRE register
dddd
ffff
Buffer transfer at overflow
Buffer transfer at overflow
Buffer transfer at overflow
GPT320.GTCCRB register
bbbb
dddd
ffff
GTIOC0A pin output
GTIOC0B pin output

Figure 23.29 Example of saw-wave PWM mode operation with up-counting, buffer operation, high output at GTCCRA/GTCCRB compare match, and low output at cycle end



Set operating mode
Set the operating mode with GTCR.MD[2:0]. In Figure 23.29, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.29, 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.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.29, GTIOA[4:0] = 00110b and GTIOB[4:0] = 00110b.

Enable GTIOC pin output
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set buffer operation
Set buffer operation with CCRA and CCRB in GTBER. In Figure 23.29, CCRA[1:0] = 01b and CCRB[1:0] = 01b.

Set compare match value
Set the GTIOCA pin transition in GTCCRA and GTIOCB pin transition in GTCCRB.

Set buffer value
For buffer operation, set the GTIOCA and GTIOCB pin transitions in 1 cycle after the current cycle in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA and GTIOCB pin transitions in 2 cycles after the current cycle in GTCCRD and GTCCRF, respectively.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
For buffer operation, set the GTIOCA and GTIOCB pin transitions in 1 cycle after the current cycle in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA and GTIOCB pin transitions in 2 cycles after the current cycle in GTCCRD and GTCCRF, respectively.

Figure 23.30 Example setting for 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:

- 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 is stopped, the buffer is transferred forcibly from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B. By setting GTDTCR 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 setting for saw-wave one-shot pulse mode.

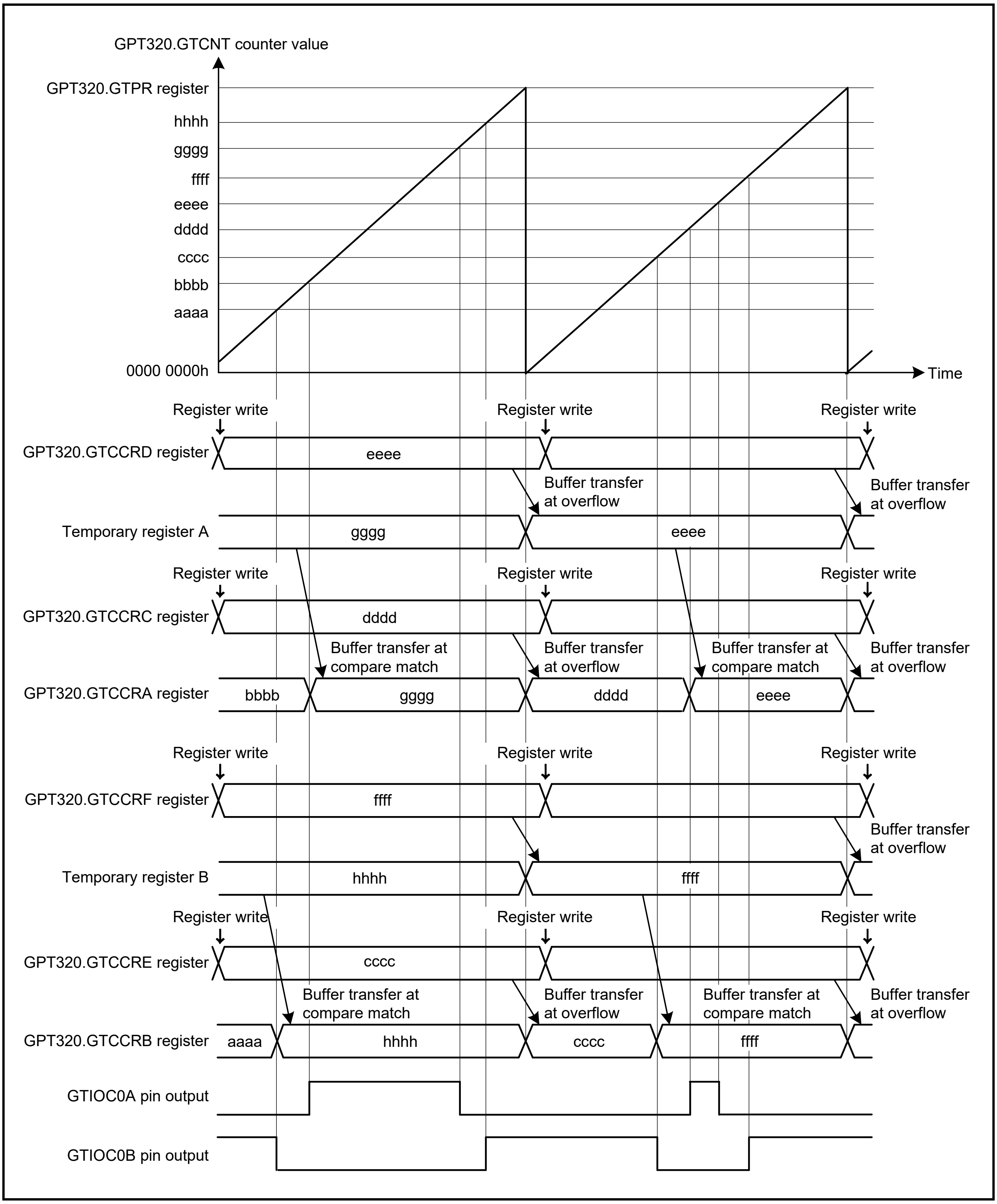
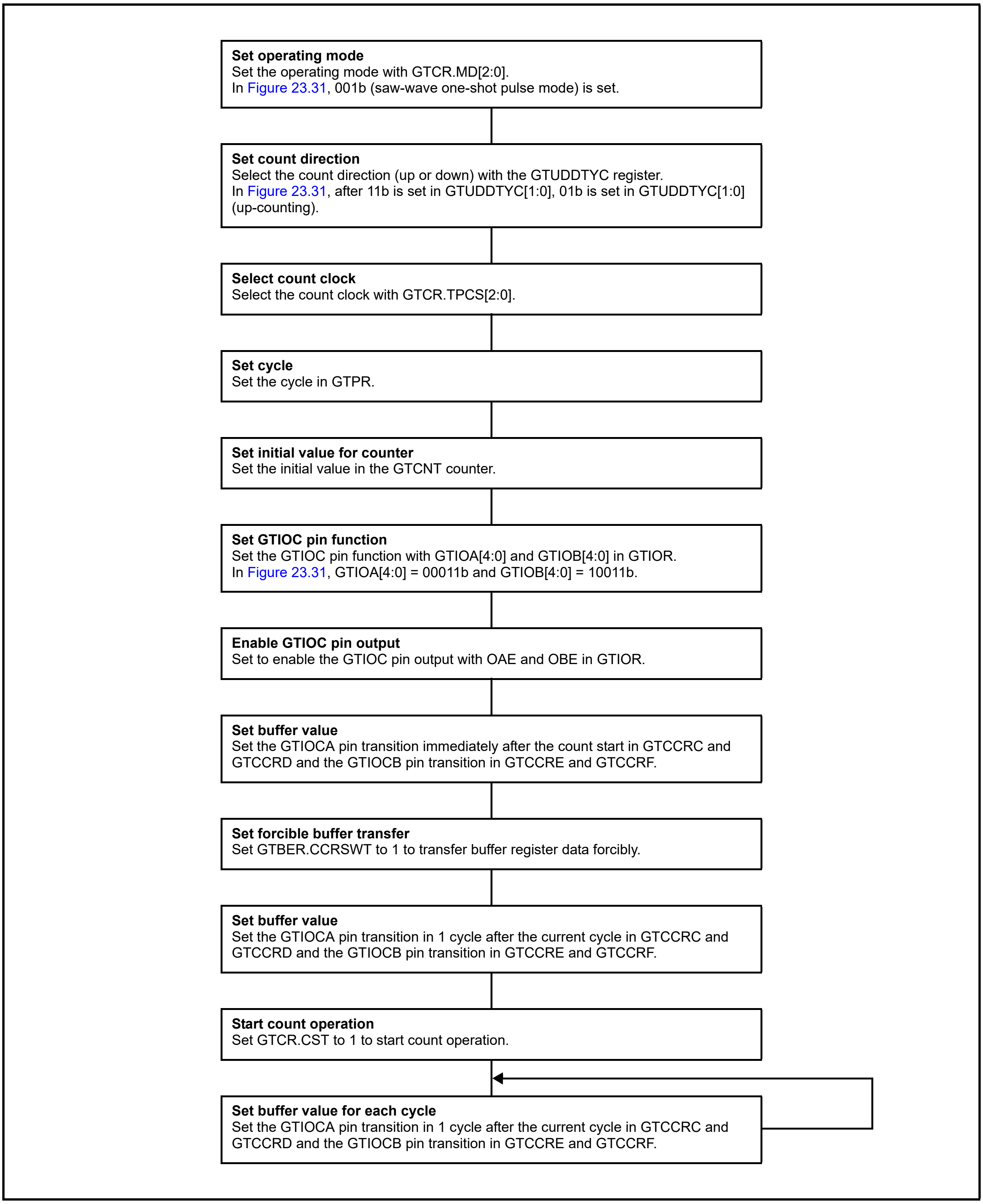

GPT320.GTCNT counter value
GPT320.GTPR register
hhhh
gggg
ffff
eeee
dddd
cccc
bbbb
aaaa
0000 0000h
Time
Register write
GPT320.GTCCRD register
eeee
Buffer transfer at overflow
Temporary register A
gggg
eeee
GPT320.GTCCRC register
dddd
Buffer transfer at compare match
GPT320.GTCCRA register
bbbb
gggg
dddd
eeee
GPT320.GTCCRF register
ffff
Temporary register B
hhhh
ffff
GPT320.GTCCRE register
cccc
GPT320.GTCCRB register
aaaa
hhhh
cccc
ffff
GTIOC0A pin output
GTIOC0B pin output

Figure 23.31 Example of saw-wave one-shot pulse mode operation with 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, and output retained at cycle end



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.31, 001b (saw-wave one-shot pulse mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.31, 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.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.31, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.

Enable GTIOC pin output
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set buffer value
Set the GTIOCA pin transition immediately after the count start in GTCCRC and GTCCRD and the GTIOCB pin transition in GTCCRE and GTCCRF.

Set forcible buffer transfer
Set GTBER.CCRSWT to 1 to transfer buffer register data forcibly.

Set buffer value
Set the GTIOCA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD and the GTIOCB pin transition in GTCCRE and GTCCRF.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
Set the GTIOCA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD and the GTIOCB pin transition in GTCCRE and GTCCRF.

Figure 23.32 Example setting for 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 based on 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 setting for a triangle-wave PWM mode 1.

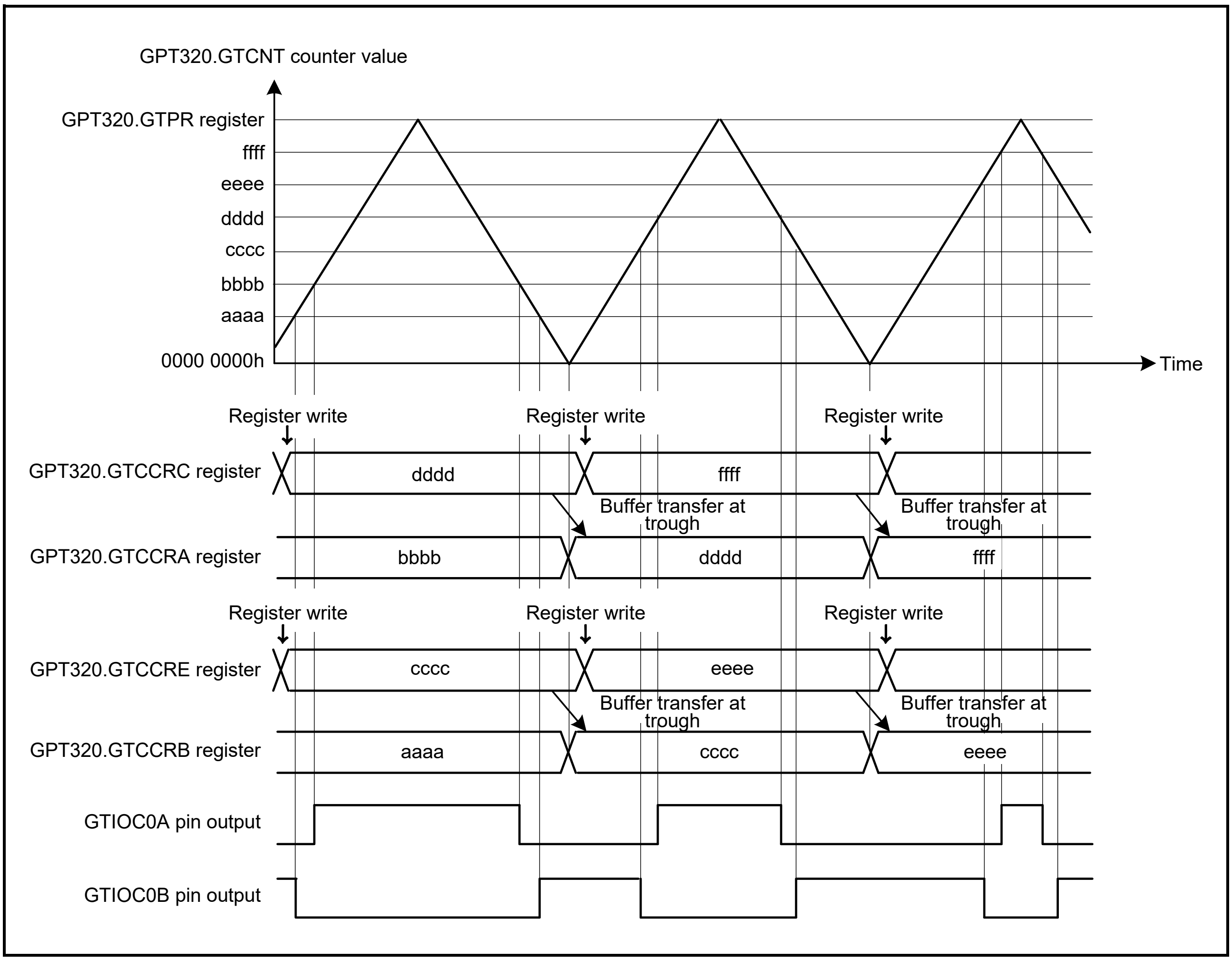
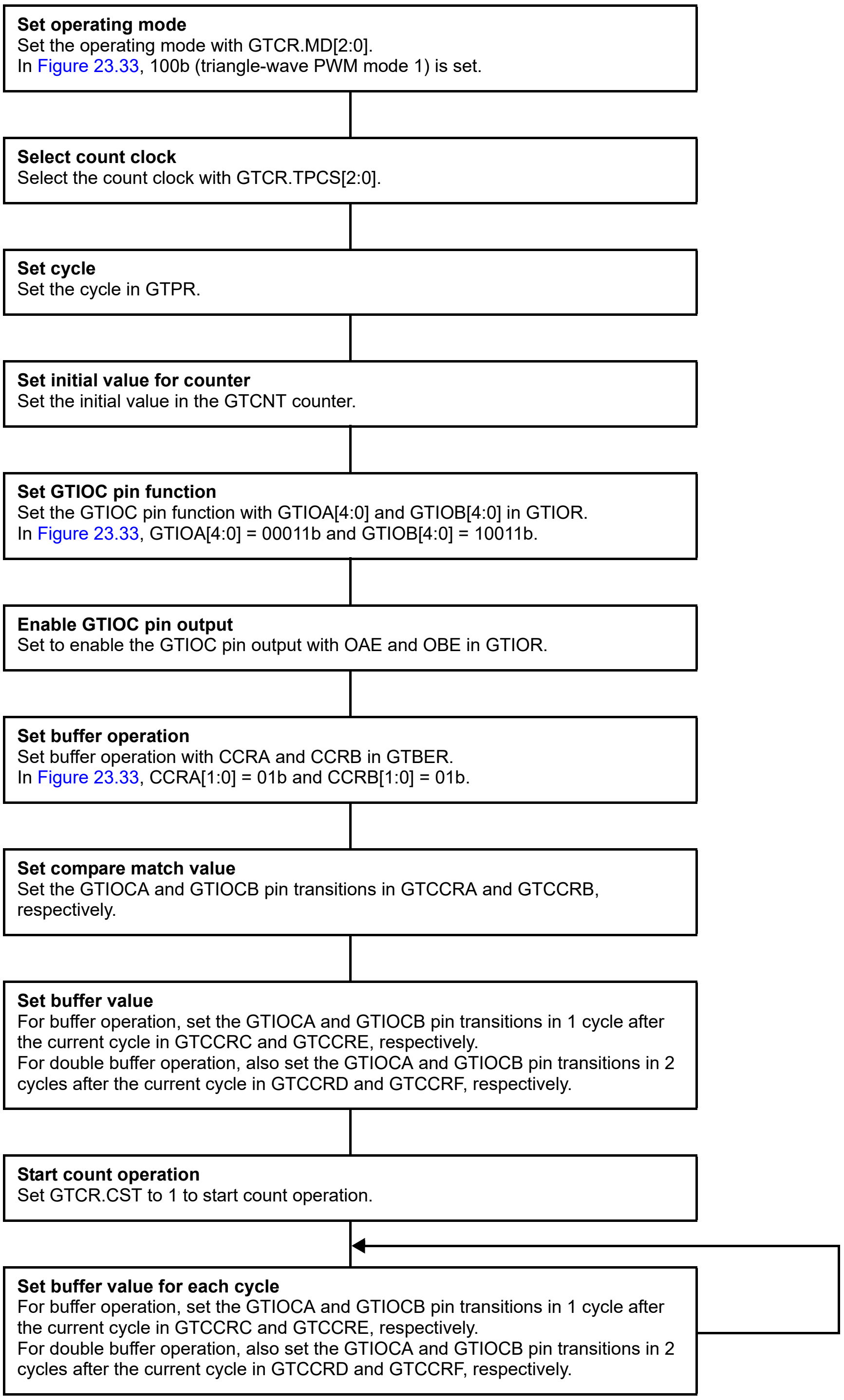

GPT320.GTCNT counter value
GPT320.GTPR register
ffff
eeee
dddd
cccc
bbbb
aaaa
0000 0000h
Time
Register write
Register write
Register write
GPT320.GTCCRC register
dddd
ffff
Buffer transfer at trough
Buffer transfer at trough
GPT320.GTCCRA register
bbbb
dddd
ffff
Register write
Register write
Register write
GPT320.GTCCRE register
cccc
eeee
Buffer transfer at trough
Buffer transfer at trough
GPT320.GTCCRB register
aaaa
cccc
eeee
GTIOC0A pin output
GTIOC0B pin output

Figure 23.33 Example of triangle-wave PWM mode 1 operation with 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, and output retained at cycle end



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.33, 100b (triangle-wave PWM mode 1) is set.

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.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.33, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.

Enable GTIOC pin output
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set buffer operation
Set buffer operation with CCRA and CCRB in GTBER.
In Figure 23.33, CCRA[1:0] = 01b and CCRB[1:0] = 01b.

Set compare match value
Set the GTIOCA and GTIOCB pin transitions in GTCCRA and GTCCRB, respectively.

Set buffer value
For buffer operation, set the GTIOCA and GTIOCB pin transitions in 1 cycle after the current cycle in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA and GTIOCB pin transitions in 2 cycles after the current cycle in GTCCRD and GTCCRF, respectively.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
For buffer operation, set the GTIOCA and GTIOCB pin transitions in 1 cycle after the current cycle in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA and GTIOCB pin transitions in 2 cycles after the current cycle in GTCCRD and GTCCRF, respectively.

Figure 23.34 Example setting for triangle-wave PWM mode 1

23.3.3.4 Triangle-wave PWM mode 2 (32-bit transfer at crest and trough)

Similar 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 based on 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 setting for triangle-wave PWM mode 2.

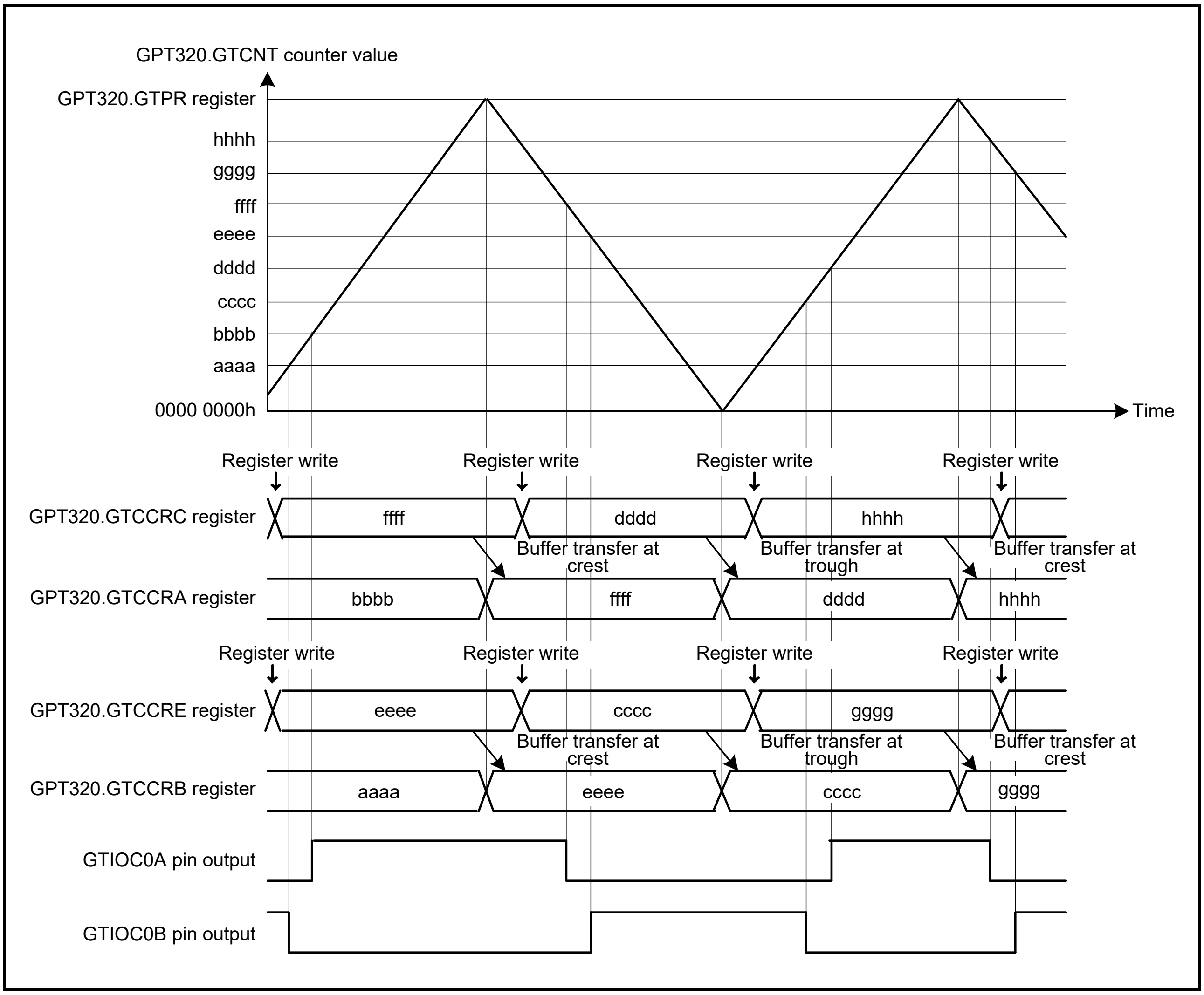
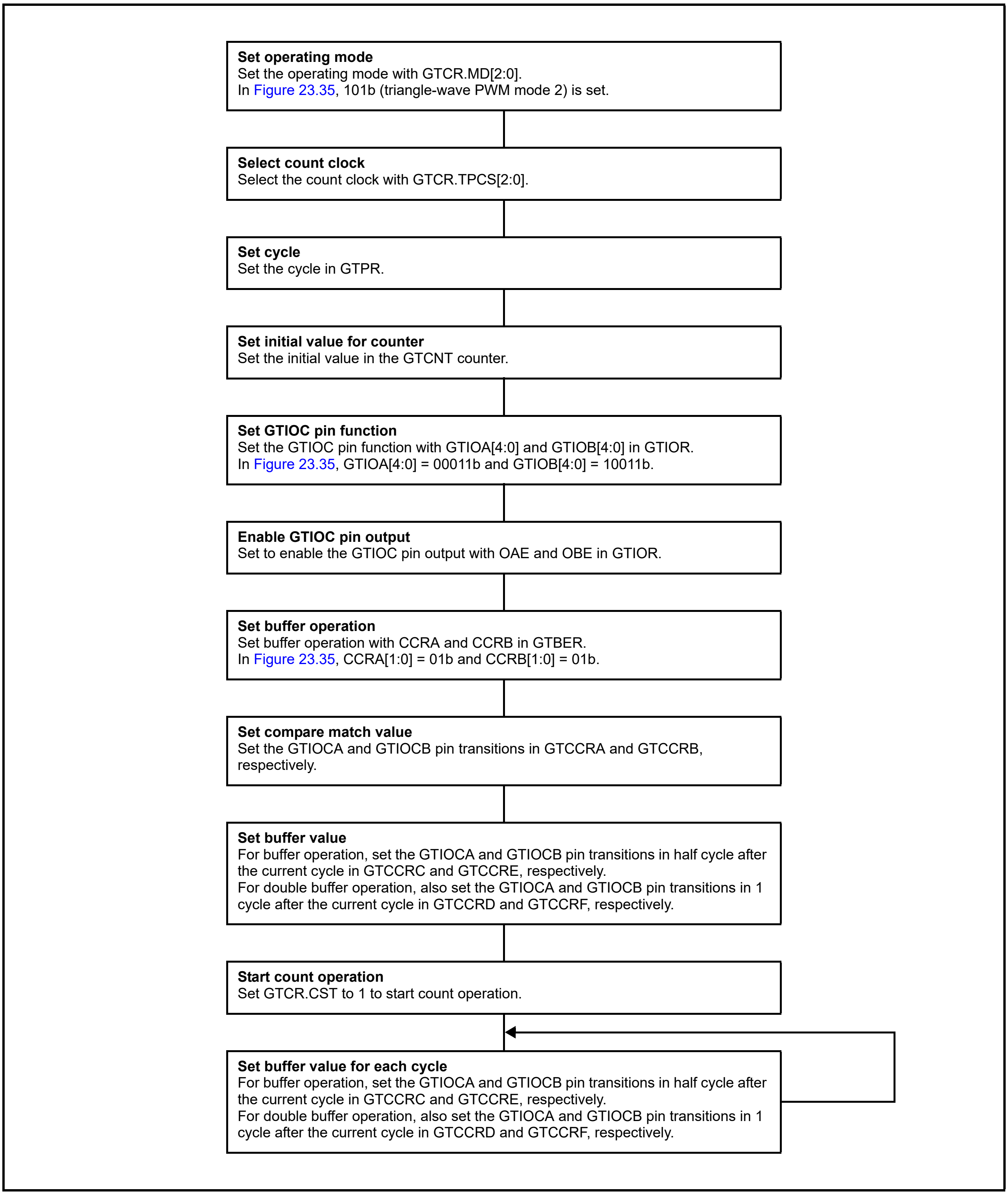

GPT320.GTCNT counter value
GPT320.GTPR register
hhhh
gggg
ffff
eeee
dddd
cccc
bbbb
aaaa
0000 0000h
Time
Register write
Register write
Register write
Register write
GPT320.GTCCRC register
ffff
dddd
hhhh
Buffer transfer at crest
Buffer transfer at trough
Buffer transfer at crest
GPT320.GTCCRA register
bbbb
ffff
dddd
hhhh
Register write
Register write
Register write
Register write
GPT320.GTCCRE register
eeee
cccc
gggg
Buffer transfer at crest
Buffer transfer at trough
Buffer transfer at crest
GPT320.GTCCRB register
aaaa
eeee
cccc
gggg
GTIOC0A pin output
GTIOC0B pin output

Figure 23.35 Example of triangle-wave PWM mode 2 operation with 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, and output retained at cycle end



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.35, 101b (triangle-wave PWM mode 2) is set.

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.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.35, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.

Enable GTIOC pin output
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set buffer operation
Set buffer operation with CCRA and CCRB in GTBER.
In Figure 23.35, CCRA[1:0] = 01b and CCRB[1:0] = 01b.

Set compare match value
Set the GTIOCA and GTIOCB pin transitions in GTCCRA and GTCCRB, respectively.

Set buffer value
For buffer operation, set the GTIOCA and GTIOCB pin transitions in half cycle after the current cycle in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA and GTIOCB pin transitions in 1 cycle after the current cycle in GTCCRD and GTCCRF, respectively.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
For buffer operation, set the GTIOCA and GTIOCB pin transitions in half cycle after the current cycle in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA and GTIOCB pin transitions in 1 cycle after the current cycle in GTCCRD and GTCCRF, respectively.

Figure 23.36 Example setting for 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:

- 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 setting for triangle-wave PWM mode 3.

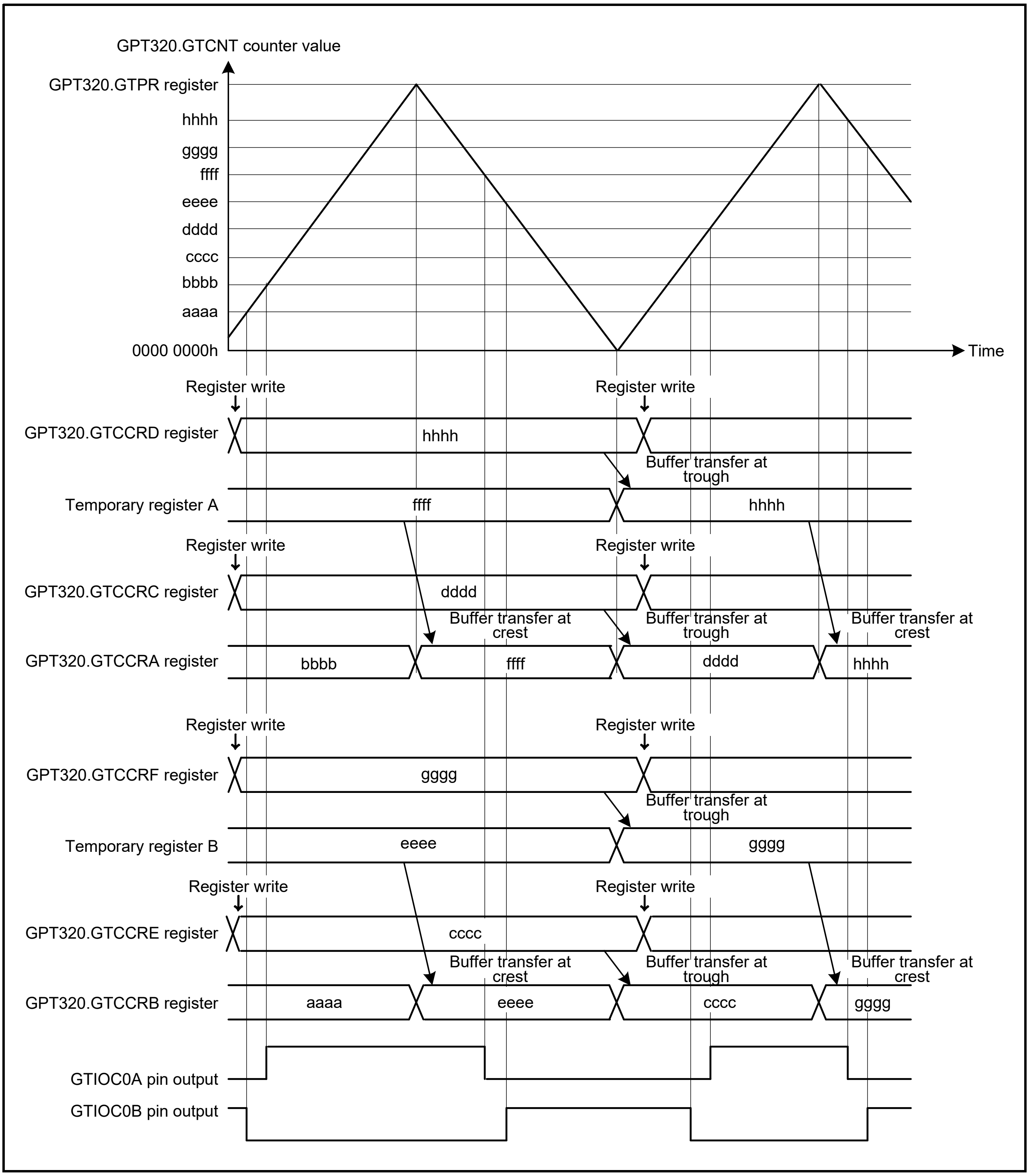
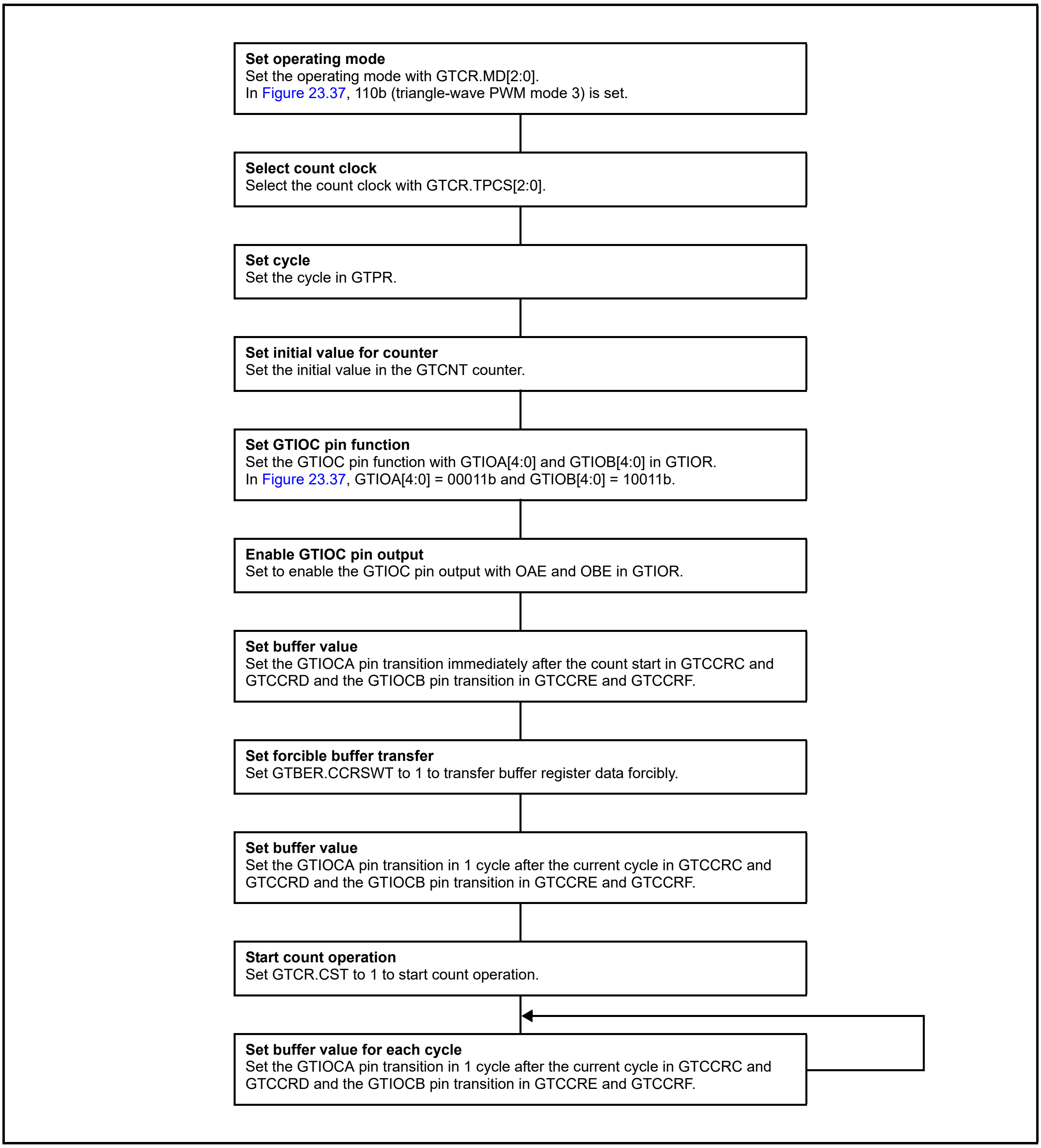

GPT320.GTCNT counter value
GPT320.GTPR register
hhhh
gggg
ffff
eeee
dddd
cccc
bbbb
aaaa
0000 0000h
Time
Register write
GPT320.GTCCRD register
hhhh
Buffer transfer at trough
Temporary register A
ffff
hhhh
GPT320.GTCCRC register
dddd
Buffer transfer at crest
GPT320.GTCCRA register
bbbb
ffff
dddd
hhhh
GPT320.GTCCRF register
gggg
Temporary register B
eeee
gggg
GPT320.GTCCRE register
cccc
GPT320.GTCCRB register
aaaa
eeee
cccc
gggg
GTIOC0A pin output
GTIOC0B pin output

Figure 23.37 Example of triangle-wave PWM mode 3 operation with low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/GTCCRB compare match, and output retained at cycle end



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.37, 110b (triangle-wave PWM mode 3) is set.

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.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.37, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.

Enable GTIOC pin output
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set buffer value
Set the GTIOCA pin transition immediately after the count start in GTCCRC and GTCCRD and the GTIOCB pin transition in GTCCRE and GTCCRF.

Set forcible buffer transfer
Set GTBER.CCRSWT to 1 to transfer buffer register data forcibly.

Set buffer value
Set the GTIOCA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD and the GTIOCB pin transition in GTCCRE and GTCCRF.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
Set the GTIOCA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD and the GTIOCB pin transition in GTCCRE and GTCCRF.

Figure 23.38 Example setting for 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.

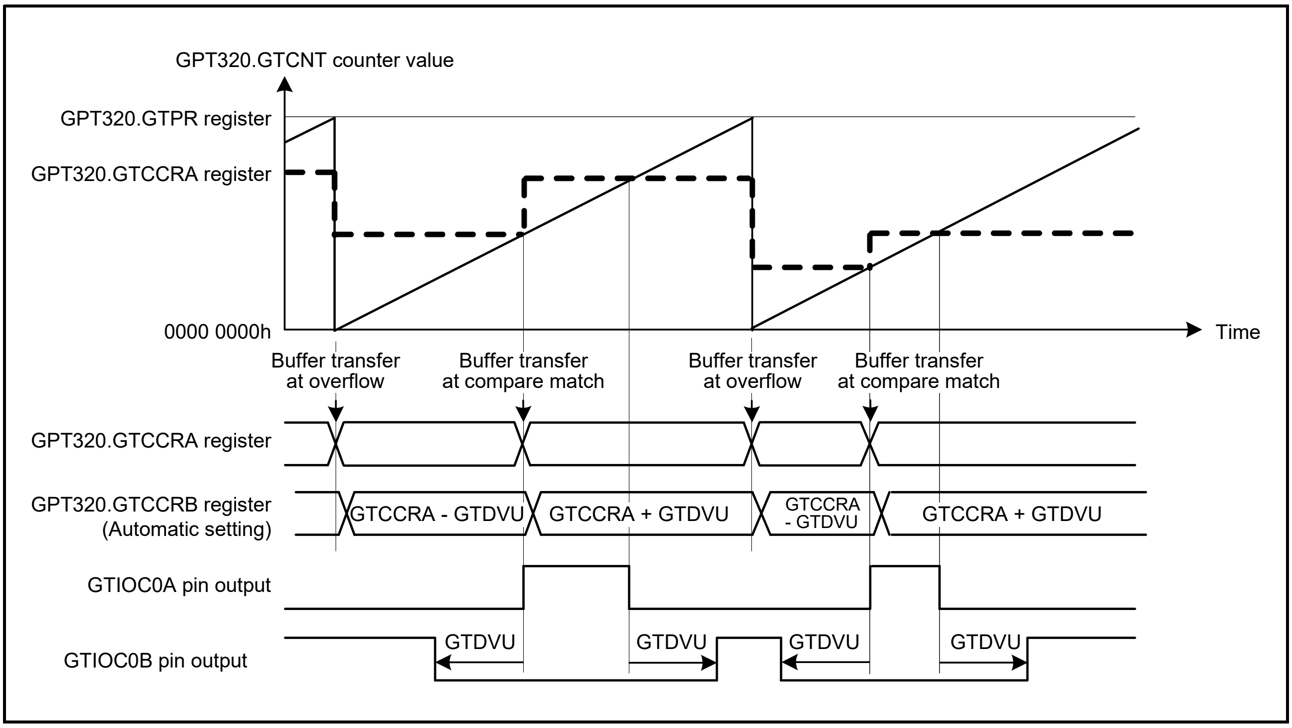

GPT320.GTCNT counter value
GPT320.GTPR register
GPT320.GTCCRA register
0000 0000h
Time
Buffer transfer at overflow
Buffer transfer at compare match
Buffer transfer at overflow
Buffer transfer at compare match
GPT320.GTCCRA register
GPT320.GTCCRB register (Automatic setting)
GTCCRA - GTDVU
GTCCRA + GTDVU
GTCCRA - GTDVU
GTCCRA + GTDVU
GTIOC0A pin output
GTIOC0B pin output
GTDVU
GTDVU
GTDVU
GTDVU

Figure 23.39 Example of automatic dead time setting function operation with saw-wave one-shot pulse mode, up-counting, and active-high

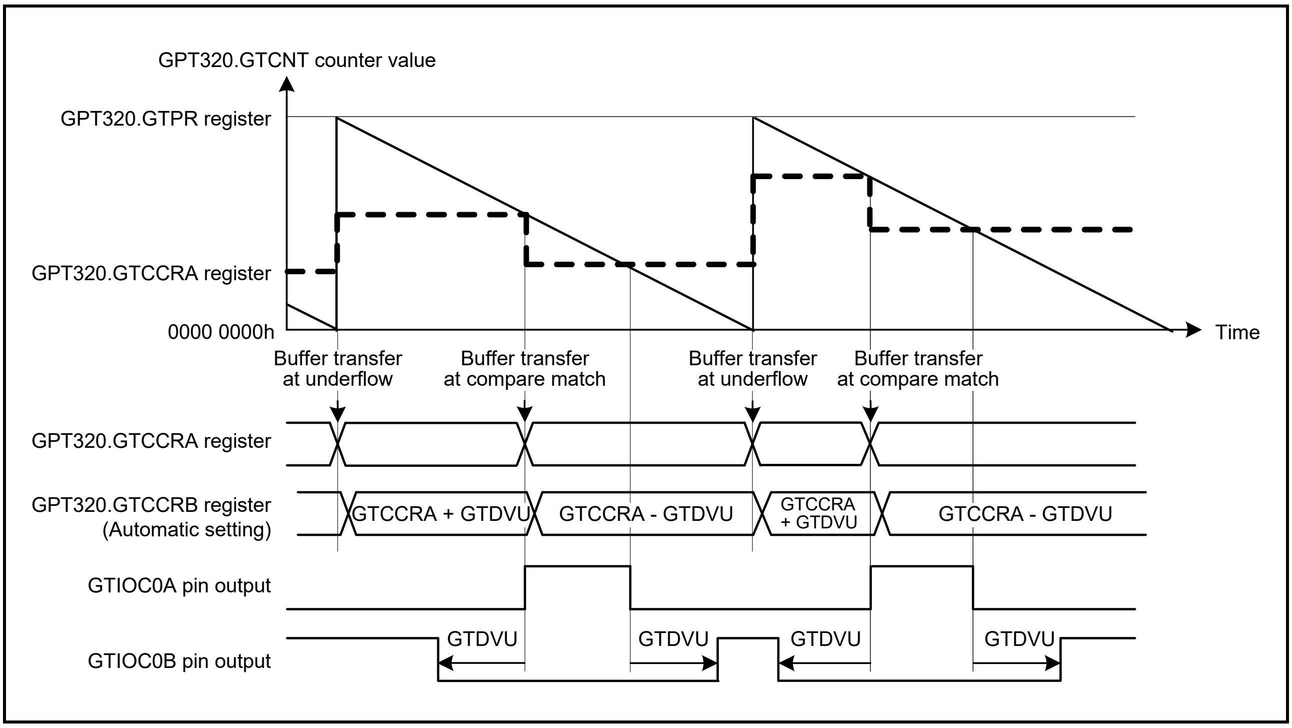

GPT320.GTCNT counter value
GPT320.GTPR register
GPT320.GTCCRA register
0000 0000h
Time
Buffer transfer at underflow
Buffer transfer at compare match
Buffer transfer at underflow
Buffer transfer at compare match
GPT320.GTCCRA register
GPT320.GTCCRB register (Automatic setting)
GTCCRA + GTDVU
GTCCRA - GTDVU
GTCCRA + GTDVU
GTCCRA - GTDVU
GTIOC0A pin output
GTIOC0B pin output
GTDVU
GTDVU
GTDVU
GTDVU

Figure 23.40 Example of automatic dead time setting function operation with saw-wave one-shot pulse mode, down-counting, and active-high

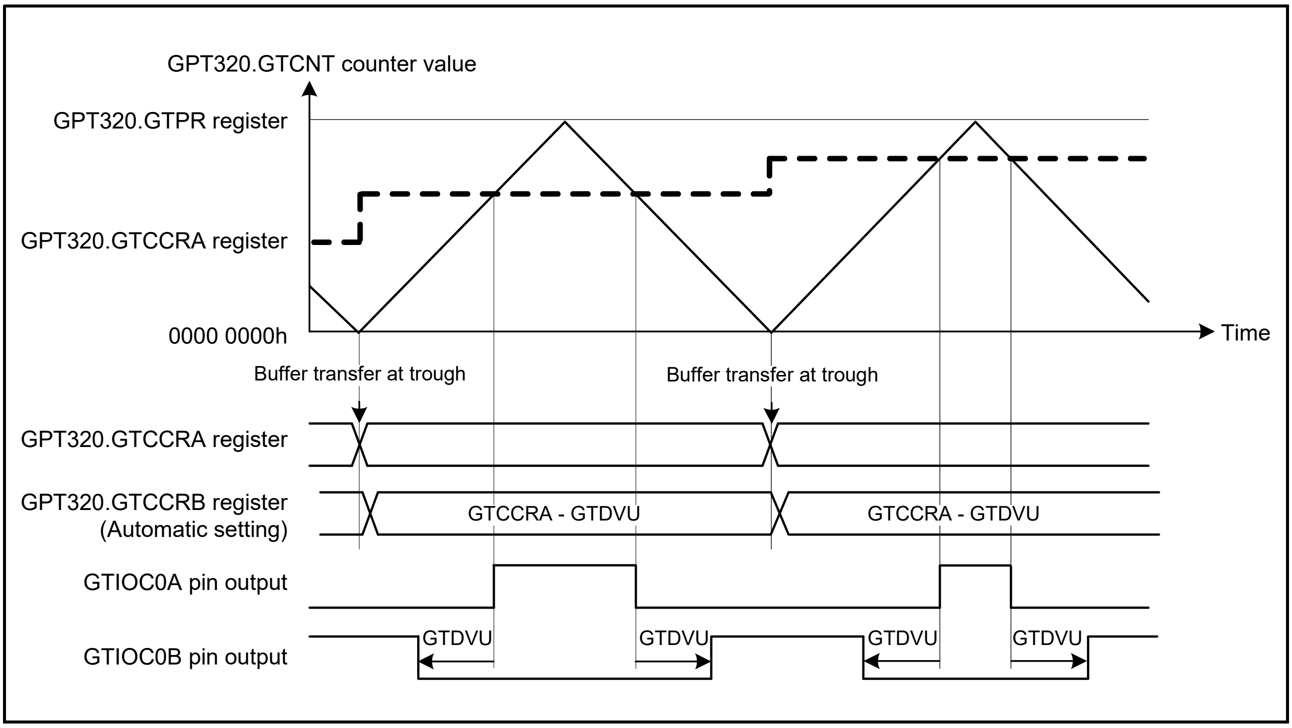

GPT320.GTCNT counter value
GPT320.GTPR register
GPT320.GTCCRA register
0000 0000h
Time
Buffer transfer at trough
Buffer transfer at trough
GPT320.GTCCRA register
GPT320.GTCCRB register
(Automatic setting)
GTCCRA - GTDVU
GTCCRA - GTDVU
GTIOC0A pin output
GTIOC0B pin output
GTDVU
GTDVU
GTDVU
GTDVU

Figure 23.41 Example of automatic compare-match value setting function with dead time with triangle-wave PWM mode 1, and active-high

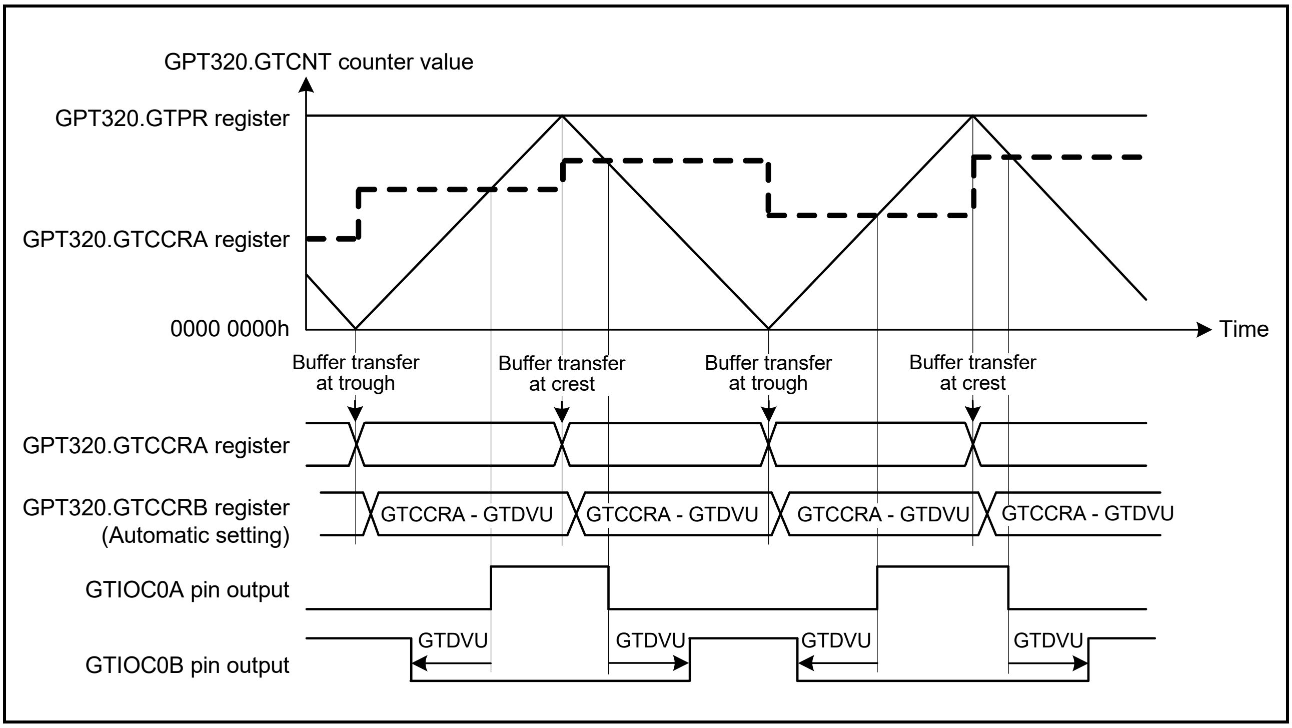
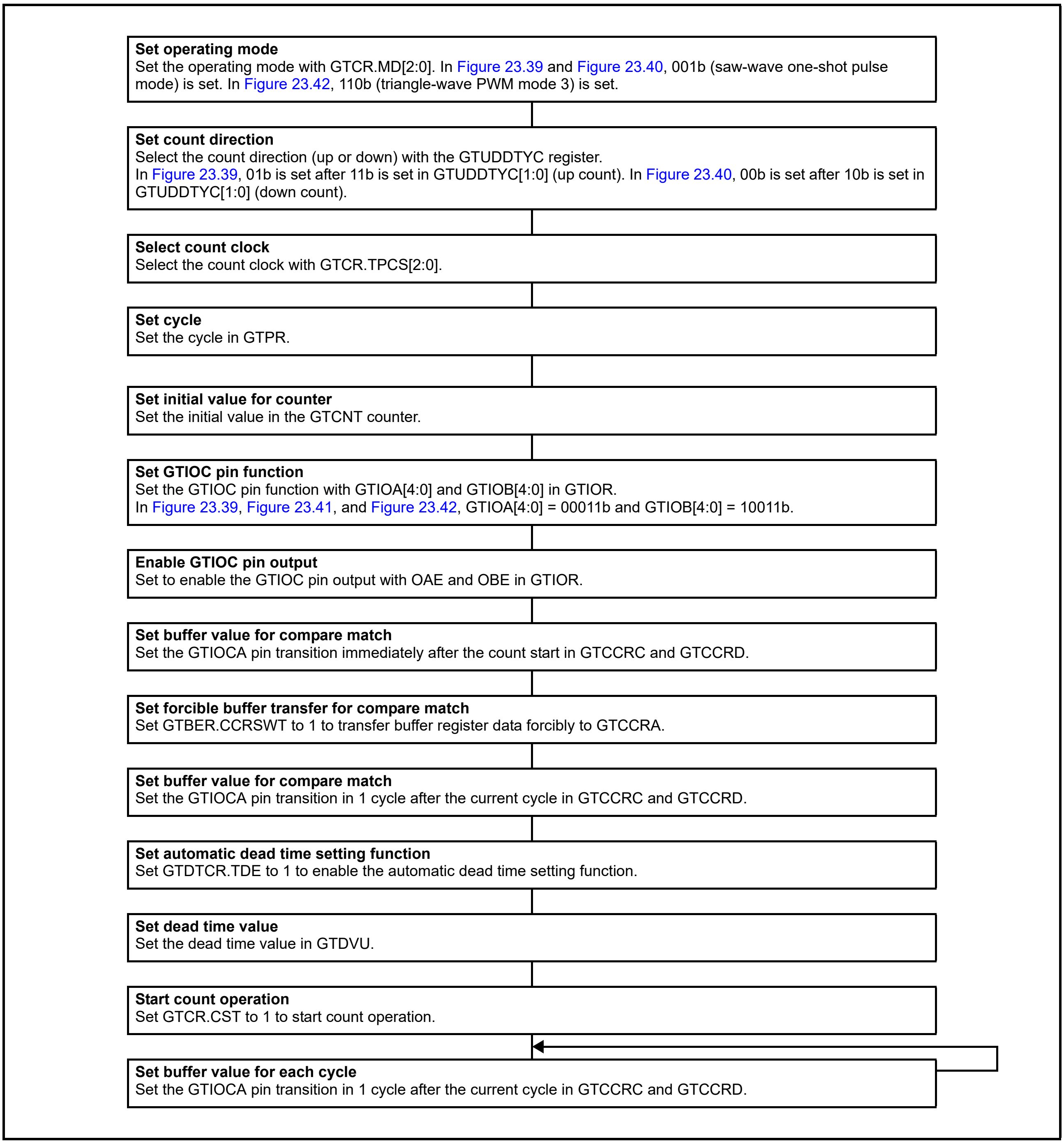

GPT320.GTCNT counter value
GPT320.GTPR register
GPT320.GTCCRA register
0000 0000h
Time
Buffer transfer at trough
Buffer transfer at crest
Buffer transfer at trough
Buffer transfer at crest
GPT320.GTCCRA register
GPT320.GTCCRB register
(Automatic setting)
GTCCRA - GTDVU
GTCCRA - GTDVU
GTCCRA - GTDVU
GTCCRA - GTDVU
GTIOC0A pin output
GTIOC0B pin output
GTDVU
GTDVU
GTDVU
GTDVU

Figure 23.42 Example of automatic compare-match value setting function with dead time with triangle-wave PWM mode 2 or 3, and active-high



Set operating mode
Set the operating mode with GTCR.MD[2:0]. In Figure 23.39 and Figure 23.40, 001b (saw-wave one-shot pulse mode) is set. In Figure 23.42, 110b (triangle-wave PWM mode 3) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.39, 01b is set after 11b is set in GTUDDTYC[1:0] (up count). In Figure 23.40, 00b is set after 10b is set in GTUDDTYC[1:0] (down count).

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.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.39, Figure 23.41, and Figure 23.42, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.

Enable GTIOC pin output
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set buffer value for compare match
Set the GTIOCA pin transition immediately after the count start in GTCCRC and GTCCRD.

Set forcible buffer transfer for compare match
Set GTBER.CCRSWT to 1 to transfer buffer register data forcibly to GTCCRA.

Set buffer value for compare match
Set the GTIOCA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD.

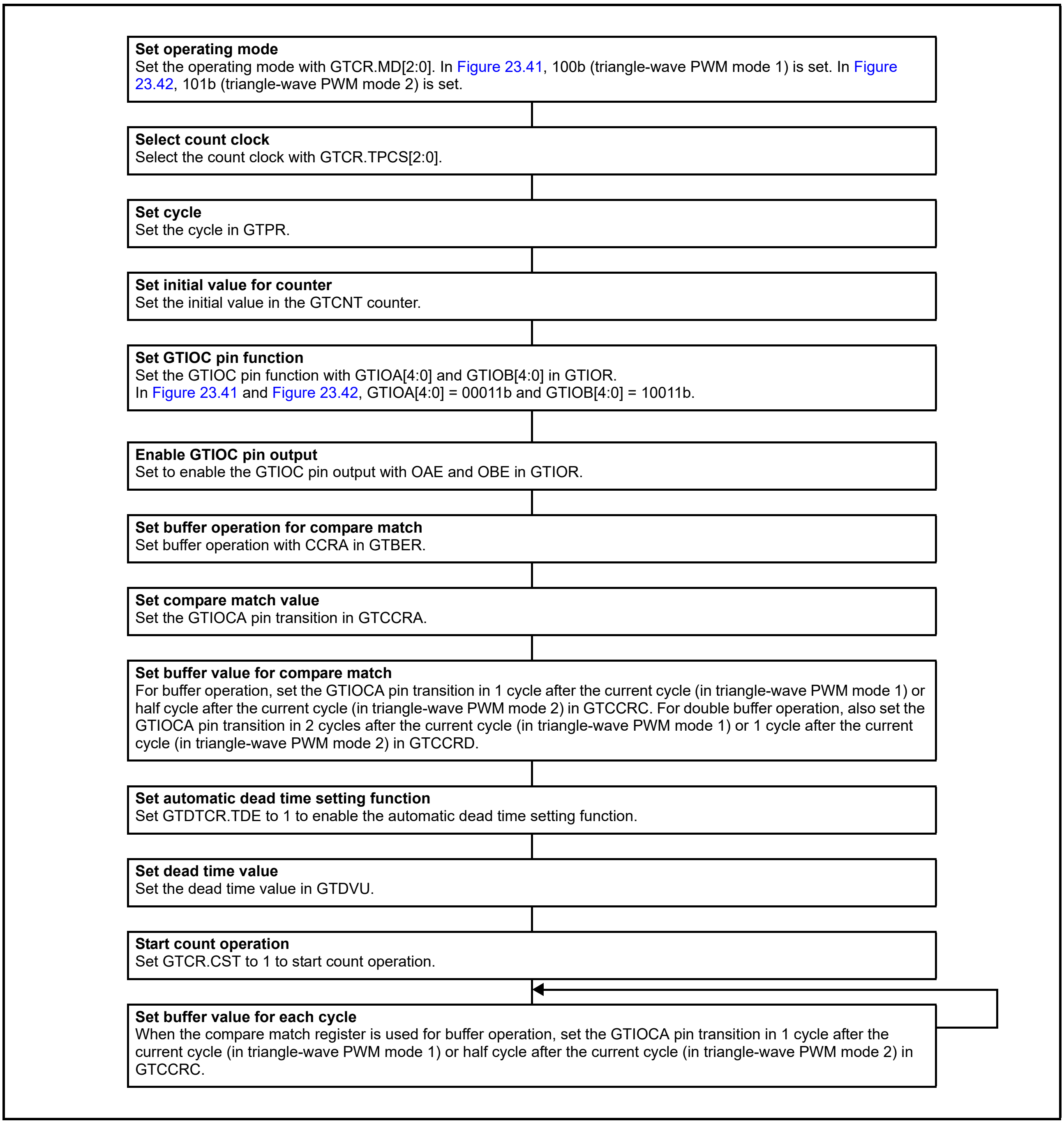
Set automatic dead time setting function
Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.

Set dead time value
Set the dead time value in GTDVU.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
Set the GTIOCA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD.

Figure 23.43 Example setting for automatic dead time setting function with saw-wave one-shot pulse mode, and triangle-wave PWM mode 3



Set operating mode
Set the operating mode with GTCR.MD[2:0]. In Figure 23.41, 100b (triangle-wave PWM mode 1) is set. In Figure 23.42, 101b (triangle-wave PWM mode 2) is set.

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.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.41 and Figure 23.42, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.

Enable GTIOC pin output
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set buffer operation for compare match
Set buffer operation with CCRA in GTBER.

Set compare match value
Set the GTIOCA pin transition in GTCCRA.

Set buffer value for compare match
For buffer operation, set the GTIOCA pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRC. For double buffer operation, also set the GTIOCA pin transition in 2 cycles after the current cycle (in triangle-wave PWM mode 1) or 1 cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRD.

Set automatic dead time setting function
Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.

Set dead time value
Set the dead time value in GTDVU.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
When the compare match register is used for buffer operation, set the GTIOCA pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRC.

Figure 23.44 Example setting for automatic dead time setting function in 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 is stopped 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 is stopped 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 is stopped and GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit value is not reflected to the count operation. If the GTUDDTYC.UDF bit is set to 1 while the count operation is stopped, the GTUDDTYC.UD bit value at that time is reflected at the start of counting.

If the count direction changes during a saw-wave count operation, the GTPR value after the start of up-counting is reflected in the count cycle during up-counting and the GTPR value before the start of down-counting is reflected during down-counting.

Figure 23.45 shows an example of count direction changing function operation.

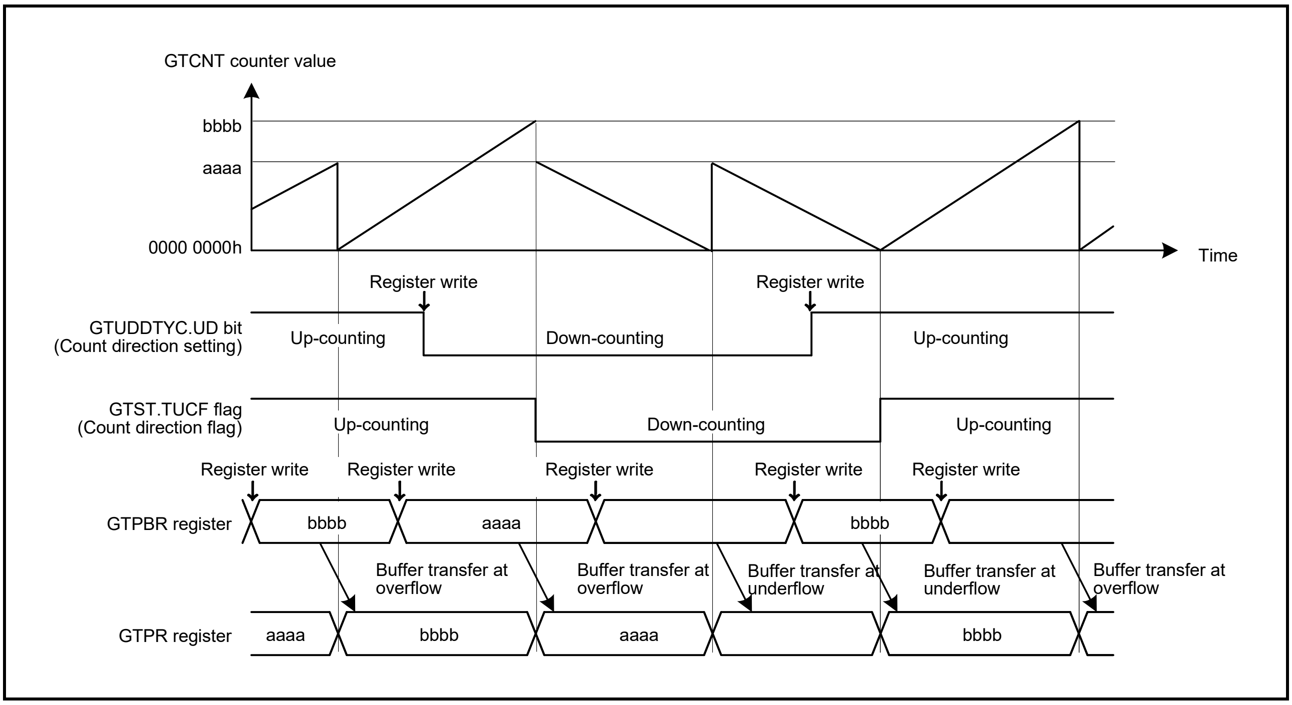

GTCNT counter value
bbbb
aaaa
0000 0000h
Time
Register write
Register write
GTUDDTYC.UD bit
(Count direction setting)
Up-counting
Down-counting
Up-counting
GTST.TUCF flag
(Count direction flag)
Up-counting
Down-counting
Up-counting
Register write
Register write
Register write
Register write
Register write
GTPBR register
bbbb
aaaa
bbbb
Buffer transfer at overflow
Buffer transfer at overflow
Buffer transfer at underflow
Buffer transfer at underflow
Buffer transfer at overflow
GTPR register
aaaa
bbbb
aaaa
bbbb

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 is 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 is stopped and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 0, the output duty modification is not reflected at the start of counting. The output duty changes at an overflow or an underflow. If the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is set to 1 while the count operation is stopped, 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 at an underflow.

If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 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 is stopped and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 1, the output duty modification is reflected at the start of counting.

In performing 0% or 100% duty operation, GPT internally continues to:

- Perform compare match operation
- Set compare match flag

- Output interrupt
- Perform buffer operation.

When the control is changed from 0% or 100% duty setting to compare match, the output value of GTIOCA pin at cycle end is determined 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.

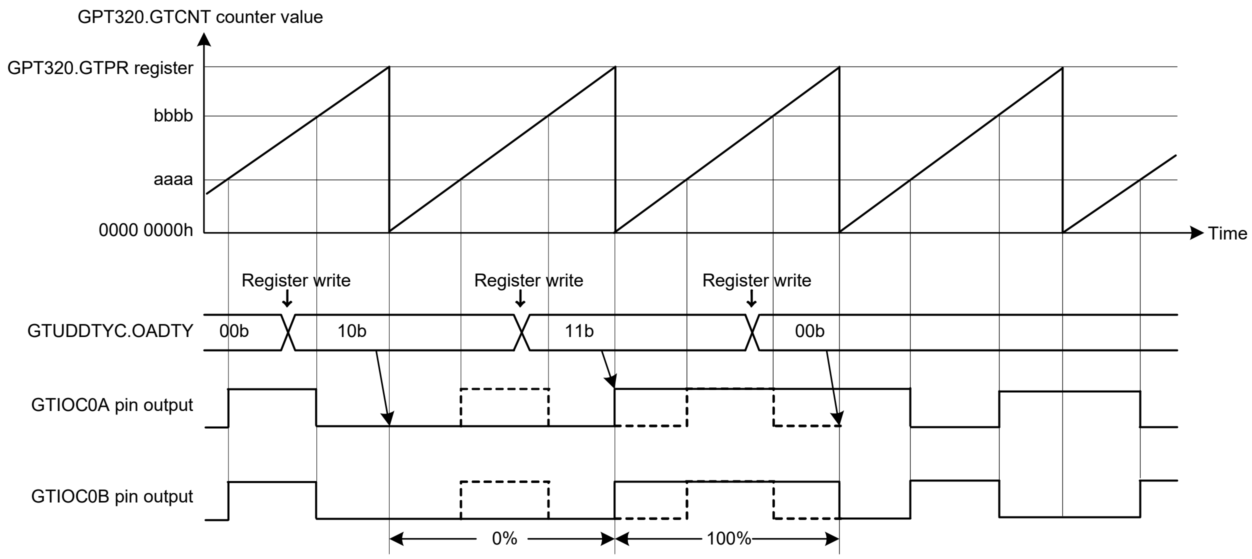
When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 01b, the output pins output low at cycle end. When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 10b, the output pins output high at cycle end.

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% or 100% duty setting (m = A, B)

GTIOR.GTIOm[3:2]	Compare match value at cycle end masked by 0% or 100% duty setting	GTUDDTYC.OmDTYR in duty 0% setting		GTUDDTYC.OmDTYR in duty 100% setting	
		0	**1**	**0**	**1**
00 (output retained at cycle end)	0	0	0	1	0
	1	0	1	1	1
01 (low output at cycle end)	—	0	0	0	0
10 (high output at cycle end)	—	1	1	1	1
11 (output toggled at cycle end)	0	1	1	0	1
	1	1	0	0	0

Figure 23.46 shows an example of output duty 0% and 100% function.


GPT320.GTCNT counter value
GPT320.GTPR register
bbbb
aaaa
0000 0000h
Time
Register write
Register write
Register write
GTUDDTYC.OADTY
00b
10b
11b
00b
GTIOC0A pin output
GTIOC0B pin output
0%
100%

[Setting examples]

GPT320.GTIOR.GTIOA[4:0] bits: 00011b
Initial low output, output toggled at compare match, output retained at cycle end

GPT320.GTUDDTYC.OADTYR bit: 0b
Applied the value of duty 0% or 100% output to GTIOA[3:2] bits function after 0% or 100% duty setting is released

GPT320.GTIOR.GTIOB[4:0] bits: 00011b
Initial low output, output toggled at compare match, output retained at cycle end

GPT320.GTUDDTYC.OBDTYR bit: 1
Applied the value of masked compare match output to GTIOB[3:2] bits function after 0% or 100% duty setting is released

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.

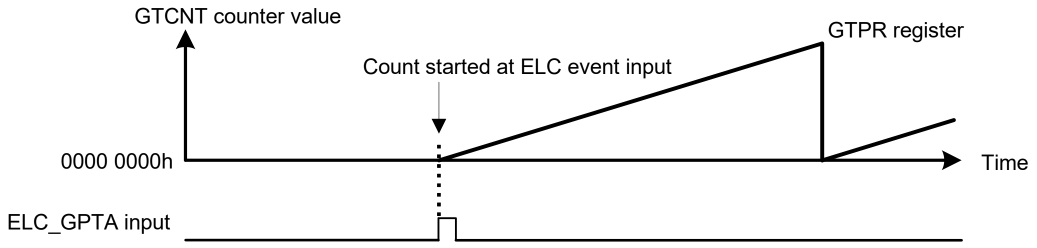
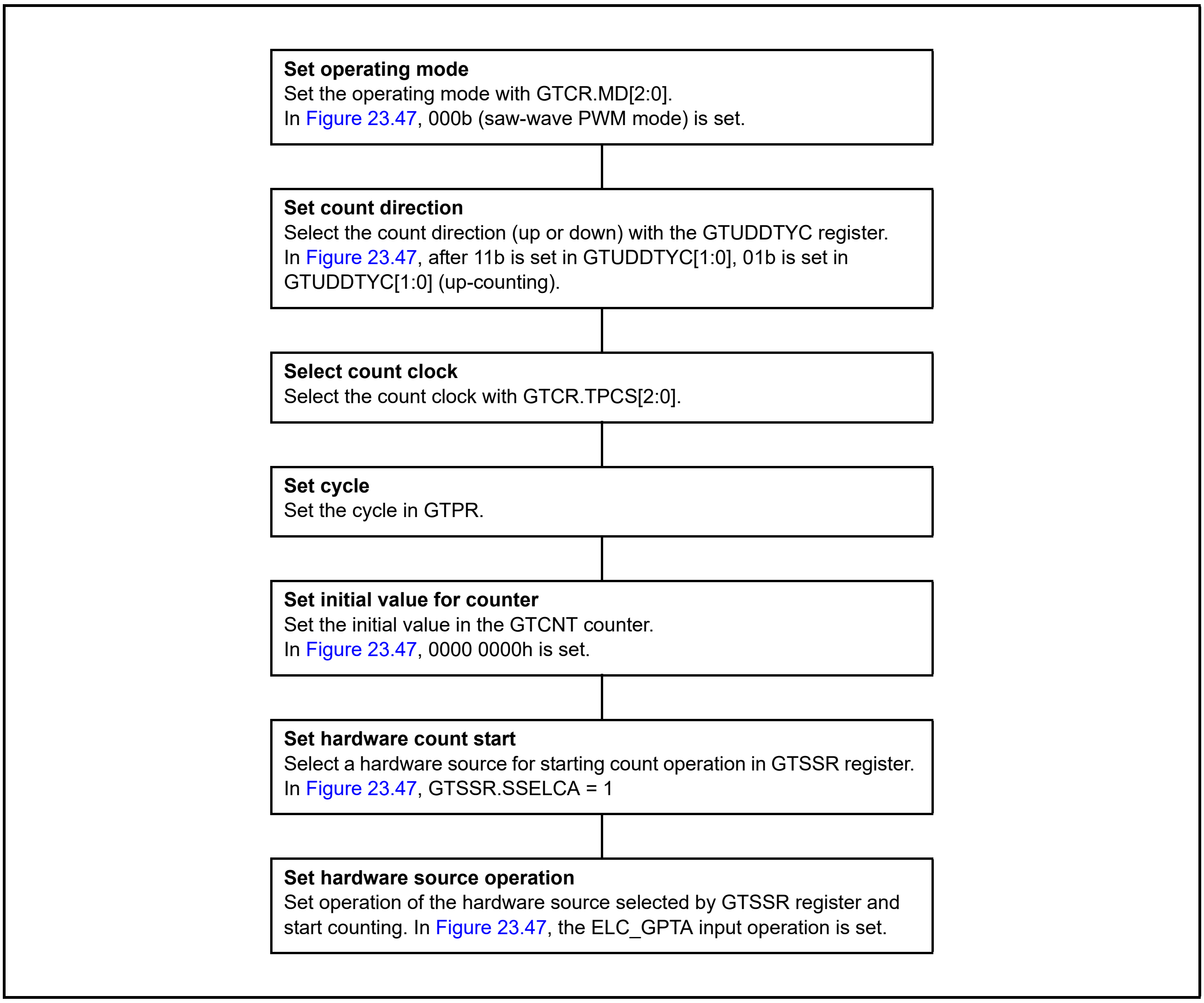

GTCNT counter value
GTPR register
Count started at ELC event input
0000 0000h
Time
ELC_GPTA input

Figure 23.47 Example of count start operation by a hardware source started, at the input of the signal from the ELC_GPTA event



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.47, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.47, 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 23.47, 0000 0000h is set.

Set hardware count start
Select a hardware source for starting count operation in GTSSR register.
In Figure 23.47, GTSSR.SSELCA = 1

Set hardware source operation
Set operation of the hardware source selected by GTSSR register and start counting. In Figure 23.47, the ELC_GPTA input operation is set.

Figure 23.48 Example setting for count start operation by a 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 and restarts at the edge of the ELC event input.

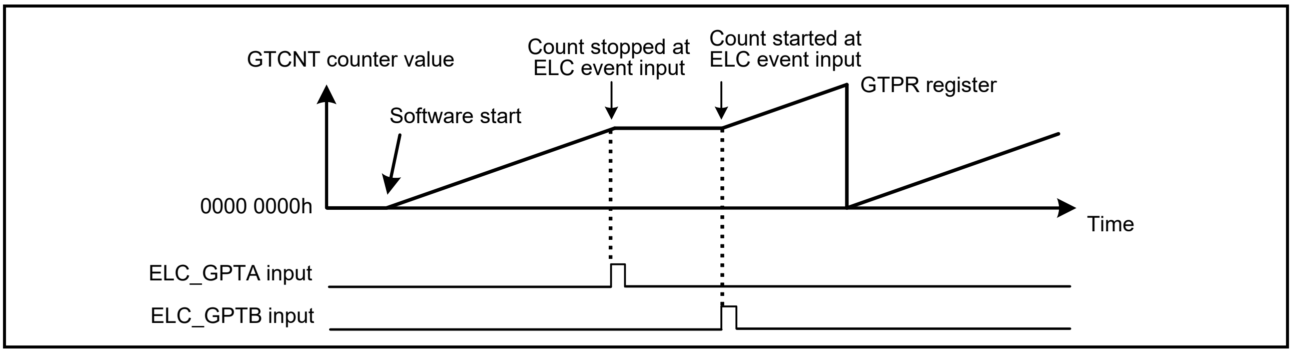
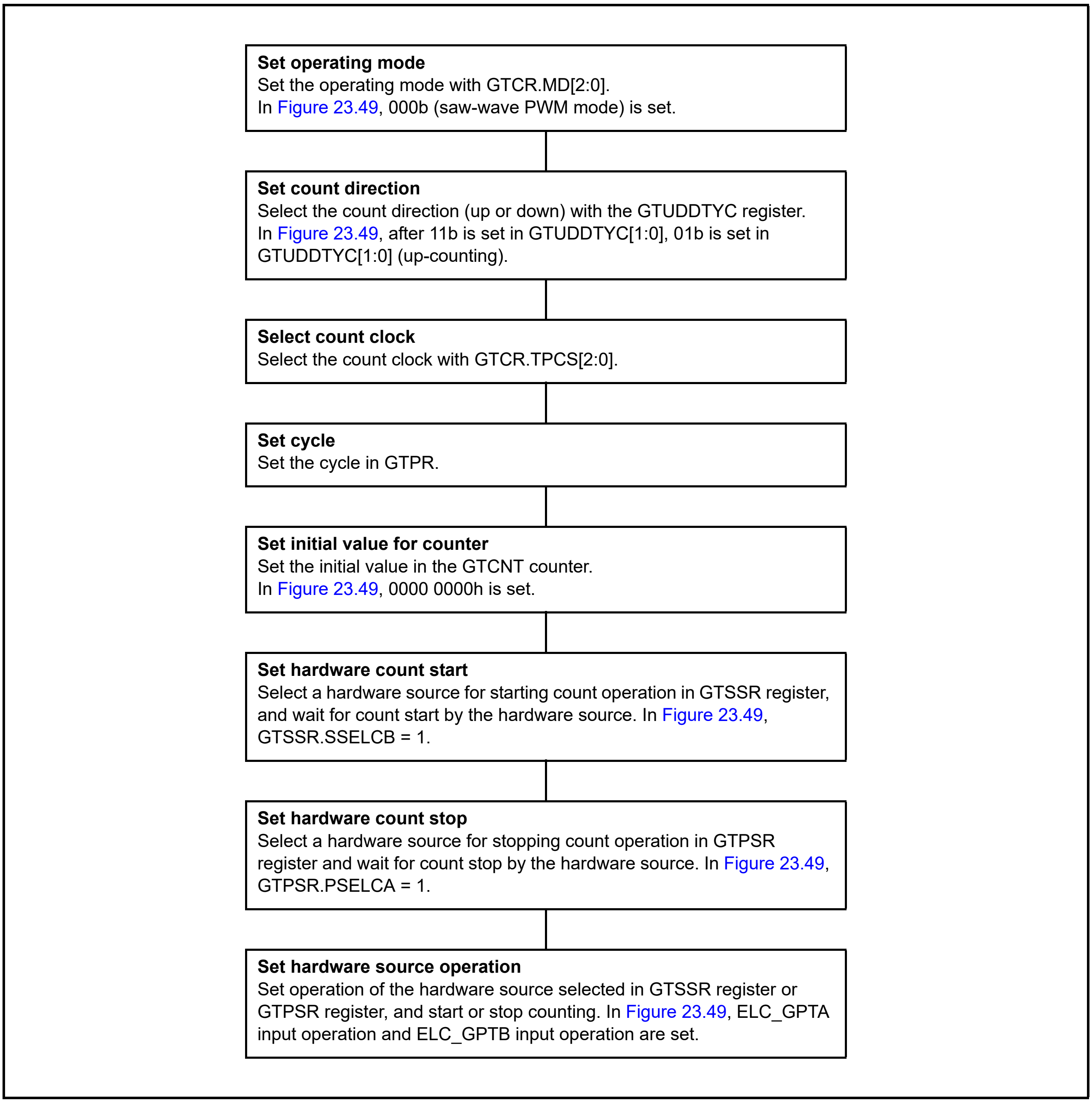

GTCNT counter value
Count stopped at ELC event input
Count started at ELC event input
GTPR register
Software start
0000 0000h
Time
ELC_GPTA input
ELC_GPTB input

Figure 23.49 Example of count stop operation by a hardware source started by software, stopped at ELC_GPTA input, and 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 setting for count stop operation by a 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.

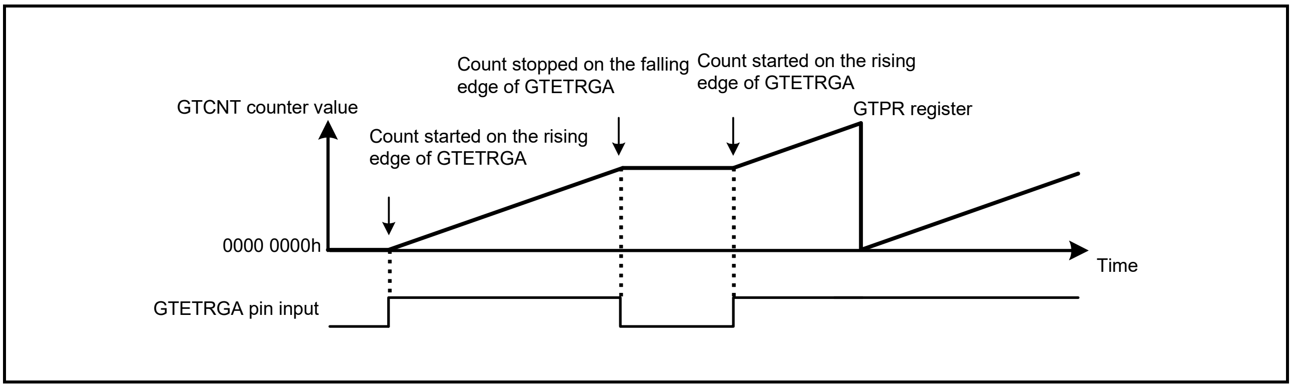
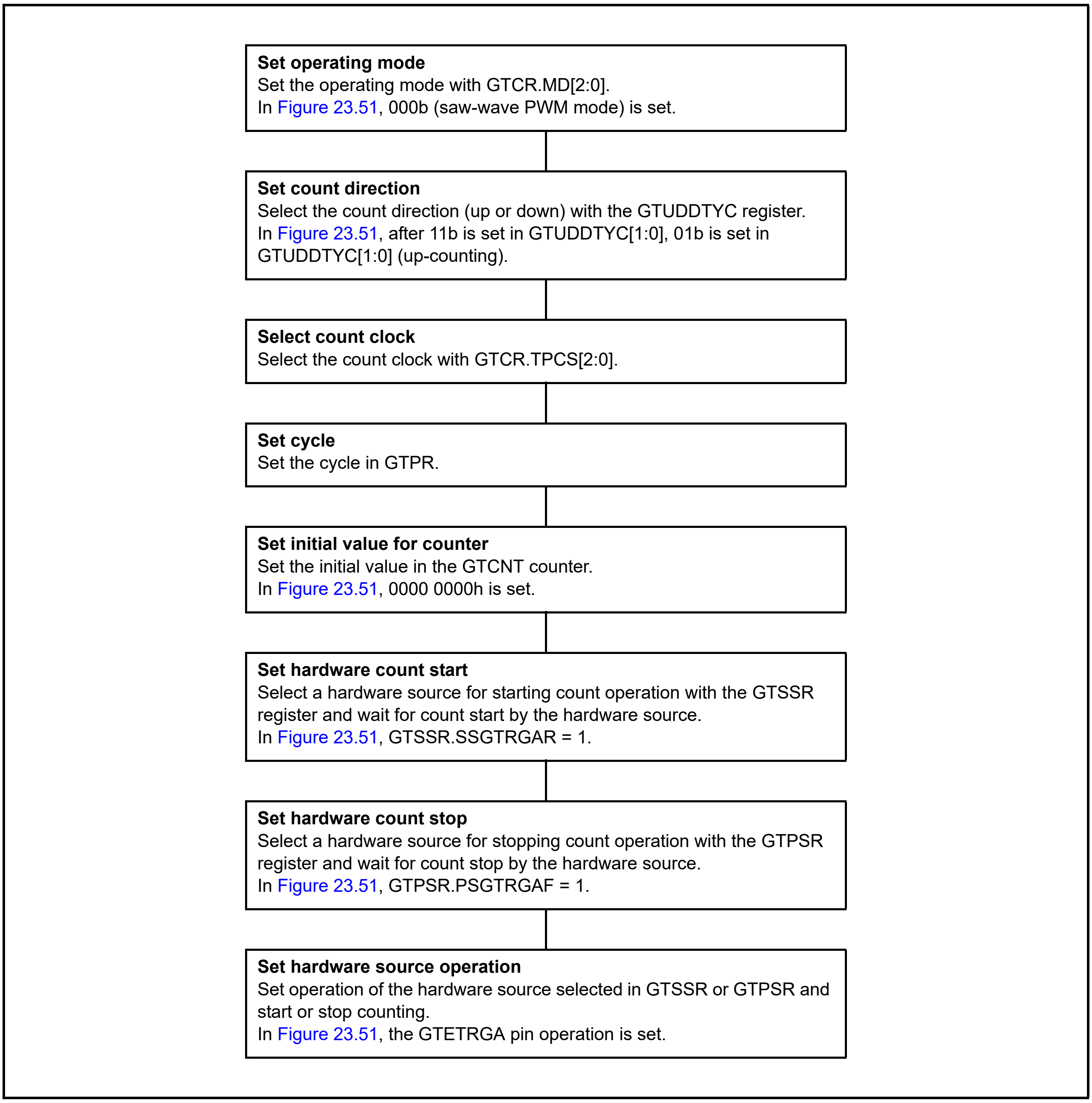

Count stopped on the falling edge of GTETRGA
Count started on the rising edge of GTETRGA
GTCNT counter value
GTPR register
Count started on the rising edge of GTETRGA
0000 0000h
Time
GTETRGA pin input

Figure 23.51 Example of count start/stop operation by a hardware source started on the rising edge of GTETRGA pin input, and stopped on the falling edge of GTETRGA pin input



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.51, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.51, 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 23.51, 0000 0000h is set.

Set hardware count start
Select a hardware source for starting count operation with the GTSSR register and wait for count start by the hardware source.
In Figure 23.51, GTSSR.SSGTRGAR = 1.

Set hardware count stop
Select a hardware source for stopping count operation with the GTPSR register and wait for count stop by the hardware source.
In Figure 23.51, GTPSR.PSGTRGAF = 1.

Set hardware source operation
Set operation of the hardware source selected in GTSSR or GTPSR and start or stop counting.
In Figure 23.51, the GTETRGA pin operation is set.

Figure 23.52 Example setting for count start/stop operation by a hardware source

23.3.7.3 Hardware clear operation

The GTCNT counter can be cleared by selecting a hardware source using GTCSR. The GPTn_OVF/GPTn_UDF (n = 0 to 5, 8) 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.

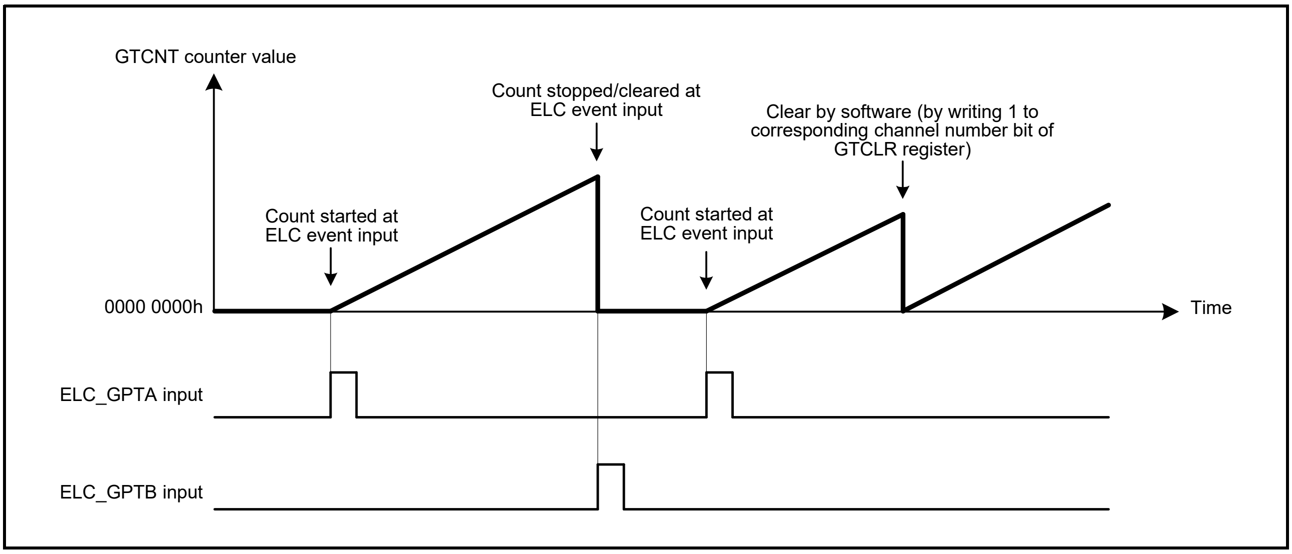

GTCNT counter value
Count stopped/cleared at ELC event input
Clear by software (by writing 1 to corresponding channel number bit of GTCLR register)
Count started at ELC event input
Count started at ELC event input
0000 0000h
Time
ELC_GPTA input
ELC_GPTB input

Figure 23.53 Examples of count clearing operation by hardware source with saw wave up-counting, started at ELC_GPTA input, and stopped/cleared at ELC_GPTB input

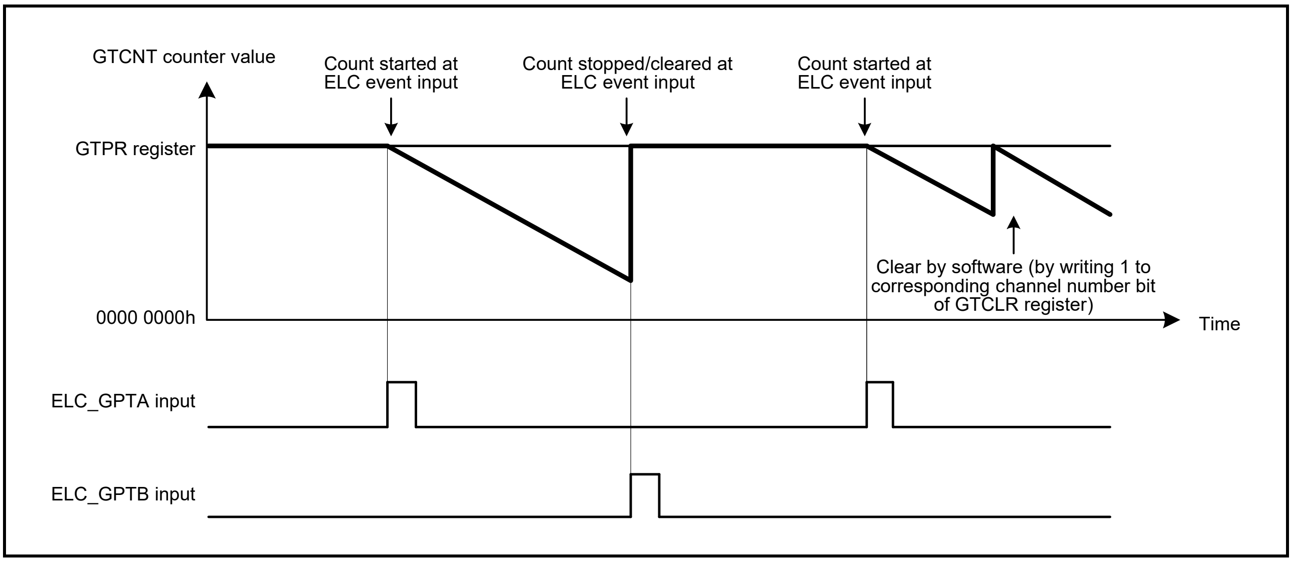
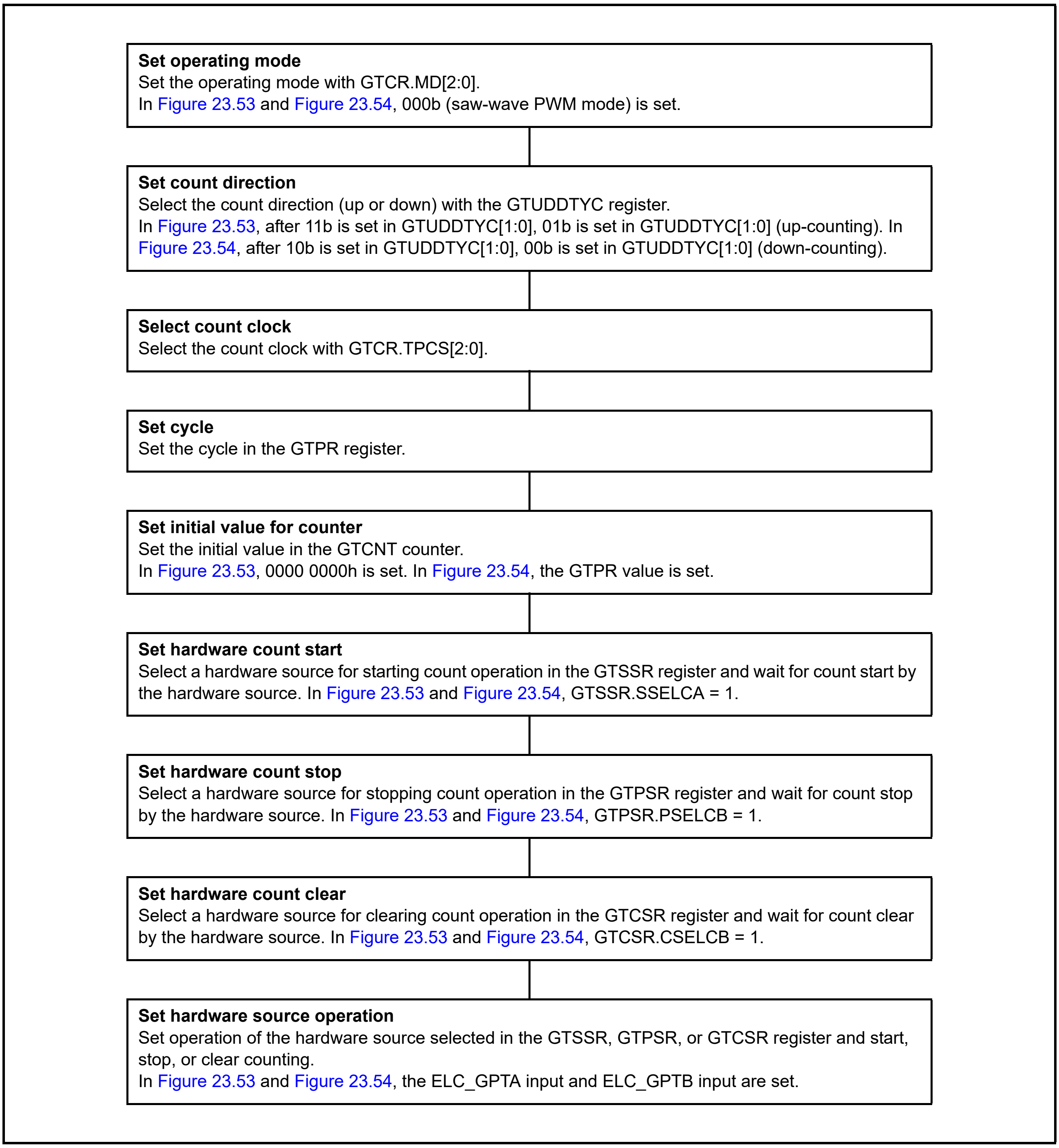

GTCNT counter value
Count started at ELC event input
Count stopped/cleared at ELC event input
Count started at ELC event input
GTPR register
Clear by software (by writing 1 to corresponding channel number bit of GTCLR register)
0000 0000h
Time
ELC_GPTA input
ELC_GPTB input

Figure 23.54 Examples of count clearing operation by hardware source with saw wave down-counting, started at ELC_GPTA input, and stopped/cleared at ELC_GPTB input



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.53 and Figure 23.54, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.53, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting). In Figure 23.54, after 10b is set in GTUDDTYC[1:0], 00b is set in GTUDDTYC[1:0] (down-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in the GTPR register.

Set initial value for counter
Set the initial value in the GTCNT counter.
In Figure 23.53, 0000 0000h is set. In Figure 23.54, the GTPR value is set.

Set hardware count start
Select a hardware source for starting count operation in the GTSSR register and wait for count start by the hardware source. In Figure 23.53 and Figure 23.54, GTSSR.SSELCA = 1.

Set hardware count stop
Select a hardware source for stopping count operation in the GTPSR register and wait for count stop by the hardware source. In Figure 23.53 and Figure 23.54, GTPSR.PSELCB = 1.

Set hardware count clear
Select a hardware source for clearing count operation in the GTCSR register and wait for count clear by the hardware source. In Figure 23.53 and Figure 23.54, GTCSR.CSELCB = 1.

Set hardware source operation
Set operation of the hardware source selected in the GTSSR, GTPSR, or GTCSR register and start, stop, or clear counting.
In Figure 23.53 and Figure 23.54, the ELC_GPTA input and ELC_GPTB input are set.

Figure 23.55 Example for setting count clearing operation by a hardware source

The GPTn_OVF/GPTn_UDF (n = 0 to 5, 8) 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 5, 8) interrupt.

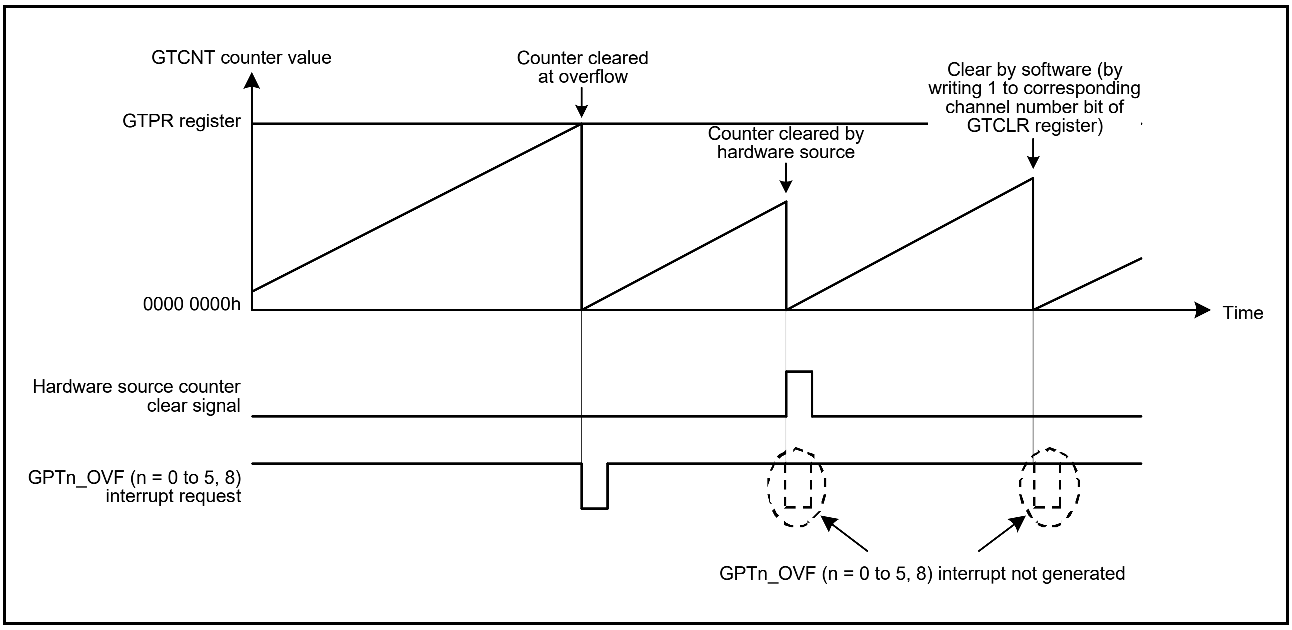

GTCNT counter value
Counter cleared at overflow
Clear by software (by writing 1 to corresponding channel number bit of GTCLR register)
GTPR register
Counter cleared by hardware source
0000 0000h
Time
Hardware source counter clear signal
GPTn_OVF (n = 0 to 5, 8) interrupt request
GPTn_OVF (n = 0 to 5, 8) interrupt not generated

Figure 23.56 Relationship between counter clearing by hardware source and GPTn_OVF (n = 0 to 5, 8) 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 of a phase start operation by software.

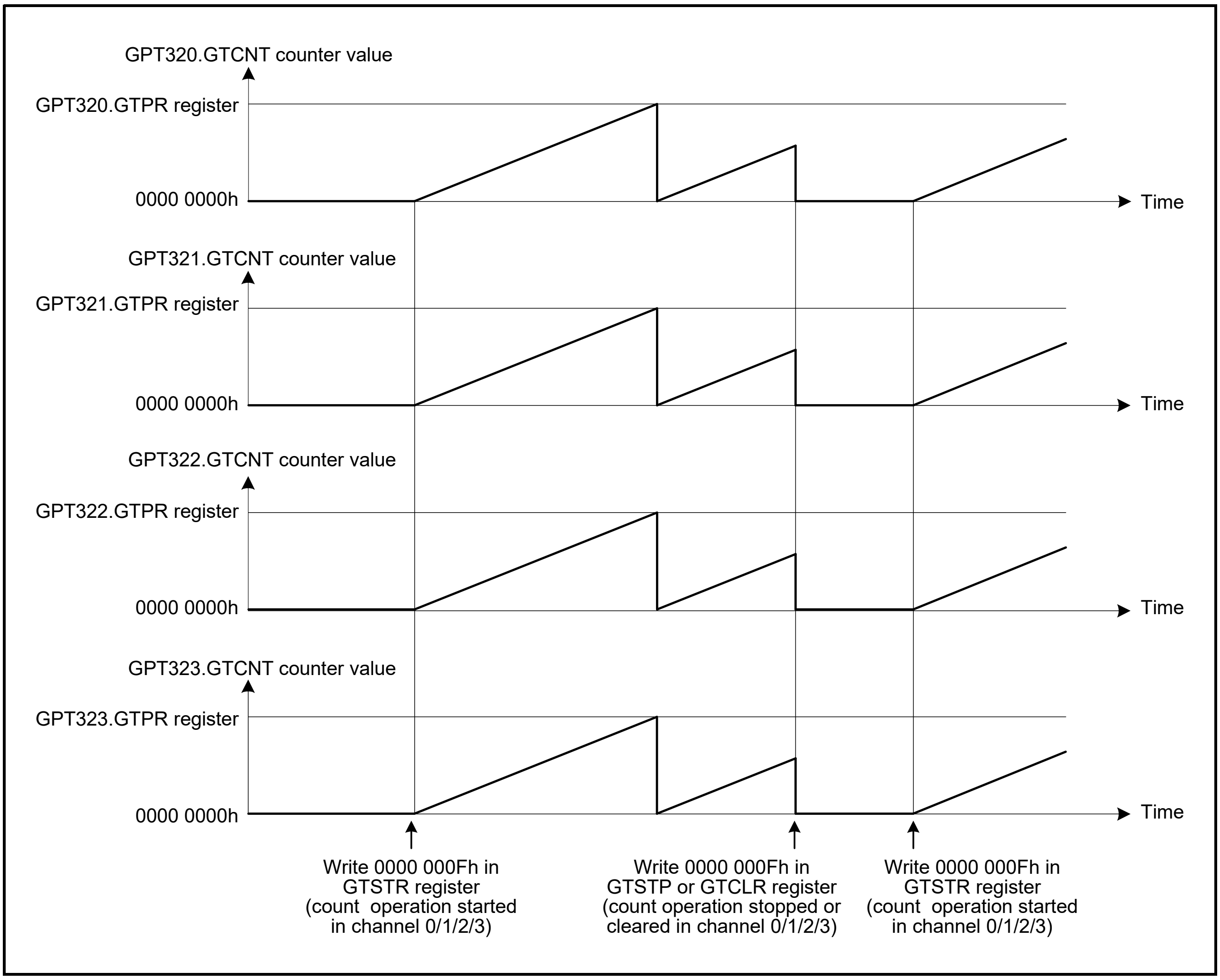

GPT320.GTCNT counter value
GPT320.GTPR register
0000 0000h
Time
GPT321.GTCNT counter value
GPT321.GTPR register
0000 0000h
Time
GPT322.GTCNT counter value
GPT322.GTPR register
0000 0000h
Time
GPT323.GTCNT counter value
GPT323.GTPR register
0000 0000h
Time
Write 0000 000Fh in GTSTR register (count operation started in channel 0/1/2/3)
Write 0000 000Fh in GTSTP or GTCLR register (count operation stopped or cleared in channel 0/1/2/3)
Write 0000 000Fh in GTSTR register (count operation started in channel 0/1/2/3)

Figure 23.57 **Example of a simultaneous start, stop, and clear by software with the same count cycle (GTPR register value)**

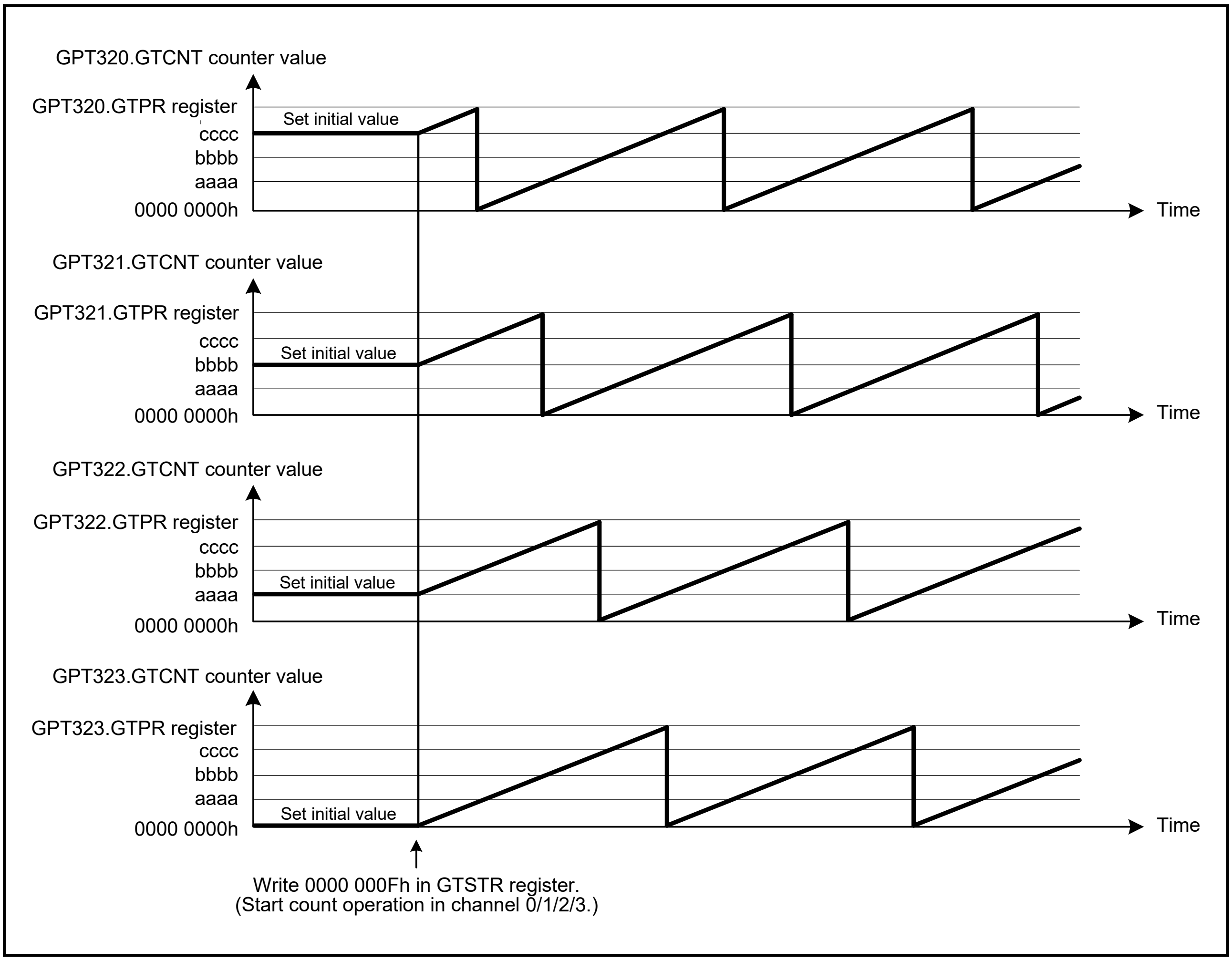

GPT320.GTCNT counter value
GPT320.GTPR register
cccc
bbbb
aaaa
0000 0000h
Set initial value
Time
GPT321.GTCNT counter value
GPT321.GTPR register
cccc
bbbb
aaaa
0000 0000h
Set initial value
Time
GPT322.GTCNT counter value
GPT322.GTPR register
cccc
bbbb
aaaa
0000 0000h
Set initial value
Time
GPT323.GTCNT counter value
GPT323.GTPR register
cccc
bbbb
aaaa
0000 0000h
Set initial value
Time
Write 0000 000Fh in GTSTR register.
(Start count operation in channel 0/1/2/3.)

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.

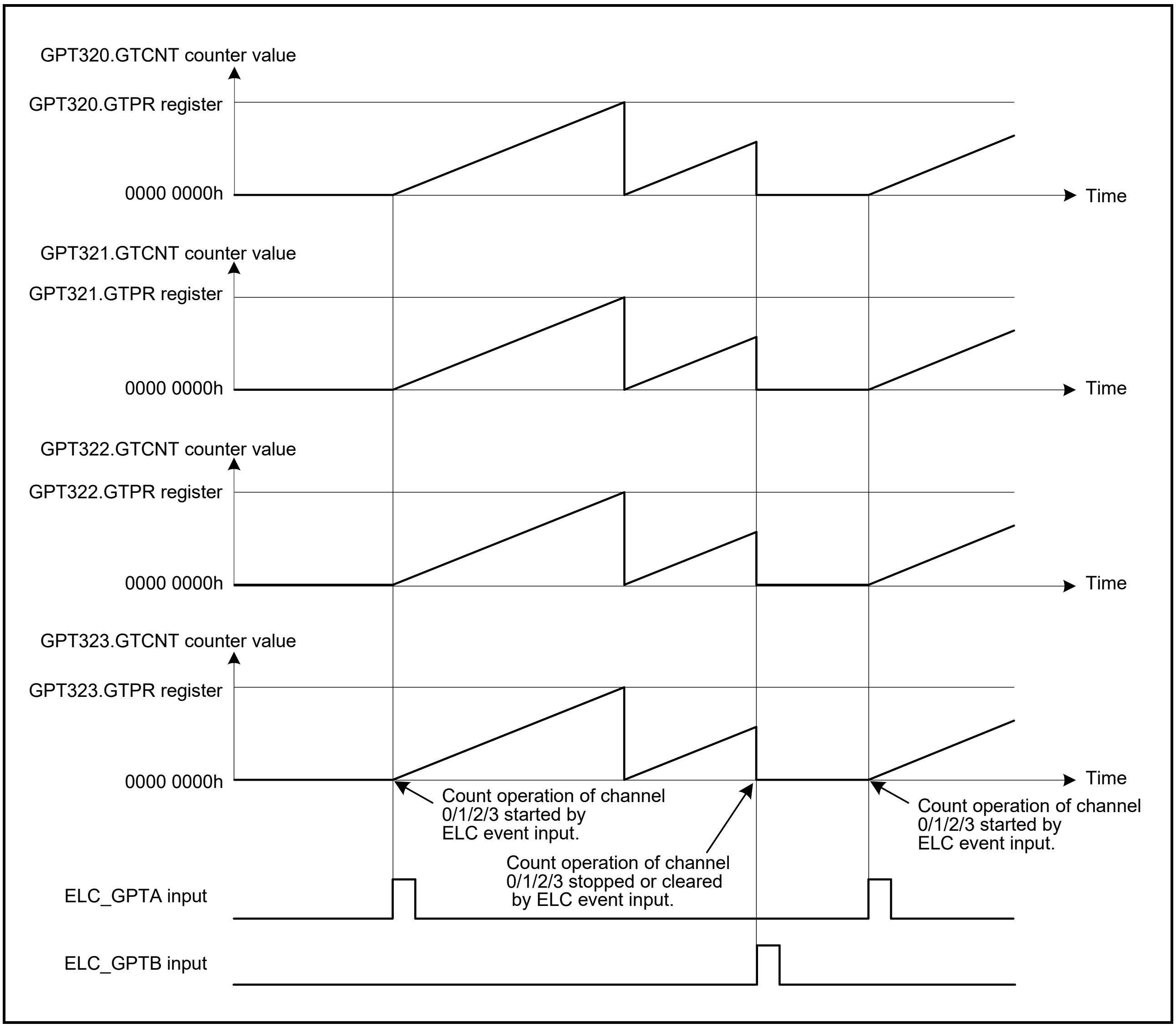
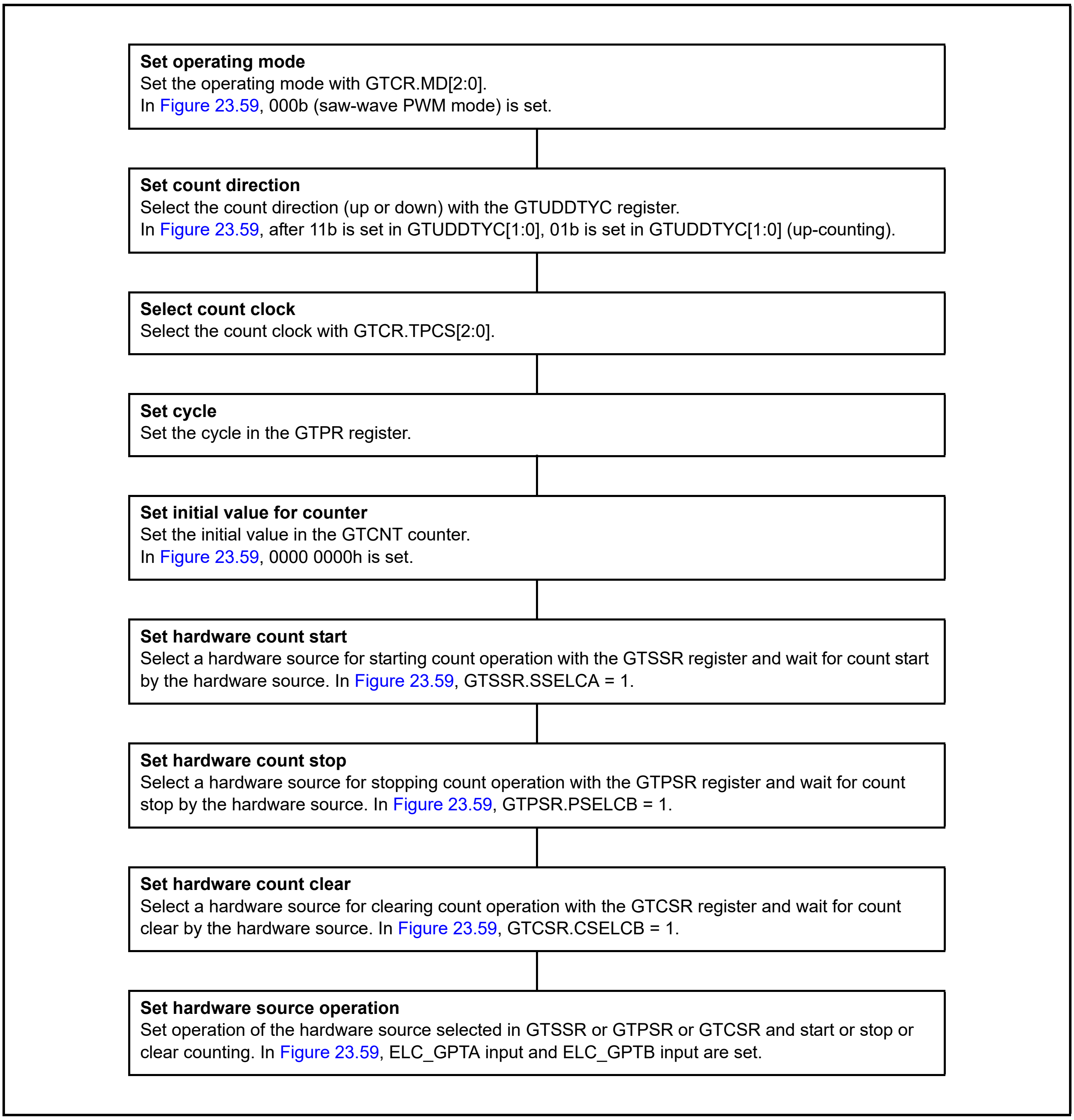

GPT320.GTCNT counter value
GPT320.GTPR register
0000 0000h
Time
GPT321.GTCNT counter value
GPT321.GTPR register
0000 0000h
Time
GPT322.GTCNT counter value
GPT322.GTPR register
0000 0000h
Time
GPT323.GTCNT counter value
GPT323.GTPR register
0000 0000h
Time
Count operation of channel 0/1/2/3 started by ELC event input.
Count operation of channel 0/1/2/3 stopped or cleared by ELC event input.
Count operation of channel 0/1/2/3 started by ELC event input.
ELC_GPTA input
ELC_GPTB input

Figure 23.59 Example of a simultaneous start, stop, and clear by a hardware source with the same count cycle (GTPR register value)



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.59, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.59, 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 the GTPR register.

Set initial value for counter
Set the initial value in the GTCNT counter.
In Figure 23.59, 0000 0000h is set.

Set hardware count start
Select a hardware source for starting count operation with the GTSSR register and wait for count start by the hardware source. In Figure 23.59, GTSSR.SSELCA = 1.

Set hardware count stop
Select a hardware source for stopping count operation with the GTPSR register and wait for count stop by the hardware source. In Figure 23.59, GTPSR.PSELCB = 1.

Set hardware count clear
Select a hardware source for clearing count operation with the GTCSR register and wait for count clear by the hardware source. In Figure 23.59, GTCSR.CSELCB = 1.

Set hardware source operation
Set operation of the hardware source selected in GTSSR or GTPSR or GTCSR and start or stop or clear counting. In Figure 23.59, ELC_GPTA input and ELC_GPTB input are set.

Figure 23.60 Example setting for simultaneous start by a hardware source

23.3.9 PWM Output Operation Examples

(1) Synchronized PWM output

The GPT outputs 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.

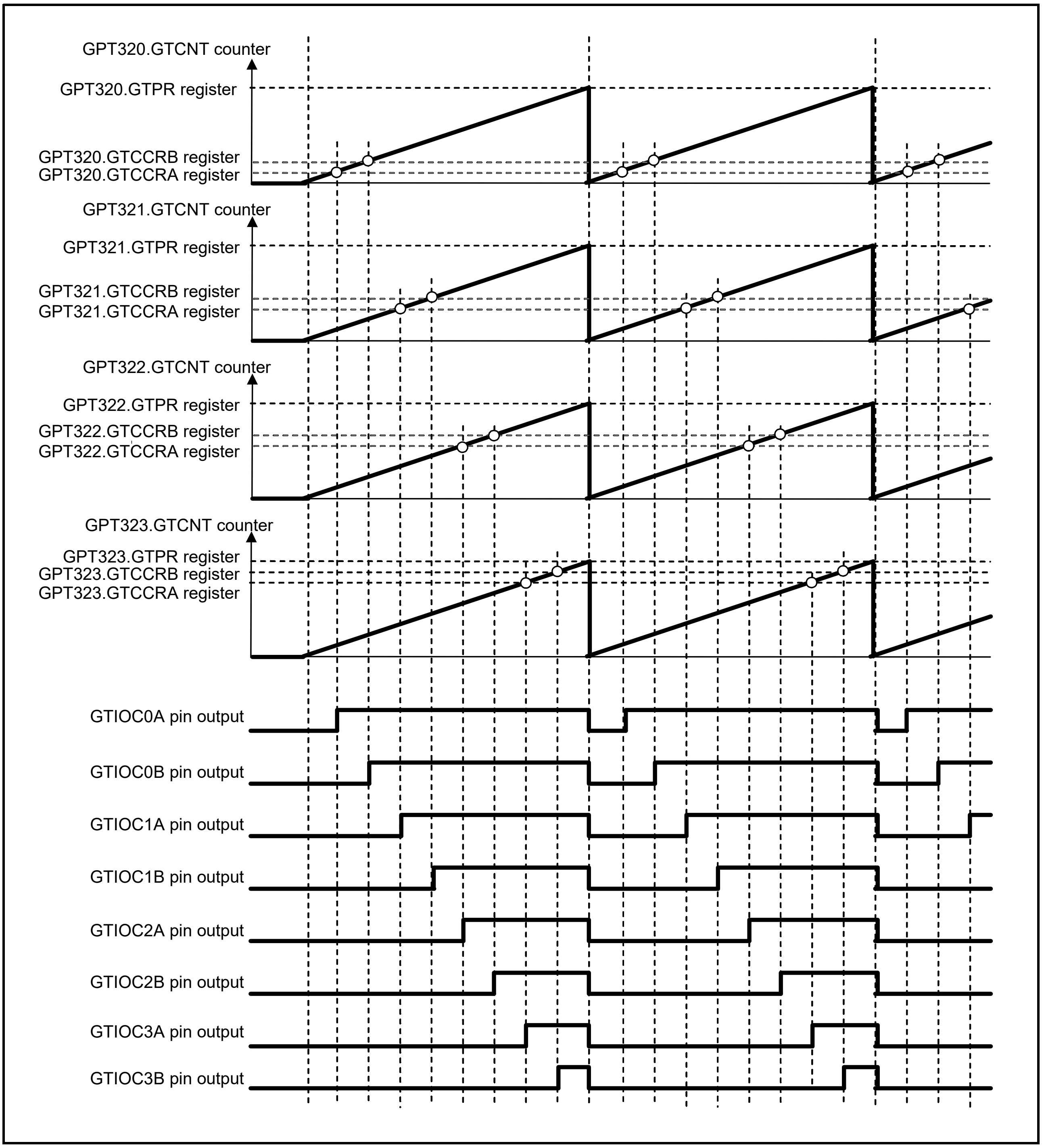

GPT320.GTCNT counter
GPT320.GTPR register
GPT320.GTCCRB register
GPT320.GTCCRA register
GPT321.GTCNT counter
GPT321.GTPR register
GPT321.GTCCRB register
GPT321.GTCCRA register
GPT322.GTCNT counter
GPT322.GTPR register
GPT322.GTCCRB register
GPT322.GTCCRA register
GPT323.GTCNT counter
GPT323.GTPR register
GPT323.GTCCRB register
GPT323.GTCCRA register
GTIOC0A pin output
GTIOC0B pin output
GTIOC1A pin output
GTIOC1B pin output
GTIOC2A pin output
GTIOC2B pin output
GTIOC3A pin output
GTIOC3B pin output

Figure 23.61 Example of synchronized PWM output

(2) 3-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.

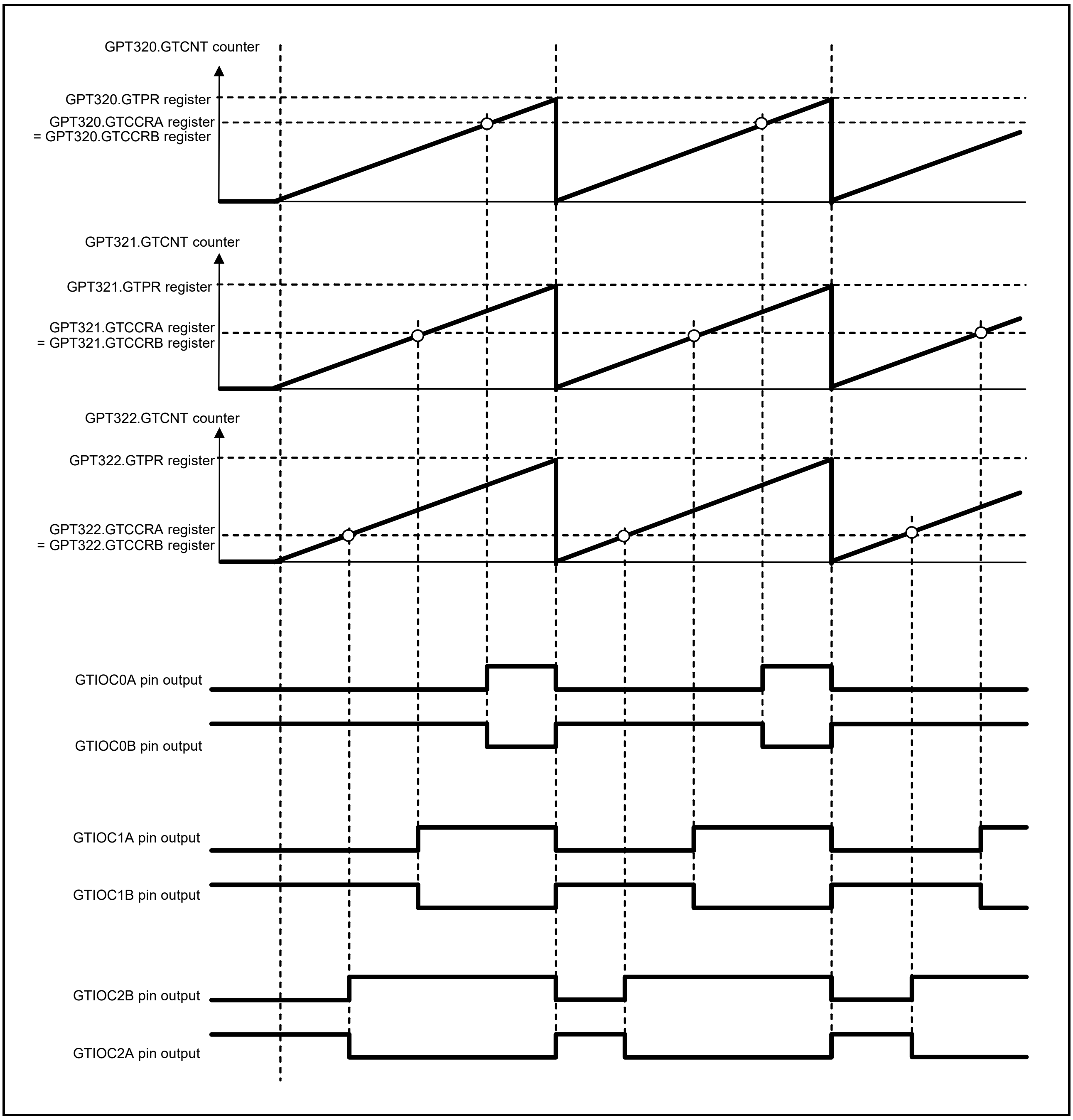

GPT320.GTCNT counter
GPT320.GTPR register
GPT320.GTCCRA register
= GPT320.GTCCRB register
GPT321.GTCNT counter
GPT321.GTPR register
GPT321.GTCCRA register
= GPT321.GTCCRB register
GPT322.GTCNT counter
GPT322.GTPR register
GPT322.GTCCRA register
= GPT322.GTCCRB register
GTIOC0A pin output
GTIOC0B pin output
GTIOC1A pin output
GTIOC1B pin output
GTIOC2B pin output
GTIOC2A pin output

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, 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.

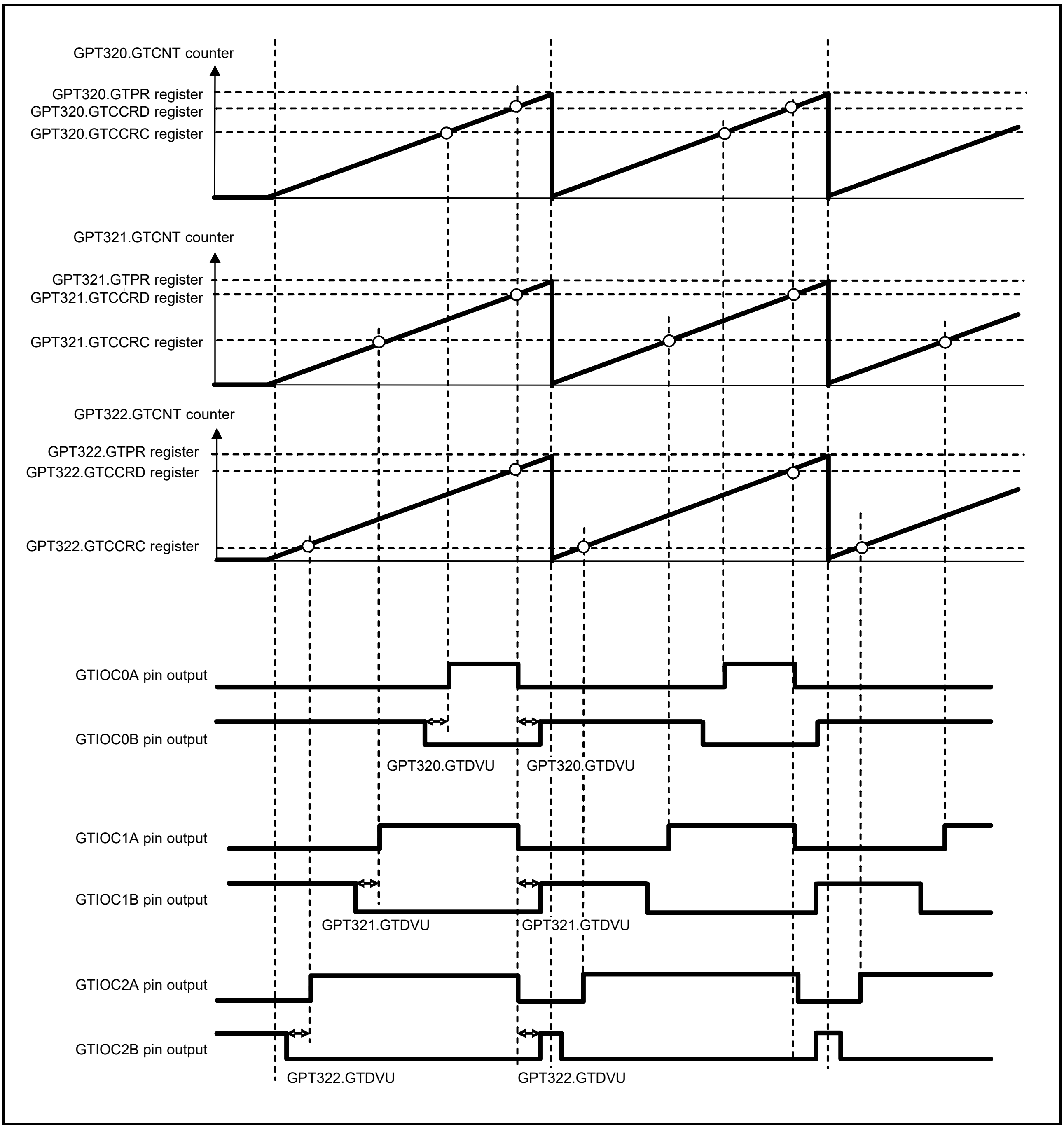

GPT320.GTCNT counter
GPT320.GTPR register
GPT320.GTCCRD register
GPT320.GTCCRC register
GPT321.GTCNT counter
GPT321.GTPR register
GPT321.GTCCRD register
GPT321.GTCCRC register
GPT322.GTCNT counter
GPT322.GTPR register
GPT322.GTCCRD register
GPT322.GTCCRC register
GTIOC0A pin output
GTIOC0B pin output
GPT320.GTDVU
GPT320.GTDVU
GTIOC1A pin output
GTIOC1B pin output
GPT321.GTDVU
GPT321.GTDVU
GTIOC2A pin output
GTIOC2B pin output
GPT322.GTDVU
GPT322.GTDVU

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.

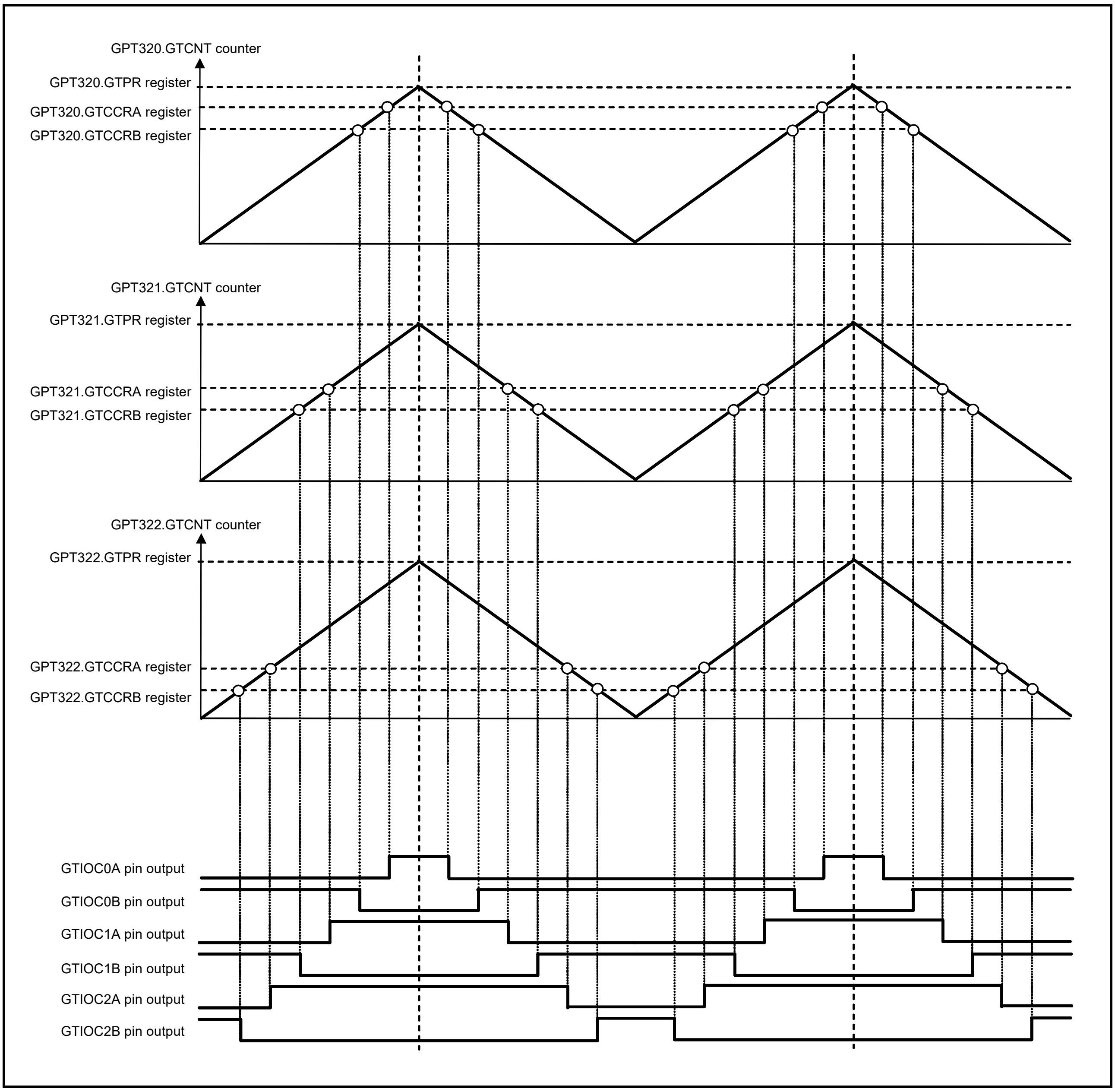

GPT320.GTCNT counter
GPT320.GTPR register
GPT320.GTCCRA register
GPT320.GTCCRB register
GPT321.GTCNT counter
GPT321.GTPR register
GPT321.GTCCRA register
GPT321.GTCCRB register
GPT322.GTCNT counter
GPT322.GTPR register
GPT322.GTCCRA register
GPT322.GTCCRB register
GTIOC0A pin output
GTIOC0B pin output
GTIOC1A pin output
GTIOC1B pin output
GTIOC2A pin output
GTIOC2B pin output

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.

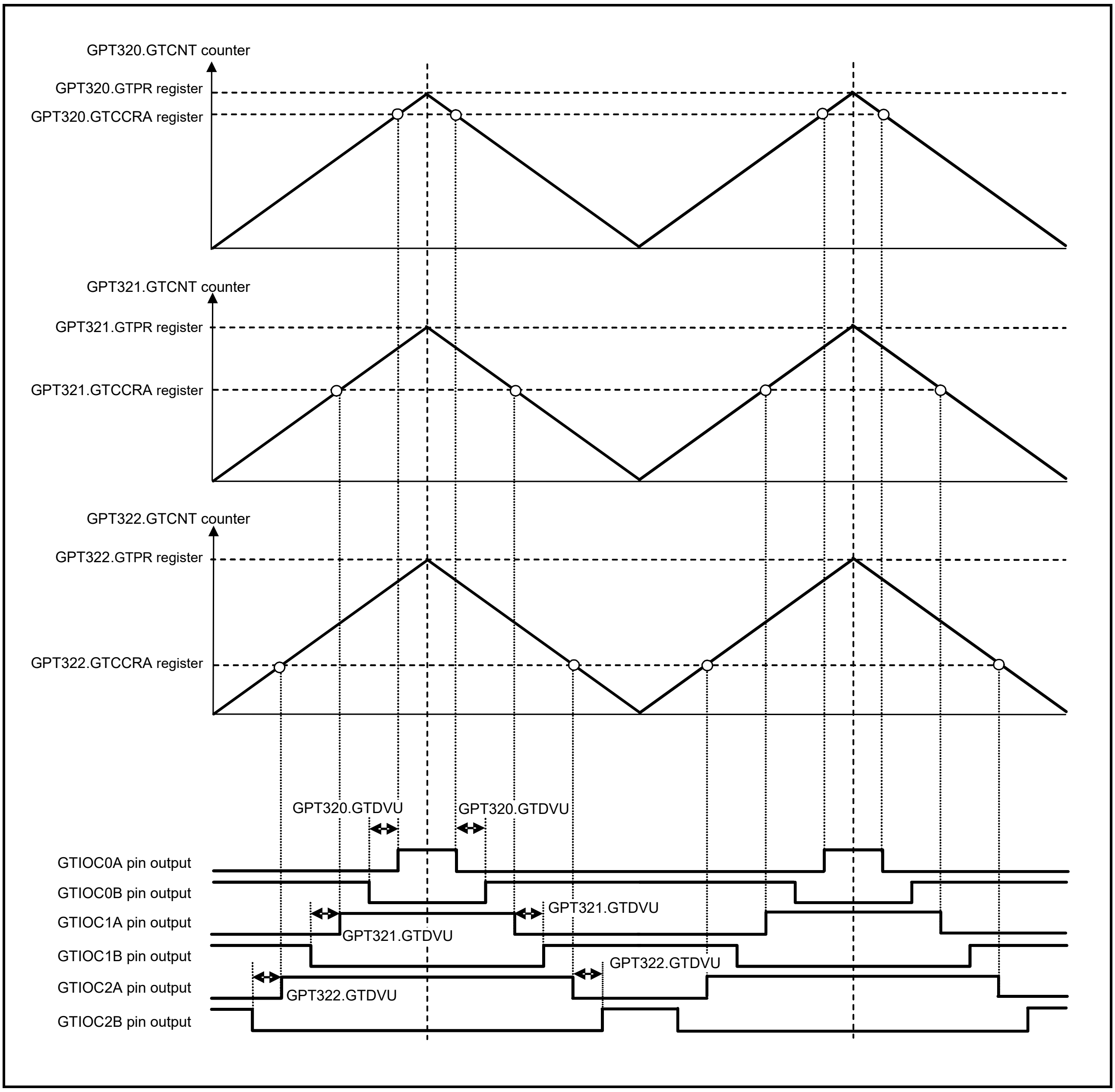

GPT320.GTCNT counter
GPT320.GTPR register
GPT320.GTCCRA register
GPT321.GTCNT counter
GPT321.GTPR register
GPT321.GTCCRA register
GPT322.GTCNT counter
GPT322.GTPR register
GPT322.GTCCRA register
GPT320.GTDVU
GPT320.GTDVU
GTIOC0A pin output
GTIOC0B pin output
GTIOC1A pin output
GPT321.GTDVU
GPT321.GTDVU
GTIOC1B pin output
GPT322.GTDVU
GTIOC2A pin output
GPT322.GTDVU
GTIOC2B pin output

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.

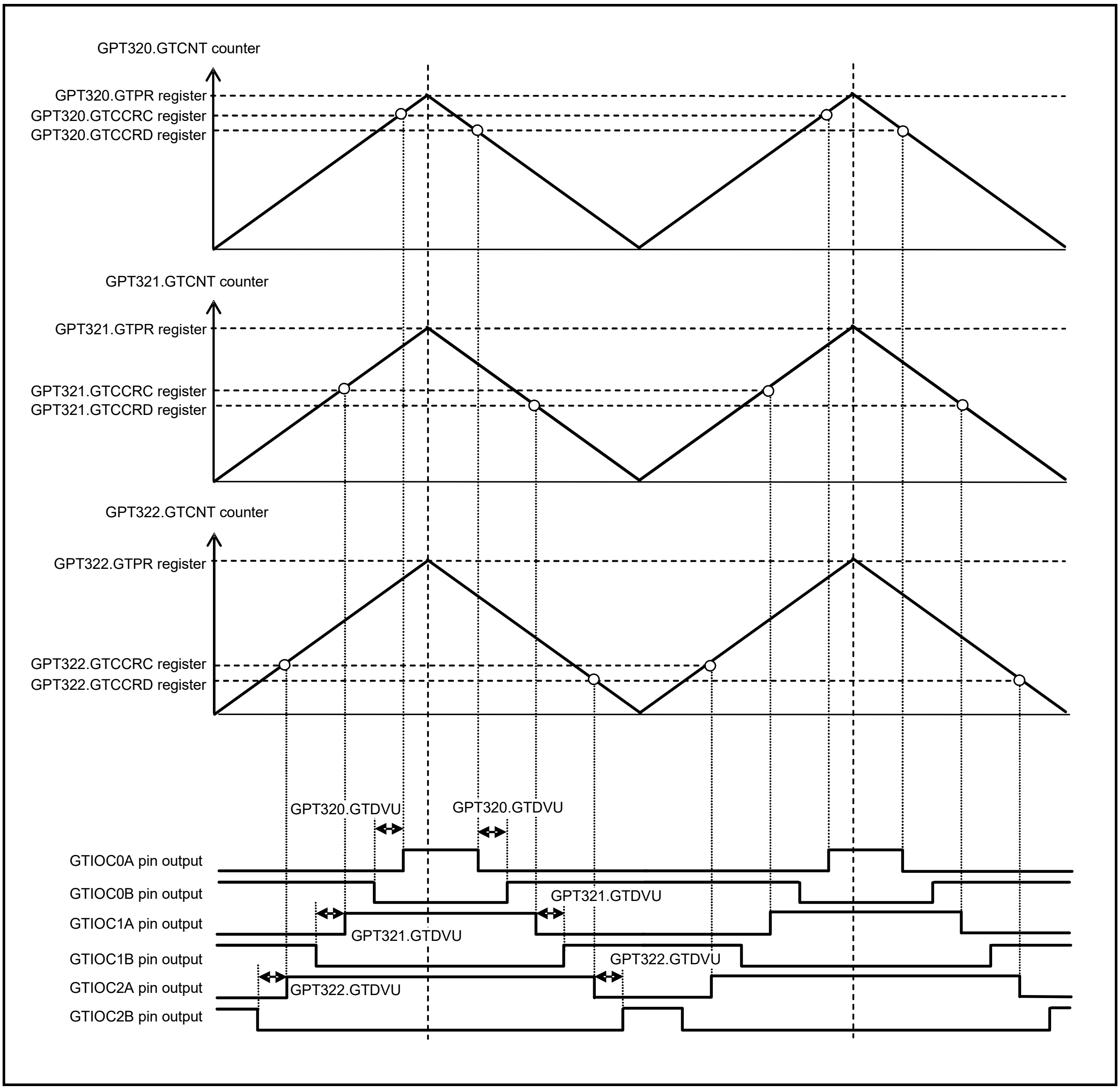

GPT320.GTCNT counter
GPT320.GTPR register
GPT320.GTCCRC register
GPT320.GTCCRD register
GPT321.GTCNT counter
GPT321.GTPR register
GPT321.GTCCRC register
GPT321.GTCCRD register
GPT322.GTCNT counter
GPT322.GTPR register
GPT322.GTCCRC register
GPT322.GTCCRD register
GPT320.GTDVU
GPT320.GTDVU
GTIOC0A pin output
GTIOC0B pin output
GPT321.GTDVU
GTIOC1A pin output
GPT321.GTDVU
GTIOC1B pin output
GPT322.GTDVU
GTIOC2A pin output
GPT322.GTDVU
GTIOC2B pin output

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 list settings for the GTUPSR and GTDNSR registers.

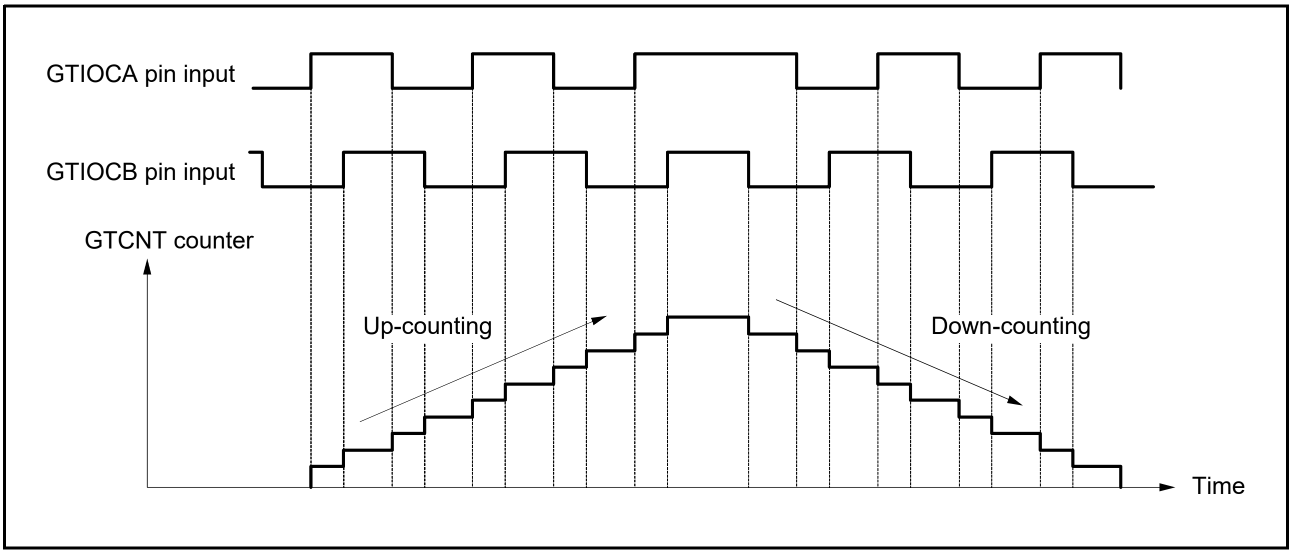

GTIOCA pin input
GTIOCB pin input
GTCNT counter
Up-counting
Down-counting
Time

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 = 0000 6900h GTDNSR = 0000 9600h
Low	↓		
↑	Low		
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High		
↓	Low		

↑ : Rising edge

↓ : Falling edge

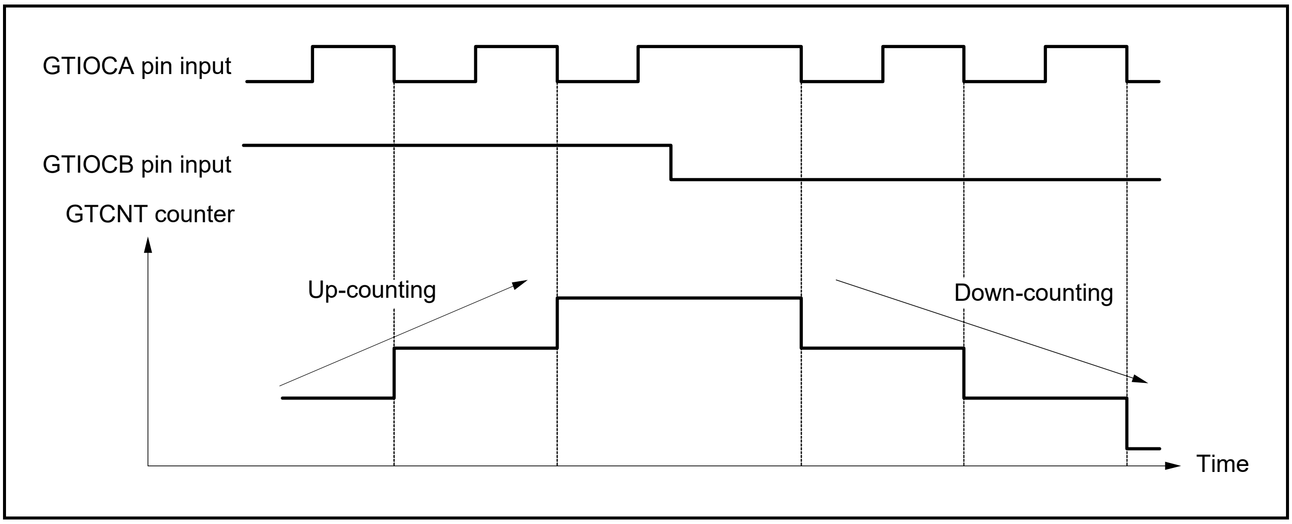

GTIOCA pin input
GTIOCB pin input
GTCNT counter
Up-counting
Down-counting
Time

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 = 0000 0800h GTDNSR = 0000 0400h
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High		
↓	Low	Down-counting	

↑ : Rising edge

↓ : Falling edge

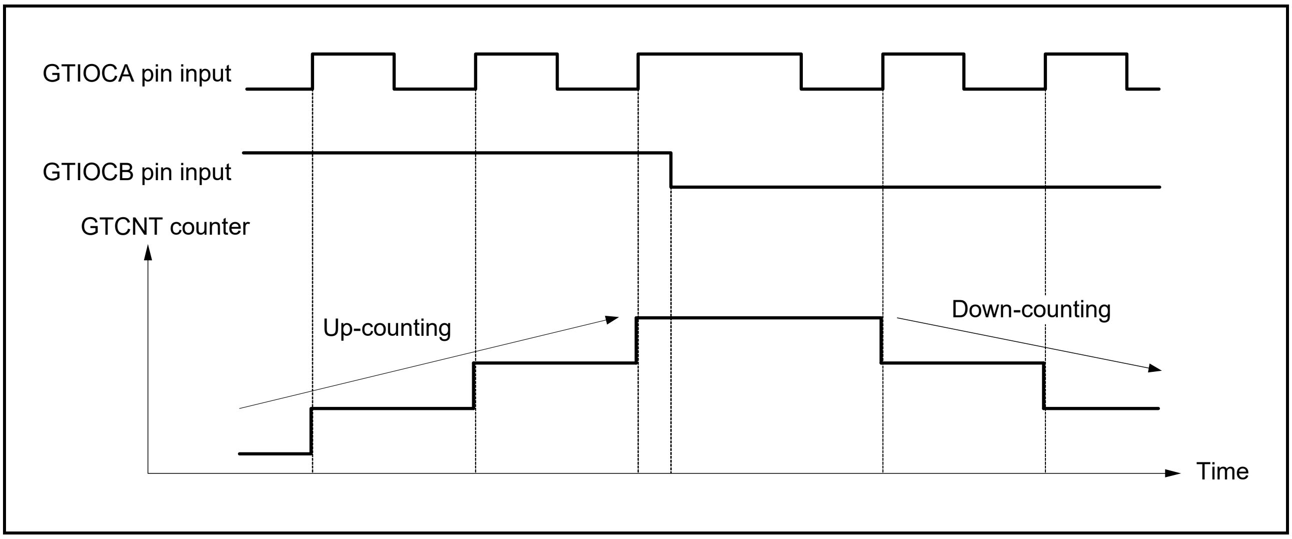

GTIOCA pin input
GTIOCB pin input
GTCNT counter
Up-counting
Down-counting
Time

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 = 0000 0200h GTDNSR = 0000 0100h
Low	↓		
↑	Low	Down-counting	
↓	High	Don't care	
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Don't care	

↑ : Rising edge

↓ : Falling edge

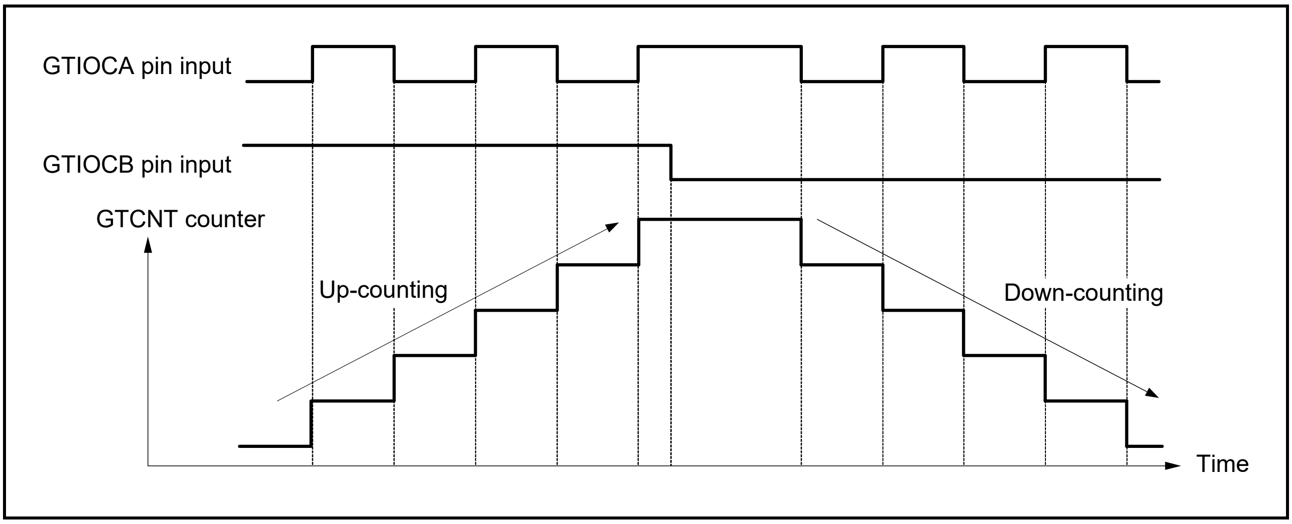

GTIOCA pin input
GTIOCB pin input
GTCNT counter
Up-counting
Down-counting
Time

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 = 0000 0A00h GTDNSR = 0000 0500h
Low	↓		
↑	Low	Down-counting	
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High	Up-counting	
↓	Low	Down-counting	

↑ : Rising edge

↓ : Falling edge

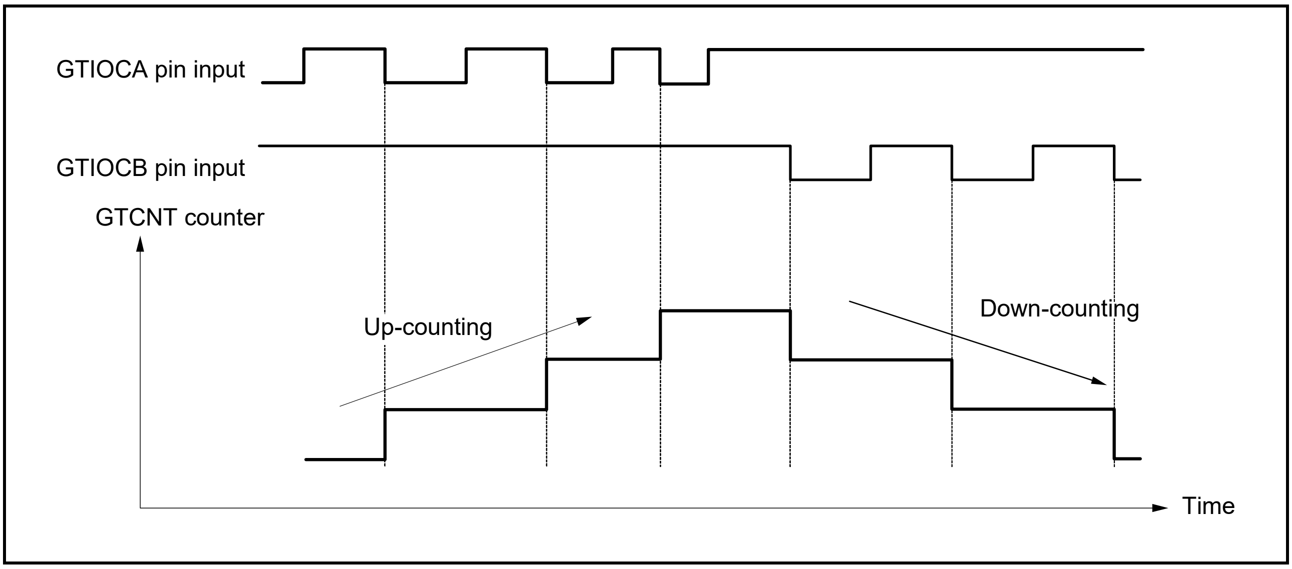

GTIOCA pin input
GTIOCB pin input
GTCNT counter
Up-counting
Down-counting
Time

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 = 0000 0800h GTDNSR = 0000 8000h
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Don't care	
↑	High		
↓	Low		

↑ : Rising edge

↓ : Falling edge

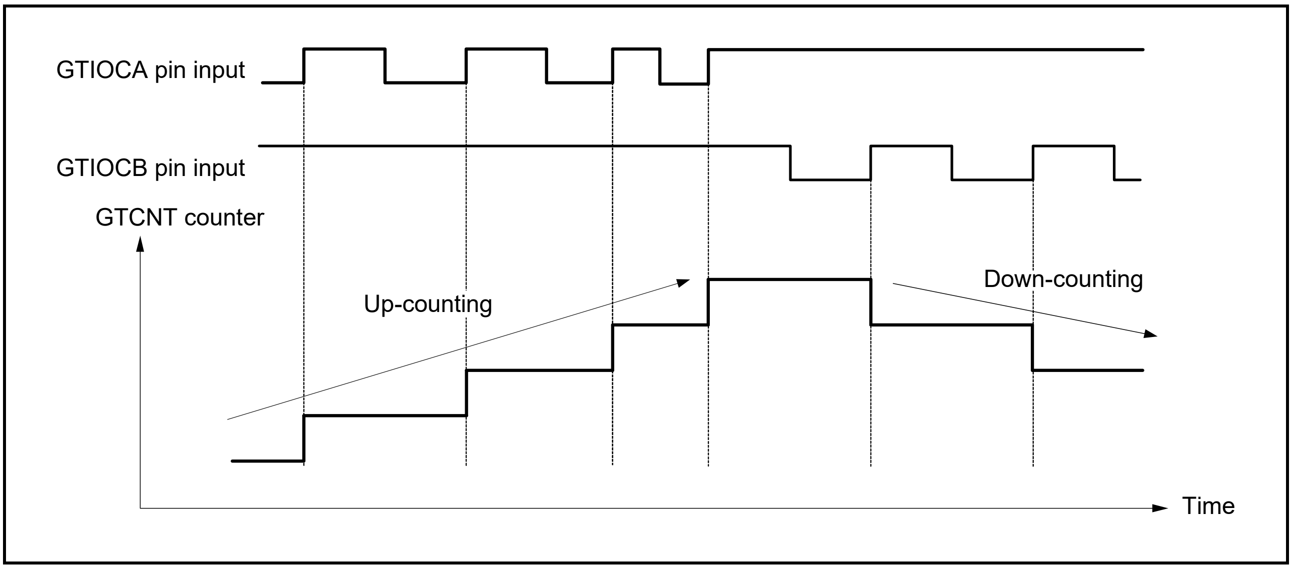

GTIOCA pin input
GTIOCB pin input
GTCNT counter
Up-counting
Down-counting
Time

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 = 0000 0200h GTDNSR = 0000 2000h
Low	↓	Don't care	
↑	Low		
↓	High		
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Don't care	

↑ : Rising edge

↓ : Falling edge

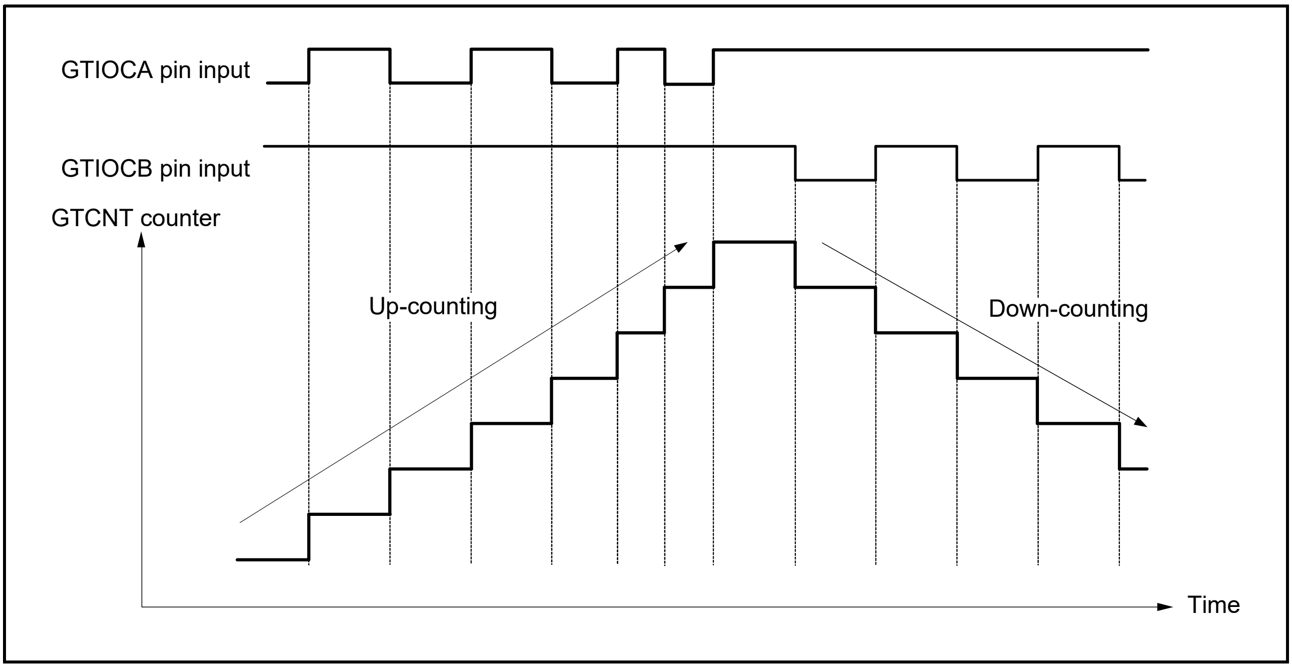

GTIOCA pin input
GTIOCB pin input
GTCNT counter
Up-counting
Down-counting
Time

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 = 0000 0A00h GTDNSR = 0000 A000h
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

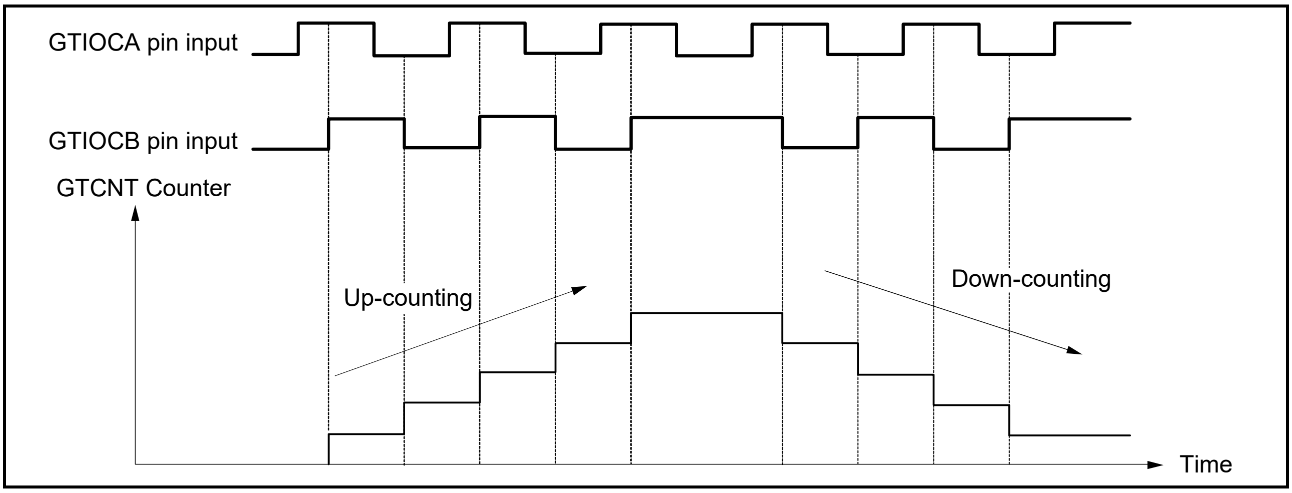

GTIOCA pin input
GTIOCB pin input
GTCNT Counter
Up-counting
Down-counting
Time

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 = 0000 6000h GTDNSR = 0000 9000h
Low	↓		
↑	Low	Don't care	
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High	Don't care	
↓	Low		

↑ : Rising edge

↓ : Falling edge

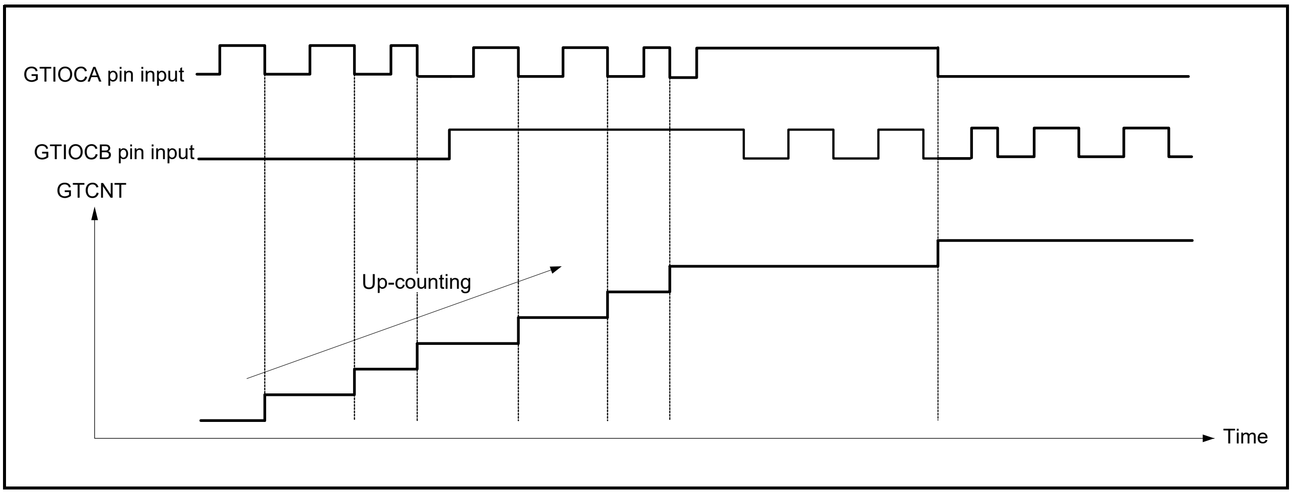


Figure 23.75 Example of phase counting mode 5 (A)

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 = 0000 0C00h GTDNSR = 0000 0000h
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High		
↓	Low	Up-counting	

↑ : Rising edge

↓ : Falling edge

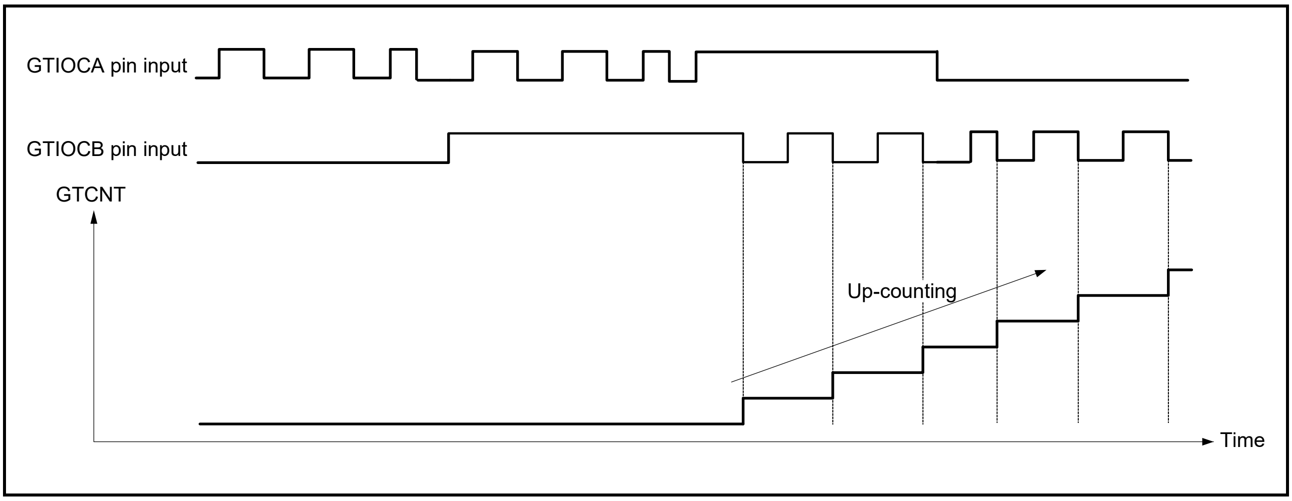


Figure 23.76 Example of phase counting mode 5 (B)

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 = 0000 C000h GTDNSR = 0000 0000h
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.

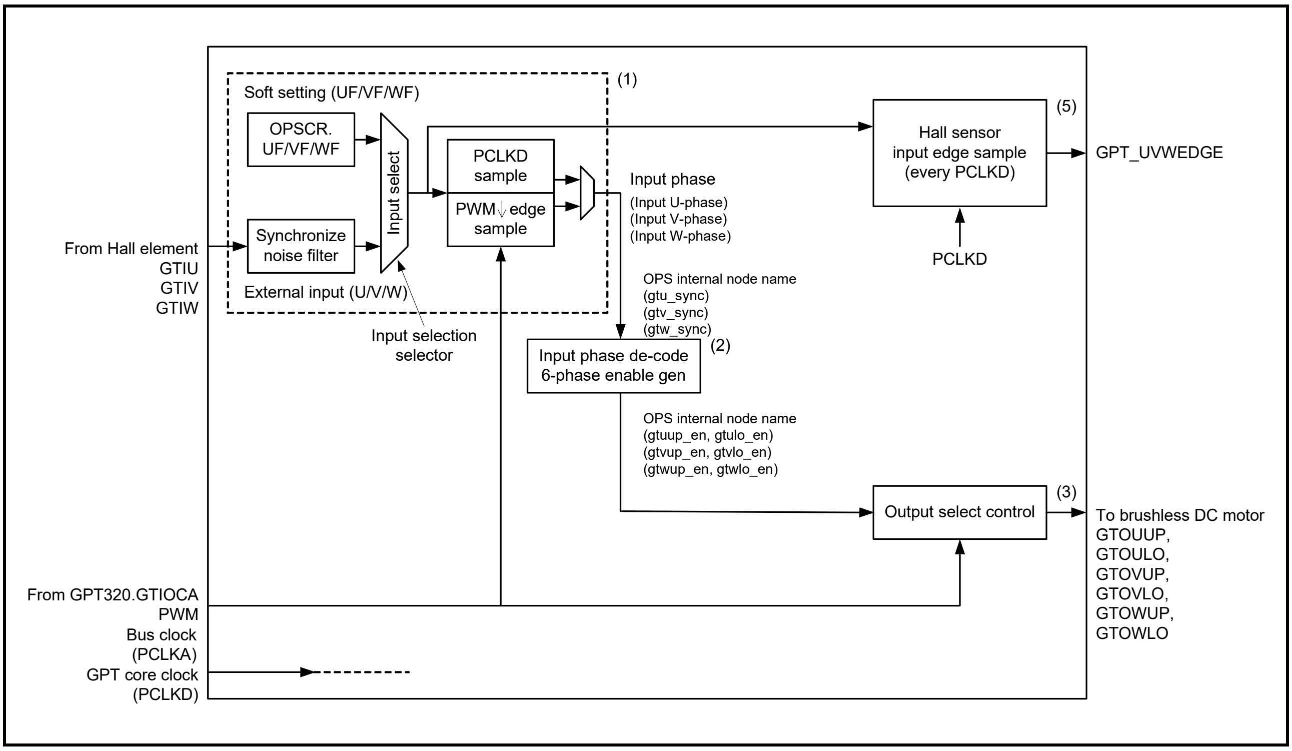
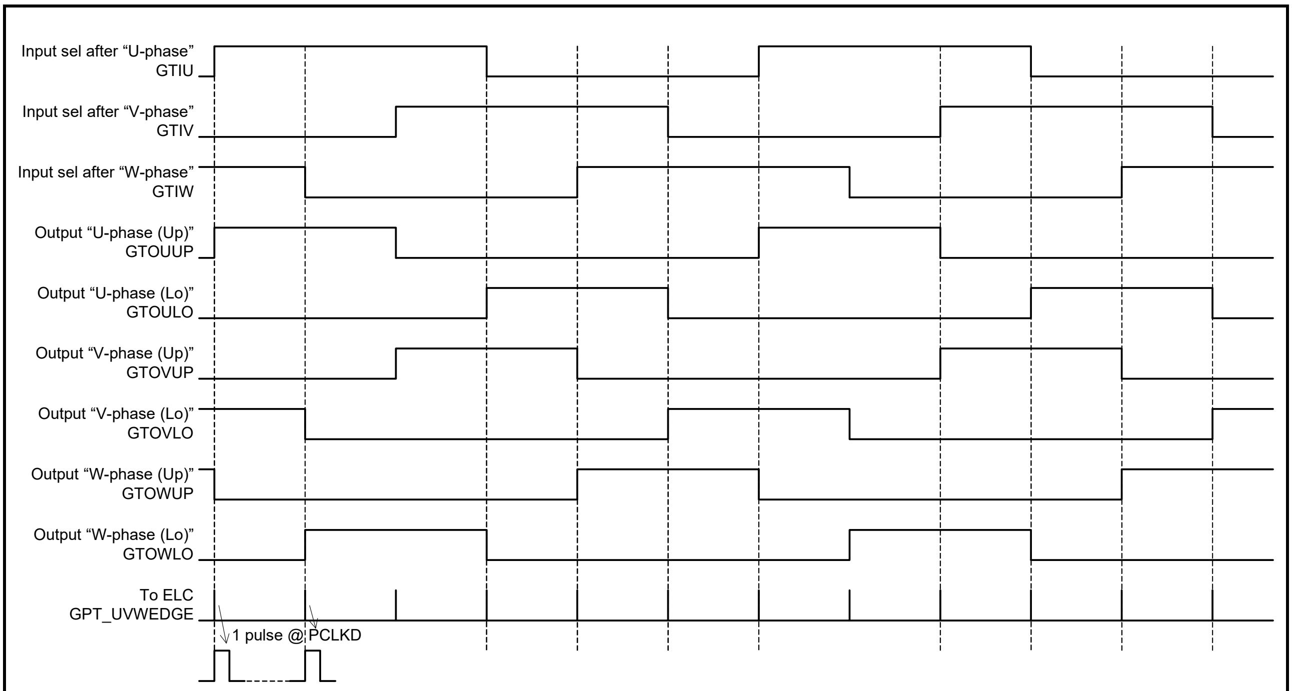

Soft setting (UF/VF/WF)
(1)
OPSCR.
UF/VF/WF
Input select
PCLKD
sample
PWM↓edge
sample
Input phase
(Input U-phase)
(Input V-phase)
(Input W-phase)
Hall sensor
input edge sample
(every PCLKD)
(5)
GPT_UVWEDGE
PCLKD
From Hall element
GTIU
GTIV
GTIW
Synchronize
noise filter
External input (U/V/W)
Input selection
selector
OPS internal node name
(gtu_sync)
(gtv_sync)
(gtw_sync)
(2)
Input phase de-code
6-phase enable gen
OPS internal node name
(gtuup_en, gtulo_en)
(gtvup_en, gtvlo_en)
(gtwup_en, gtwlo_en)
Output select control
(3)
To brushless DC motor
GTOUUP,
GTOULO,
GTOVUP,
GTOVLO,
GTOWUP,
GTOWLO
From GPT320.GTIOCA
PWM
Bus clock
(PCLKA)
GPT core clock
(PCLKD)

Figure 23.77 Conceptual diagram of GPT_OPS control flow

Figure 23.78 shows a 6-phase level signals output example of a GPT_OPS operation.

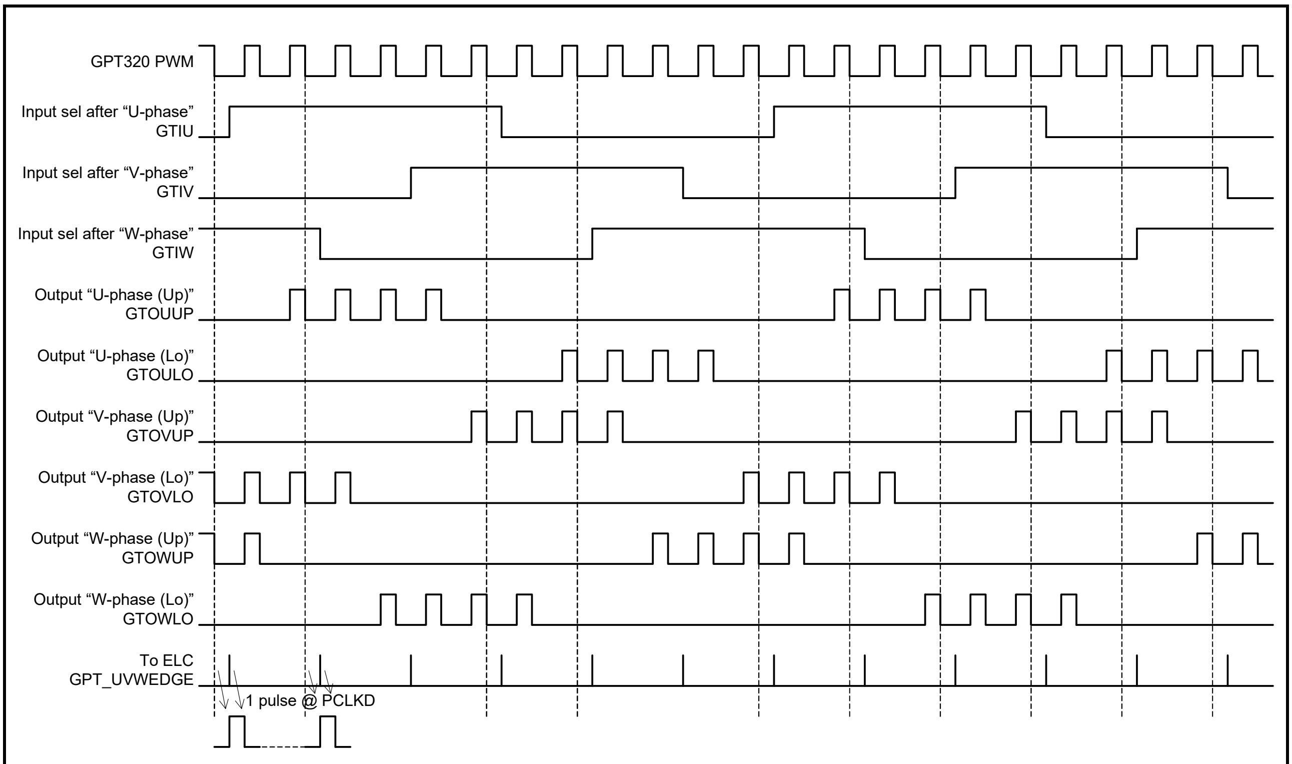
The GPT_UVWEDGE signal in Figure 23.78 is the Hall sensor input edge to ELC output.


Input sel after "U-phase"
GTIU
Input sel after "V-phase"
GTIV
Input sel after "W-phase"
GTIW
Output "U-phase (Up)"
GTOUUP
Output "U-phase (Lo)"
GTOULO
Output "V-phase (Up)"
GTOVUP
Output "V-phase (Lo)"
GTOVLO
Output "W-phase (Up)"
GTOWUP
Output "W-phase (Lo)"
GTOWLO
To ELC
GPT_UVWEDGE
1 pulse @ PCLKD

Note: Register settings: OPSCR.ALIGN = 0, OPSCR.EN = 1, OPSCR.P = 0, OPSCR.N = 0, OPSCR.INV = 0

Figure 23.78 Example of 6-phase level output operation

Figure 23.79 shows a 6-phase PWM output example of a GPT_OPS operation with chopper control.


GPT320 PWM
Input sel after "U-phase" GTIU
Input sel after "V-phase" GTIV
Input sel after "W-phase" GTIW
Output "U-phase (Up)" GTOUUP
Output "U-phase (Lo)" GTOULO
Output "V-phase (Up)" GTOVUP
Output "V-phase (Lo)" GTOVLO
Output "W-phase (Up)" GTOWUP
Output "W-phase (Lo)" GTOWLO
To ELC GPT_UVWEDGE
1 pulse @ PCLKD

Note: Register settings: OPSCR.ALIGN = 1, OPSCR.EN = 1, OPSCR.P = 1, OPSCR.N = 1, OPSCR.INV = 0

Figure 23.79 Example of 6-phase PWM output operation with chopper control

Figure 23.80 shows an example of output disable control (6-phase PWM output operation).

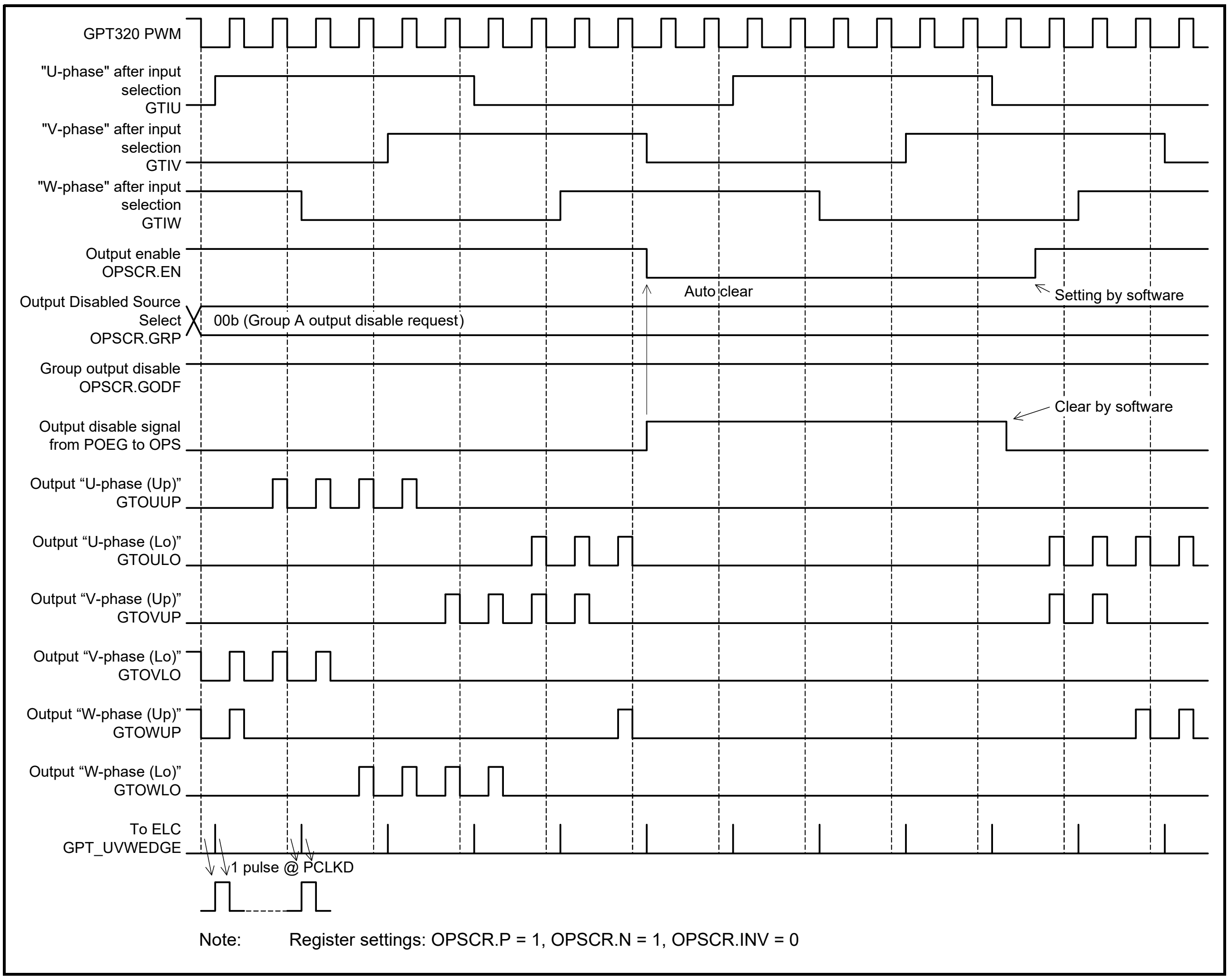

GPT320 PWM
"U-phase" after input selection GTIU
"V-phase" after input selection GTIV
"W-phase" after input selection GTIW
Output enable OPSCR.EN
Auto clear
Setting by software
Output Disabled Source Select OPSCR.GRP
00b (Group A output disable request)
Group output disable OPSCR.GODF
Clear by software
Output disable signal from POEG to OPS
Output "U-phase (Up)" GTOUUP
Output "U-phase (Lo)" GTOULO
Output "V-phase (Up)" GTOVUP
Output "V-phase (Lo)" GTOVLO
Output "W-phase (Up)" GTOWUP
Output "W-phase (Lo)" GTOWLO
To ELC GPT_UVWEDGE
1 pulse @ PCLKD
Note: Register settings: OPSCR.P = 1, OPSCR.N = 1, OPSCR.INV = 0

Figure 23.80 Example of group output disable control operation

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 software settings and external input by the OPSCR.FB bit.

When OPSCR.FB bit is 0, select the external input. Enable the input signal after synchronization with the GPT core 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 is 1.

When OPSCR.FB bit is 1, select the soft setting (OPSCR.UF, VF, WF) with the value of the input phase of PWM (PWM of GPT320.GTIOCA) using falling edge sampling with OPSCR.ALIGN bit is 1.

When OPSCR.ALIGN bit is 0, GPT_OPS operates with the input phase of PCLKD synchronization with either OPSCR.FB bit is 0 or OPSCR.FB bit is 1. However, in some situations, the PWM pulse width of the output U/V/W phases (PWM output mode) of switch timing (just before or 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

OPSCR register		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 synchronization + 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 synchronization + 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 bits indicate the PCLKD sampling results of the input selected by the OPSCR.FB bit.

When OPSCR.FB bit is 0 and after synchronization with the GPT core clock (PCLKD) and noise filtering (optional), OPSCR.U, V, W bits indicate the sampling results of the external input. When OPSCR.FB bit is 1, OPSCR.U, V, W bits have 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 select 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)	
OPSCR.EN bit	OPSCR.P bit	OPSCR.INV bit	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)	
OPSCR.EN bit	OPSCR.N bit	OPSCR.INV bit	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 is 1 and the signal value selected by the OPSCR.GRP[1:0] 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, set the OPSCR.EN to 1 after clearing the output disable request with software.

The timing of the OPSCR.EN bit cleared 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 ELC.

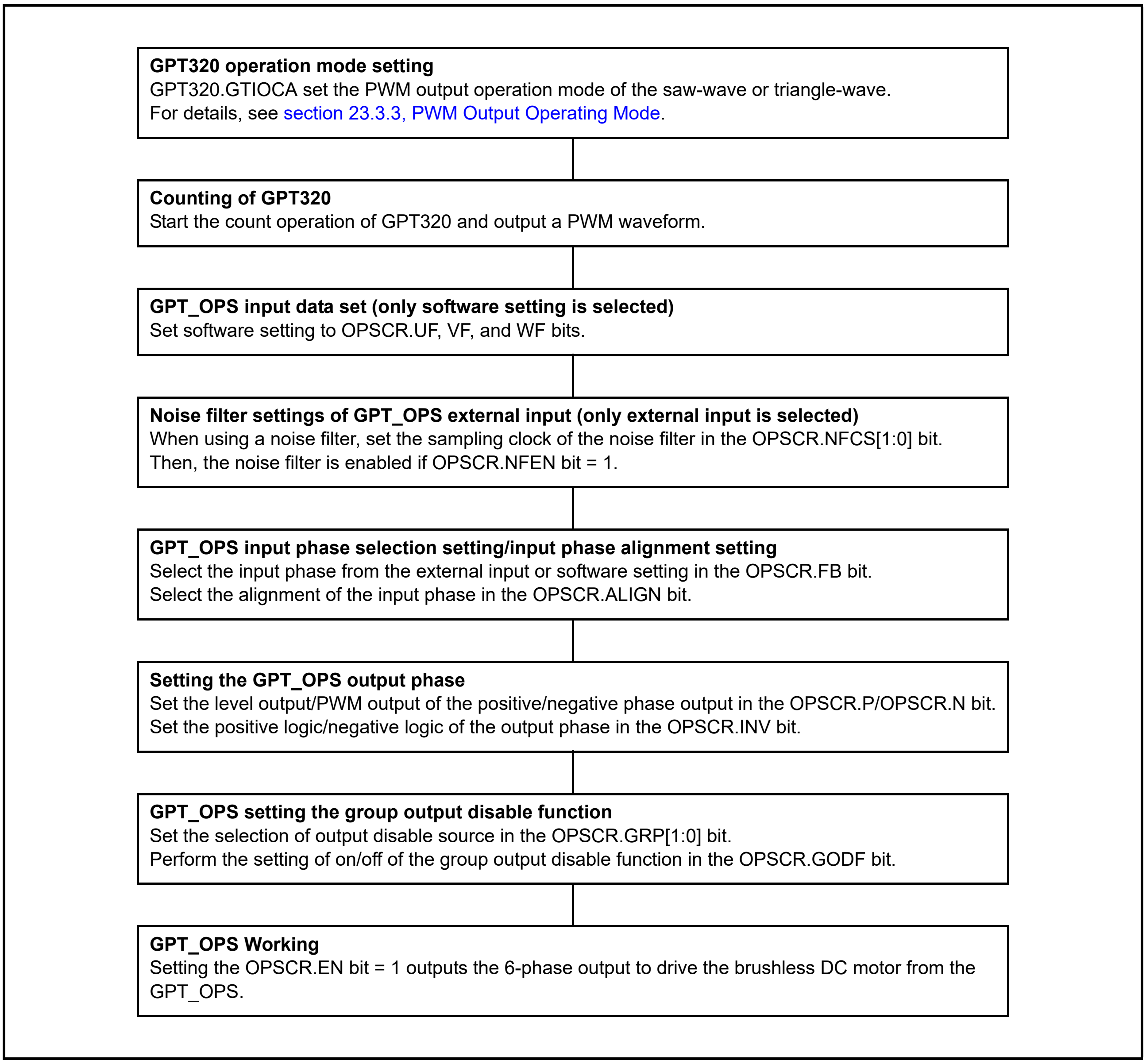
The Hall sensor input edge signal is the logical OR of the rising and falling edge signals of each U-phase/V-phase/W-phase input sampled at PCLKD. That is, if the high period of each of the U-phase/V-phase/W-phase of the input phase is short in duration, the Hall sensor edge input signal is not output at that time.

When the OPSCR.FB bit is 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 is 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



GPT320 operation mode setting
GPT320.GTIOCA set the PWM output operation mode of the saw-wave or triangle-wave.
For details, see section 23.3.3, PWM Output Operating Mode.

Counting of GPT320
Start the count operation of GPT320 and output a PWM waveform.

GPT_OPS input data set (only software setting is selected)
Set software setting to OPSCR.UF, VF, and WF bits.

Noise filter settings of GPT_OPS external input (only external input is selected)
When using a noise filter, set the sampling clock of the noise filter in the OPSCR.NFCS[1:0] bit.
Then, the noise filter is enabled if OPSCR.NFEN bit = 1.

GPT_OPS input phase selection setting/input phase alignment setting
Select the input phase from the external input or software setting in the OPSCR.FB bit.
Select the alignment of the input phase in the OPSCR.ALIGN bit.

Setting the GPT_OPS output phase
Set the level output/PWM output of the positive/negative phase output in the OPSCR.P/OPSCR.N bit.
Set the positive logic/negative logic of the output phase in the OPSCR.INV bit.

GPT_OPS setting the group output disable function
Set the selection of output disable source in the OPSCR.GRP[1:0] bit.
Perform the setting of on/off of the group output disable function in the OPSCR.GODF bit.

GPT_OPS Working
Setting the OPSCR.EN bit = 1 outputs the 6-phase output to drive the brushless DC motor from the GPT_OPS.

Figure 23.81 Example 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 2)

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
4	GPT4_CCMPA	GPT164.GTCCRA input capture/compare match	TCFA	Possible
	GPT4_CCMPB	GPT164.GTCCRB input capture/compare match	TCFB	Possible
	GPT4_CMPC	GPT164.GTCCRC compare match	TCFC	Possible
	GPT4_CMPD	GPT164.GTCCRD compare match	TCFD	Possible
	GPT4_CMPE	GPT164.GTCCRE compare match	TCFE	Possible
	GPT4_CMPF	GPT164.GTCCRF compare match	TCFF	Possible
	GPT4_OVF	GPT164.GTCNT overflow (GPT164.GTPR compare match)	TCFPO	Possible
	GPT4_UDF	GPT164.GTCNT underflow	TCFPU	Possible

Table 23.21 Interrupt sources (2 of 2)

Channel	Name	Interrupt source	Interrupt flag	DMAC/DTC activation
5	GPT5_CCMPA	GPT165.GTCCRA input capture/compare match	TCFA	Possible
	GPT5_CCMPB	GPT165.GTCCRB input capture/compare match	TCFB	Possible
	GPT5_CMPC	GPT165.GTCCRC compare match	TCFC	Possible
	GPT5_CMPD	GPT165.GTCCRD compare match	TCFD	Possible
	GPT5_CMPE	GPT165.GTCCRE compare match	TCFE	Possible
	GPT5_CMPF	GPT165.GTCCRF compare match	TCFF	Possible
	GPT5_OVF	GPT165.GTCNT overflow (GPT165.GTPR compare match)	TCFPO	Possible
	GPT5_UDF	GPT165.GTCNT underflow	TCFPU	Possible
8	GPT8_CCMPA	GPT168.GTCCRA input capture/compare match	TCFA	Possible
	GPT8_CCMPB	GPT168.GTCCRB input capture/compare match	TCFB	Possible
	GPT8_CMPC	GPT168.GTCCRC compare match	TCFC	Possible
	GPT8_CMPD	GPT168.GTCCRD compare match	TCFD	Possible
	GPT8_CMPE	GPT168.GTCCRE compare match	TCFE	Possible
	GPT8_CMPF	GPT168.GTCCRF compare match	TCFF	Possible
	GPT8_OVF	GPT168.GTCNT overflow (GPT168.GTPR compare match)	TCFPO	Possible
	GPT8_UDF	GPT168.GTCNT underflow	TCFPU	Possible

(1) GPTn_CCMPA interrupt (n = 0 to 5, 8)

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 5, 8)

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 5, 8)

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 5, 8)

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 5, 8)

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 5, 8)

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 5, 8)

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 5, 8)

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 (1 of 2)

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)

Table 23.22 Interrupt signals and interrupt status flags (2 of 2)

Interrupt signal	Interrupt status flag
GPTn_CCMPA	GTST[0] (TCFA)

Note: n = 0 to 5, 8

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:

- Generation of compare match A interrupt (GPTn_CCMPA)
- Generation of compare match B interrupt (GPTn_CCMPB)
- Generation of compare match C interrupt (GPTn_CMPC)
- Generation of compare match D interrupt (GPTn_CMPD)
- Generation of compare match E interrupt (GPTn_CMPE)
- Generation of compare match F interrupt (GPTn_CMPF)
- Generation of overflow interrupt (GPTn_OVF)
- Generation of underflow interrupt (GPTn_UDF).

Note: n = 0 to 5, 8

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 details 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.

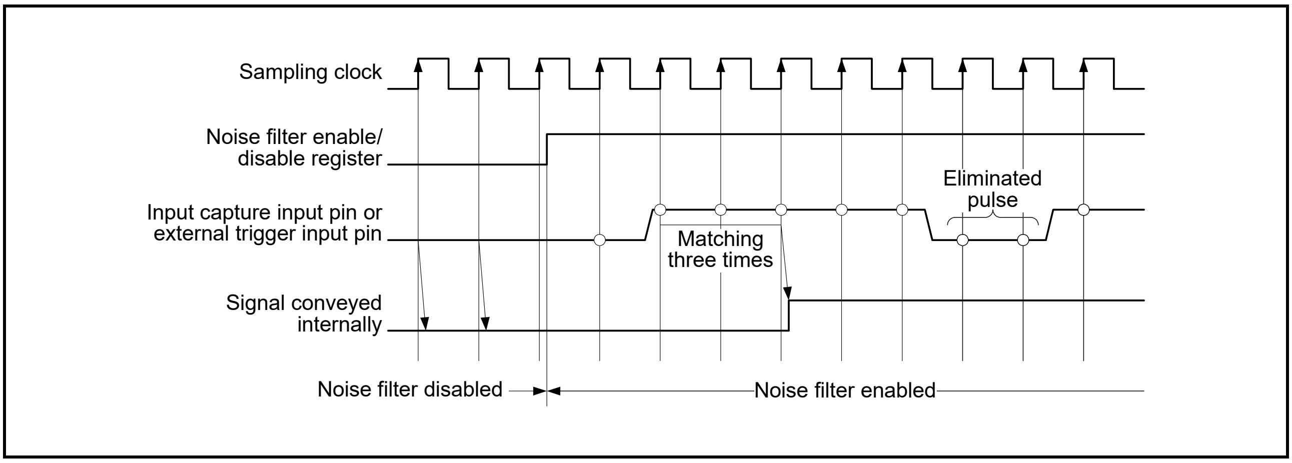

Sampling clock
Noise filter enable/
disable register
Input capture input pin or
external trigger input pin
Eliminated
pulse
Matching
three times
Signal conveyed
internally
Noise filter disabled
Noise filter enabled

Figure 23.82 Timing of noise filtering

If noise filtering is enabled, the input capture operation or external trigger operation is performed on the edges of the noise filtered signal after a delay of a sampling interval × 3 + PCLKD. This is caused by 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, GTICBSR, 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 the buffer register write is delayed 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 buffer register write and 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.

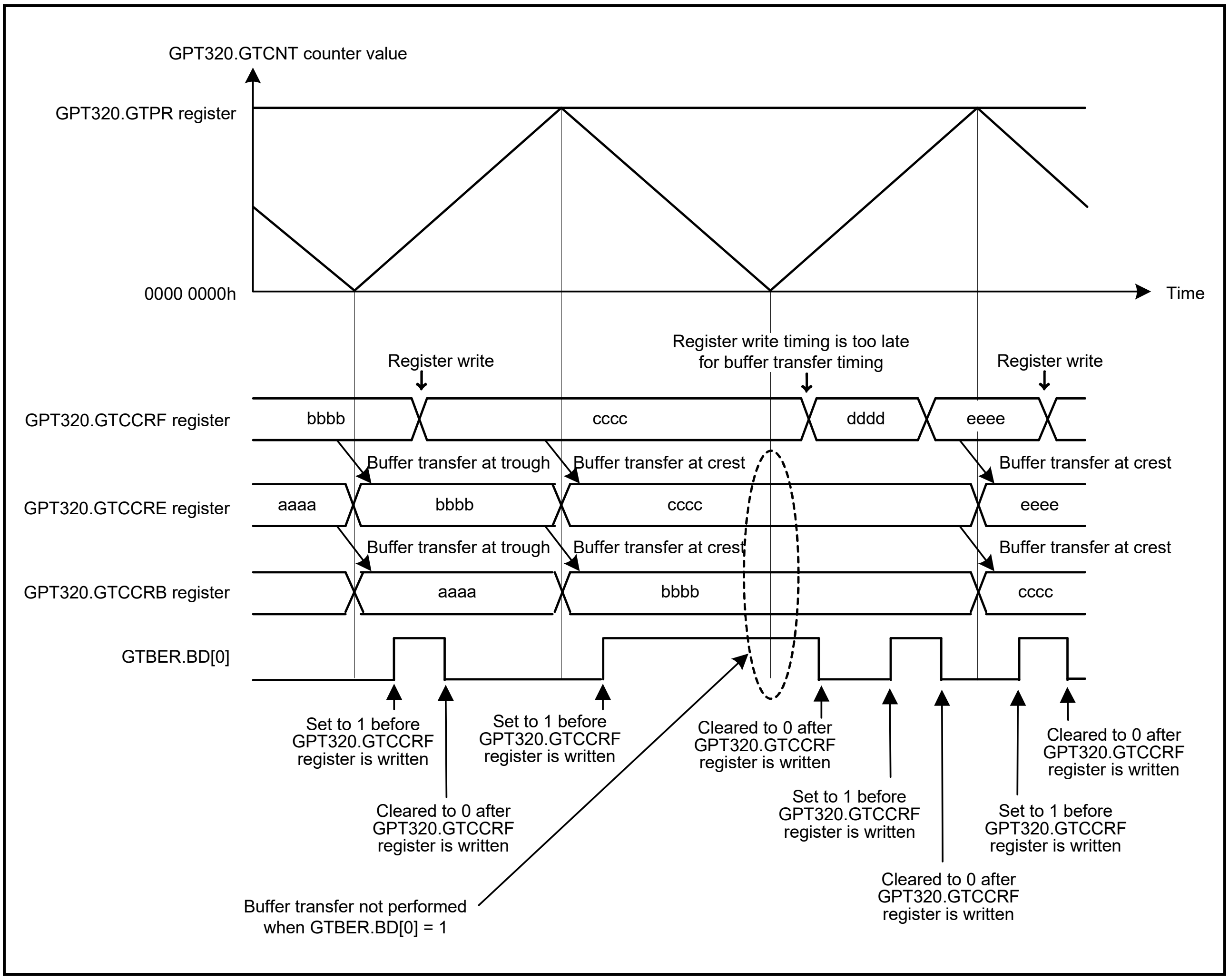

GPT320.GTCNT counter value
GPT320.GTPR register
0000 0000h
Time
Register write
Register write timing is too late for buffer transfer timing
Register write
GPT320.GTCCRF register
bbbb
cccc
dddd
eeee
Buffer transfer at trough
Buffer transfer at crest
Buffer transfer at crest
GPT320.GTCCRE register
aaaa
bbbb
cccc
eeee
Buffer transfer at trough
Buffer transfer at crest
Buffer transfer at crest
GPT320.GTCCRB register
aaaa
bbbb
cccc
GTBER.BD[0]
Set to 1 before GPT320.GTCCRF register is written
Set to 1 before GPT320.GTCCRF register is written
Cleared to 0 after GPT320.GTCCRF register is written
Cleared to 0 after GPT320.GTCCRF register is written
Cleared to 0 after GPT320.GTCCRF register is written
Set to 1 before GPT320.GTCCRF register is written
Set to 1 before GPT320.GTCCRF register is written
Cleared to 0 after GPT320.GTCCRF register is written
Cleared to 0 after GPT320.GTCCRF register is written
Buffer transfer not performed when GTBER.BD[0] = 1

Figure 23.83 **Example of operation for disabling buffer operation with triangle waves, double buffer operation, and 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 based on the settings in the 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 shared 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 based on the GTIOR.OADF[1:0] setting for the GTIOCA pin and the GTIOR.OBDF[1:0] setting for the 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 should be released immediately without waiting for the 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.

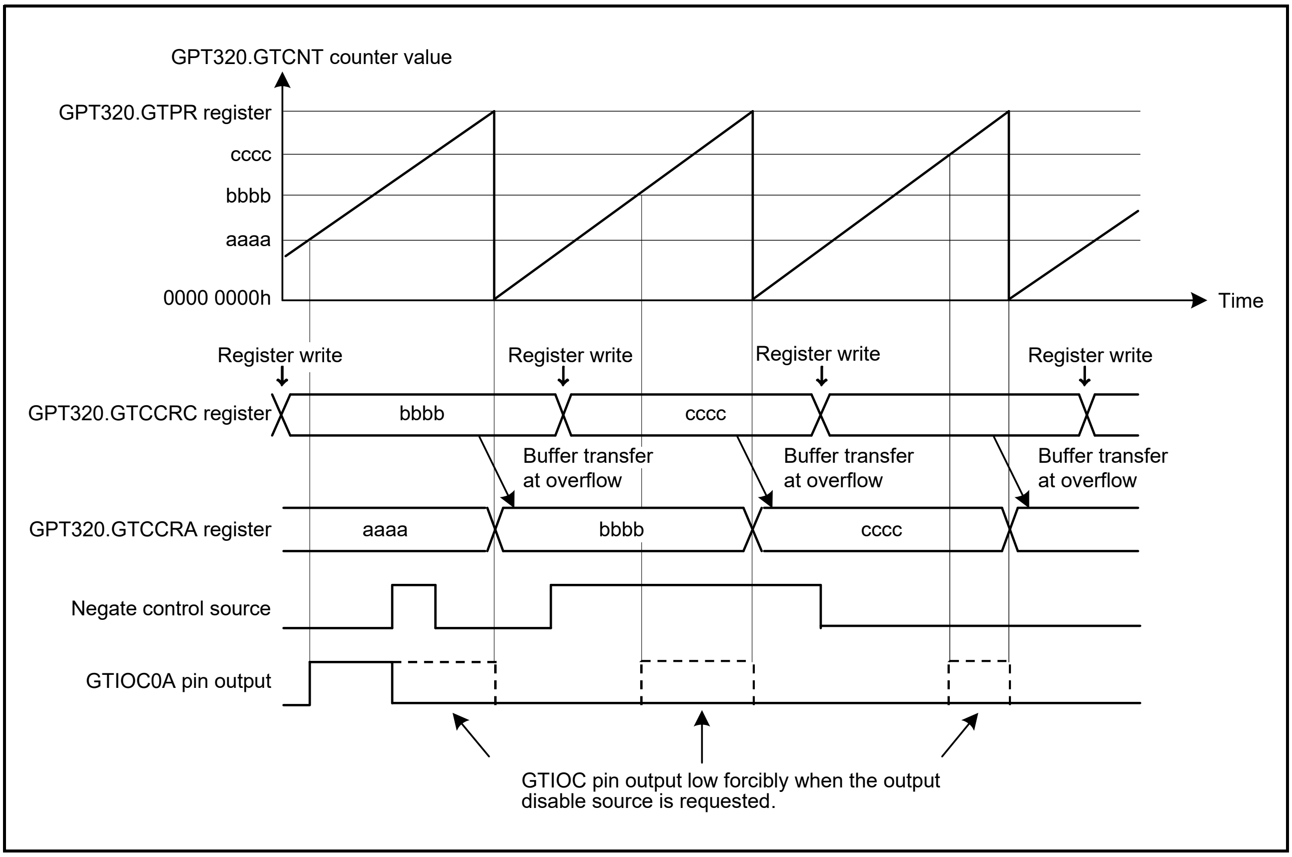

GPT320.GTCNT counter value
GPT320.GTPR register
cccc
bbbb
aaaa
0000 0000h
Time
Register write
Register write
Register write
Register write
GPT320.GTCCRC register
bbbb
cccc
Buffer transfer at overflow
Buffer transfer at overflow
Buffer transfer at overflow
GPT320.GTCCRA register
aaaa
bbbb
cccc
Negate control source
GTIOC0A pin output
GTIOC pin output low forcibly when the output disable source is requested.

Figure 23.84 Example of GTIOC pin output disable control operation in saw-wave up-counting, buffer operation, active level 1, high output at GTCCRA compare match, low output at cycle end, and low output at output disable

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.

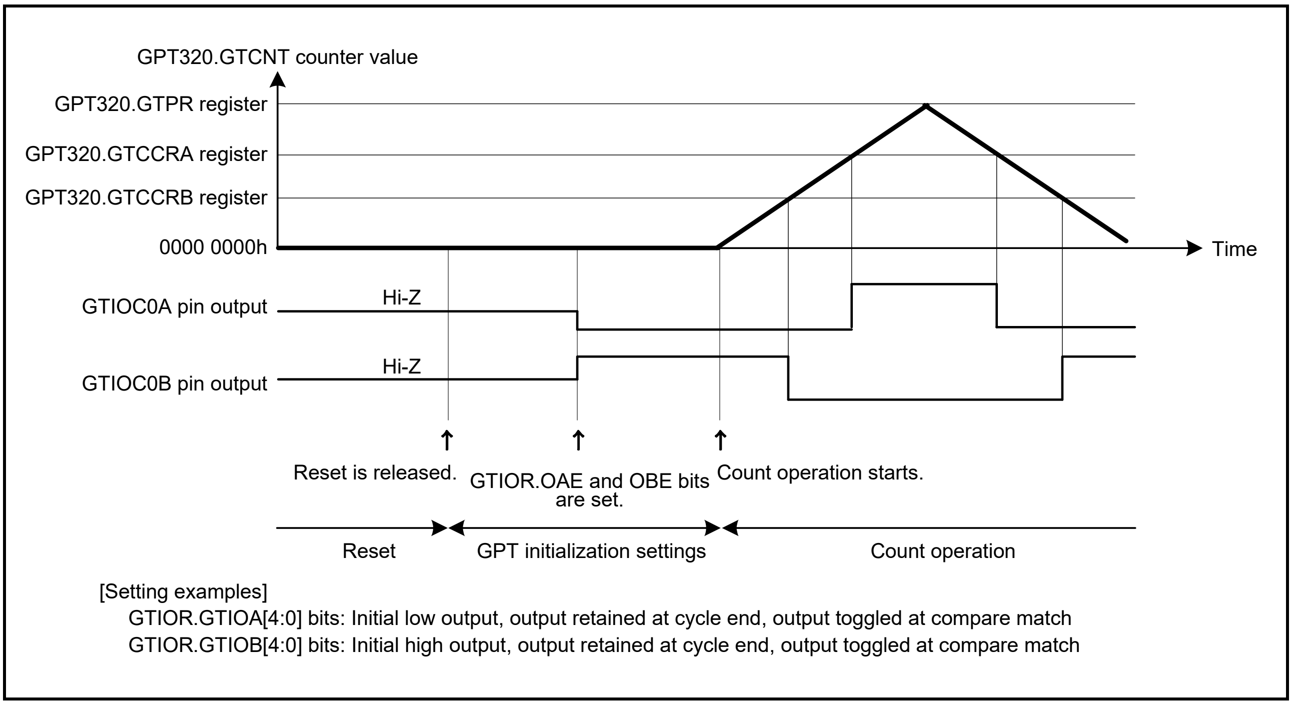

GPT320.GTCNT counter value
GPT320.GTPR register
GPT320.GTCCRA register
GPT320.GTCCRB register
0000 0000h
Time
GTIOC0A pin output
Hi-Z
GTIOC0B pin output
Hi-Z
Reset is released.
GTIOR.OAE and OBE bits are set.
Count operation starts.
Reset
GPT initialization settings
Count operation
[Setting examples]
GTIOR.GTIOA[4:0] bits: Initial low output, output retained at cycle end, output toggled at compare match
GTIOR.GTIOB[4:0] bits: Initial high output, output retained at cycle end, output toggled at compare match

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 the OAHLD and OBHLD bits in GTIOR to 1 and retain the outputs at count stop
- Set the OAHLD and OBHLD bits in GTIOR to 0, specify arbitrary output values at OADFLT and OBDFLT in GTIOR, and output the arbitrary values at count stop
- Set the pin to output an arbitrary value as a general output port by setting the PDR, PODR, 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 stopped. If counting stops, registers should be initialized before counting starts.

23.9 Usage Notes

23.9.1 Module-Stop Function Setting

The Module Stop Control Register can enable or disable GPT operation. The GPT 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.

23.9.2 GTCCRn Settings during Compare Match Operation (n = A to F)

(1) When automatic dead time setting is made in triangle-wave PWM mode

The GTCCRA register must satisfy 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 Range for GTCNT Counter

The GTCNT counter register must be set with the range of 0 ≤ GTCNT ≤ GTPR.

23.9.4 Starting and Stopping the GTCNT Counter

The control timing of starting and stopping the GTCNT counter by the GTCR.CST bit synchronizes the count clock that is selected in GTCR.TPCS[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. On the other side, there might be 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 the GTCNT register.

Table 23.23 Priority order of sources updating GTCNT

Source 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), writing by CPU has 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 writing to GTCCRm registers, writing to GTCCRm registers has priority over input capture/buffer transfer operation. When there is a conflict between input capture and writing to the counter register by the CPU or updating the counter register by hardware sources, the pre-update counter value is captured. When there is a conflict between updating the GTCCRm registers and reading by the CPU, pre-update data is read.

(4) GTPR registers

When there is a conflict between buffer transfer operation and writing to the GTPR register, writing to GTPR register has priority over buffer transfer operation. When there is a conflict between updating GTPR register and reading by the CPU, pre-update data is read.

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.

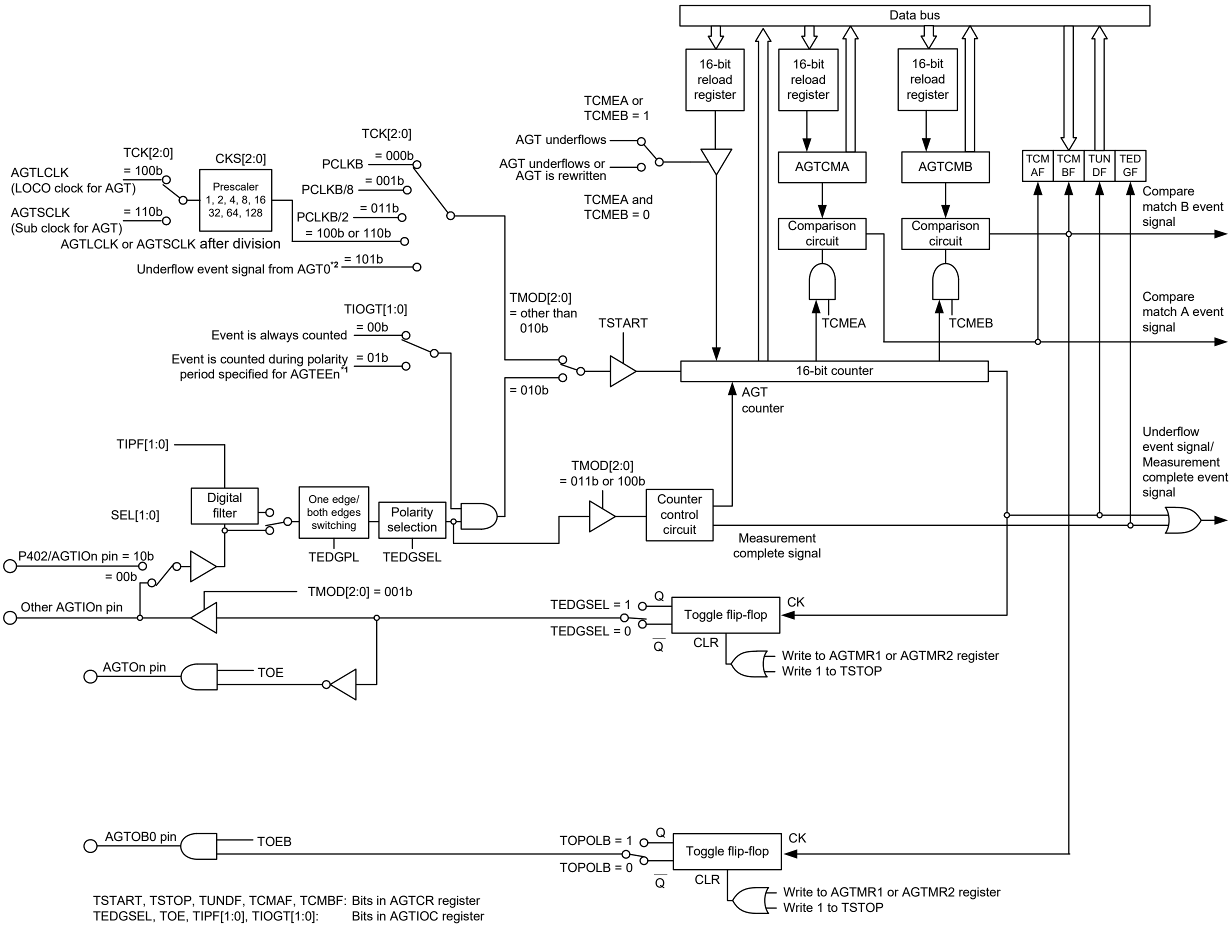
This 16-bit timer consists of a reload register and a down counter. The reload register and the down counter are allocated in the same address, and can be accessed with the AGT register.

Table 24.1 lists the AGT specifications, Figure 24.1 shows the AGT block diagram, and Table 24.2 lists the AGT pin configuration.

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, PCLKB/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)[*3] • 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 3. Compare match A does not have a compare match output pin.


Data bus
16-bit reload register
16-bit reload register
16-bit reload register
TCMEA or TCMEB = 1
AGT underflows
AGT underflows or AGT is rewritten
TCMEA and TCMEB = 0
AGTCMA
AGTCMB
TCM AF
TCM BF
TUN DF
TED GF
Compare match B event signal
Comparison circuit
Comparison circuit
TCK[2:0]
TCK[2:0]
CKS[2:0]
AGTLCLK (LOCO clock for AGT)
= 100b
Prescaler 1, 2, 4, 8, 16 32, 64, 128
PCLKB
= 000b
= 001b
PCLKB/8
= 011b
PCLKB/2
= 100b or 110b
AGTSCLK (Sub clock for AGT)
= 110b
AGTLCLK or AGTSCLK after division
Underflow event signal from AGT0*2
= 101b
TMOD[2:0] = other than 010b
TSTART
TCMEA
TCMEB
Compare match A event signal
TIOGT[1:0]
Event is always counted
= 00b
Event is counted during polarity period specified for AGTEEn*1
= 01b
16-bit counter
= 010b
AGT counter
Underflow event signal/ Measurement complete event signal
TIPF[1:0]
TMOD[2:0] = 011b or 100b
SEL[1:0]
Digital filter
One edge/ both edges switching
Polarity selection
Counter control circuit
Measurement complete signal
TEDGPL
TEDGSEL
P402/AGTIOn pin = 10b
= 00b
Other AGTIOn pin
TMOD[2:0] = 001b
TEDGSEL = 1
TEDGSEL = 0
Q
Q̄
Toggle flip-flop
CK
CLR
Write to AGTMR1 or AGTMR2 register
Write 1 to TSTOP
AGTOn pin
TOE
AGTOB0 pin
TOEB
TOPOLB = 1
TOPOLB = 0
Q
Q̄
Toggle flip-flop
CK
CLR
Write to AGTMR1 or AGTMR2 register
Write 1 to TSTOP

TSTART, TSTOP, TUNDF, TCMAF, TCMBF: Bits in AGTCR register
TEDGSEL, TOE, TIPF[1:0], TIOGT[1:0]: Bits in AGTIOC register
TMOD[2:0], TEDGPL, TCK[2:0]: Bits in AGTMR1 register
CKS[2:0]: Bits in AGTMR2 register
TCMEA, TCMEB, TOEB, TOPOLB: Bits in AGTCMSR register
SEL[1:0]: Bits in AGTIOSEL register

Note 1. The polarity can be selected by the EEPS bit in the AGTISR register.
Note 2. AGT0 cannot use AGT underflow event. 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
AGTOB0	Output	Output compare match B output for AGT

Note: Channel number (n = 0, 1)
Note 1. When AGTIO1 is assigned P402, AGTIO1 can only be used as inputs.

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) or 011b (PCLKB/2), 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 with 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 stopped 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 bit is set to 0 (count stopped) 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: PCLKB 0 0 1: PCLKB/8 0 1 1: PCLKB/2 1 0 0: Divided clock AGTLCLK specified by CKS[2:0] bits in the AGTR2 register 1 0 1: Underflow event signal from AGT0*6 1 1 0: Divided clock AGTSCLK specified by CKS[2:0] bits in the AGTR2 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, and AGTOB0 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. Only switch count sources when both the TSTART and TCSTF bits in the AGTCR register are set to 0 (count is stopped).

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 is stopped). Do not change the operating mode during count operation.

Note 4. The TEDGPL bit is enabled only in event counter mode.

Note 5. To run AGT in Software Standby mode, select AGTLCLK or AGTSCLK.

Note 6. AGT0 cannot use AGT0 underflow. 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. Only rewrite CKS[2:0] 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)

The LPM bit sets the low power operation, which impacts access to certain AGT registers. Set this bit to 1 to operate in low power. When this bit is 1, access to the following registers is prohibited:

- AGT/AGTCMA/AGTCMB/AGTCR.

After this bit is switched from 1 to 0, the first access to the register is constrained as follows:

- 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

Bit	Symbol	Bit name	Description	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	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 is used, the polarity to count an event can be selected with the EEPS bit in the AGTISR register.
Note 2. TIOGT[1:0] bits are enabled only in event counter mode.
Note 3. When event counter mode operation is performed during Software Standby mode, the digital filter function cannot be used.

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 on rising edge 1: Count on falling edge.
Pulse width measurement mode	0: Low-level width is measured 1: High-level width is measured.
Pulse period measurement mode	0: Measure from one rising edge to the next rising edge 1: Measure from one falling edge to the next falling edge.

Table 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	—	—	—	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: Compare match A register disabled 1: Compare match A register enabled.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	TCMEB	Compare match B register enable [*1, *2]	0: Compare match B register disabled 1: Compare match B register enabled.	R/W
b5	TOEB	AGTOBn output enable [*1, *2, *3]	0: AGTOBn output disabled 1: AGTOBn output enabled.	R/W
b6	TOPOLB	AGTOBn polarity select [*1, *2, *3]	0: AGTOBn output is started on low 1: AGTOBn output is started on 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. Only rewrite the AGTCMSR register when both the TSTART and TCSTF bits in the AGTCR register are set to 0 (count is stopped).
Note 2. Do not set 1 when in pulse width measurement mode or pulse period measurement mode.
Note 3. This bit is not available in AGT1.

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 [*1]	b1 b0 0 0: Select the AGTIOn except for the following pins 0 1: Setting prohibited 1 0: Select the P402/AGTIOn. P402/AGTIOn is input only. It is not possible to output 1 1: Setting prohibited.	R/W
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. You must set the Pin Function Select Register. See section 20, I/O Ports.

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)

The SEL[1:0] bits select the AGTIOn pin function.

TIES bit (AGTIOn Input Enable)

The TIES bit 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 differs depending on the value of the TSTART bit in the AGTCR register and of the TCMEA or TCMEB bit in the AGTCMSR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and the counter. When the TSTART bit is 1 (count starts) and the TCMEA and TCMEB bits are 0 (compare match A/B register are invalid), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the next count source. When the TSTART bit is 1 (count starts) and the TCMEA bit or 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.

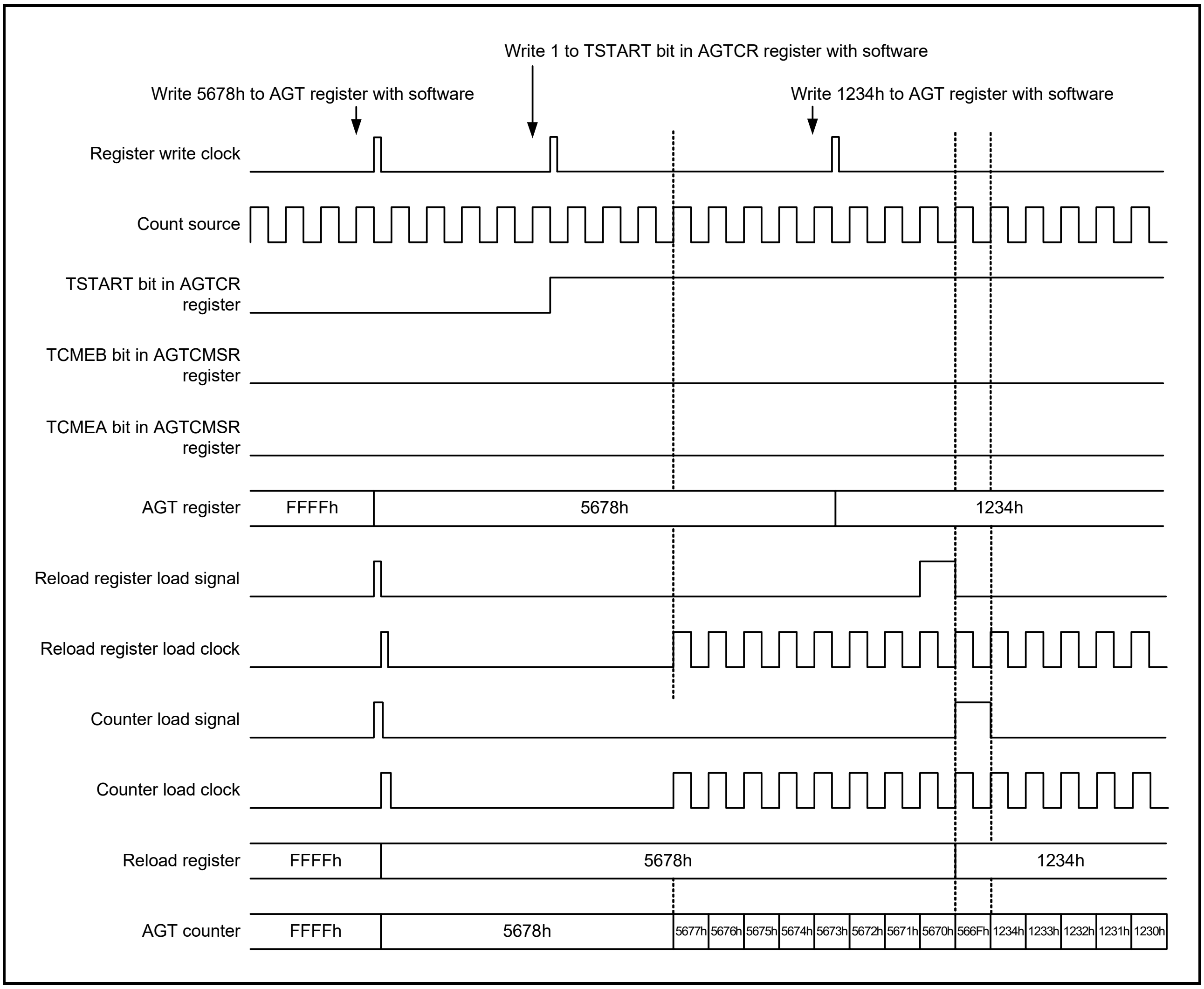

Write 1 to TSTART bit in AGTCR register with software
Write 5678h to AGT register with software
Write 1234h to AGT register with software
Register write clock
Count source
TSTART bit in AGTCR register
TCMEB bit in AGTCMSR register
TCMEA bit in AGTCMSR register
AGT register
FFFFh
5678h
1234h
Reload register load signal
Reload register load clock
Counter load signal
Counter load clock
Reload register
FFFFh
5678h
1234h
AGT counter
FFFFh
5678h
5677h
5676h
5675h
5674h
5673h
5672h
5671h
5670h
566Fh
1234h
1233h
1232h
1231h
1230h

Figure 24.2 Timing of rewrite operation with TSTART bit value and TCMEA or TCMEB bit value when compare match A register or compare match B register is invalid

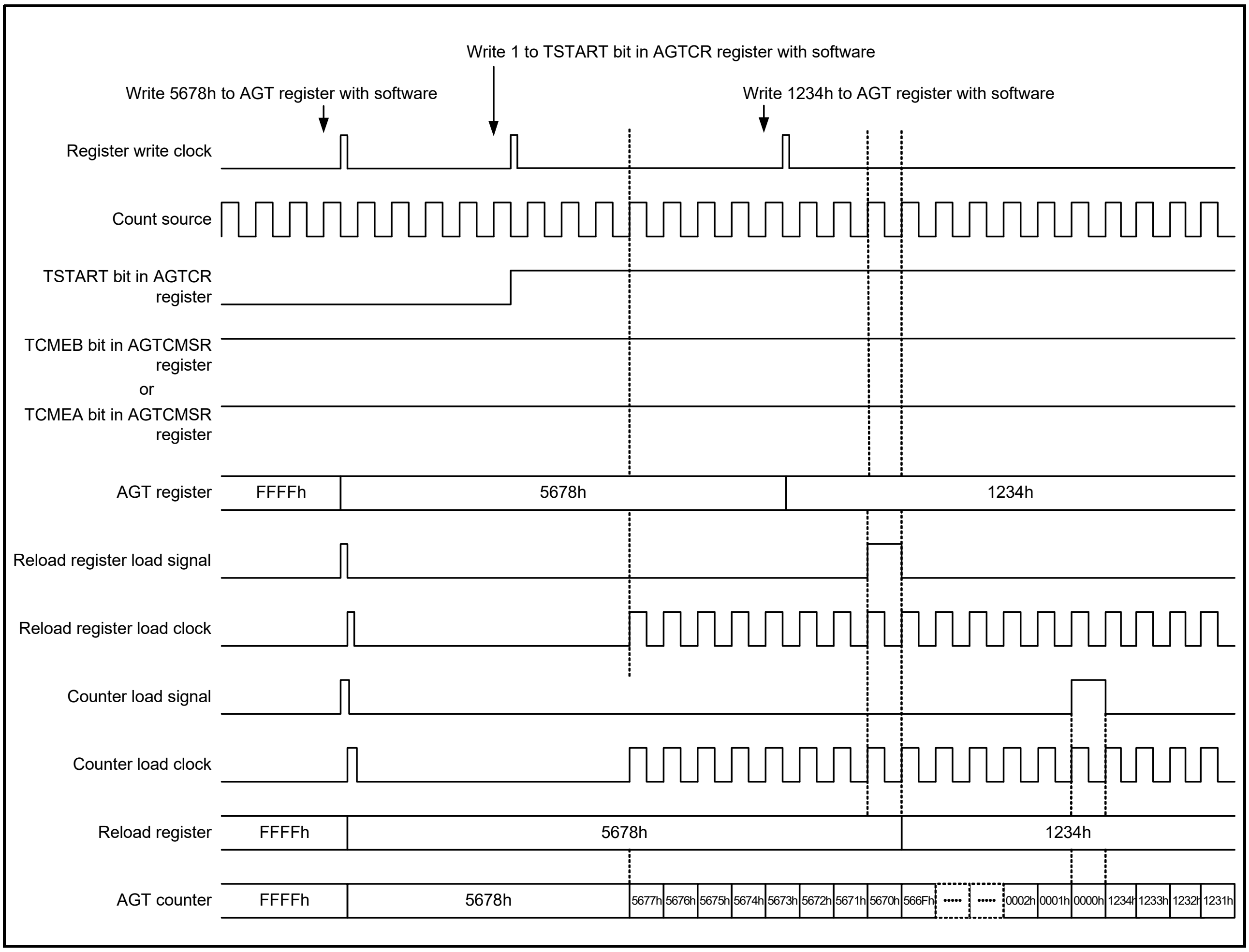

Write 1 to TSTART bit in AGTCR register with software
Write 5678h to AGT register with software
Write 1234h to AGT register with software
Register write clock
Count source
TSTART bit in AGTCR register
TCMEB bit in AGTCMSR register
or
TCMEA bit in AGTCMSR register
AGT register
FFFFh
5678h
1234h
Reload register load signal
Reload register load clock
Counter load signal
Counter load clock
Reload register
FFFFh
5678h
1234h
AGT counter
FFFFh
5678h
5677h
5676h
5675h
5674h
5673h
5672h
5671h
5670h
566Fh
·····
·····
0002h
0001h
0000h
1234h
1233h
1232h
1231h

Figure 24.3 **Timing of rewrite operation with TSTART bit value and TCMEA or TCMEB bit value 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 of the TSTART bit in the AGTCR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and 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.

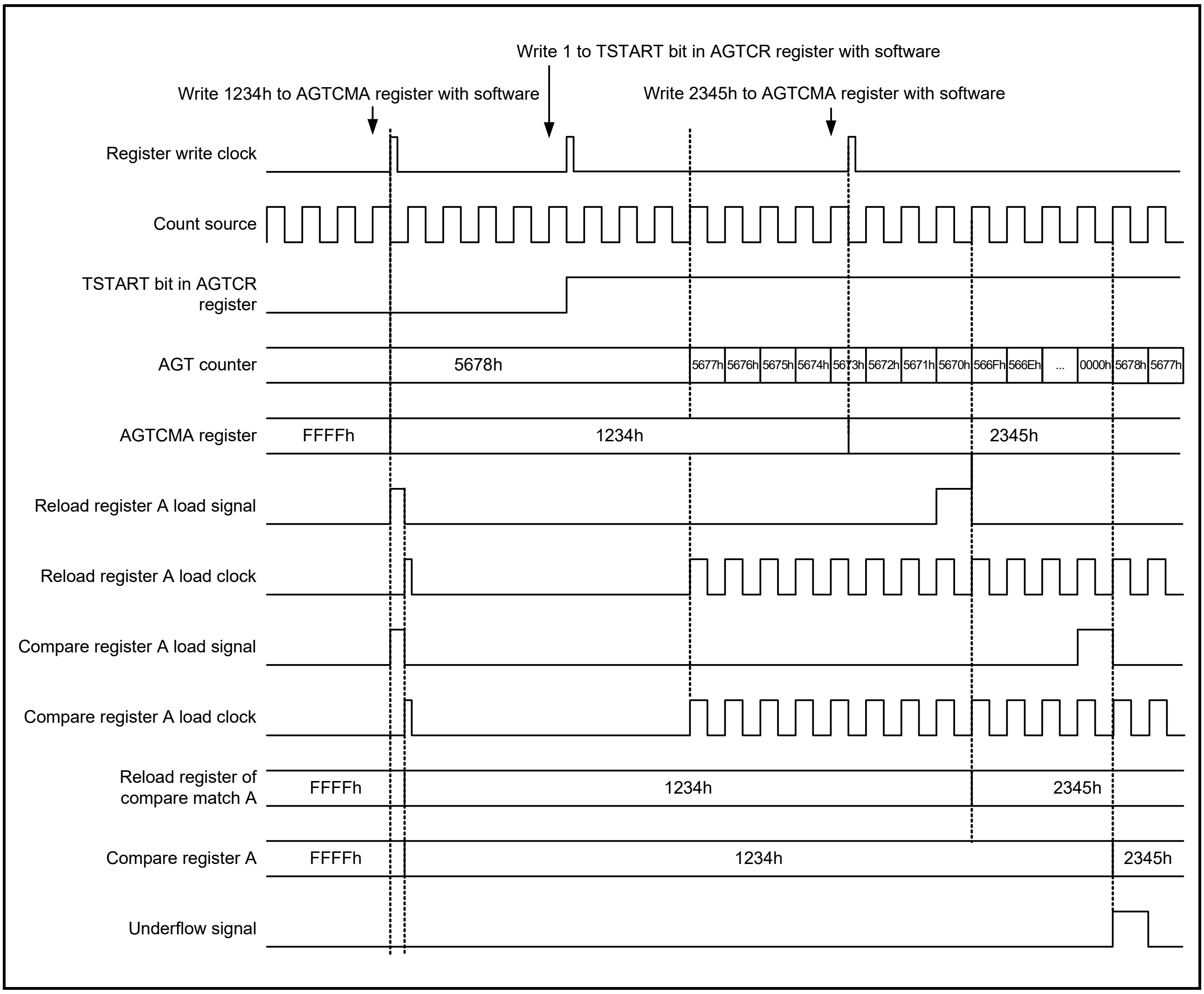

Write 1 to TSTART bit in AGTCR register with software
Write 1234h to AGTCMA register with software
Write 2345h to AGTCMA register with software
Register write clock
Count source
TSTART bit in AGTCR register
AGT counter
5678h
5677h
5676h
5675h
5674h
5673h
5672h
5671h
5670h
566Fh
566Eh
...
0000h
5678h
5677h
AGTCMA register
FFFFh
1234h
2345h
Reload register A load signal
Reload register A load clock
Compare register A load signal
Compare register A load clock
Reload register of compare match A
FFFFh
1234h
2345h
Compare register A
FFFFh
1234h
2345h
Underflow signal

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 with the TCK[2:0] bits in the AGTMR1 register. In timer mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 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.

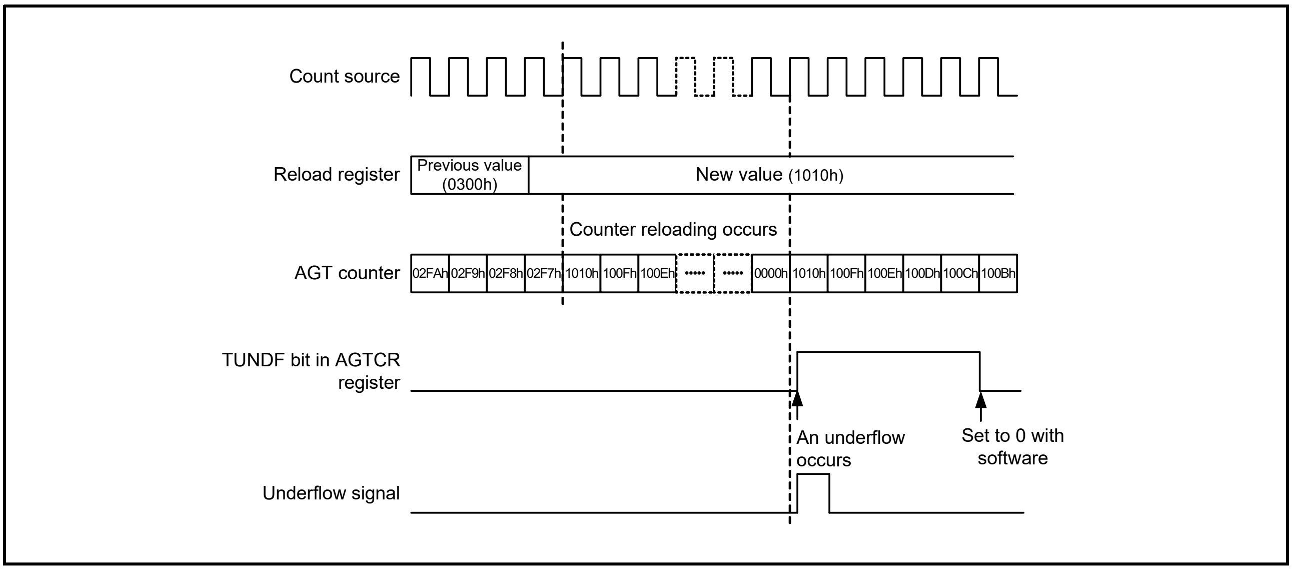

Count source
Reload register
Previous value (0300h)
New value (1010h)
Counter reloading occurs
AGT counter
02FAh
02F9h
02F8h
02F7h
1010h
100Fh
100Eh
•••••
•••••
0000h
1010h
100Fh
100Eh
100Dh
100Ch
100Bh
TUNDF bit in AGTCR register
An underflow occurs
Set to 0 with software
Underflow signal

Figure 24.5 Operation example in timer mode

24.3.4 Pulse Output Mode

In pulse output mode, the counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register, and the output level of the AGTIOn and AGTOn pins inverted each time an underflow occurs.

In pulse output mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 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.

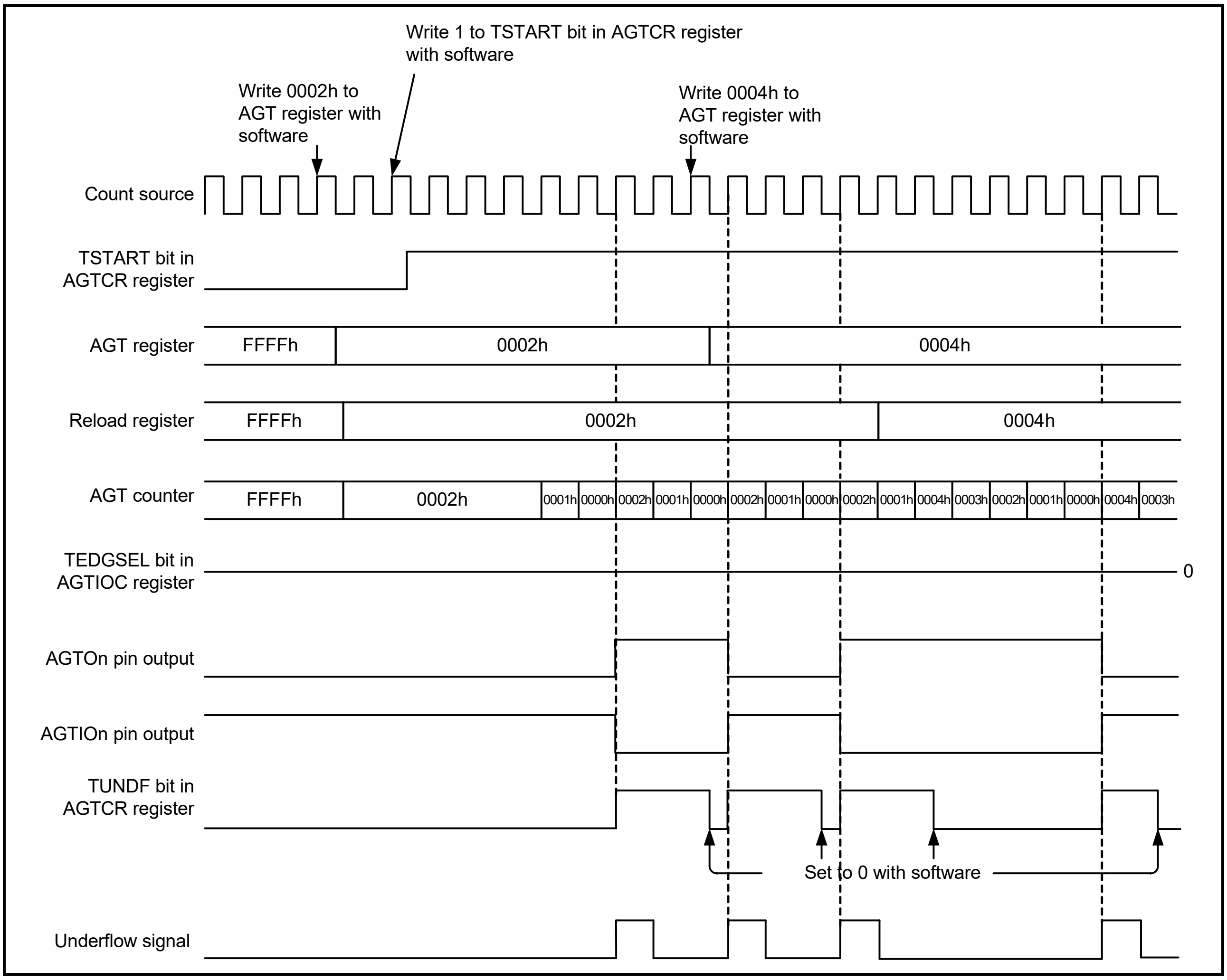

Write 1 to TSTART bit in AGTCR register with software
Write 0002h to AGT register with software
Write 0004h to AGT register with software
Count source
TSTART bit in AGTCR register
AGT register
FFFFh
0002h
0004h
Reload register
FFFFh
0002h
0004h
AGT counter
FFFFh
0002h
0001h
0000h
0002h
0001h
0000h
0002h
0001h
0000h
0002h
0001h
0004h
0003h
0002h
0001h
0000h
0004h
0003h
TEDGSEL bit in AGTIOC register
0
AGTOn pin output
AGTIOn pin output
TUNDF bit in AGTCR register
Set to 0 with software
Underflow signal

Figure 24.6 Operation example in pulse output mode

24.3.5 Event Counter Mode

In event counter 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.

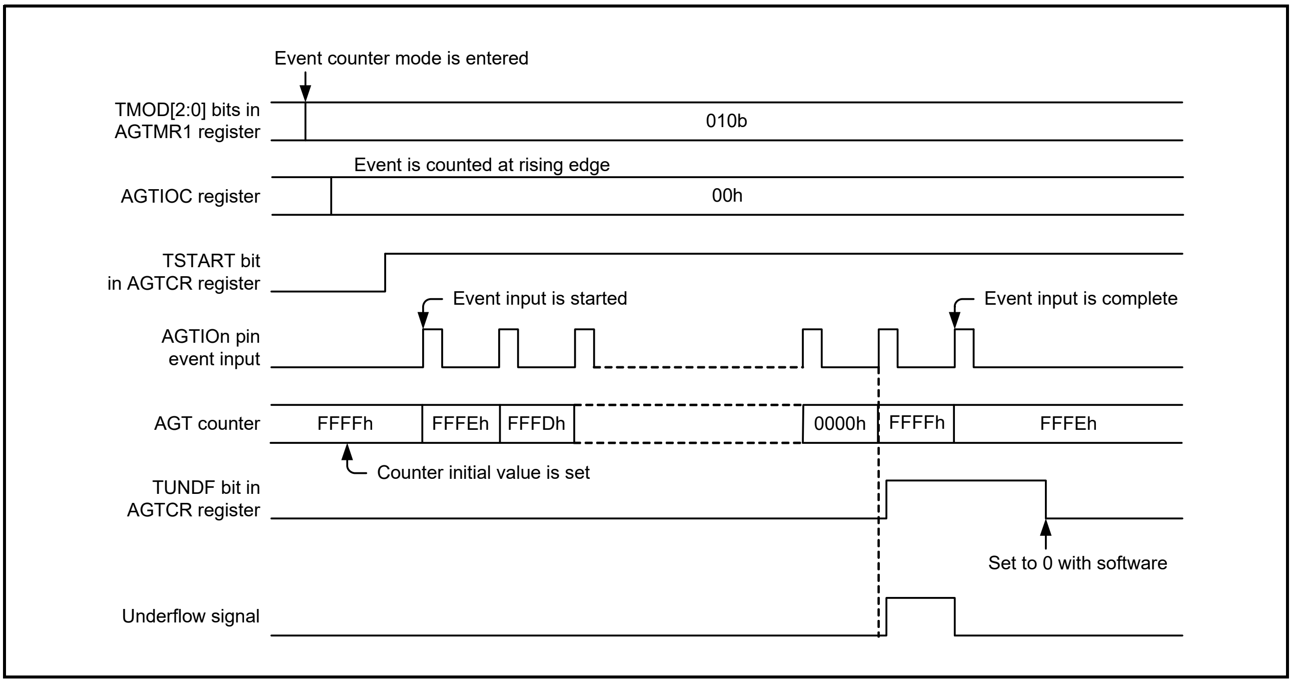
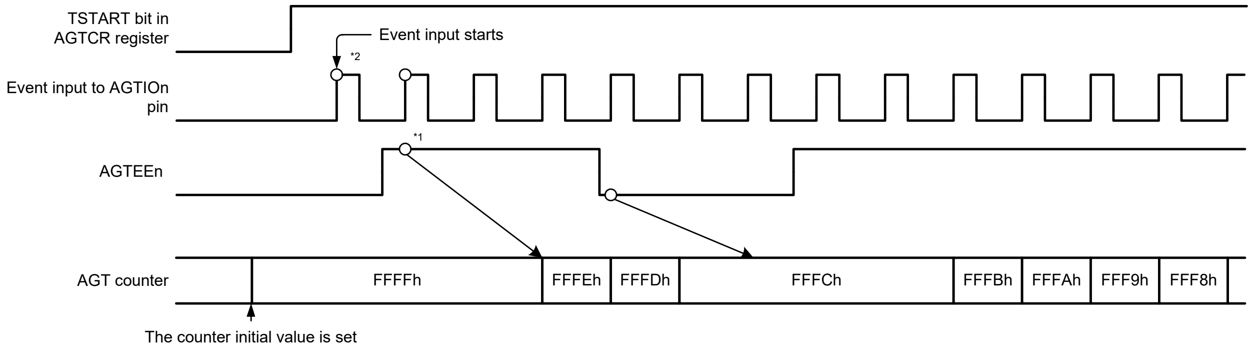

Event counter mode is entered
TMOD[2:0] bits in AGTMR1 register
010b
Event is counted at rising edge
AGTIOC register
00h
TSTART bit in AGTCR register
Event input is started
Event input is complete
AGTIOn pin event input
AGT counter
FFFFh
FFFEh
FFFDh
0000h
FFFFh
FFFEh
Counter initial value is set
TUNDF bit in AGTCR register
Set to 0 with software
Underflow signal

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).

Timing example when the setting of operating mode is as follows :
AGTMR1 register: TMOD[2:0] = 010b (event counter mode)
AGTIOC register: TIOGT[1:0] = 01b (event is counted during specified period for external interrupt pin)
TIPF[1:0] = 00b (no filter)
TEDGSEL = 1 (count at rising edge)
AGTISR register: EEPS = 1 (high-level period is counted)


TSTART bit in AGTCR register
Event input starts
*2
Event input to AGTIOn pin
*1
AGTEEn
AGT counter
FFFFh
FFFEh
FFFDh
FFFCh
FFFBh
FFFAh
FFF9h
FFF8h
The counter initial value is set

Note 1. To control synchronization, there is a delay of 2 cycles of the count source until the count operation is affected. It is also possible that the count start timing is shifted by 1 cycle because of the phase difference between the AGTEEn 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 pulse width measurement mode, the pulse width of an external signal input to the AGTIOn pin is measured. When the level specified by the TEDGSEL bit in the AGTIOC register is input to the AGTIOn pin, the counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register. When the specified level on the AGTIOn pin ends, the counter is stopped, the TEDGF 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.

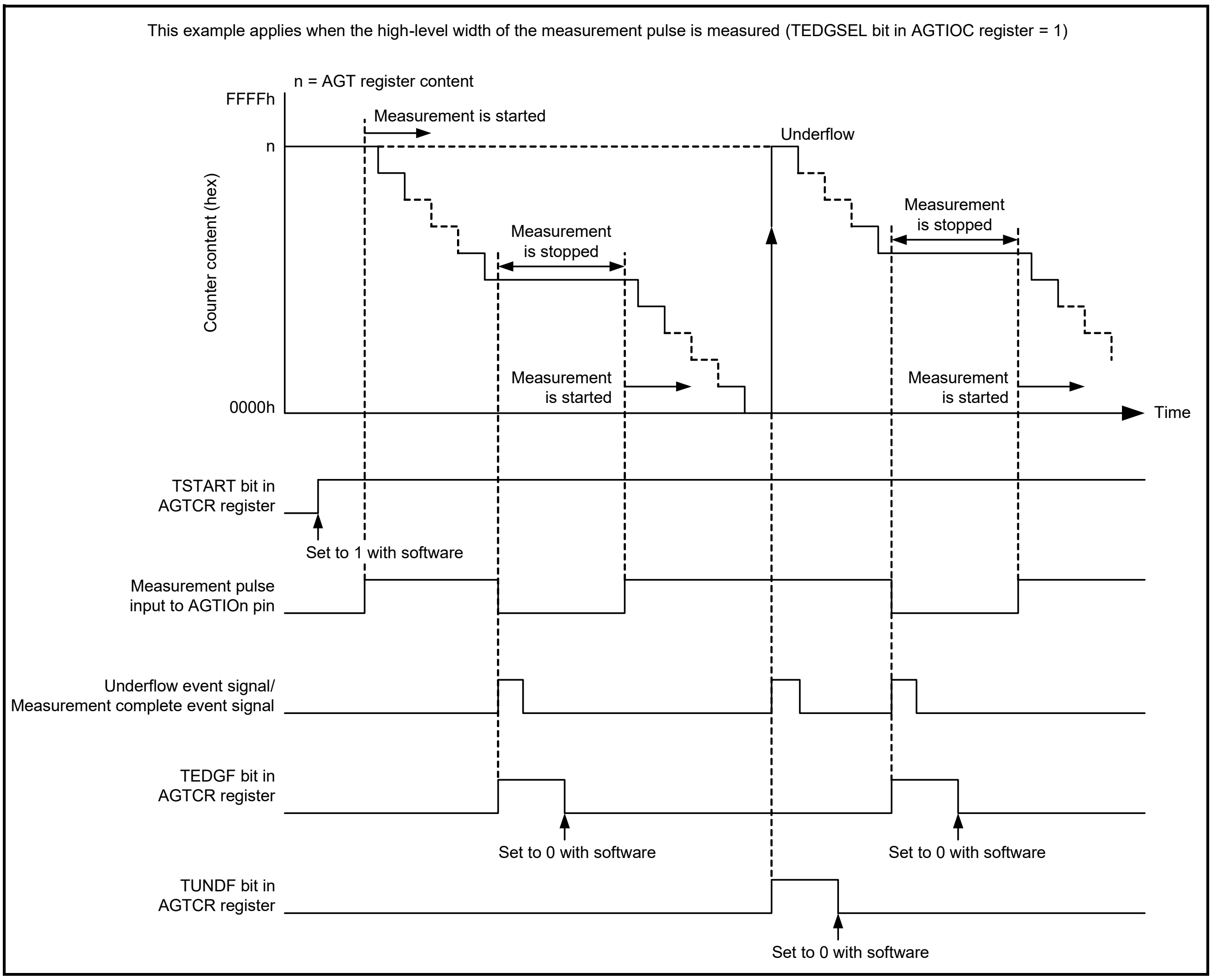

This example applies when the high-level width of the measurement pulse is measured (TEDGSEL bit in AGTIOC register = 1)
n = AGT register content
FFFFh
n
0000h
Counter content (hex)
Measurement is started
Underflow
Measurement is stopped
Measurement is stopped
Measurement is started
Measurement is started
Time
TSTART bit in AGTCR register
Set to 1 with software
Measurement pulse input to AGTIOn pin
Underflow event signal/ Measurement complete event signal
TEDGF bit in AGTCR register
Set to 0 with software
Set to 0 with software
TUNDF bit in AGTCR register
Set to 0 with software

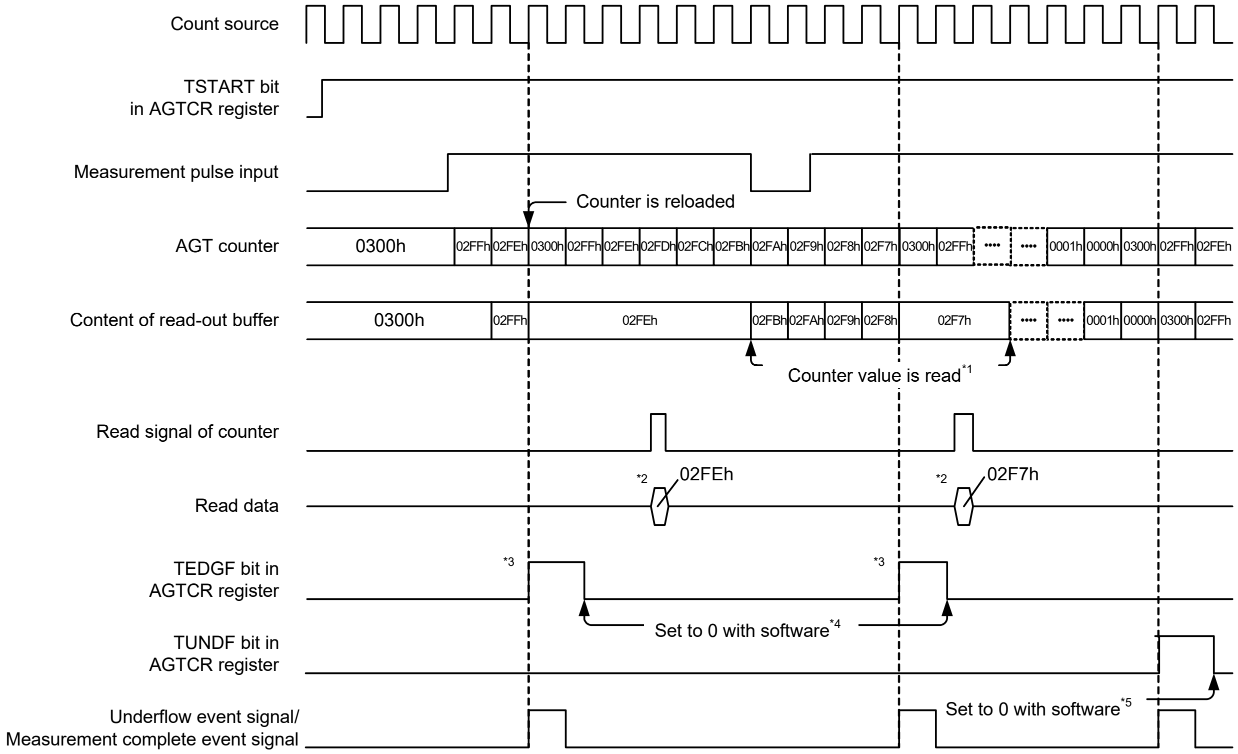
Figure 24.9 Operation example in pulse width measurement mode

24.3.7 Pulse Period Measurement Mode

In pulse period measurement mode, the pulse period of an external signal input to the AGTIOn pin is measured. The counter is decremented by the count source selected with TCK[2:0] bits in the AGTMR1 register. When a pulse with the period specified by the TEDGSEL bit in the AGTIOC register is input to the AGTIOn pin, the count value is transferred to the read-out buffer on the rising edge of the count source. The value in the reload register is loaded to the counter at the next rising edge. Simultaneously, the TEDGF 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.


Count source
TSTART bit in AGTCR register
Measurement pulse input
Counter is reloaded
AGT counter
0300h
02FFh
02FEh
0300h
02FFh
02FEh
02FDh
02FCh
02FBh
02FAh
02F9h
02F8h
02F7h
0300h
02FFh
....
....
0001h
0000h
0300h
02FFh
02FEh
Content of read-out buffer
0300h
02FFh
02FEh
02FBh
02FAh
02F9h
02F8h
02F7h
....
....
0001h
0000h
0300h
02FFh
Counter value is read*1
Read signal of counter
Read data
*2
02FEh
*2
02F7h
TEDGF bit in AGTCR register
*3
*3
Set to 0 with software*4
TUNDF bit in AGTCR register
Set to 0 with software*5
Underflow event signal/ Measurement complete event signal

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 1. 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 2. When the AGT register is read in pulse period measurement mode, the content of the read-out buffer is read.

Note 3. When the active edge of the measurement pulse 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 4. To set to 0 with software, write 0 to the TEDGF bit in the AGTCR register using an 8-bit memory manipulation instruction.

Note 5. 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

The compare match function detects matches (compare match) between the content of the AGTCMA or AGTCMB register and the content of the AGT register. This function is enabled when the TCMEA or TCMEB bit in the AGTCMSR register is 1 (compare match A register or compare match B register is valid). The counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register, and when the values of AGT and AGTCMA or AGTCMB match, the TCMAF/TCMBF bit in the AGTCR register is set to 1 (match), and an interrupt request is generated.

When the compare match function is enabled, the timing of the rewrite operation to the reload register and the counter differs. See section 24.3.1, Reload Register and Counter Rewrite Operation for details. In addition, the output level of the AGTOB0 pin is inverted by the match and by the underflow. The output level can be selected with the TOPOLB bit in the AGTCMSR register.

Figure 24.11 shows the operation example in compare match mode.

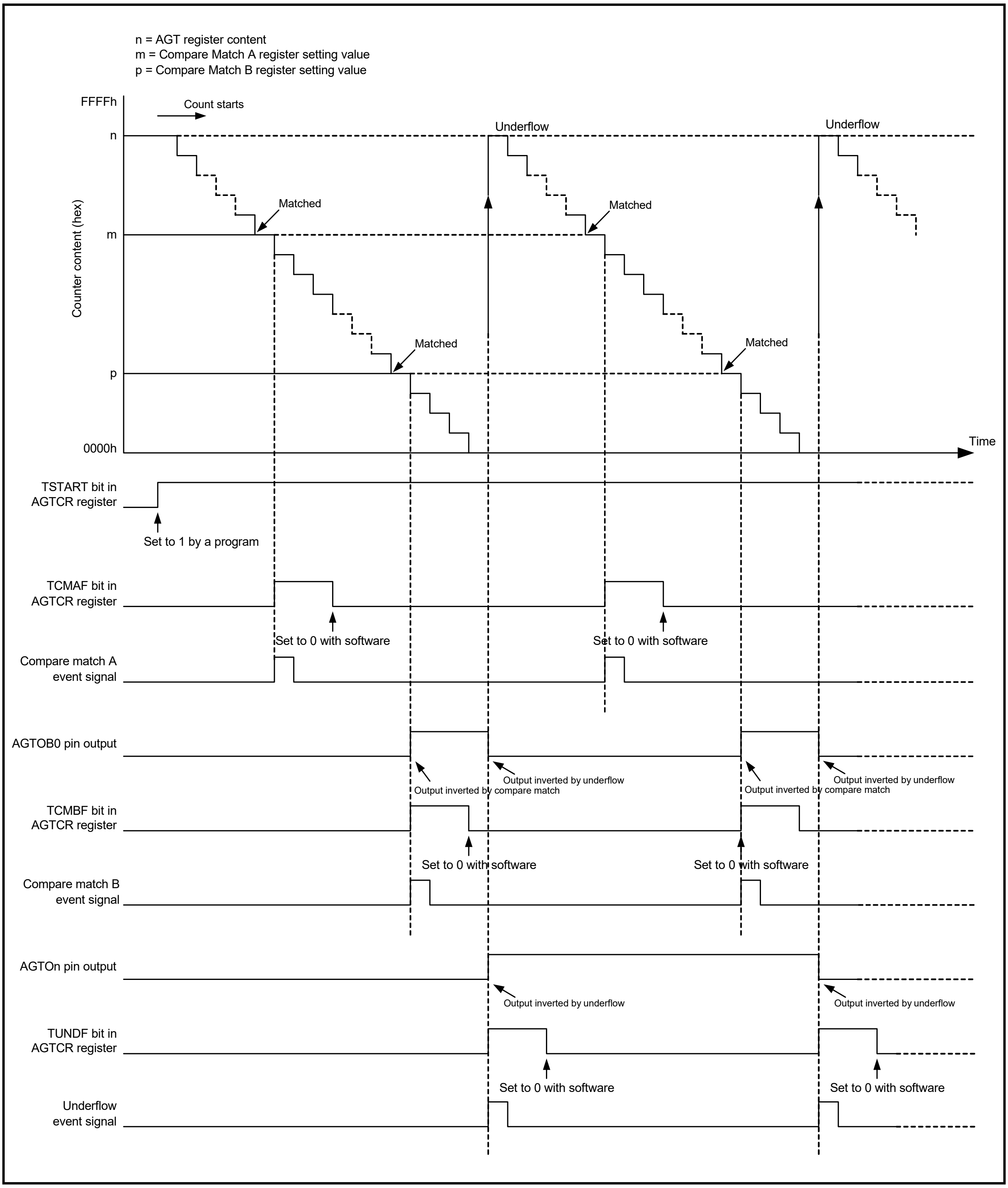

n = AGT register content
m = Compare Match A register setting value
p = Compare Match B register setting value
FFFFh
Count starts
n
m
p
0000h
Counter content (hex)
Underflow
Underflow
Matched
Matched
Matched
Matched
Time
TSTART bit in AGTCR register
Set to 1 by a program
TCMAF bit in AGTCR register
Set to 0 with software
Set to 0 with software
Compare match A event signal
AGTOB0 pin output
Output inverted by compare match
Output inverted by underflow
Output inverted by compare match
Output inverted by underflow
TCMBF bit in AGTCR register
Set to 0 with software
Set to 0 with software
Compare match B event signal
AGTOn pin output
Output inverted by underflow
Output inverted by underflow
TUNDF bit in AGTCR register
Set to 0 with software
Set to 0 with software
Underflow event signal

Figure 24.11 Operation example in compare match mode (TOPOLB = 0)

24.3.9 Output Settings for each Mode

Table 24.5 to Table 24.7 list the states of pins AGTOn, AGTIOn, and AGTOB0 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 AGTOB0 pin setting

	AGTCMSR register		
Operating mode	**TOEB bit**	**TOPOLB bit**	**AGTOB0 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			

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 24.8 and Table 24.9 show the settings that can be used in Software Standby mode.

Table 24.8 Usable setting in Software Standby mode (AGT0)

Operating mode	AGTMR1.TCK[2:0]	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 24.9 Usable setting in Software Standby mode (AGT1)

Operating mode	AGTMR1.TCK[2:0]	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 24.8.

24.3.11 Interrupt Sources

The AGT has three interrupt sources for channels n (n = 0, 1) as listed in Table 24.10.

Table 24.10 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 PCLKB cycles. 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 PCLKB cycles. 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 the TSTART and TCSTF bits 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 1 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 the 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, PCLKB/8, or PCLKB/2 as the Count Source

When a reset is generated, the operation of the 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 bit in the MSTPCRD register must be set to 1 except when accessing the AGT1 registers. The MSTPD3 bit in the 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 Switching Source Clock

When switching a clock source by changing SCKCR.CKSEL[2:0], the clock output from the selector stops for 4 cycles of the switched clock. Therefore, when using the AGTIOn, AGTEEn, or both input as external event input, the clock source should not be switched. If switching the clock source while using the external event input, extend the input pulse width by 4 clock cycles of the switched source clock cycles.

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 oscillator 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.

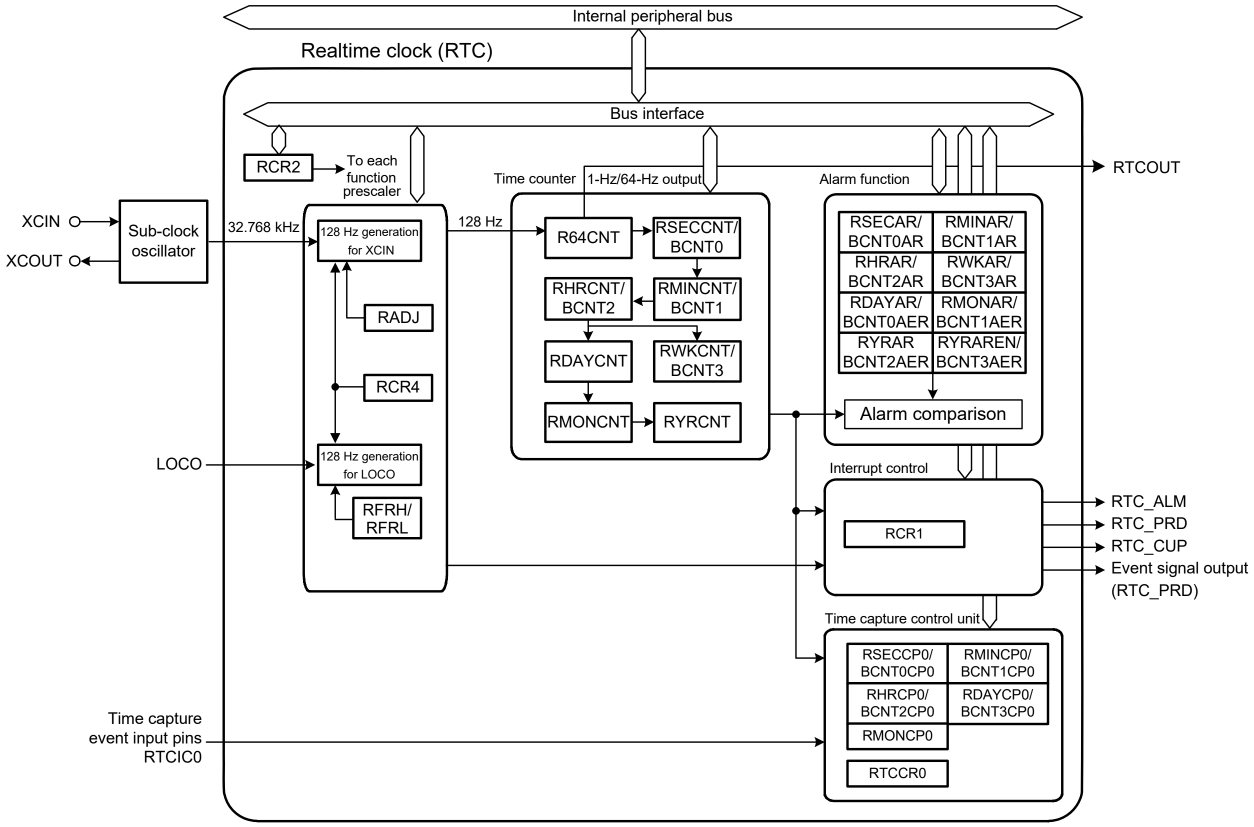
Note: Regardless of the use of VBATT function, set the VBTCR1.BPWSWSTP bit to 1 before accessing the RTC registers after cold start. For details, see Figure 12.2, Setting flow of the VBTCR1.BPWSWSTP bit, in section 12, Battery Backup Function.

Table 25.1 lists the RTC specifications, Figure 25.1 shows a block diagram, and Table 25.2 lists the I/O pins.

Table 25.1 RTC specifications

Parameter	Description
Count mode	Calendar count mode/binary count mode
Count source*1	Sub-clock oscillator (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 • Return 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. The frequency of the peripheral module clock (PCLKB) must be ≥ the frequency of the count source clock.


Internal peripheral bus
Realtime clock (RTC)
Bus interface
RCR2
To each function prescaler
Time counter
1-Hz/64-Hz output
Alarm function
RTCOUT
XCIN
XCOUT
Sub-clock oscillator
32.768 kHz
128 Hz generation for XCIN
128 Hz
R64CNT
RSECCNT/ BCNT0
RHRCNT/ BCNT2
RMINCNT/ BCNT1
RDAYCNT
RWKCNT/ BCNT3
RMONCNT
RYRCNT
RADJ
RCR4
LOCO
128 Hz generation for LOCO
RFRH/ RFRL
RSECAR/ BCNT0AR
RMINAR/ BCNT1AR
RHRAR/ BCNT2AR
RWKAR/ BCNT3AR
RDAYAR/ BCNT0AER
RMONAR/ BCNT1AER
RYRAR BCNT2AER
RYRAREN/ BCNT3AER
Alarm comparison
Interrupt control
RCR1
RTC_ALM
RTC_PRD
RTC_CUP
Event signal output (RTC_PRD)
Time capture control unit
RSECCP0/ BCNT0CP0
RMINCP0/ BCNT1CP0
RHRCP0/ BCNT2CP0
RDAYCP0/ BCNT3CP0
RMONCP0
RTCCR0
Time capture event input pins RTCIC0

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
RTCCR0: Time Capture Control Register 0
RSECCPn/BCNT0CP0: Second Capture Register n/BCNT0 Capture Register 0
RMINCPn/BCNT1CP0: Minute Capture Register n/BCNT1 Capture Register 0
RHRCPn/BCNT2CP0: Hour Capture Register n/BCNT2 Capture Register 0
RDAYCPn/BCNT3CP0: Date Capture Register n/BCNT3 Capture Register 0
RMONCP0: Month Capture Register 0

Figure 25.1 RTC block diagram

Table 25.2 RTC I/O pins

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

25.2 Register Descriptions

Write or read from the RTC registers as described in section 25.6.5, Notes on 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 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 MCU to enter Software Standby mode immediately after setting any of these registers. For details, see section 25.6.4, Transitions to Low Power Modes after Setting Registers.

25.2.1 64-Hz Counter (R64CNT)

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	Indicates 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

Bit	Symbol	Bit name	Description	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

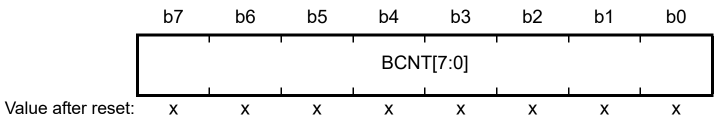
The RSECCNT counter sets and counts the BCD-coded second value. It counts the carries generated once per second in the 64-Hz counter.

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



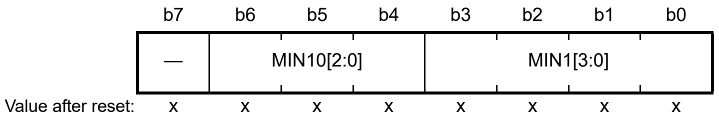
x: Undefined

BCNT0 is a read/write 32-bit binary counter b7 to b0 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.

25.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1)

(1) In calendar count mode:

Address(es): RTC.RMINCNT 4004 4004h



x: Undefined

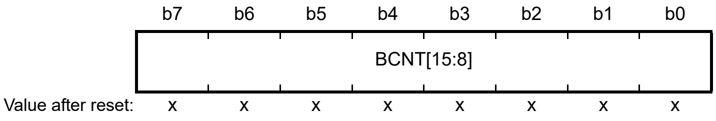
Bit	Symbol	Bit name	Description	R/W
b3 to b0	MIN1[3:0]	1-Minute Count	Counts from 0 to 9 every minute. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	MIN10[2:0]	10-Minute Count	Counts from 0 to 5 for 60-minute counting	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set 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



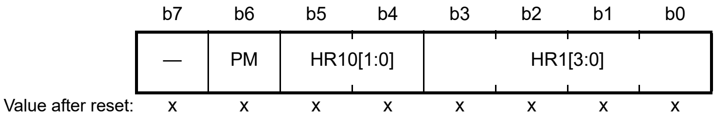
x: Undefined

BCNT1 is a read/write 32-bit binary counter b15 to b8 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.

25.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2)

(1) In calendar count mode:

Address(es): RTC.RHRCNT 4004 4006h



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: 0: a.m. 1: p.m.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

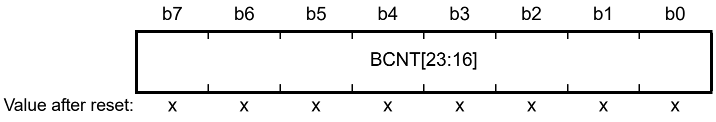
The RHRCNT counter sets and counts the BCD-coded hour value. It counts carries generated once every hour in the minute counter. The specifiable time differs based on the setting in the hours mode bit (RCR2.HR24):

- 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



x: Undefined

The BCNT2 counter is a read/write 32-bit binary counter b23 to b16 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.

25.2.5 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

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	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	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:

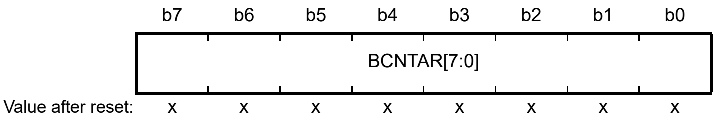
- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR

- 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 cleared to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT0AR 4004 4010h



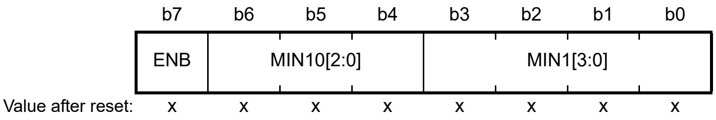
x: Undefined

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



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: 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 cleared 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 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 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 to set 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 to set 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 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 settings: 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)

The AIE bit enables or disables alarm interrupt requests.

CIE bit (Carry Interrupt Enable)

The CIE bit enables or 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)

The PIE bit enables or disables a periodic interrupt.

RTCOS bit (RTCOUT Output Select)

The RTCOS 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 the I/O ports, see section 20.5.1, Procedure for Specifying the Pin Functions.

PES[3:0] bits (Periodic Interrupt Select)

The PES[3:0] 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: Invalid (writing 0 has no effect) 1: The prescaler and the target registers for RTC software reset *1 are initialized. • In reading: 0: Normal time operation in progress, or an RTC software reset has completed 1: RTC software reset in progress.	R/W
b2	ADJ30	30-Second Adjustment	• In writing: 0: Invalid (writing 0 has no effect) 1: 30-second adjustment is executed. • In reading: 0: Normal time operation in progress, or 30-second adjustment has completed 1: 30-second adjustment in progress.	R/W
b3	RTCOE	RTCOUT Output Enable	0: RTCOUT output disabled 1: RTCOUT output enabled.	R/W
b4	AADJE	Automatic Adjustment Enable*3	0: Automatic adjustment is disabled 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select*3	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: RTC operates in 12-hour mode 1: 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, RTCCR0, RSECCP0/BCNT0CP0, RMINCP0/
Note 2. BCNT1CP0, RHRCP0/BCNT2CP0, RDAYCP0/BCNT3CP0, RMONCP0, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP.
Note 3. 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)

The START bit stops or restarts the prescaler or time counter operation. This 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)

The RESET 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)

The ADJ30 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 completes. 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 cleared to 0 by an RTC software reset.

RTCOE bit (RTCOUT Output Enable)

The RTCOE 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*[3])

The AADJE 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*[3])

The AADJP 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)

The HR24 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)

The CNTMD 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: Invalid (writing 0 has no effect) 1: The prescaler and the target registers for RTC software reset*1 are initialized. • In reading 0: Normal time operation in progress, or an RTC software reset has completed 1: RTC software reset in progress.	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 is disabled 1: RTCOUT output is enabled.	R/W
b4	AADJE	Automatic Adjustment Enable*3	0: Automatic adjustment is disabled 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select*3	0: Add or subtract the RADJ.ADJ[5:0] bits from the prescaler count value every 32 seconds 1: Add or subtract 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: 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, RTCCR0, RSECCP0/BCNT0CP0, RMINCP0/
Note 2. BCNT1CP0, RHRCP0/BCNT2CP0, RDAYCP0/BCNT3CP0, RMONCP0, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP
Note 3. When LOCO is selected, the setting of this bit is disabled.

START bit (Start)

The START bit stops or restarts the prescaler or counter (clock) operation. This 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)

The RESET 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 complete, 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)

The RTCOE 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 in addition to setting this bit.

AADJE bit (Automatic Adjustment Enable)

The AADJE 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 cleared to 0 by an RTC software reset.

AADJP bit (Automatic Adjustment Period Select)

The AADJP 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 cleared to 0 by an RTC software reset.

CNTMD bit (Count Mode Select)

The CNTMD 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.

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 in binary count mode.

When the RCKSEL bit is set to 0, the time is counted with the sub-clock oscillator. When the bit is set to 1, the time is counted with LOCO.

RCKSEL bit (Count Source Select)

The RCKSEL bit selects the count source from the sub-clock oscillator and LOCO.

The count source must be selected only once before specifying 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 ≥ 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 must 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 additional processing. This register is cleared to 00h by an RTC software reset. The setting of this register is enabled only when the sub-clock oscillator is selected. When LOCO is selected, adjustment is not performed.

ADJ[5:0] bits (Adjustment Value)

The ADJ[5:0] bits specify the adjustment value (the number of sub-clock cycles) from the prescaler.

PMADJ[1:0] bits (Plus–Minus)

The PMADJ[1:0] 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 0 (RTCCR0)

Address(es): RTC.RTCCR0 4004 4040h

	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 RTCIC0 pin is disabled as the time capture event input 1: The RTCIC0 pin is enabled as the time capture event input.	R/W

Note 1. Indicates that an event is detected. Writing 1 to this bit has no effect. Writing 0 sets this bit to 0.

The RTCCR0 register is used both in calendar count mode and in binary count mode. RTCCR0 controls the RTCIC0 pin, respectively.

RTCCR0 is updated in synchronization with the count source. When RTCCR0 is modified, check that all the bits except for the TCST bit are updated before continuing with more processing. This register is cleared to 00h by an RTC software reset. When RTCIC0 is used as the time capture pin, VBTICTLR.VCH0IEN must be set to 1. For more information, see section 12, Battery Backup Function.

TCCT[1:0] bits (Time Capture Control)

The TCCT[1:0] bits control the edge detection of the time capture event input pin, RTCIC0. The detection edge is selectable. The TCCT[1:0] bits must be set while the VBTICTLR.VCH0IEN bit is 1.

TCST bit (Time Capture Status)

The TCST bit indicates that an event of the time capture event input pin, RTCIC0 was detected. When the TCST bit is 0, no event is detected. When the TCST bit is 1, this bit indicates that an event was detected on the associated pin and the capture register is valid. When multiple events are detected, the capture time for the first event is retained.

If an event is detected while the count operation is stopped (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 additional processing.

TCNF[1:0] bits (Time Capture Noise Filter Control)

The TCNF[1:0] bits control the noise filter of the time capture event input pin (RTCIC0).

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.VCH0IEN bit is 1.

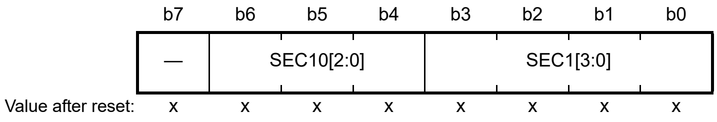
TCEN bit (Time Capture Event Input Pin Enable)

The TCEN bit enables or disables the time capture event input pin (RTCIC0). When the functions of the time capture event input pin (RTCIC0) 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 0 (RSECCP0) /BCNT0 Capture Register 0 (BCNT0CP0)

(1) In calendar count mode:

Address(es): RTC.RSECCP0 4004 4052h


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

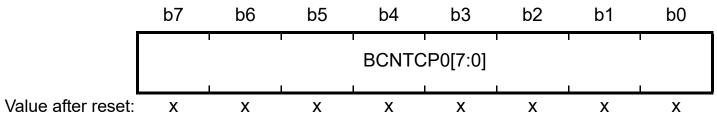
Bit	Symbol	Bit name	Description	R/W
b3 to b0	SEC1[3:0]	1-Second Capture	Capture value for the ones place of seconds	R
b6 to b4	SEC10[2:0]	10-Second Capture	Capture value for the tens place of seconds	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset	R

RSECCP0 is a read-only register that captures the RSECCNT value when a time capture event is detected.

The event detection time detected by the RTCIC0 pin is stored in the RSECCP0 register. This register is cleared to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCR0.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT0CP0 4004 4052h


b7
b6
b5
b4
b3
b2
b1
b0
BCNTCP0[7:0]
Value after reset:
x x x x x x x x

x: Undefined

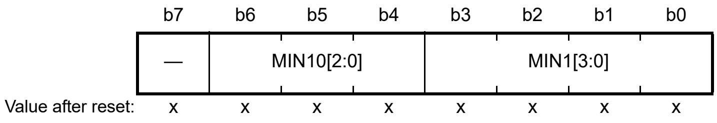
BCNT0CPy is a read-only register that captures the BCNT0 value when a time capture event is detected. The event detection times detected by the RTCIC0 pin is stored in the BCNT0CP0 register, respectively.

This register is cleared to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCR0.TCCT[1:0] bits.

25.2.24 Minute Capture Register 0 (RMINCP0)/BCNT1 Capture Register 0 (BCNT1CP0)

(1) In calendar count mode:

Address(es): RTC.RMINCP0 4004 4054h


b7
b6
b5
b4
b3
b2
b1
b0
—
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 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

Bit	Symbol	Bit name	Description	R/W
b7	—	Reserved	This bit is read as 0 after an RTC software reset	R

RMINCP0 is a read-only register that captures the RMINCNT value when a time capture event is detected.

The event detection time detected by the RTCIC0 pin is stored in the RMINCP0 register.

This register is cleared to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCR0.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT1CP0 4004 4054h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTCP0[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT1CP0 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 pin is stored in the BCNT1CP0 register, respectively.

This register is cleared to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCR0.TCCT[1:0] bits.

25.2.25 Hour Capture Register 0 (RHRCP0) /BCNT2 Capture Register 0 (BCNT2CP0)

(1) In calendar count mode:

Address(es): RTC.RHRCP0 4004 4056h

	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: AM. 1: PM.	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset	R

RHRCP0 is a read-only register that captures the RHRCNT value when a time capture event is detected.

The event detection time detected by the RTCIC0 pin is stored in the RHRCP0 register. The PM bit is only enabled when the RCR2.HR24 bit is 0 (in 12-hour mode).

This register is cleared to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCR0.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT2CP0 4004 4056h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTCP0[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT2CP0 is a read-only register that captures the BCNT2 value when a time capture event is detected.

The event detection time detected by the RTCIC0 pin is stored in the BCNT2CP0 register.

This register is cleared to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCR0.TCCT[1:0] bits.

25.2.26 Date Capture Register 0 (RDAYCP0) /BCNT3 Capture Register 0 (BCNT3CP0)

(1) In calendar count mode:

Address(es): RTC.RDAYCP0 4004 405Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	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 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

RDAYCP0 is a read-only register that captures the RDAYCNT value when a time capture event is detected.

The event detection time detected by the RTCIC0 pin is stored in the RDAYCP0 register.

This register is cleared to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCR0.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT3CP0 4004 405Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTCP0[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT3CP0 is a read-only register that captures the BCNT3 value when a time capture event is detected.

The event detection time detected by the RTCTC0 pin is stored in the BCNT3CP0 register.

This register is cleared to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCR0.TCCT[1:0] bits.

25.2.27 Month Capture Register 0 (RMONCP0)

(1) In calendar count mode:

Address(es): RTC.RMONCP0 4004 405Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	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 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

RMONCP0 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 pin is stored in the RMONCP0 register, respectively.

This register is cleared to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCR0.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, perform the initial settings for the clock setting, count mode setting, time error adjustment, time setting, alarm, interrupt, and time capture control register.

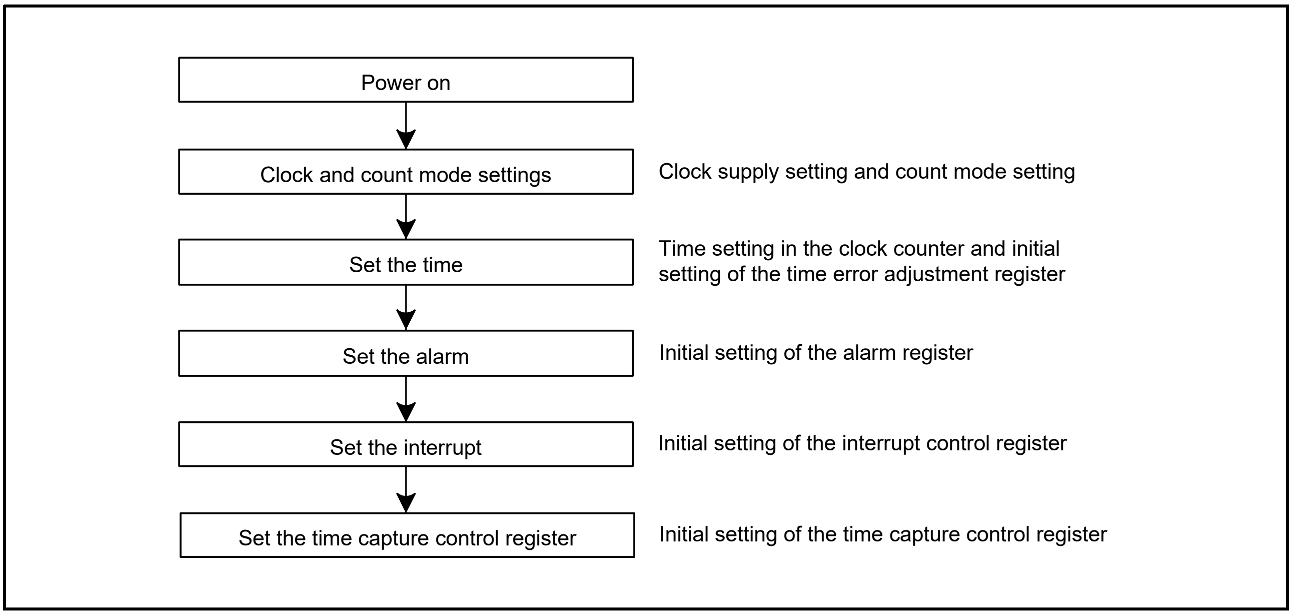
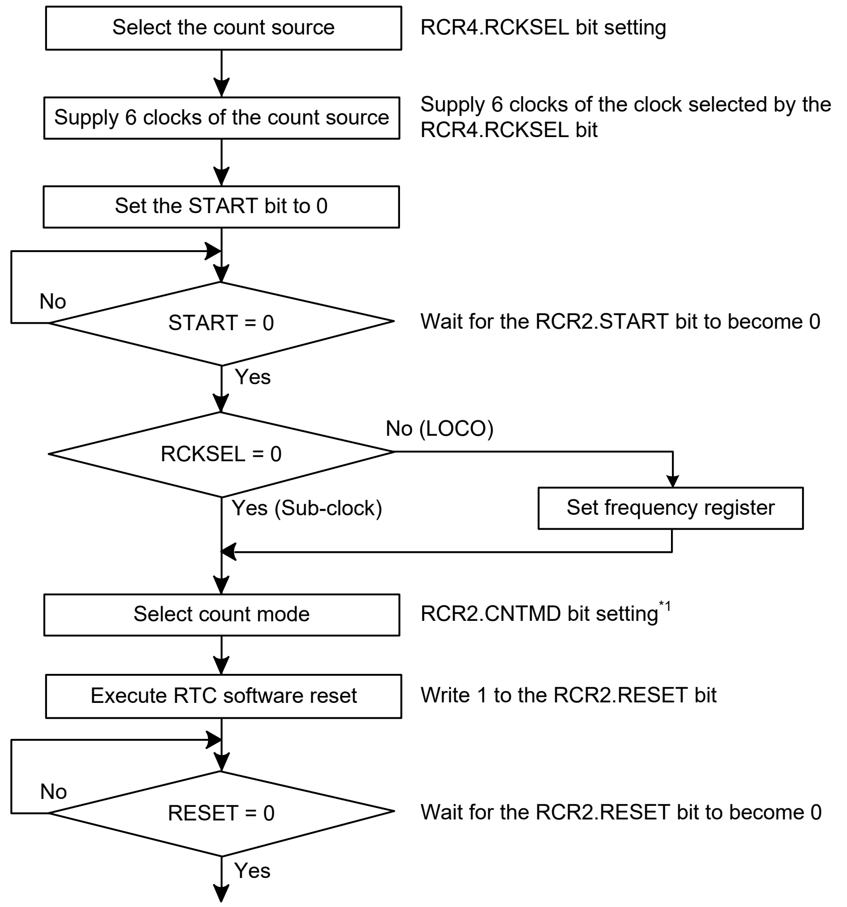

Power on
Clock and count mode settings
Clock supply setting and count mode setting
Set the time
Time setting in the clock counter and initial setting of the time error adjustment register
Set the alarm
Initial setting of the alarm register
Set the interrupt
Initial setting of the interrupt control register
Set the time capture control register
Initial setting of the time capture control register

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.

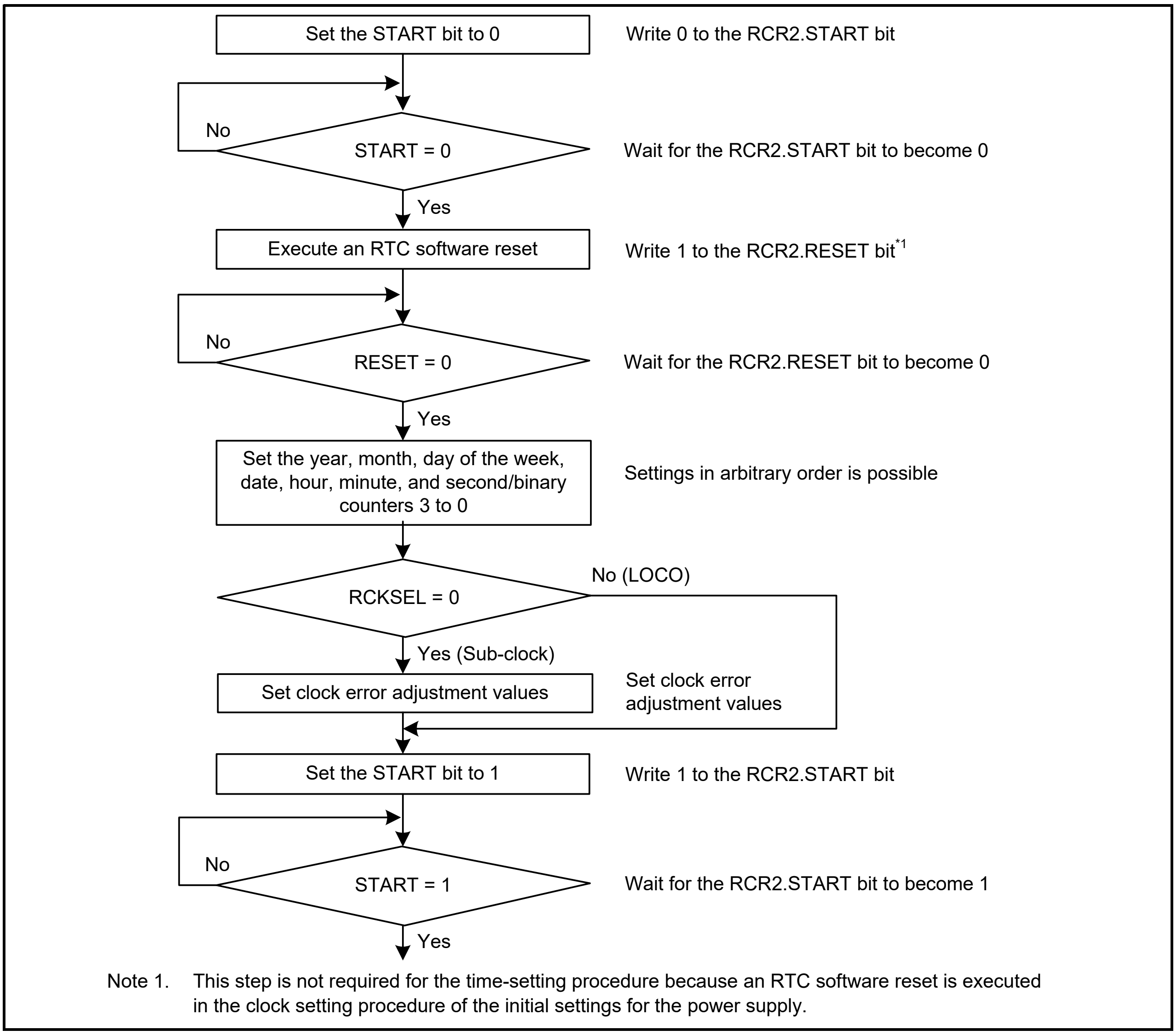

Select the count source
RCR4.RCKSEL bit setting
Supply 6 clocks of the count source
Supply 6 clocks of the clock selected by the RCR4.RCKSEL bit
Set the START bit to 0
No
START = 0
Wait for the RCR2.START bit to become 0
Yes
RCKSEL = 0
No (LOCO)
Yes (Sub-clock)
Set frequency register
Select count mode
RCR2.CNTMD bit setting*1
Execute RTC software reset
Write 1 to the RCR2.RESET bit
No
RESET = 0
Wait for the RCR2.RESET bit to become 0
Yes

Note 1. This step is not required if the count mode is set concurrently by setting the START bit to 0. A value associated with the count mode setting must be written to the RCR2.CNTMD bit.

Figure 25.3 Clock and count mode setting procedure

25.3.3 Setting the Time

Figure 25.4 shows how to set the time.


Set the START bit to 0
Write 0 to the RCR2.START bit
No
START = 0
Wait for the RCR2.START bit to become 0
Yes
Execute an RTC software reset
Write 1 to the RCR2.RESET bit*1
No
RESET = 0
Wait for the RCR2.RESET bit to become 0
Yes
Set the year, month, day of the week, date, hour, minute, and second/binary counters 3 to 0
Settings in arbitrary order is possible
RCKSEL = 0
No (LOCO)
Yes (Sub-clock)
Set clock error adjustment values
Set clock error adjustment values
Set the START bit to 1
Write 1 to the RCR2.START bit
No
START = 1
Wait for the RCR2.START bit to become 1
Yes

Note 1. This step is not required 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.

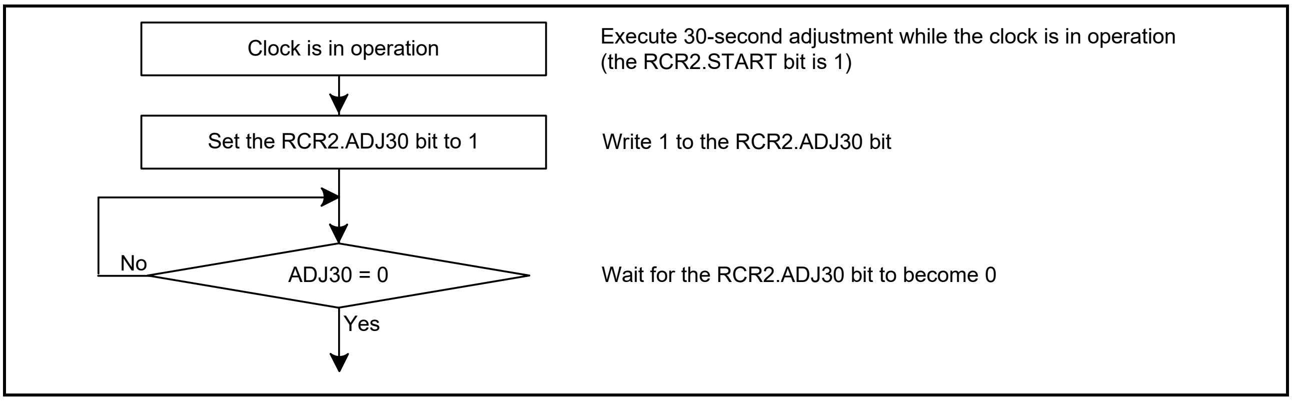
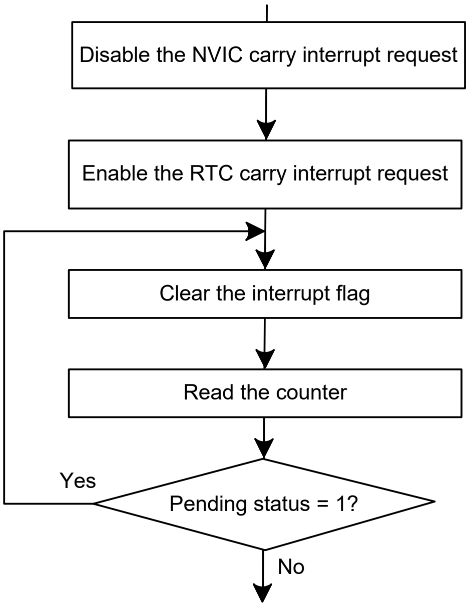

Clock is in operation
Execute 30-second adjustment while the clock is in operation (the RCR2.START bit is 1)
Set the RCR2.ADJ30 bit to 1
Write 1 to the RCR2.ADJ30 bit
No
ADJ30 = 0
Wait for the RCR2.ADJ30 bit to become 0
Yes

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.

(a) To read the time without using interrupt


Disable the NVIC carry interrupt request
Enable the RTC carry interrupt request
Clear the interrupt flag
Read the counter
Yes
Pending status = 1?
No

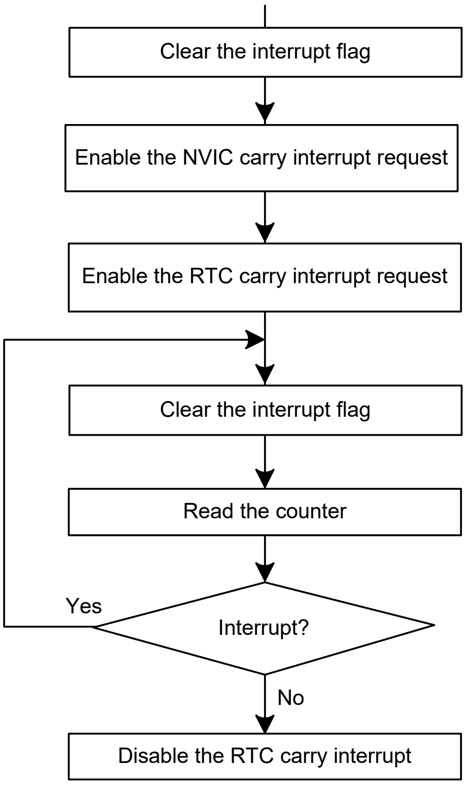
Write 1 to the Interrupt Clear-Enable Register corresponding to the RTC_CUP interrupt

Write 1 to the RCR1.CIE bit

Write 0 to the IELSRn.IR bit and write 1 to the Interrupt Clear-Pending Register corresponding to the RTC_CUP interrupt

Read the counter again when the Interrupt Set-pending Register corresponding to the RTC_CUP interrupt is 1

(b) To read the time using interrupts


Clear the interrupt flag
Enable the NVIC carry interrupt request
Enable the RTC carry interrupt request
Clear the interrupt flag
Read the counter
Yes
Interrupt?
No
Disable the RTC carry interrupt

Write 0 to the IELSRn.IR bit and write 1 to the Interrupt Clear-Pending Register corresponding to the RTC_CUP interrupt

Write 1 to the Interrupt Set-Enable Register corresponding to the RTC_CUP interrupt

Write 1 to the RCR1.CIE bit

Write 0 to the IELSRn.IR bit and write 1 to the Interrupt Clear-Pending Register corresponding to the RTC_CUP interrupt

Write 0 to the RCR1.CIE bit*[1]

Note 1. Disable interrupts if required.

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.

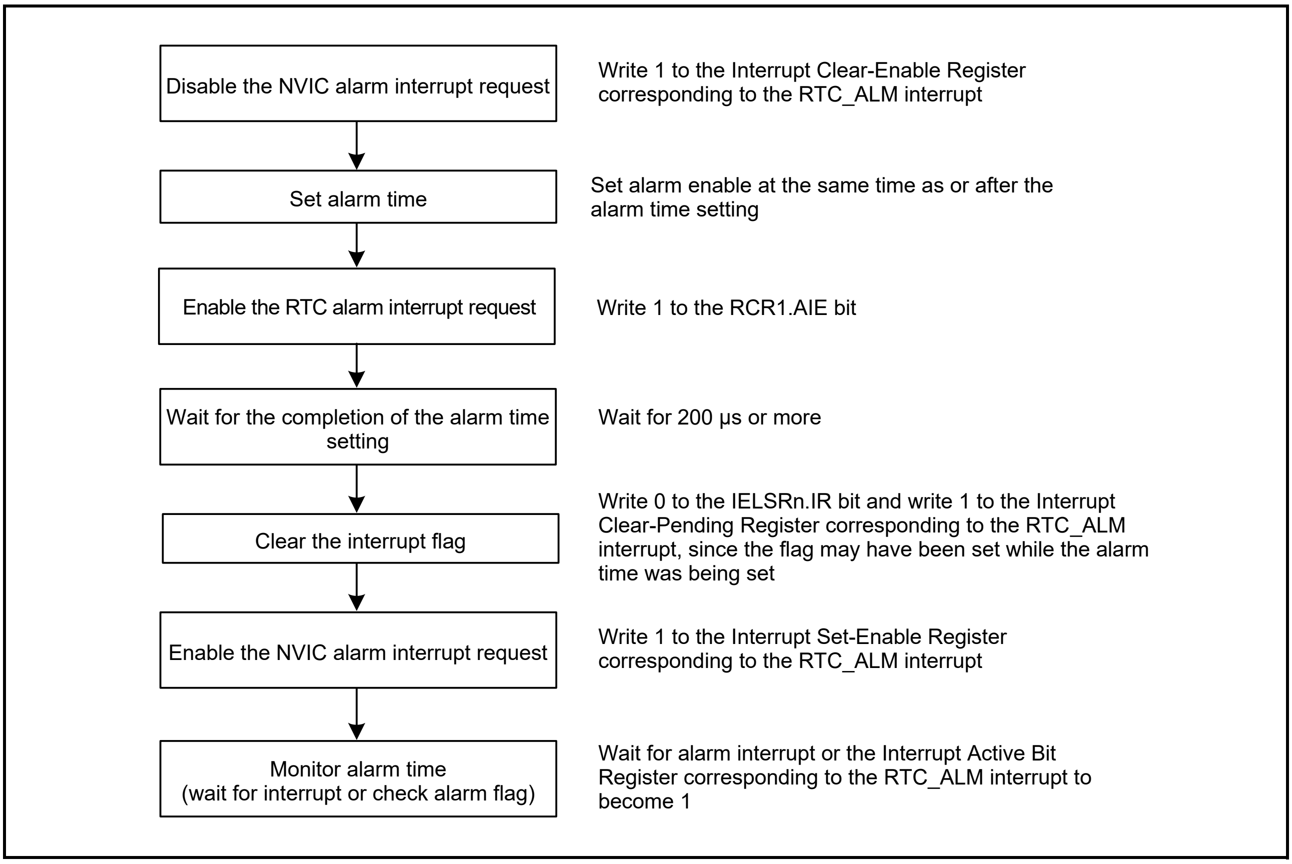

Disable the NVIC alarm interrupt request
Write 1 to the Interrupt Clear-Enable Register corresponding to the RTC_ALM interrupt
Set alarm time
Set alarm enable at the same time as or after the alarm time setting
Enable the RTC alarm interrupt request
Write 1 to the RCR1.AIE bit
Wait for the completion of the alarm time setting
Wait for 200 µs or more
Clear the interrupt flag
Write 0 to the IELSRn.IR bit and write 1 to the Interrupt Clear-Pending Register corresponding to the RTC_ALM interrupt, since the flag may have been set while the alarm time was being set
Enable the NVIC alarm interrupt request
Write 1 to the Interrupt Set-Enable Register corresponding to the RTC_ALM interrupt
Monitor alarm time
(wait for interrupt or check alarm flag)
Wait for alarm interrupt or the Interrupt Active Bit Register corresponding to the RTC_ALM interrupt to become 1

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 associated with the RTC_ALM interrupt are set to 1. Alarm detection can be confirmed by reading the Interrupt Set-Pending Register associated with 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 associated with 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 state, the MCU returns from the low power state.

25.3.7 Procedure for Disabling Alarm Interrupt

Figure 25.8 shows the procedure for disabling the enabled alarm interrupt request.

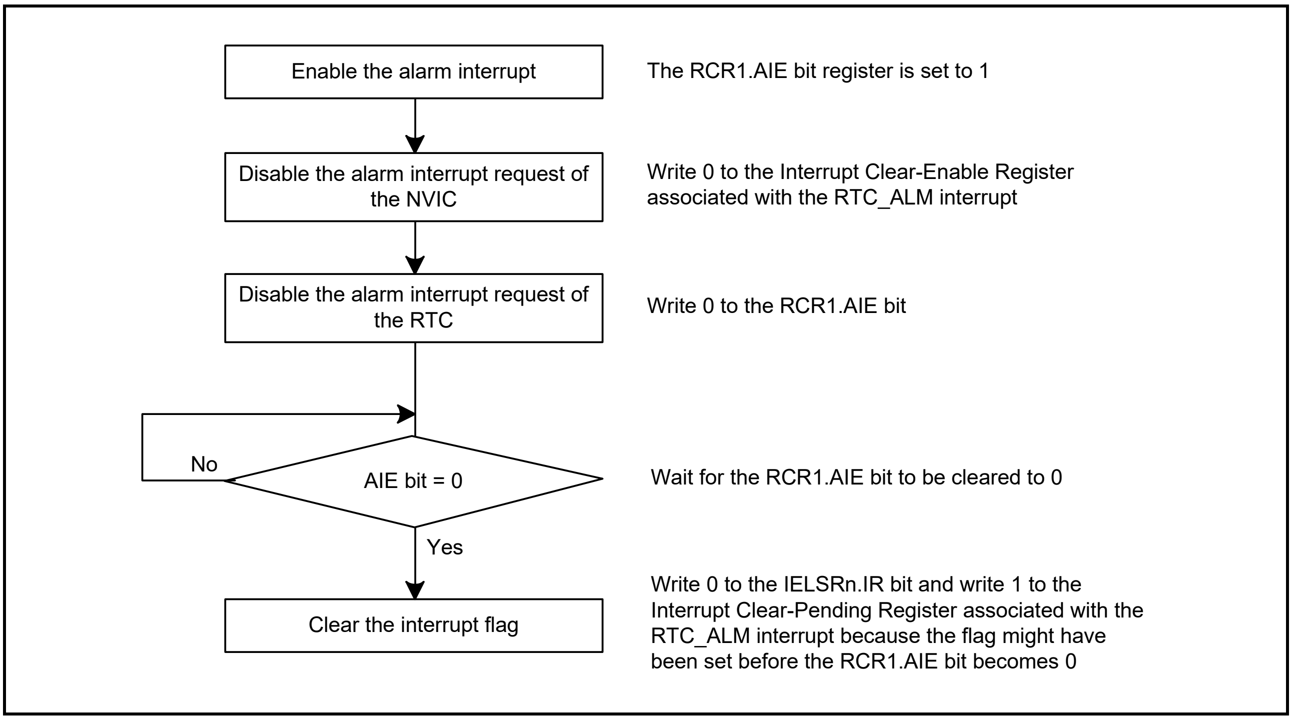

Enable the alarm interrupt
The RCR1.AIE bit register is set to 1
Disable the alarm interrupt request of the NVIC
Write 0 to the Interrupt Clear-Enable Register associated with the RTC_ALM interrupt
Disable the alarm interrupt request of the RTC
Write 0 to the RCR1.AIE bit
No
AIE bit = 0
Wait for the RCR1.AIE bit to be cleared to 0
Yes
Clear the interrupt flag
Write 0 to the IELSRn.IR bit and write 1 to the Interrupt Clear-Pending Register associated with the RTC_ALM interrupt because the flag might have been set before the RCR1.AIE bit becomes 0

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 oscillator. Because 32,768 cycles of the sub-clock oscillator constitute 1 second of operation when the sub-clock oscillator is selected, the clock runs fast if the sub-clock frequency is high and slow if the sub-clock frequency is low.

The time error adjustment functions include:

- Automatic adjustment
- Adjustment by software.

Use the RCR2.AADJP 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 oscillator running at 32.769 kHz

(a) Adjustment procedure

When the sub-clock oscillator 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 1 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 oscillator running at 32.766 kHz

(a) Adjustment procedure

When the sub-clock oscillator 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 oscillator 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 advancing 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 oscillator running at 32.769 kHz

(a) Adjustment procedure

When the sub-clock oscillator 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 1 clock cycle every second. The time on the clock is fast by 1 clock cycle per second, so adjustment can take the form of setting the clock back by 1 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.VCH0IEN should be set to 1 to enable the RTCIC0 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.

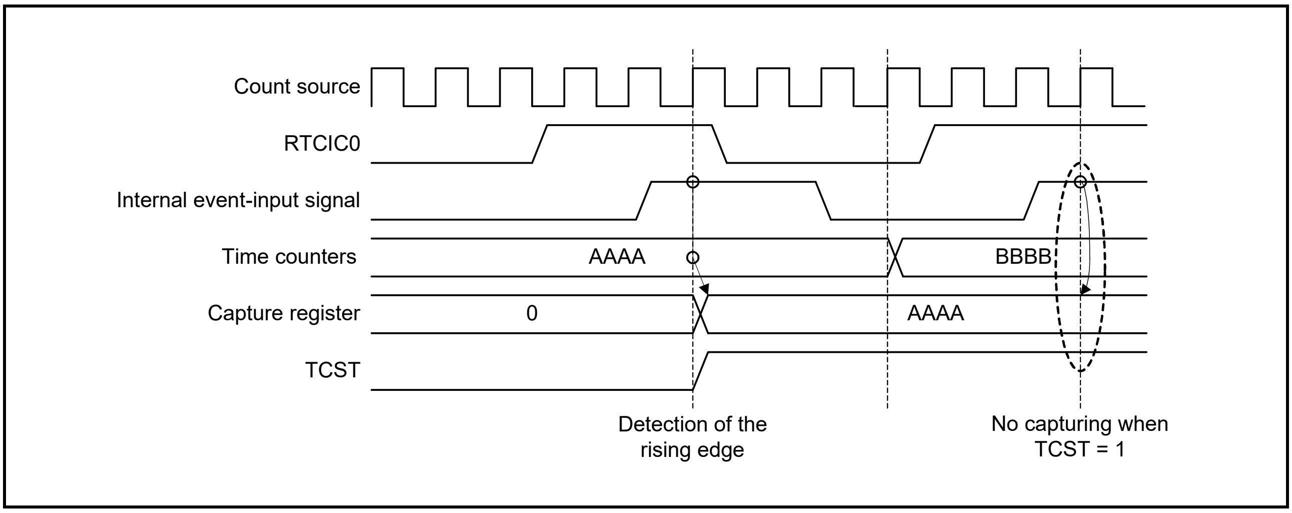

Count source
RTCIC0
Internal event-input signal
Time counters
AAAA
BBBB
Capture register
0
AAAA
TCST
Detection of the rising edge
No capturing when TCST = 1

Figure 25.9 Timing of a time capture operation with the filter off

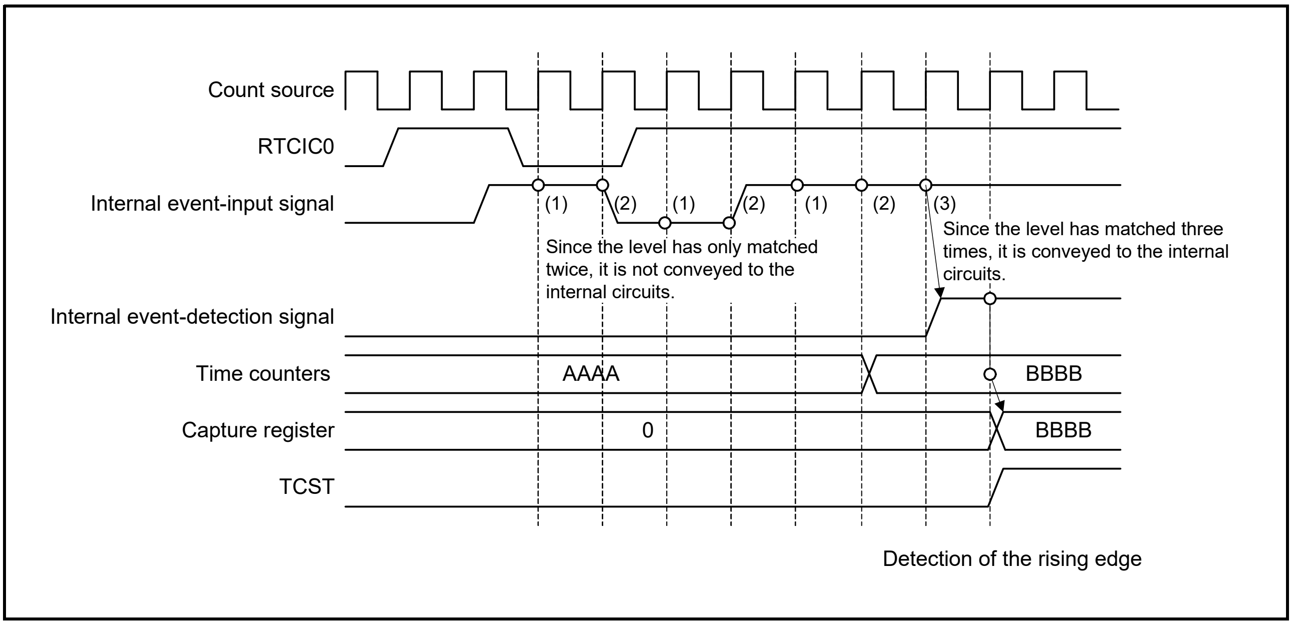

Count source
RTCIC0
Internal event-input signal
Internal event-detection signal
Time counters
Capture register
TCST
(1)
(2)
(1)
(2)
(1)
(2)
(3)
Since the level has only matched twice, it is not conveyed to the internal circuits.
Since the level has matched three times, it is conveyed to the internal circuits.
AAAA
BBBB
0
BBBB
Detection of the rising edge

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.

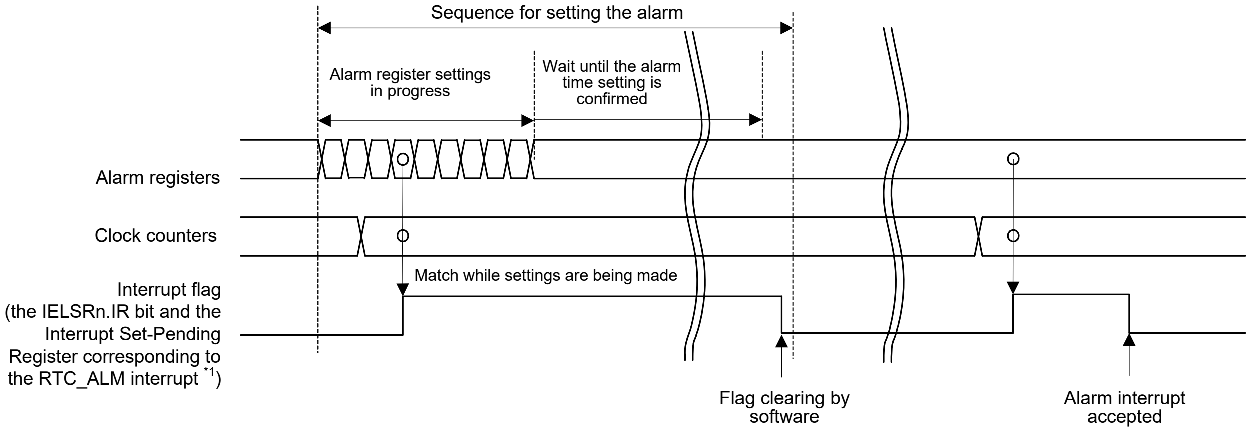
Table 25.3 RTC Interrupt sources

Name	Interrupt sources
RTC_ALM	Alarm interrupt
RTC_PRD	Periodic interrupt
RTC_CUP	Carry interrupt

(1) Alarm interrupt (RTC_ALM)

This interrupt is generated based on the comparison result between the alarm registers and RTC 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 associated with 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 mismatching of the alarm registers and clock counters, the flag is not set again until there is another match or the values of the alarm registers are modified again.


Sequence for setting the alarm
Alarm register settings in progress
Wait until the alarm time setting is confirmed
Alarm registers
Clock counters
Match while settings are being made
Interrupt flag (the IELSRn.IR bit and the Interrupt Set-Pending Register corresponding to the RTC_ALM interrupt *1)
Flag clearing by software
Alarm interrupt accepted

Note 1. See section 14., Interrupt Controller Unit (ICU) for details on the associated interrupt vector number.

Figure 25.11 Timing 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.

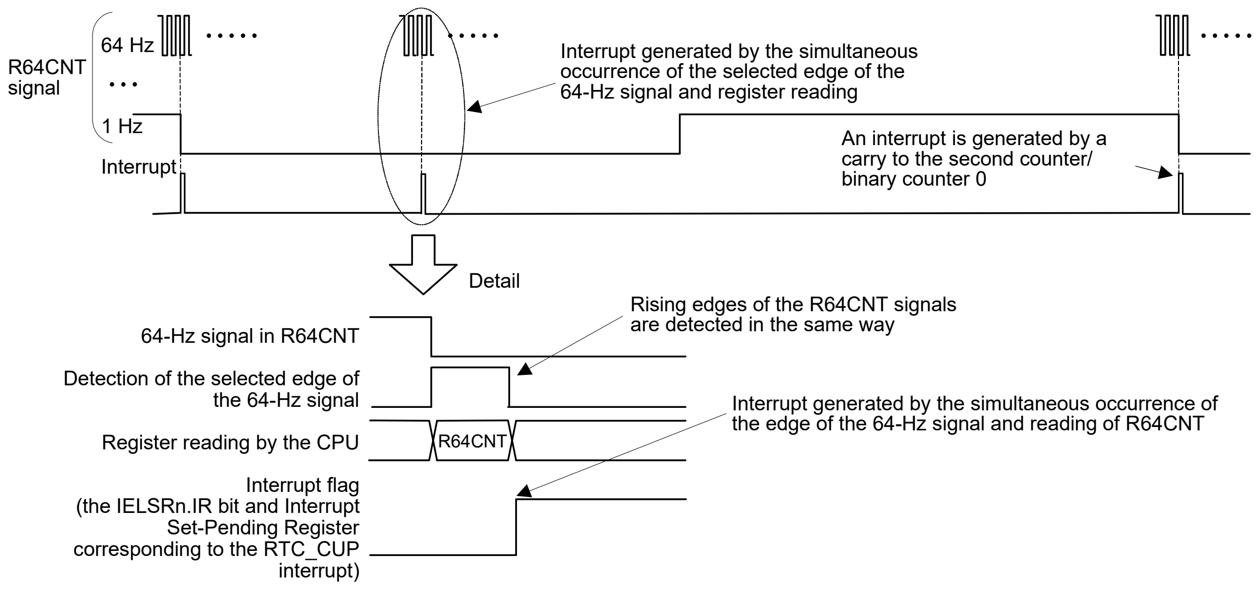

R64CNT signal
64 Hz
1 Hz
Interrupt
Interrupt generated by the simultaneous occurrence of the selected edge of the 64-Hz signal and register reading
An interrupt is generated by a carry to the second counter/ binary counter 0
Detail
64-Hz signal in R64CNT
Rising edges of the R64CNT signals are detected in the same way
Detection of the selected edge of the 64-Hz signal
Register reading by the CPU
R64CNT
Interrupt generated by the simultaneous occurrence of the edge of the 64-Hz signal and reading of R64CNT
Interrupt flag (the IELSRn.IR bit and Interrupt Set-Pending Register corresponding to the RTC_CUP interrupt)

Figure 25.12 Timing for the carry interrupt (RTC_CUP)

25.5 Event Link Output

The RTC generates periodic event output (RTC_PRD) event signals for the Event Link Controller (ELC) that 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 associated 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.

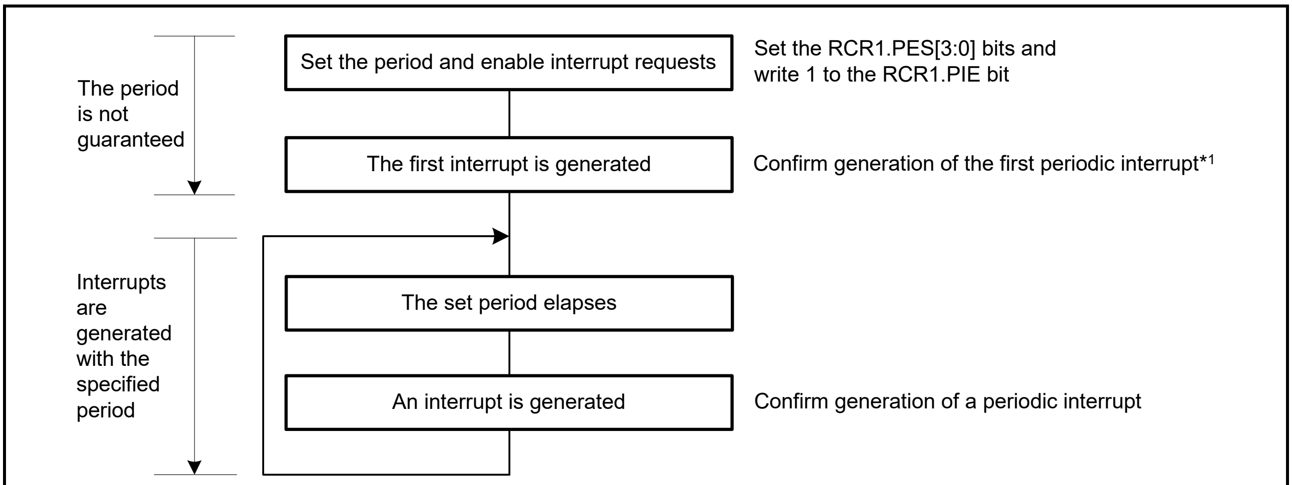
25.6.2 Use of Periodic Interrupts

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 based on the adjustment value.


The period is not guaranteed
Set the period and enable interrupt requests
Set the RCR1.PES[3:0] bits and write 1 to the RCR1.PIE bit
The first interrupt is generated
Confirm generation of the first periodic interrupt*1
Interrupts are generated with the specified period
The set period elapses
An interrupt is generated
Confirm generation of a periodic interrupt

Note 1. When a interrupt generation period is changed while the periodic interrupt is used, an interrupt may be generated at the completion of the setting. If the interrupt is generated immediately after the setting, the period is not guaranteed for two interrupts including the current interrupt.

Figure 25.13 Using the 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 based on the adjustment value.

25.6.4 Transitions to Low Power Modes after Setting Registers

A transition to a low power state (Software Standby mode or battery backup) during a write to an RTC register might corrupt the value in the register. After setting the register, confirm that the setting is in place before initiating a transition to a low power state.

25.6.5 Notes on Writing to and Reading from Registers

- When reading a counter register such as the second counter after writing 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 returning from a reset, a 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 have elapsed.

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.

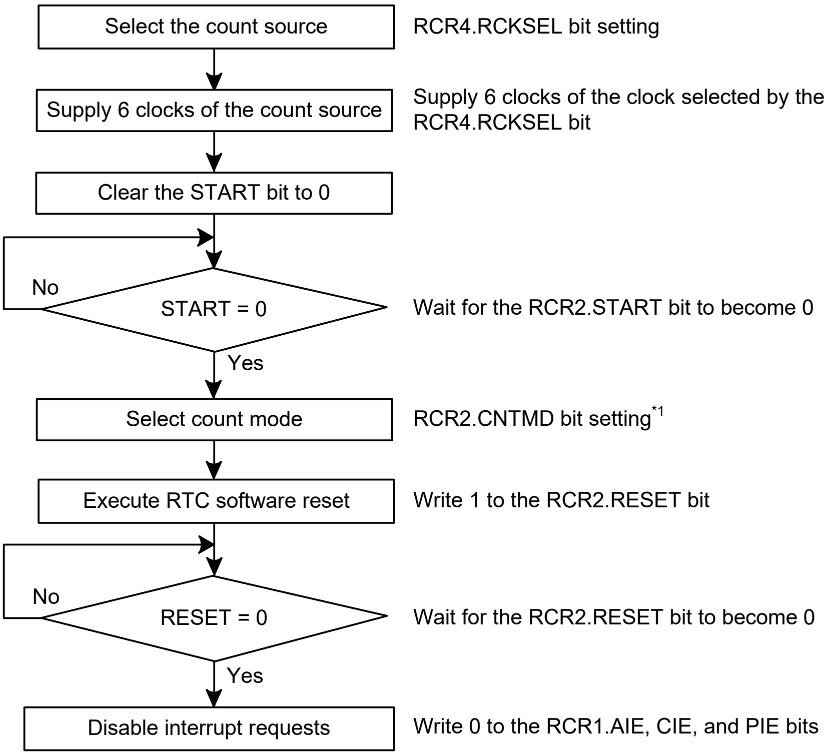
25.6.7 Initialization Procedure when the RTC is not to be Used

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 oscillator 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 oscillator. To stop the sub-clock oscillator, write 1 to the SOSCCR.SOSTP bit.

For details on the setting of the SOSCCR.SOSTP bit, see section 9, Clock Generation Circuit.


Select the count source
RCR4.RCKSEL bit setting
Supply 6 clocks of the count source
Supply 6 clocks of the clock selected by the RCR4.RCKSEL bit
Clear the START bit to 0
No
START = 0
Wait for the RCR2.START bit to become 0
Yes
Select count mode
RCR2.CNTMD bit setting*1
Execute RTC software reset
Write 1 to the RCR2.RESET bit
No
RESET = 0
Wait for the RCR2.RESET bit to become 0
Yes
Disable interrupt requests
Write 0 to the RCR1.AIE, CIE, and PIE bits

Note 1. This step is not required if the count mode is set concurrently by setting the START bit to 0.

Figure 25.14 Initialization procedure

25.6.8 When Switching Source Clock

When switching a clock source by changing SCKCR.CKSEL[2:0], the clock output from the selector stops for 4 cycles of the switched clock. If the RTC periodical interrupt or RTC periodical event output was generated at this time, the interrupt or event is invalid.

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 a 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.

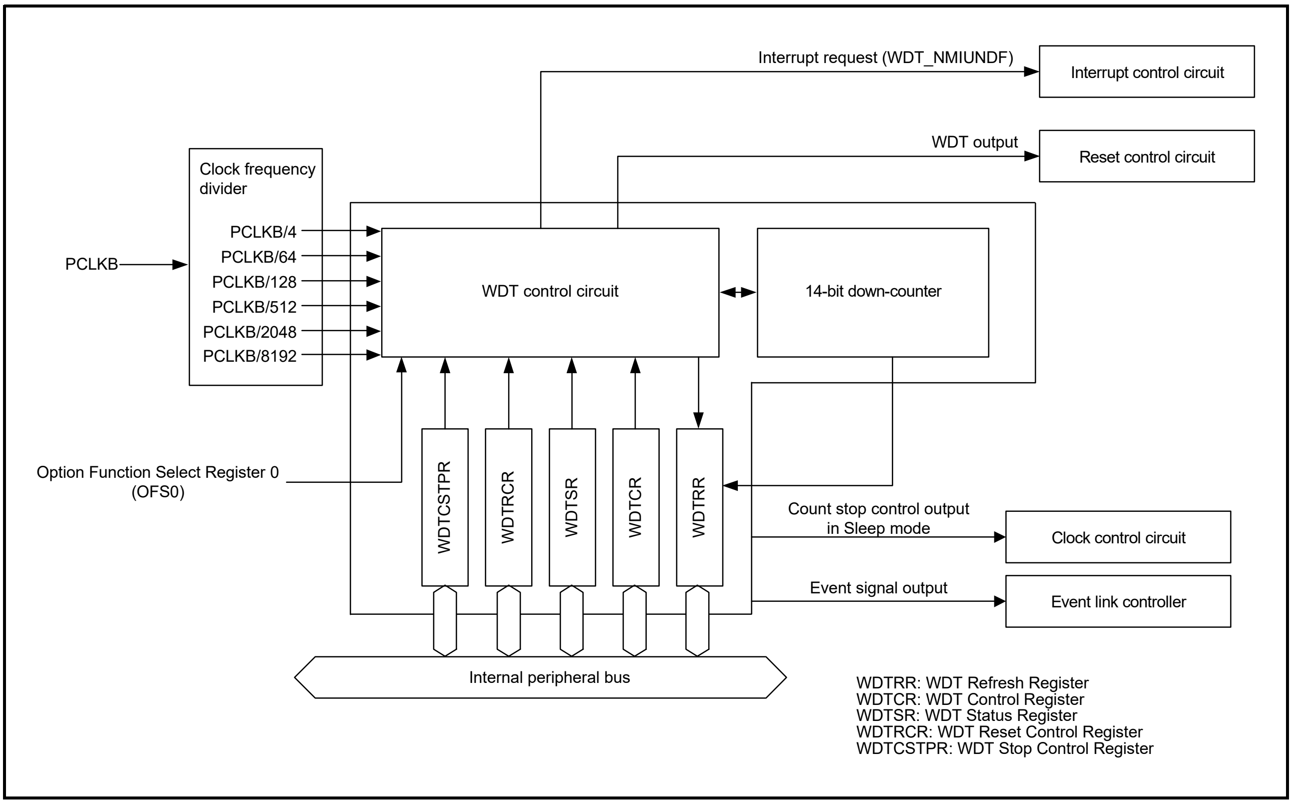

Interrupt request (WDT_NMIUNDF)
Interrupt control circuit
WDT output
Reset control circuit
Clock frequency divider
PCLKB
PCLKB/4
PCLKB/64
PCLKB/128
PCLKB/512
PCLKB/2048
PCLKB/8192
WDT control circuit
14-bit down-counter
Option Function Select Register 0 (OFS0)
WDTCSTPR
WDTRCR
WDTSR
WDTCR
WDTRR
Count stop control output in Sleep mode
Clock control circuit
Event signal output
Event link controller
Internal peripheral bus
WDTRR: WDT Refresh Register
WDTCR: WDT Control Register
WDTSR: WDT Status Register
WDTRCR: WDT Reset Control Register
WDTCSTPR: WDT Stop Control Register

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 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 the Timeout Period Select 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 Select	b1 b0 0 0: 1,024 cycles (03FFh) 0 1: 4,096 cycles (0FFFh) 1 0: 8,192 cycles (1FFFh) 1 1: 16,384 cycles (3FFFh).	R/W
b3, b2	—	Reserved	These bits are read as 0 and cannot be modified	R/W
b7 to b4	CKS[3:0]	Clock Division Ratio Select	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 Select	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 and cannot be modified	R/W
b13, b12	RPSS[1:0]	Window Start Position Select	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 and cannot be modified	R/W

Some constraints apply to writes to the WDTCR register. For details, see section 26.3.2, Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the settings in the WDTCR register are disabled, and the settings in the Option Function Select Register 0 (OFS0) are enabled. The settings for the WDTCR register can also be made for the OFS0 register. For details, see section 26.3.7, Associations between Option Function Select Register 0 (OFS0) and WDT Registers.

TOPS[1:0] bits (Timeout Period Select)

The TOPS[1:0] bits select the timeout period (the period until the down-counter underflows) from 1,024, 4,096, 8,192, and 16,384 cycles, taking the divided clock specified by the CKS[3:0] bits as 1 cycle.

After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the 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	1024	4096
				0	1		4096	16384
				1	0		8192	32768
				1	1		16384	65536
0	1	0	0	0	0	PCLKB/64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	PCLKB/128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	1	0	0	0	PCLKB/512	1024	524288
				0	1		4096	2097152
				1	0		8192	4194304
				1	1		16384	8388608
0	1	1	1	0	0	PCLKB/2048	1024	2097152
				0	1		4096	8388608
				1	0		8192	16777216
				1	1		16384	33554432
1	0	0	0	0	0	PCLKB/8192	1024	8388608
				0	1		4096	33554432
				1	0		8192	67108864
				1	1		16384	134217728

CKS[3:0] bits (Clock Division Ratio Select)

The CKS[3:0] bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from the 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 PCLKB clock cycles can be selected for the WDT.

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

The RPES[1:0] bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. The selected window end position should be a value smaller than the value for the window start position (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.

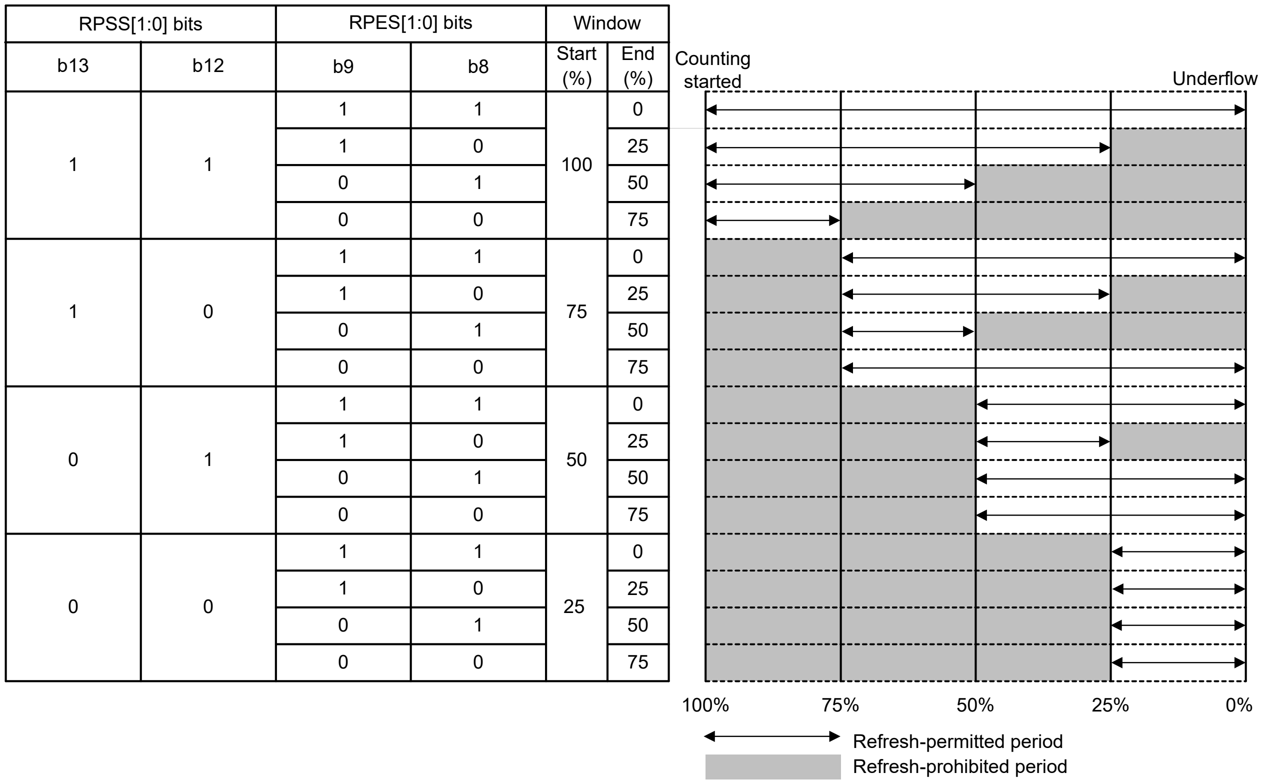
RPSS[1:0] bits (Window Start Position Select)

The RPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window 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 the CNTVAL[13:0] 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 the UNDFF flag to confirm whether an underflow occurred in the down-counter. A value of 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 the flag is ignored for (N+1) PCLKB cycles following 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 the REFEF flag to confirm whether a refresh error occurred. A value of 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 following 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 and cannot be modified	R/W
b7	RSTIRQS	Reset Interrupt Request Select	0: Non-maskable interrupt request or interrupt request output is enabled 1: Reset output is enabled.	R/W

Some constraints apply to writes to the WDTRCR register. For details, see section 26.3.2, Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the WDTRCR register settings are disabled, and the settings in the Option Function Select Register 0 (OFS0) are enabled. The settings for the WDTCR register can also be made for the OFS0 register. For details, see section 26.3.7, Associations 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 and cannot be modified	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 state. Some constraints apply to writes to the WDTCSTPR register. For details, see section 26.3.2, Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the WDTCSTPR register settings are disabled, and the settings in the Option Function Select Register 0 (OFS0) are enabled. The settings for the WDTCSTPR register can also be made for the OFS0 register. For details, see section 26.3.7, Associations between Option Function Select Register 0 (OFS0) and WDT Registers.

SLCSTP bit (Sleep-Mode Count Stop Control)

The SLCSTP bit selects whether to stop counting when transition to Sleep mode.

26.2.6 Option Function Select Register 0 (OFS0)

For information on the OFS0 register, see section 26.3.7, Associations 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 a release from the reset state
- Register start mode, in which counting starts with a refresh by writing to the register.

In auto start mode, counting automatically starts after release from the reset state according to the settings in the Option Function Select Register 0 (OFS0) in the flash.

In register start mode, counting starts with a refresh by writing to the 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:

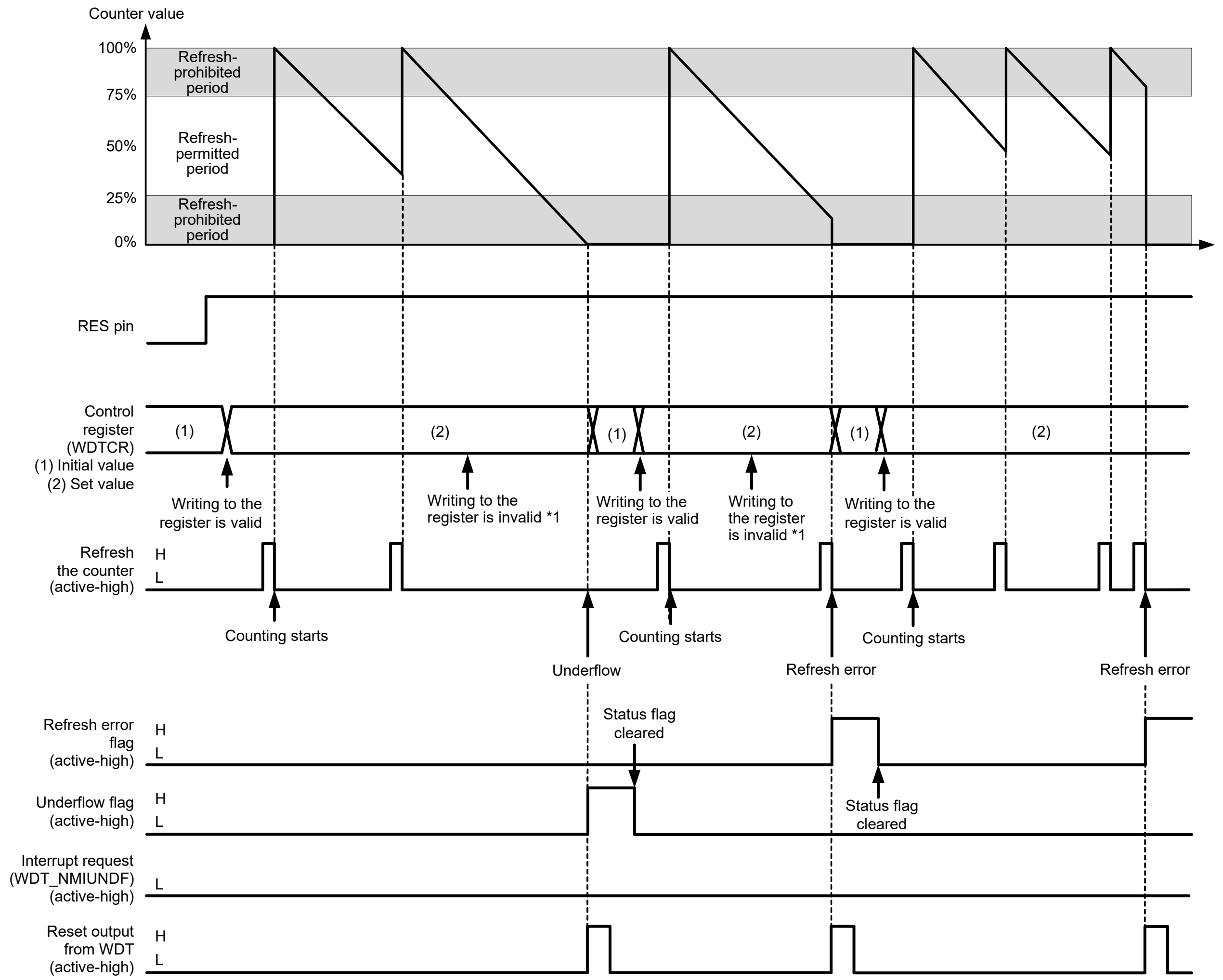
- Clock division ratio
- Window start and end positions
- Timeout period in the WDTCR register
- Reset output or interrupt request output in the WDTRCR register
- Counter stop control during transition to Sleep mode in the WDTCSTPR register.

Refresh the down-counter to start counting down from the value set in the Timeout Period Select 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 in the WDT Reset Interrupt Request Select bit (WDTRCR.RSTIRQS). Non-maskable interrupt request or interrupt request can be selected in 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).


Counter value
100%
75%
50%
25%
0%
Refresh-prohibited period
Refresh-permitted period
Refresh-prohibited period
RES pin
Control register (WDTCR)
(1) Initial value
(2) Set value
(1)
(2)
Writing to the register is valid
Writing to the register is invalid *1
Writing to the register is valid
Writing to the register is invalid *1
Writing to the register is valid
Refresh the counter (active-high)
H
L
Counting starts
Underflow
Refresh error
Status flag cleared
Refresh error flag (active-high)
Underflow flag (active-high)
Interrupt request (WDT_NMIUNDF) (active-high)
Reset output from WDT (active-high)

Note 1. See 26.3.2 Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

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 following values in the Option Function Select register 0 (OFS0) are set in the WDT registers:

- Clock division ratio
- Window start and end positions
- Timeout period
- Reset output or interrupt request
- Counter stop control during transition to Sleep mode.

When the reset state is released, the down-counter automatically starts counting down from the value set in the WDT Timeout Period Select bits (OFS0.WDTTOPS[1:0]).

Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The WDT does not output the reset signal 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 1 cycle. The value of the timeout period is set in the down-counter and counting restarts.

Reset output or interrupt request output can be selected in the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS). Non-maskable interrupt request or interrupt request can be selected in the WDT Underflow/ Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 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).

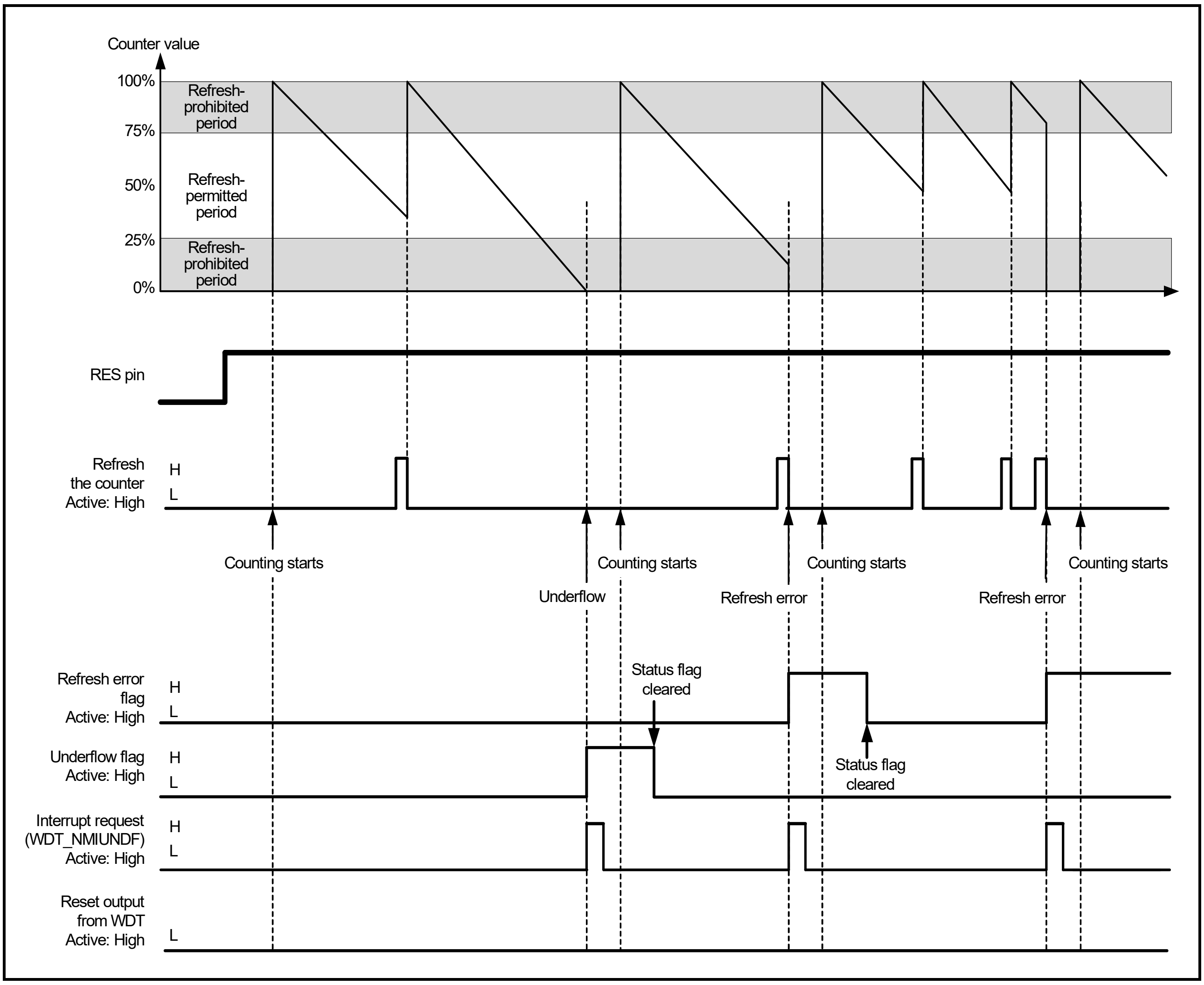

Counter value
100%
75%
50%
25%
0%
Refresh-prohibited period
Refresh-permitted period
Refresh-prohibited period
RES pin
Refresh the counter Active: High
H
L
Counting starts
Underflow
Counting starts
Refresh error
Counting starts
Refresh error
Counting starts
Refresh error flag Active: High
Status flag cleared
Underflow flag Active: High
Status flag cleared
Interrupt request (WDT_NMIUNDF) Active: High
Reset output from WDT Active: High

Figure 26.4 Operation example in auto start mode

26.3.2 Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers

Writing to the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), or WDT Count Stop Control Register (WDTCSTPR) is possible once between the release from the reset state and the first refresh operation.

After a refresh, (counting starts) or a write to WDTCR, WDTRCR or WDTCSTPR, the protection signal in the WDT becomes 1 to protect WDTCR, WDTRCR and WDTCSTPR against subsequent write 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.

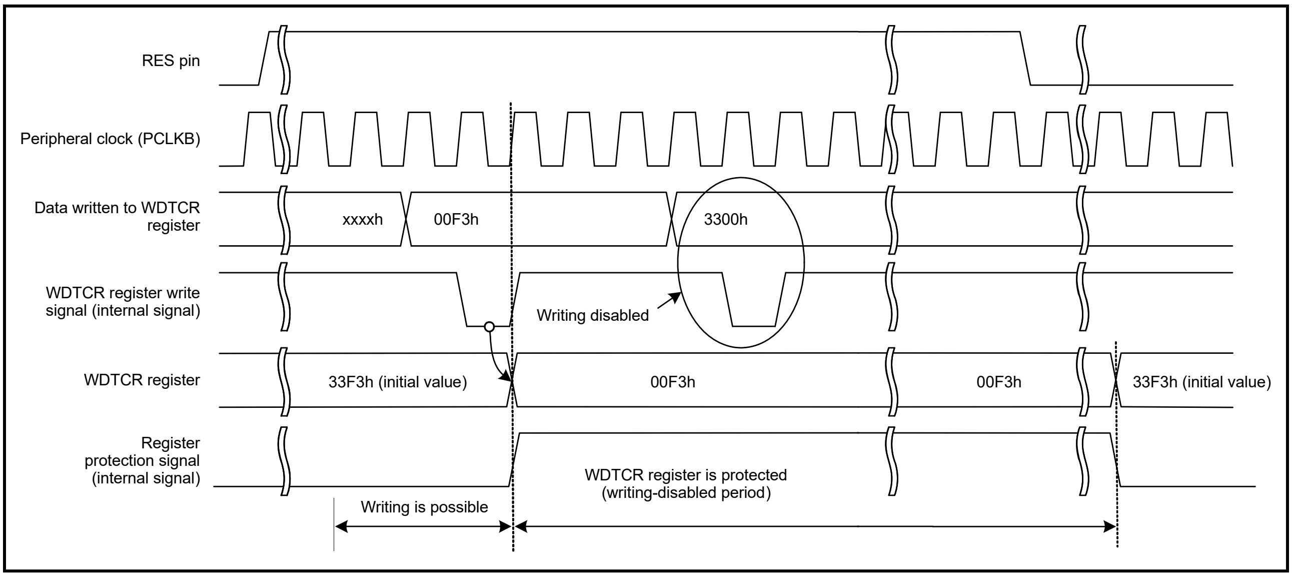

RES pin
Peripheral clock (PCLKB)
Data written to WDTCR register
xxxxh
00F3h
3300h
WDTCR register write signal (internal signal)
Writing disabled
WDTCR register
33F3h (initial value)
00F3h
00F3h
33F3h (initial value)
Register protection signal (internal signal)
WDTCR register is protected (writing-disabled period)
Writing is possible

Figure 26.5 Control waveforms produced in response to writes to the 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. If an invalid value is written, correct refreshing resumes by writing 00h and FFh to the WDTRR Register.

Correct refreshing is also performed when a register other than WDTRR is accessed or when WDTRR is read between writing 00h and writing FFh to WDTRR.

Writes to refresh the counter must be made within the refresh-permitted period, and this is determined by writing FFh. For this reason, correct refreshing is performed even when 00h is written outside the refresh-permitted period.

[Example write sequences 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.

[Example write sequences that are invalid 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. To meet this requirement, complete writing FFh to WDTRR 4 count cycles before the down-counter underflows.

Figure 26.6 shows the WDT refresh-operation waveforms when the clock division ratio = PCLKB/64.

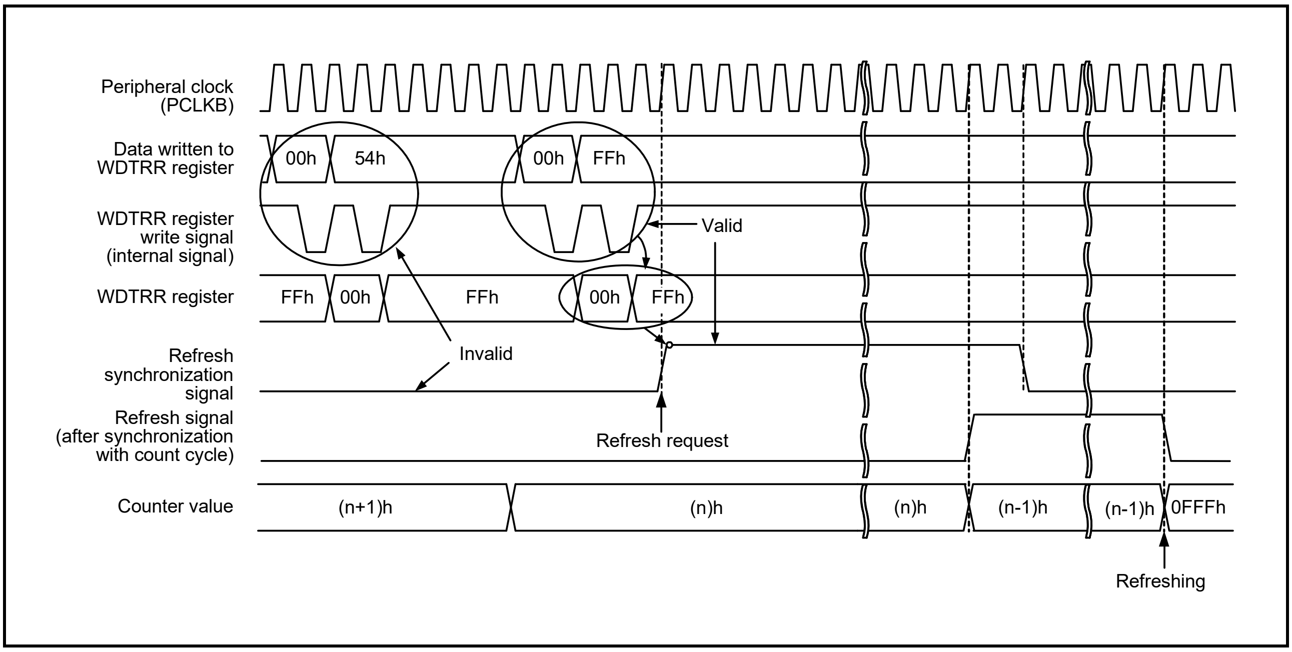

Peripheral clock (PCLKB)
Data written to WDTRR register
00h
54h
00h
FFh
WDTRR register write signal (internal signal)
Valid
WDTRR register
FFh
00h
FFh
00h
FFh
Refresh synchronization signal
Invalid
Refresh signal (after synchronization with count cycle)
Refresh request
Counter value
(n+1)h
(n)h
(n)h
(n-1)h
(n-1)h
0FFFh
Refreshing

Figure 26.6 WDT refresh operation waveforms when WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b

26.3.4 Reset Output

When the Reset Interrupt Select bit (WDTRCR.RSTIRQS) is set to 1 in register start mode or when the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 1 in auto start mode, a reset signal is output for 1 cycle count when an underflow in the down-counter or a refresh error occurs.

In register start mode, the down-counter is initialized (all bits set to 0) and stopped in that state after output of a reset signal. After the reset state is released and the program is restarted, the counter is set up 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 Select bit (WDTRCR.RSTIRQS) is set to 0 in register start mode or when the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 0 in auto start mode, an interrupt (WDT_NMIUNDF) signal is generated when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or an interrupt. For details, see section 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 the Down-Counter Value

The WDT stores the counter value in the down-counter value bits (WDTSR.CNTVAL[13:0]) of the WDT Status Register. Check these bits to obtain the counter value.

Figure 26.7 shows the processing for reading the WDT down-counter value when the clock division ratio is PCLKB/64.

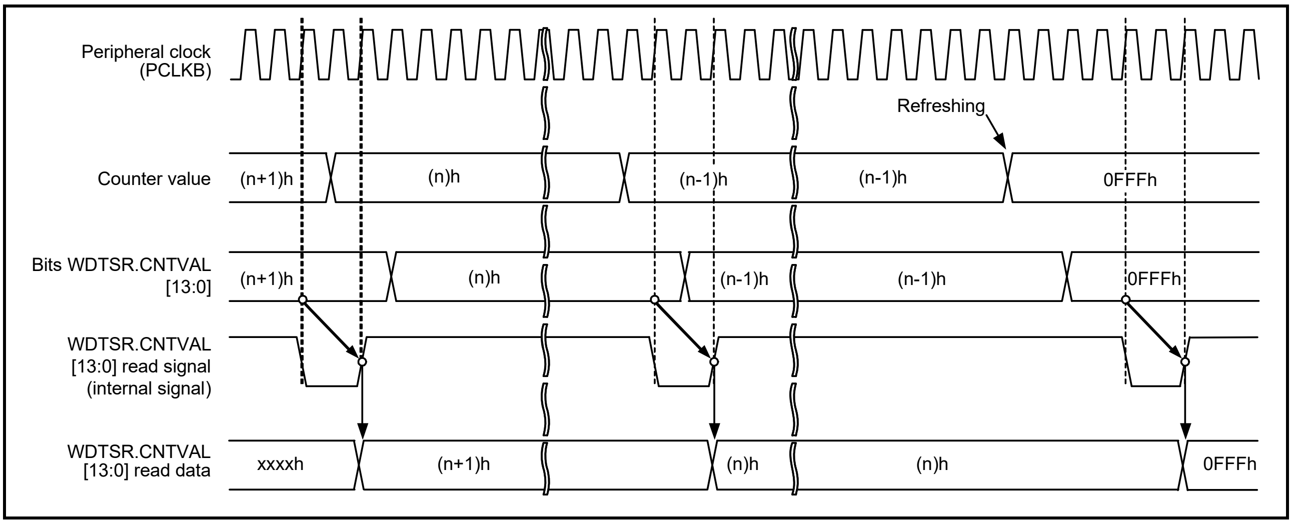


Figure 26.7 Read process for WDT down-counter value when WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b

26.3.7 Associations 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 the WDT registers

Control target	Function	OFS0 register (enabled in auto start mode) OFS0.WDTSTRT = 0	WDT registers (enabled in register start mode) OFS0.WDTSTRT = 1
Down-counter	Timeout period selection	OFS0.WDTTOPS[1:0]	WDTCR.TOPS[1:0]
	Clock division ratio selection	OFS0.WDTCKS[3:0]	WDTCR.CKS[3:0]
	Window start position selection	OFS0.WDTRPSS[1:0]	WDTCR.RPSS[1:0]
	Window end position selection	OFS0.WDTRPES[1:0]	WDTCR.RPES[1:0]
Reset output or interrupt request output	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 Select 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 25h 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 (ELSRm.ELS[7:0] = 25h).

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 IWDT can be triggered automatically by a reset, underflow, refresh error, or a refresh of the count value in the registers.

The IWDT functions differ from those of the WDT in the following respects:

- The divided IWDT-dedicated clock (IWDTCLK) is used as the count source (not affected by 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 a 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, supply the IWDT-dedicated clock (IWDTCLK). The bus interface and registers operate with PCLKB, and the 14-bit counter and control circuits operate with IWDTCLK.

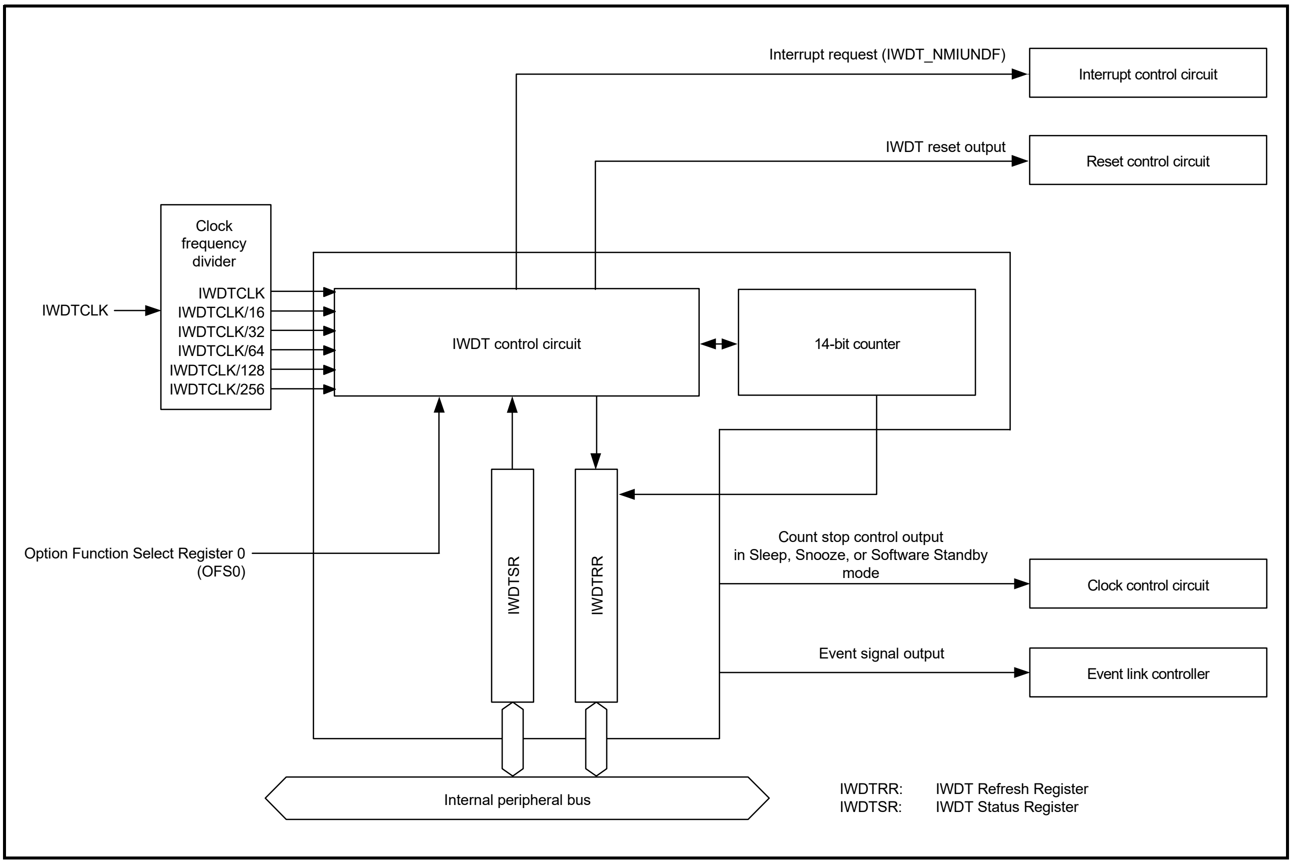

Interrupt request (IWDT_NMIUNDF)
Interrupt control circuit
IWDT reset output
Reset control circuit
Clock frequency divider
IWDTCLK
IWDTCLK
IWDTCLK/16
IWDTCLK/32
IWDTCLK/64
IWDTCLK/128
IWDTCLK/256
IWDT control circuit
14-bit counter
Option Function Select Register 0 (OFS0)
IWDTSR
IWDTRR
Count stop control output in Sleep, Snooze, or Software Standby mode
Clock control circuit
Event signal output
Event link controller
Internal peripheral bus
IWDTRR: IWDT Refresh Register
IWDTSR: IWDT Status Register

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 the CNTVAL[13:0] bits to confirm the value of the down-counter. The read value might differ from the actual count by 1.

UNDFF flag (Underflow Flag)

Read the UNDFF flag to confirm whether an underflow occurred in the down-counter. The value 1 indicates that the down-counter underflowed. Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

Clearing of the UNDFF flag takes (N+2) IWDTCLK cycles and 2 PCLKB cycles. In addition, clearing of the 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 the REFEF 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+2) IWDTCLK cycles and 2 PCLKB cycles. In addition, clearing of this flag is ignored for (N+2) IWDTCLK cycles after a refresh error. 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.

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 Selects)

The IWDTTOPS[1:0] bits select the timeout period, that is, the period until the down-counter underflows from 128, 512, 1024, or 2048 cycles, taking the divided clock specified by the IWDTCKS[3:0] bits as 1 cycle.

After the down-counter is refreshed, the combination of the IWDTCKS[3:0] and IWDTTOPS[1:0] bits determines the number of IWDTCLK cycles until the counter underflows.

Table 27.2 lists the relationship between the IWDTCKS[3:0] and IWDTTOPS[1:0] bit settings, the timeout period, and the number of IWDTCLK cycles.

Table 27.2 Timeout period settings

IWDTCKS[3:0] bits				IWDTTOPS[1:0] bits		Clock division ratio	Timeout period (number of cycles)	IWDTCLK cycles
b7	b6	b5	b4	b1	b0			
0	0	0	0	0	0	IWDTCLK	128	128
				0	1		512	512
				1	0		1024	1024
				1	1		2048	2048
0	0	1	0	0	0	IWDTCLK/16	128	2048
				0	1		512	8192
				1	0		1024	16384
				1	1		2048	32768
0	0	1	1	0	0	IWDTCLK/32	128	4096
				0	1		512	16384
				1	0		1024	32768
				1	1		2048	65536
0	1	0	0	0	0	IWDTCLK/64	128	8192
				0	1		512	32768
				1	0		1024	65536
				1	1		2048	131072
1	1	1	1	0	0	IWDTCLK/128	128	16384
				0	1		512	65536
				1	0		1024	131072
				1	1		2048	262144
0	1	0	1	0	0	IWDTCLK/256	128	32768
				0	1		512	131072
				1	0		1024	262144
				1	1		2048	524288

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

The IWDTCKS[3:0] bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from the IWDT-dedicated clock (IWDTCLK) divided by 1, 16, 32, 64, 128, and 256. 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)

The IWDTRPES[1:0] bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. 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.

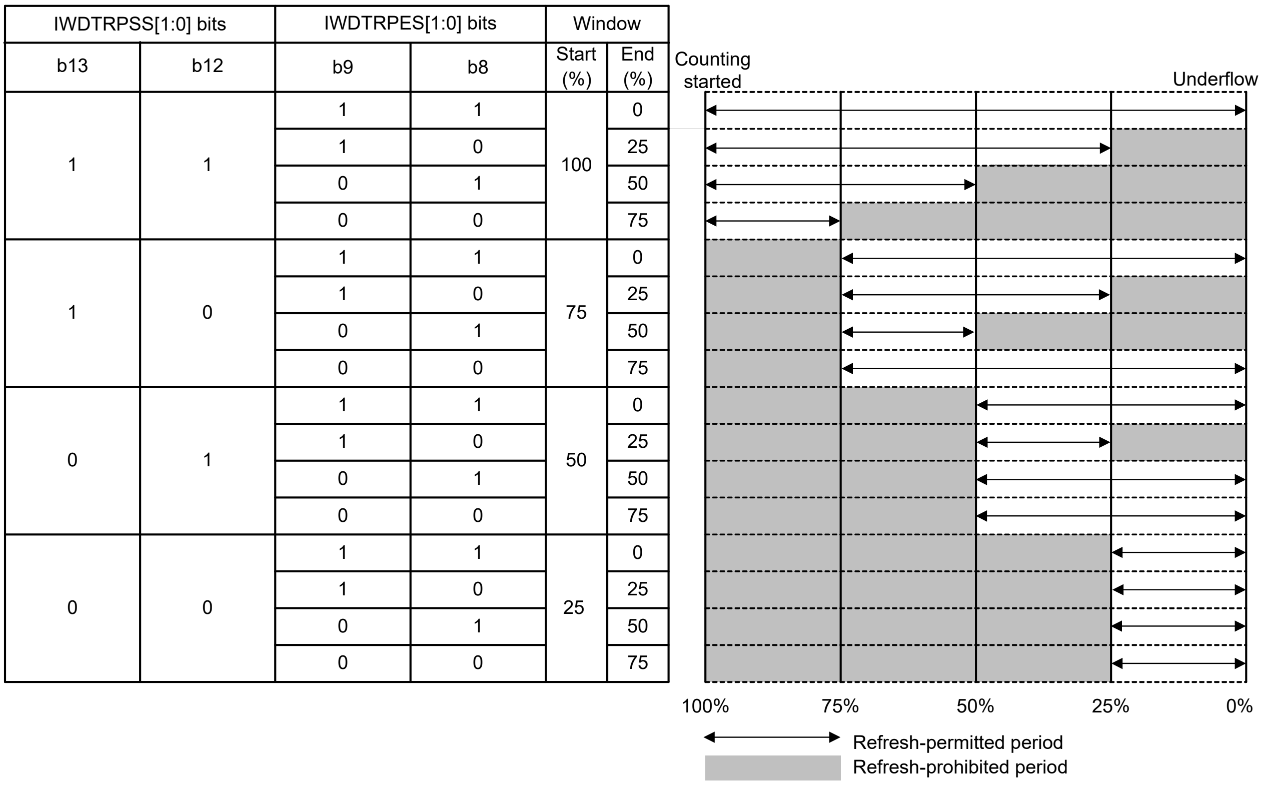
IWDTRPSS[1:0] bits (IWDT Window Start Position Select)

The IWDTRPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. The window start position should be a value greater than the value for 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 in 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)

The IWDTRSTIRQS 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)

The IWDTSTPCTL bit selects whether the stop counting on transition to Sleep, Snooze, 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 when an attempt is made to refresh outside the refresh-permitted period, the IWDT asserts the reset signal or non-maskable interrupt request/interrupt request (IWDT_NMIUNDF).

After the reset signal or non-maskable interrupt request/interrupt request is generated, the counter reloads the timeout period after counting for 1 cycle, and restarts the count. The reset output or interrupt request can be selected with the IWDT Reset Interrupt Request Select bit (OFS0.IWDTRSTIRQS). Non-maskable interrupt request or interrupt request can be selected with 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).

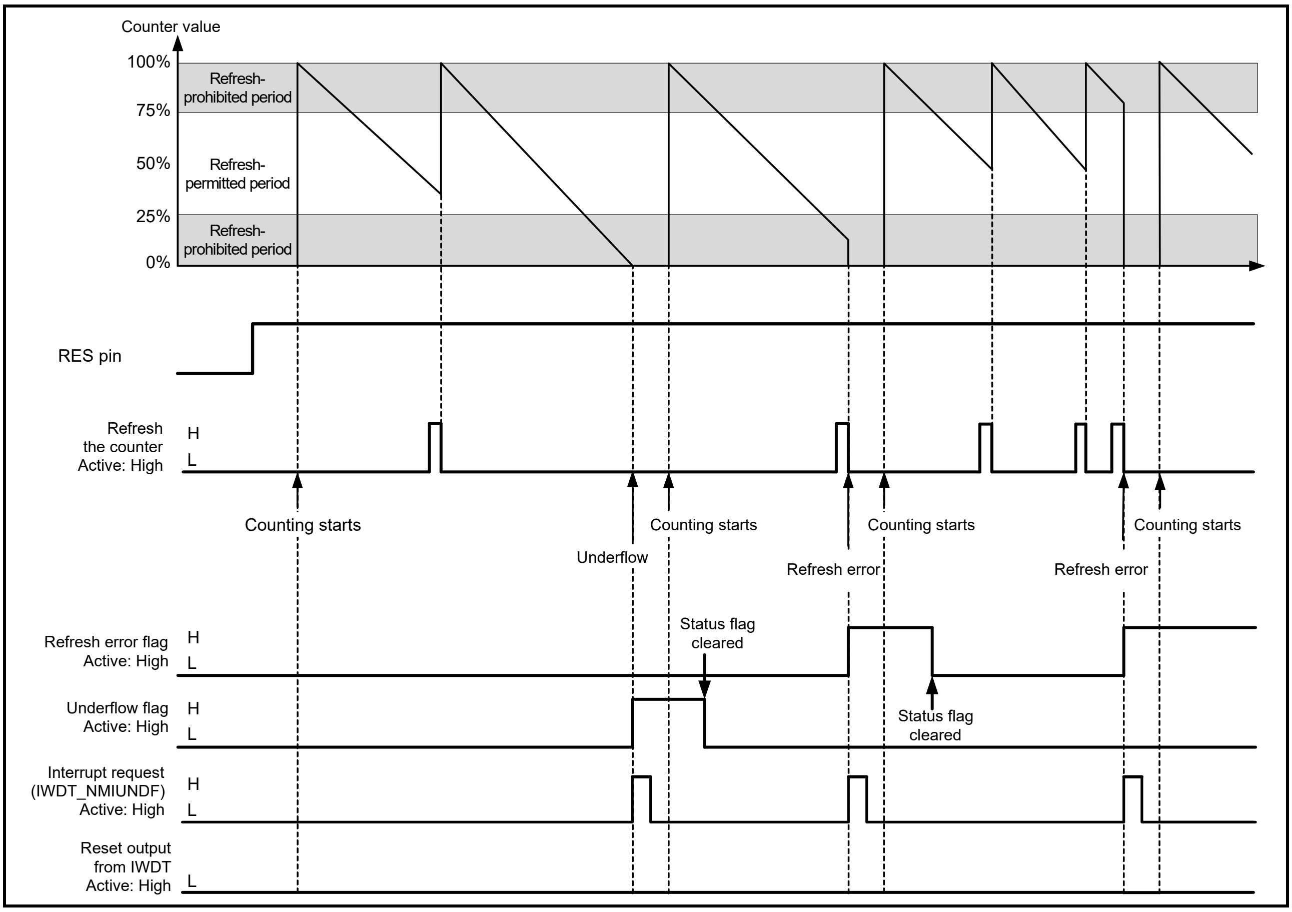

Counter value
100%
75%
50%
25%
0%
Refresh-prohibited period
Refresh-permitted period
Refresh-prohibited period
RES pin
Refresh the counter Active: High
H
L
Counting starts
Underflow
Counting starts
Refresh error
Counting starts
Refresh error
Counting starts
Status flag cleared
Refresh error flag Active: High
Underflow flag Active: High
Status flag cleared
Interrupt request (IWDT_NMIUNDF) Active: High
Reset output from IWDT Active: High

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 down-counter is not refreshed. If an invalid value is written, correct refreshing resumes on a write of 00h and FFh to the IWDTRR.

When writes are made 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-1)th time) → 00h (nth time) → FFh is valid, and the refresh is performed correctly. Even when the first value written before 00h is not 00h, correct refreshing is performed as long as the operation contains the write sequence of 00h → FFh.

Correct refreshing is also performed when a register other than IWDTRR is accessed or IWDTRR is read between writing 00h and writing FFh to IWDTRR.

[Example write sequences that are valid to refresh the counter]

- 00h → FFh
- 00h (n–1th time) → 00h (nth time) → FFh
- 00h → access to another register or read from IWDTRR → FFh.

[Example write sequences that are not valid to refresh 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 completes.

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 1 cycle for counting). Therefore, writing FFh to the IWDTRR must be completed 4 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]).

[Example 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 occurs if FFh is written to IWDTRR after the value of the IWDTSR.CNTVAL[13:0] bits reaches 1FFFh
- When the window end position is set to 1FFFh, refreshing occurs if 2003h (4 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 performed immediately before an underflow. In this case, if 0003h (4 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 performed.

Figure 27.4 shows the IWDT refresh-operation waveforms when PCLKB > IWDTCLK and the clock division ratio is IWDTCLK.

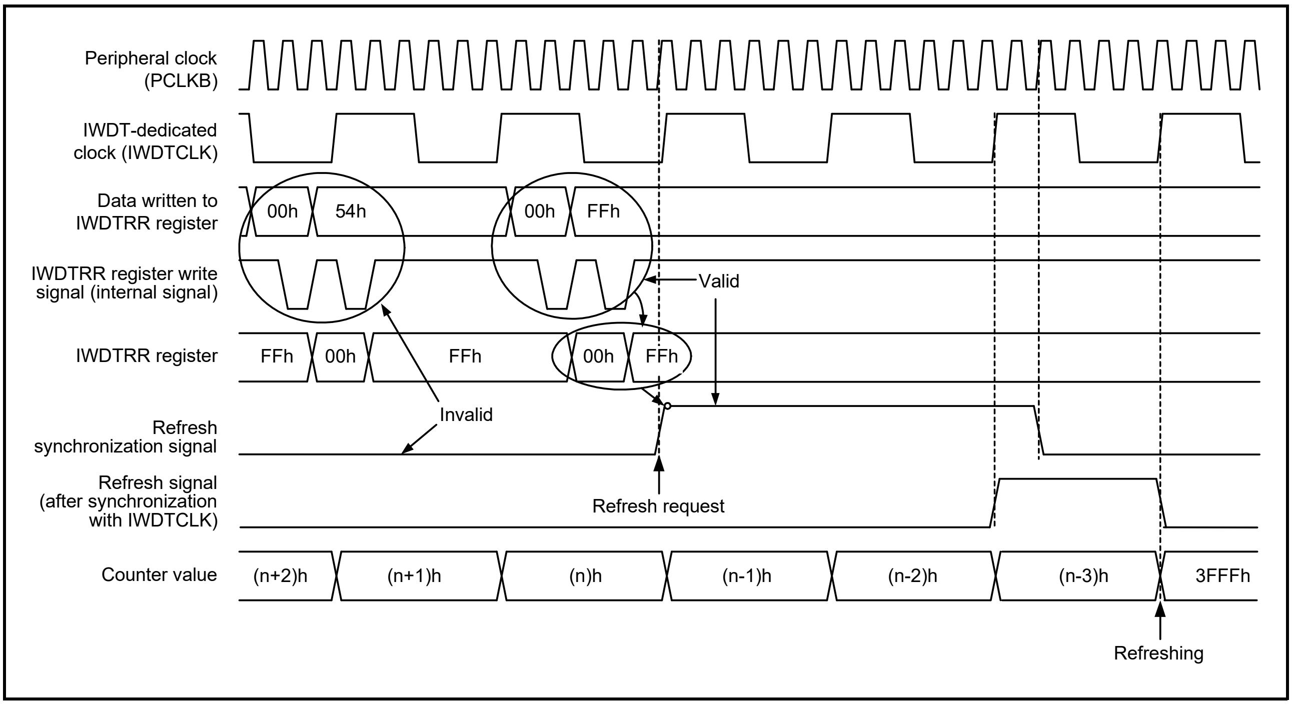

Peripheral clock (PCLKB)
IWDT-dedicated clock (IWDTCLK)
Data written to IWDTRR register
00h
54h
00h
FFh
IWDTRR register write signal (internal signal)
Valid
IWDTRR register
FFh
00h
FFh
00h
FFh
Invalid
Refresh synchronization signal
Refresh signal (after synchronization with IWDTCLK)
Refresh request
Counter value
(n+2)h
(n+1)h
(n)h
(n-1)h
(n-2)h
(n-3)h
3FFFh
Refreshing

Figure 27.4 **IWDT refresh operation waveforms when 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 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 3 IWDTCLK cycles and 2 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. Therefore, 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 4 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.

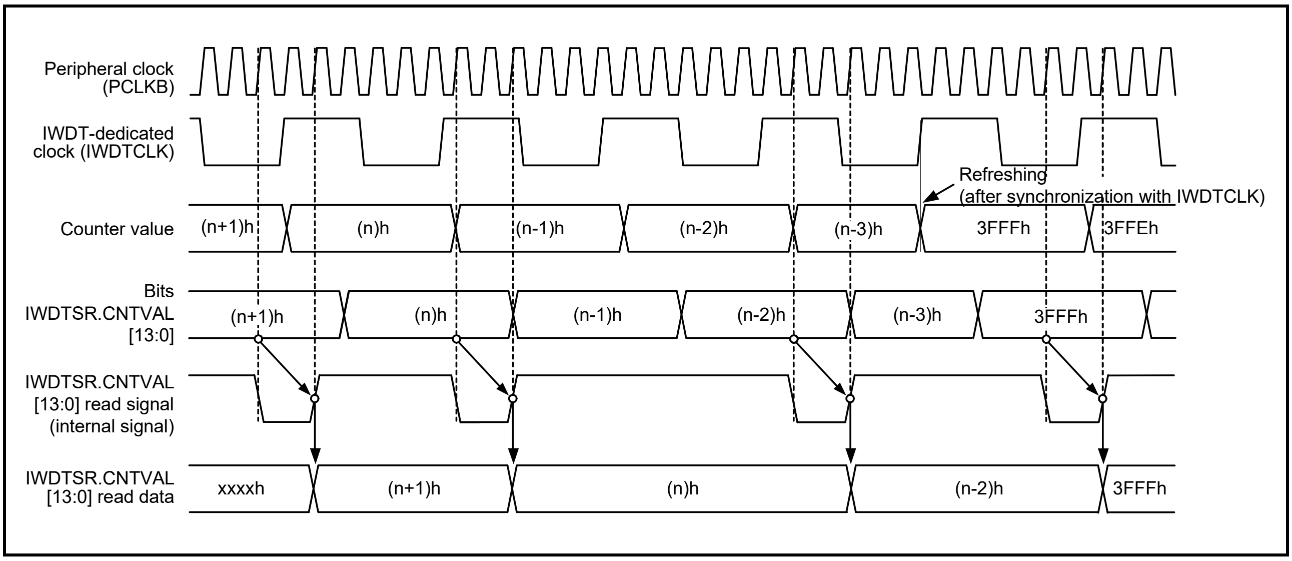

Peripheral clock (PCLKB)
IWDT-dedicated clock (IWDTCLK)
Counter value
Bits IWDTSR.CNTVAL [13:0]
IWDTSR.CNTVAL [13:0] read signal (internal signal)
IWDTSR.CNTVAL [13:0] read data
Refreshing (after synchronization with IWDTCLK)
(n+1)h
(n)h
(n-1)h
(n-2)h
(n-3)h
3FFFh
3FFEh
xxxxh

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 given the accuracy of PCLKB and IWDTCLK. Set values that 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.

Table 28.1 lists the USBFS specifications, Figure 28.1 shows a 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 • Host and device controller can be switched by software • Self-power or bus power mode can be selected • Revision 1.2 of Battery Charging specification is supported
	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 can be selected, including the default control pipe • Pipes 1 to 9 can be assigned any endpoint number.
	Transfer conditions that can be set for each pipe: • Pipe 0: Control transfer with 64-byte single buffer • Pipes 1 and 2: Bulk transfer with 64-byte double buffer 256-byte double buffer for isochronous transfer • 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 • HOCO clock that can be used as USB clock.
Module-stop function	Module-stop state can be set to reduce power consumption

Figure 28.1 shows a block diagram of the USBFS.

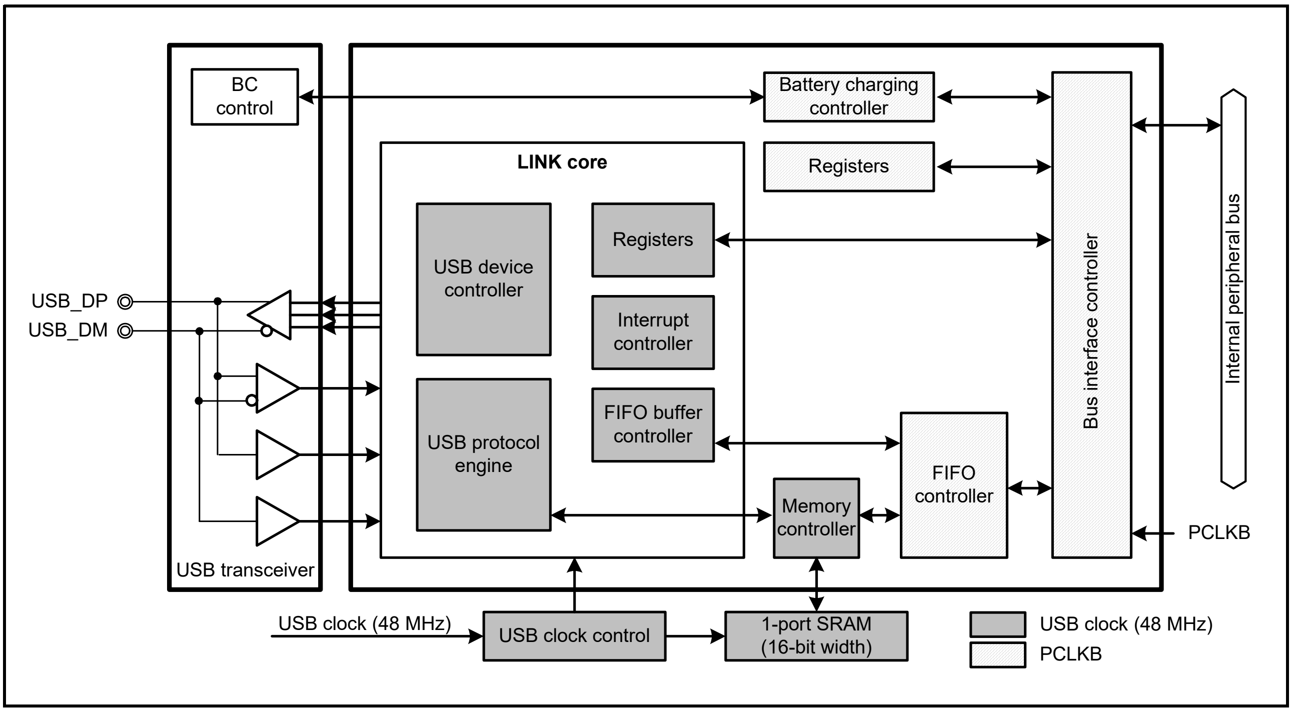

BC control
Battery charging controller
LINK core
Registers
USB device controller
Registers
Interrupt controller
FIFO buffer controller
USB protocol engine
Memory controller
FIFO controller
Bus interface controller
Internal peripheral bus
PCLKB
USB_DP
USB_DM
USB transceiver
USB clock (48 MHz)
USB clock control
1-port SRAM (16-bit width)
USB clock (48 MHz)
PCLKB

Figure 28.1 USBFS block diagram

Table 28.2 lists the I/O pins of the USBFS.

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+ pin of the USB bus.
	USB_DM	I/O	D– I/O pin for the on-chip USB transceiver. Must be connected to the D- pin of the USB bus.
	USB_VBUS	Input	USB cable connection monitor pin. Must be connected to VBUS signal on the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USBFS is a device controller.*1
	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.
Common	VCC_USB	Input	Power supply for USB transceiver.
	VCC_USB_LDO	Input	Power supply pin for USB transceiver. Apply the same voltage as VCC_USB
	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: Line pull-up disabled 1: Line pull-up enabled.	R/W
b4	DPRPU	D+ Line Resistor Control*1	0: Line pull-up disabled 1: Line pull-up enabled.	R/W
b5	DRPD	D+/D– Line Resistor Control	0: Line pull-down disabled 1: Line pull-down enabled.	R/W
b6	DCFM	Controller Function Select	0: Device controller selected 1: Host controller selected.	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: Single-ended receiver disabled 1: Single-ended receiver enabled.	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: Clock supply to the USBFS stopped 1: Clock supply to the USBFS enabled.	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 and confirm 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 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 USB.

When this bit is 0, only SYSCFG can be read from and written to. No other USB-related registers can 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	—	—	—	—	LNST[1:0]	
Value after reset:	0*1	0*1	0	0	0	0	0	0	0	0	0	0	0	x	0	0

x: Undefined

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	—	Reserved	The read value is undefined.	R
b5 to b3	—	Reserved	These bits are read as 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.	R

Bit	Symbol	Bit name	Description	R/W
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 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) or after enabling pull-down of the lines (SYSCFG.DRPD bit = 1) in host controller mode.

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 a 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
	—	—	—	—	—	—	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: Downstream port disabled (SOF transmission disabled) 1: Downstream port enabled (SOF transmission enabled).	R/W
b5	RESUME	Resume Output	0: Resume signal not output 1: Resume signal output.	R/W

Bit	Symbol	Bit name	Description	R/W
b6	USBRST	USB Bus Reset Output	0: USB bus reset signal not output 1: USB bus reset signal output.	R/W
b7	RWUPE	Wakeup Detection Enable	0: Downstream port wakeup disabled 1: Downstream port wakeup enabled.	R/W
b8	WKUP	Wakeup Output	0: Remote wakeup signal not output 1: Remote wakeup signal output.	R/W
b9	VBUSEN	USB_VBUSEN Output Pin Control	0: External USB_VBUSEN pin outputs low 1: External USB_VBUSEN pin outputs high.	R/W
b15 to b10	—	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 USB enters the idle state after the SOF packet output.

With this bit set to 0, the USB enters an idle state after outputting SOF packets.

The USB 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 suspended 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 cleared to 0 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 maintained 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). Set this bit to 0 in host controller mode.

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

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

(2) When the MBW bit is 0

Address(es): USBFS.CFIFOL 4009 0014h, USBFS.D0FIFOL 4009 0018h, USBFS.D1FIFOL 4009 001Ch

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

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 transmission 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 [15:8]	Bits [7:0]
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 [15:8]	Bits [7:0]
0	Access prohibited*1	N + 0 data
1	Access prohibited*1	N + 0 data

Note 1. Writing to or reading from an access-prohibited area 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 received 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 decrement each time the received 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 used to read or write 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, the current pipe specification is maintained until the access is complete, even if software attempts to change the CURPIPE[3:0] setting. Access continues after the current value is written back to the CURPIPE[3:0] bits.

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 the ISEL 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 reading the FIFO buffer, read with the access size 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

Bit	Symbol	Bit name	Description	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
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 used to read or write 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, the current pipe specification is maintained until the access is complete, even if software attempts to change the CURPIPE[3:0] setting. Access continues after the current value is written back to the CURPIPE[3:0] bits.

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 reading the FIFO buffer, read with the access size set in the 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	Indicates the received data length. These bits indicate different values depending on the setting of the RCNT bit setting in the port select register. For details, see the description of the DTLN[8:0] bits in this section.	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 in the CPU.	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 received 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 USB retains the length of the received 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. 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 received 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. In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

- 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.

BCLR bit (CPU Buffer Clear)

Set the BCLR bit to 1 to clear the FIFO buffer in 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 bit 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 in 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

Bit	Symbol	Bit name	Description	R/W
b8	BRDYE	Buffer Ready Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b9	NRDYE	Buffer Not Ready Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b10	BEMPE	Buffer Empty Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b11	CTRE	Control Transfer Stage Transition Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b12	DVSE	Device State Transition Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b13	SOFE	Frame Number Update Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b14	RSME	Resume Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b15	VBSE	VBUS Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled.	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

Bit	Symbol	Bit name	Description	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 Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b1	PIPE1BRDYE	BRDY Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b2	PIPE2BRDYE	BRDY Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b3	PIPE3BRDYE	BRDY Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b4	PIPE4BRDYE	BRDY Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b5	PIPE5BRDYE	BRDY Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b6	PIPE6BRDYE	BRDY Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b7	PIPE7BRDYE	BRDY Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b8	PIPE8BRDYE	BRDY Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b9	PIPE9BRDYE	BRDY Interrupt Enable for Pipe 9	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 enable bit in the BRDYENB register is changed from 0 to 1 by 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 Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b1	PIPE1NRDYE	NRDY Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b2	PIPE2NRDYE	NRDY Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b3	PIPE3NRDYE	NRDY Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b4	PIPE4NRDYE	NRDY Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b5	PIPE5NRDYE	NRDY Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b6	PIPE6NRDYE	NRDY Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b7	PIPE7NRDYE	NRDY Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b8	PIPE8NRDYE	NRDY Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b9	PIPE9NRDYE	NRDY Interrupt Enable for Pipe 9	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 NRDYENB register 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 enable bit in the NRDYENB register is changed from 0 to 1 by 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 Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled.	R/W

Bit	Symbol	Bit name	Description	R/W
b1	PIPE1BEMPE	BEMP Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b2	PIPE2BEMPE	BEMP Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b3	PIPE3BEMPE	BEMP Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b4	PIPE4BEMPE	BEMP Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b5	PIPE5BEMPE	BEMP Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b6	PIPE6BEMPE	BEMP Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b7	PIPE7BEMPE	BEMP Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b8	PIPE8BEMPE	BEMP Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b9	PIPE9BEMPE	BEMP Interrupt Enable for Pipe 9	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 enable bit in the BEMPENB register is changed from 0 to 1 by 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 this bit is 0 before stopping the clock supply to the USBFS.

EDGESTS bit (Edge Interrupt Output Status Monitor)

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 the timing for clearing the BRDY interrupt status for each pipe.

TRNENSEL bit (Transaction-Enabled Time Select)

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
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

Bit	Symbol	Bit name	Description	R/W
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 bits, 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 are 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 through software.

CTSQ[2:0] bits (Control Transfer Stage)

In host controller mode, the read value of the CTSQ[2:0] bits is invalid.

VALID bit (USB Request Reception)

In host controller mode, the read value of the VALID bit 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)

The BRDY 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. The BRDY bit cannot be set to 0 even if 0 is written to this bit by software.

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 for the PIPENRDY bits associated with the PIPEnNRDYE bits (n = 0 to 9) that is set to 1 (when the USBFS detects the NRDY interrupt status in at least one pipe from 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. The NRDY bit cannot be set to 0 even if 0 is written to this bit by software.

BEMP bit (Buffer Empty Interrupt Status)

The BEMP 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 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 when detecting a change 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 when 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 when updating the frame number (when the DVSTCTR0.UACT bit is 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 when updating the frame number. An SOFR 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 when 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 when 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 bit 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

Bit	Symbol	Bit name	Description	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	Overcurrent 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 through 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.

PDDETINTE0 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 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.

Software must terminate all pipes in which communications are currently being carried out and re-enumerate the USB port.

Values read from the EOFERR bit 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 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 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.

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)

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 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 (Overcurrent Input Change Interrupt Status)

The OVRCR bit indicates the status of the USB_OVRCURA and USB_OVRCURB input pin change interrupts.

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 Pipe 0*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b1	PIPE1BRDY	BRDY Interrupt Status for Pipe 1*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b2	PIPE2BRDY	BRDY Interrupt Status for Pipe 2*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b3	PIPE3BRDY	BRDY Interrupt Status for Pipe 3*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b4	PIPE4BRDY	BRDY Interrupt Status for Pipe 4*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b5	PIPE5BRDY	BRDY Interrupt Status for Pipe 5*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b6	PIPE6BRDY	BRDY Interrupt Status for Pipe 6*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b7	PIPE7BRDY	BRDY Interrupt Status for Pipe 7*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b8	PIPE8BRDY	BRDY Interrupt Status for Pipe 8*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b9	PIPE9BRDY	BRDY Interrupt Status for Pipe 9*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 Pipe 0	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b1	PIPE1NRDY	NRDY Interrupt Status for Pipe 1	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b2	PIPE2NRDY	NRDY Interrupt Status for Pipe 2	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b3	PIPE3NRDY	NRDY Interrupt Status for Pipe 3	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1

Bit	Symbol	Bit name	Description	R/W
b4	PIPE4NRDY	NRDY Interrupt Status for Pipe 4	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b5	PIPE5NRDY	NRDY Interrupt Status for Pipe 5	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b6	PIPE6NRDY	NRDY Interrupt Status for Pipe 6	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b7	PIPE7NRDY	NRDY Interrupt Status for Pipe 7	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b8	PIPE8NRDY	NRDY Interrupt Status for Pipe 8	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b9	PIPE9NRDY	NRDY Interrupt Status for Pipe 9	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 Pipe 0	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b1	PIPE1BEMP	BEMP Interrupt Status for Pipe 1	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b2	PIPE2BEMP	BEMP Interrupt Status for Pipe 2	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b3	PIPE3BEMP	BEMP Interrupt Status for Pipe 3	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b4	PIPE4BEMP	BEMP Interrupt Status for Pipe 4	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b5	PIPE5BEMP	BEMP Interrupt Status for Pipe 5	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b6	PIPE6BEMP	BEMP Interrupt Status for Pipe 6	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b7	PIPE7BEMP	BEMP Interrupt Status for Pipe 7	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b8	PIPE8BEMP	BEMP Interrupt Status for Pipe 8	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b9	PIPE9BEMP	BEMP Interrupt Status for Pipe 9	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 FRNM[10:0] bits indicate the latest frame number for the USBFS after issuing of an SOF packet every 1 ms or writing to the FRNM[10:0] bits at the SOF packet reception.

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	These bits store the USB request bmRequestType value	R/W*1
b15 to b8	BREQUEST[7:0]	Request	These bits store the 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 read/write.

USBREQ stores setup requests for control transfers.

In device controller mode, the USBREQ stores the received values of bRequest and bmRequestType. 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 the setup transactions for transmission. 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 the setup transactions for reception. Writing to these bits has no effect.

BREQUEST[7:0] bits (Request)

The BREQUEST[7:0] bits store the 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 the setup transactions for reception. Writing to these 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 read/write.

In device controller mode, USBVAL stores the received value of wValue. 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)

The WVALUE[15:0] bits store the wValue value of the USB request.

- In host controller mode:
  Set these bits to the wValue value of USB requests in the setup transactions for transmission. 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 the setup transactions for reception. Writing to these 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 read/write.

USBINDX stores setup requests for control transfers.

In device controller mode, the USBINDX stores the received wIndex value. In host controller mode, the USBINDX sets the wIndex value to be transmitted.

USBINDX is initialized by a USB bus reset.

WINDEX[15:0] bits (Index)

The WINDEX[15:0] bits hold the value of a USB request.

- In host controller mode:
  Set these bits to the wIndex value of USB requests in the setup transactions for transmission. 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 of USB requests in the setup transactions for reception. Writing to these bits has no effect.

28.2.22 USB Request Length Register (USBLENG)

Address(es): USBFS.USBLENG 4009 005Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WLENTUH[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	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 read/write.

USBLENG stores setup requests for control transfers.

In device controller mode, the USBLENG stores the received value of wLength. In host controller mode, the USBLENG sets the value of wLength to be transmitted.

USBLENG is initialized by a USB bus reset.

WLENTUH[15:0] bits (Length)

The WLENTUH[15:0] 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 kept open 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 USBFS changes the PID[1:0] bits to NAK, checking the PBUSY bit through software is not required.

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 a transfer has ended. The USBFS determines that the transfer has ended 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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	—	—	—	—	MXPS[6:0]						
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
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 USBFS changes the PID[1:0] bits to NAK, checking the PBUSY bit through software is not required. After modifying the MXPS[6:0] bits 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 USBFS changes the PID[1:0] bits to NAK, checking the PBUSY bit through software is not required.

MXPS[6:0] bits (Maximum Packet Size*1)

The MXPS[6:0] bits specify the maximum data payload (maximum packet size) for the DCP. The initial value 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*2)

In host controller mode, the DEVSEL[3:0] 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	SQMO N	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.	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.	R/W*1
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. Only set the SQSET and SQCLR bits to 1 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 USBFS changes the PID[1:0] bits to NAK, checking the PBUSY bit through software is not required.

PID[1:0] bits (Response PID)

The PID[1:0] bits control the USBFS 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 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 bit to 1, and the PID[1:0] setting cannot be changed until software clears the VALID bit to 0.
- When the PID[1:0] bits are set to BUF (01b) by 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 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. Software cannot write 1 to the bit while the INTSTS0.VALID bit is 1. The CCPL 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 at the start of a USBFS transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

After PID is set to NAK by 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 SQMON bit on successful 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*2)

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*2)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer. The SQCLR bit 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 software. This is not required 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 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 target 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)

The BSTS 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 received data can be read from the buffer
- When ISEL = 1, the bit indicates whether transmitted 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: 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
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, set the pipe functions using PIPECFG, PIPEMAXP, and PIPEPERI registers.The PIPEnCTR, PIPEnTRE, and PIPEnTRN registers can be set independently of the pipe selection in the PIPESEL 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	These bits specify 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: Pipe operation continued after transfer ends 1: Pipe operation disabled 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 generated on transmitting or receiving data 1: BRDY interrupt generated 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 USBFS changes the PID[1:0] bits to 00 (NAK), checking the PBUSY bit through software is not required.

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 software is not required.

Note 3. To change 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 then 0 to the PIPEnCTR.ACLRM bit continuously through software to clear the FIFO buffer assigned to the selected 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*1)

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*2,*3)

The DIR bit specifies the transfer direction for the selected pipe.

When software sets this bit to 0, the USBFS uses the selected pipe for receiving. When software sets this bit to 1, the USBFS uses the selected pipe for transmitting.

SHTNAK bit (Pipe Disabled at End of Transfer*1)

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 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*2,*3)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe. The bit is valid for pipes 1 to 5.

BFRE bit (BRDY Interrupt Operation Specification*2,*3)

The BFRE bit specifies the BRDY interrupt generation timing from the USBFS to the CPU for the selected pipe.

When software sets the BFRE bit to 1 and the selected pipe is receiving, the USBFS detects the transfer completion and generates the BRDY interrupt on reading the packet.

When a BRDY interrupt is generated with this setting, write 1 to the BCLR bit in the Port Control Register with software. 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 software and the selected pipe is transmitting, the USBFS does not generate the BRDY interrupt. For details, see section 28.3.3.1, BRDY interrupt.

TYPE[1:0] bits (Transfer Type*1)

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 software is not required.

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 USBFS changes the PID[1:0] bits to 00 (NAK), checking the PBUSY bit through software is not required.

PIPEMAXP specifies the maximum packet size for pipes 1 to 9.

MXPS[8:0] bits (Maximum Packet Size*2)

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*3)

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 the n-th 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 software is not required.

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 to 0. When the selected pipe is not for isochronous transfer, set this bit to 0.

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: The selected pipe is not used for the transaction 1: The selected pipe is used 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.	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.	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: There are no data to be transmitted in the FIFO buffer 1: There is data to be transmitted 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 software is not required.

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 USBFS changes the PID[1:0] bits to 00 (NAK), checking the PBUSY bit through software is not required.

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 on 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 USBFS 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 software during USBFS communication on the selected pipe, check that the PBUSY bit is 1 to determine if USBFS 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 software is not required.

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 receiving and the PIPECFG.SHTNAK bit for the selected pipe is set to 1 by 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

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)	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.
	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 currently used for the transaction.

The USBFS changes the PBUSY bit from 0 to 1 at the start of the USBFS transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

Reading the PBUSY bit with 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*2)

Setting the SQSET bit to 1 through software allows the USBFS to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the selected pipe. The USBFS sets the SQSET bit to 0.

SQCLR bit (Sequence Toggle Bit Clear*2)

Setting the SQCLR bit to 1 through software allows the USBFS to clear the expected value of the sequence toggle bit for the next transaction of the selected pipe to DATA0. The USBFS sets the SQCLR bit to 0.

ACLRM bit (Auto Buffer Clear Mode*3)

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 (1 of 2)

Number	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

Table 28.9 Data cleared by the USBFS when ACLRM = 1 (2 of 2)

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
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*2)

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 for bulk transfer. 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 transmitting (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 it 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 it 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 receiving (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	Set to 1 when received data can be read from the FIFO buffer, and set to 0 on completion of data read
		1	Setting prohibited
	1	0	Set to 1 when received data can be read from the FIFO buffer, and set to 0 when software sets the BCLR bit in the Port Control Register to 1 after the data read is complete
		1	Set to 1 when received data can be read from the FIFO buffer, and set to 0 on completion of data read
1	0	0	Set to 1 when transmitted data can be written to the FIFO buffer, and 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: The selected pipe is not used for the transaction 1: The selected pipe is used 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 software is not required.

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 software is not required.

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 selected pipe for USB transfer. Table 28.7 and Table 28.7 show the basic operation of the USBFS depending on the PID[1:0] setting when there are no errors in the transmitted and received packets.

After changing the PID[1:0] setting from BUF to NAK through software during USB communication on the selected pipe, check that the PBUSY bit is 1 to determine 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 software is not required.

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 currently used for the transaction.

The USBFS changes the PBUSY bit from 0 to 1 at the start of the USBFS transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

Reading the PBUSY bit through 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 bit indicates the expected value of the sequence toggle bit for the next transaction of the selected pipe.

The USBFS toggles the SQMON bit 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*2)

Setting the SQSET bit to 1 through software allows the USBFS to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the selected pipe. The USBFS sets the SQSET bit to 0.

SQCLR bit (Sequence Toggle Bit Clear*2)

Setting the SQCLR bit to 1 through software allows the USBFS to clear the expected value of the sequence toggle bit for the next transaction of the selected pipe to DATA0. The USBFS sets the SQCLR bit to 0.

ACLRM bit (Auto Buffer Clear Mode*2,*3)

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

Number	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, check that the PIPEnCTR.PBUSY bit is 0, then change the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK. If the USBFS changes the PID[1:0] bits to NAK, checking the PBUSY bit through software is not required.

TRCLR bit (Transaction Counter Clear)

When the TRCLR bit is set to 1, the USBFS 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 software allows the USBFS to control hardware on having received the number of packets equal to the TRNCNT[15:0] setting, as follows:

- 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 number of packets (number of transactions) to be received by the selected pipe • When read from: Indicates the specified number of transactions if the PIPEnTRE.TRENB bit is 0. Indicates the number of currently counted transactions if the PIPEnTRE.TRENB bit is 1.	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 sets 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 aligns 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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	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
b15 to b2	—	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.

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 the RPDME0 bit to 1 to control the pull-down resistor of the D- pin.

IDPSRCE0 bit (D+ Pin IDPSRC Output Control)

When the IDPSRCE0 bit is 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)

When the IDMSINKE0 bit is set to 1 in device controller mode, the USBFS detects whether VDMSRC (0.6 V), output from the host to D- on primary detection, is connected, or whether VDPSRC (0.6 V), output from the device to D+, is connected to D- by the host.

VDPSRCE0 bit (D+ Pin VDPSRC (0.6 V) Output Control)

When the VDPSRCE0 bit is 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)

When the IDPSINKE0 bit is set to 1 in device controller mode, the USBFS detects whether VDMSRC (0.6 V), output from the device to D-, is connected to D+ (DCP) by the host. In host controller mode, the USBFS detects whether VDPSRC (0.6 V), output from the device to D+ on primary detection, is connected.

VDMSRCE0 bit (D- Pin VDMSRC (0.6 V) Output Control)

When the VDMSRCE0 bit is 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, the CHGDETSTS0 flag is set to 1 if the USBFS detects whether VDMSRC (0.6 V), output from the host to D- during primary detection, is connected, or whether VDPSRC (0.6 V), output from the device to D+, is connected to D- by the host.

PDDETSTS0 flag (D+ Pin 0.6 V Input Detection Status)

In device controller mode, the PDDETSTS0 flag is set to 1 if the USBFS detects whether VDMSRC (0.6 V), output from the device to D- during secondary detection, is connected to D+ (DCP) by the host.

In host controller mode, this bit is set to 1 if the USBFS detects whether VDPSRC (0.6 V), output from the device 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 the USBFS-related registers

Setting the SYSCFG.USBE bit to 1 after starting the clock supply to the USB (SYSCFG.SCKE bit = 1) enables and starts USBFS operation.

28.3.1.2 Selecting the controller function

The USBFS can operate as either a host or device controller.

Use the SYSCFG.DCFM bit to select one of these 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 USBFS 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, SYSCFG.DMRPU, and SYSCFG.DRPD bits.

In device controller mode, confirm that connection to the USB host is made, then set the SYSCFG.DPRPU bit to 1 to pull up the D+ line (in full-speed communication), or set the SYSCFG.DMRPU bit to 1 to 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 Control settings for the USBFS data bus resistors

SYSCFG register settings			USB data bus control		
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

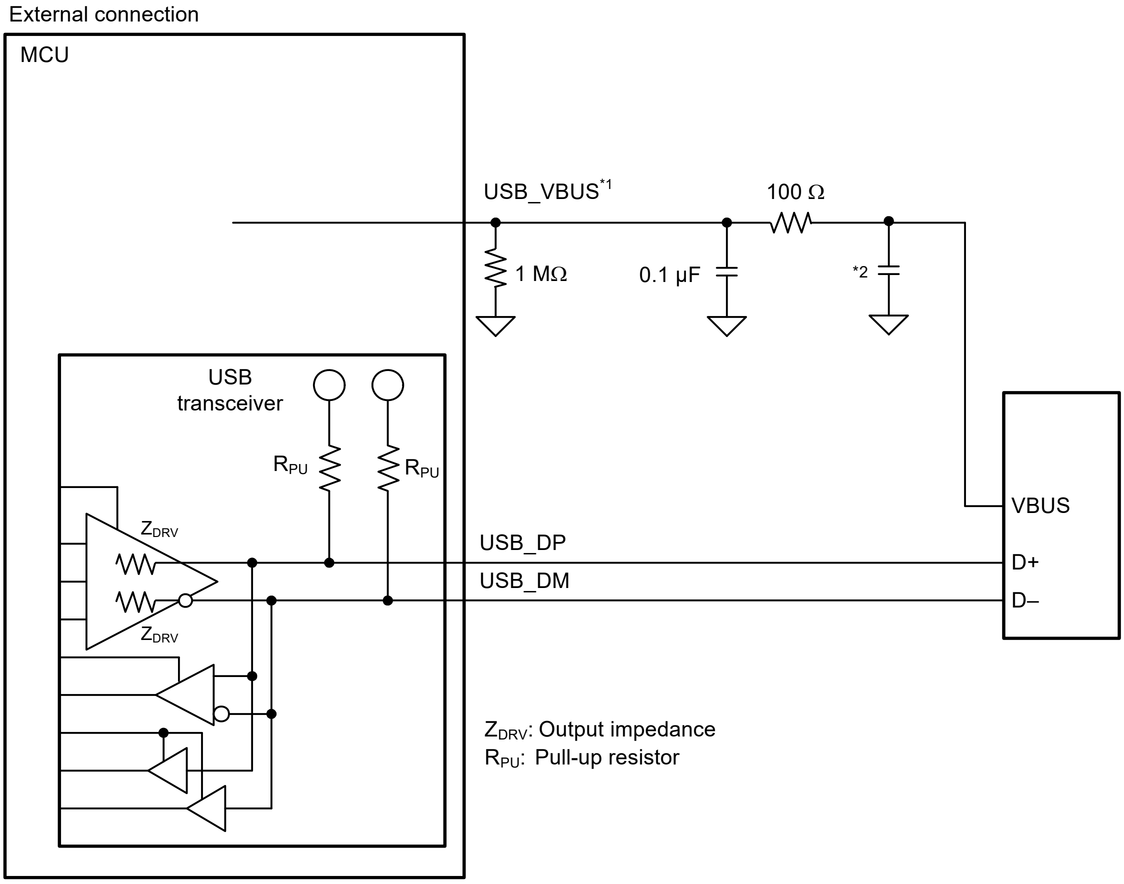
28.3.1.4 Example of USB external connection circuits

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 provides a power supply in the USB-PHY.

Figure 28.2 show examples of external circuits for USB connection.

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.

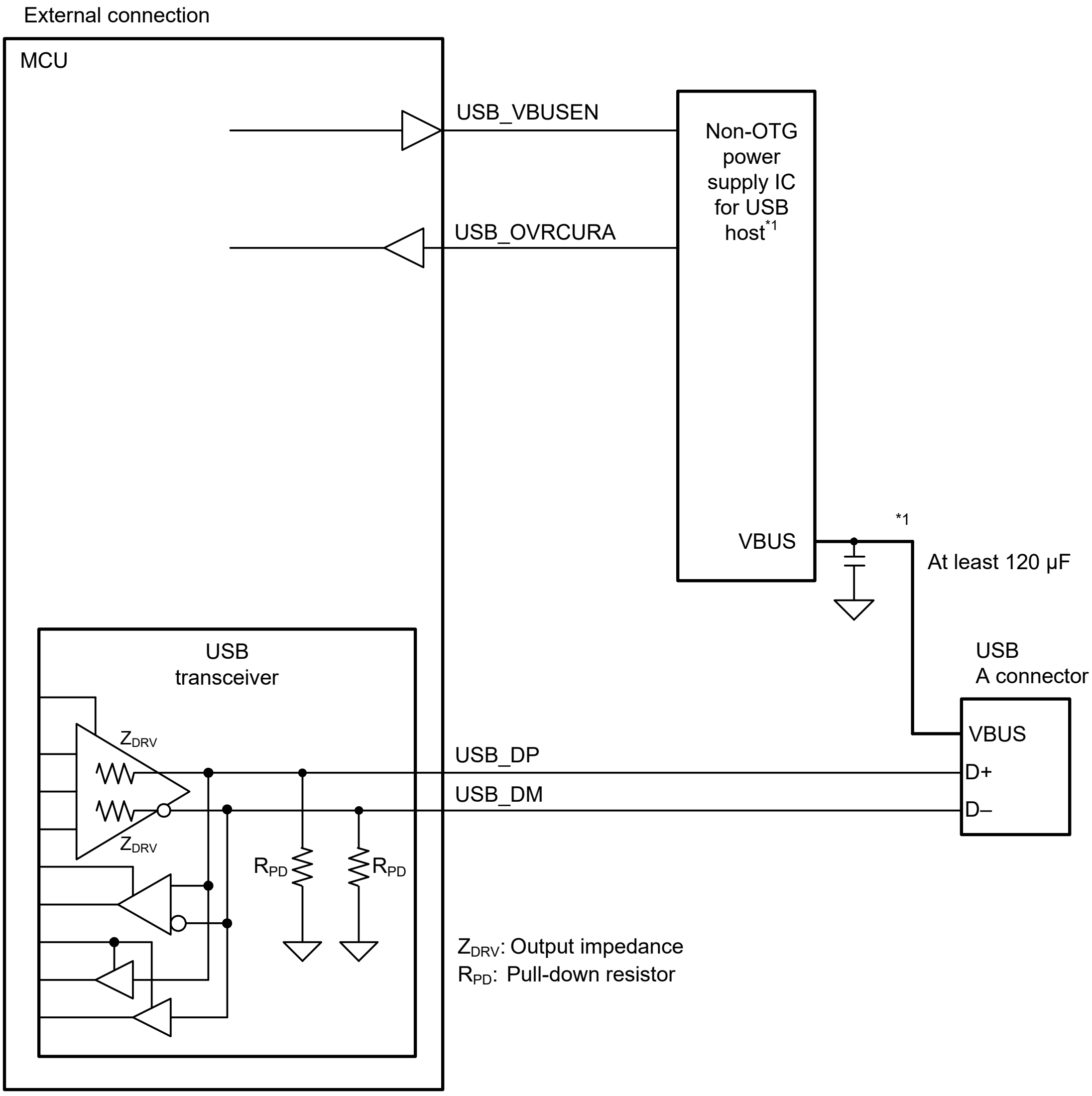
Figure 28.2 shows an example of functional connection of the USB connector in the self-powered state.


External connection
MCU
USB_VBUS*1
100 Ω
1 MΩ
0.1 µF
*2
USB transceiver
RPU
RPU
ZDRV
ZDRV
VBUS
USB_DP
D+
USB_DM
D–
ZDRV: Output impedance
RPU: Pull-up resistor

Note 1. USB_VBUS is 5 V tolerant.
Note 2. Design the board so that the total VBUS capacitance ranges from 1.0 to 10 µF.

Figure 28.2 Example device connection in self-powered state

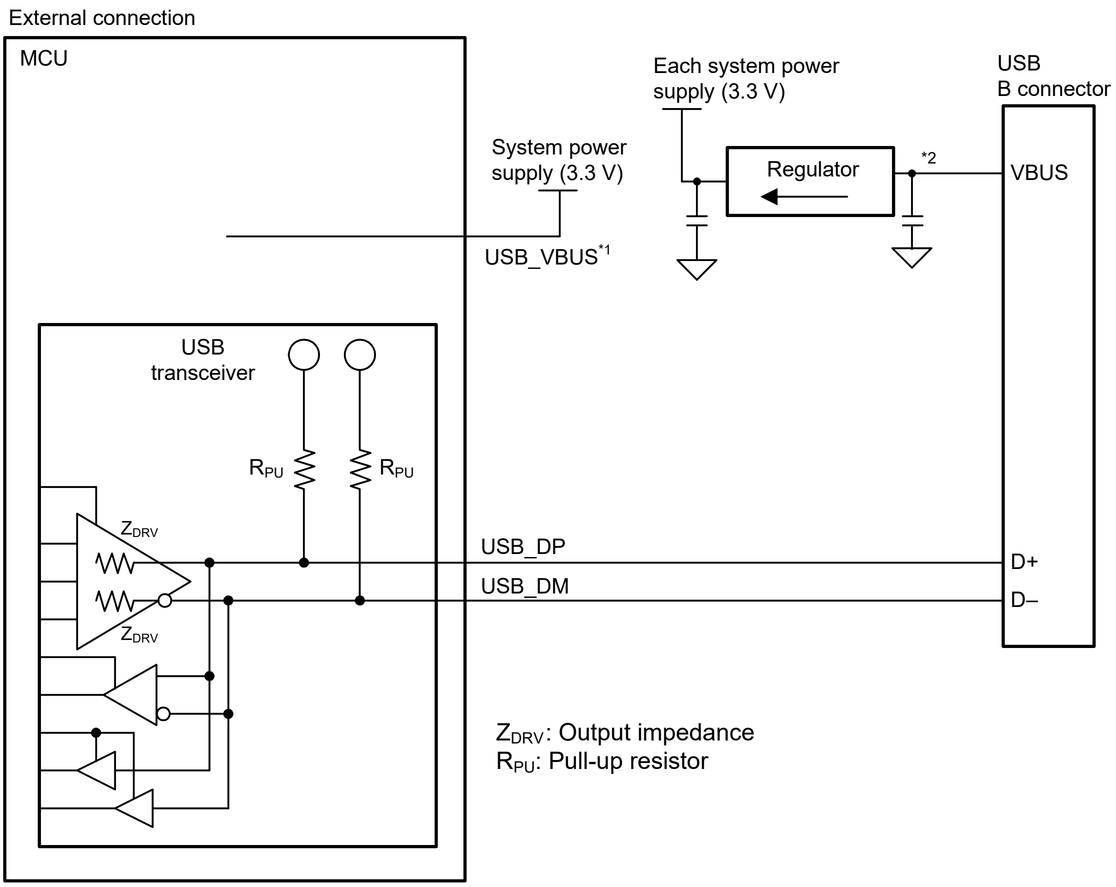
Figure 28.3 shows an example of host connection of the USB connector.


External connection
MCU
USB_VBUSEN
USB_OVRCURA
Non-OTG power supply IC for USB host*1
VBUS
*1
At least 120 μF
USB A connector
VBUS
D+
D–
USB transceiver
ZDRV
ZDRV
USB_DP
USB_DM
RPD
RPD
ZDRV: Output impedance
RPD: Pull-down resistor

Note 1. When Battery Charging Specification Rev 1.2 is to be supported, ensure that the power supply IC and the VBUS wiring width are enough for at least 1.5 A.

Figure 28.3 **Example host connection**

Figure 28.4 shows an example of functional connection of the USB connector in bus-powered state.


External connection
MCU
System power supply (3.3 V)
USB_VBUS*1
Each system power supply (3.3 V)
Regulator
*2
USB B connector
VBUS
USB transceiver
R_{PU}
R_{PU}
Z_{DRV}
Z_{DRV}
USB_DP
USB_DM
D+
D–
Z_{DRV}: Output impedance
R_{PU}: Pull-up resistor

Note 1. USB_VBUS is 5 V tolerant.

Note 2. Design the board so that the total VBUS capacitance ranges from 1.0 to 10 μF.

Figure 28.4 Example device connection in bus-powered state 1

Figure 28.5 shows an example of functional connection of the USB connector in bus-powered state 2.

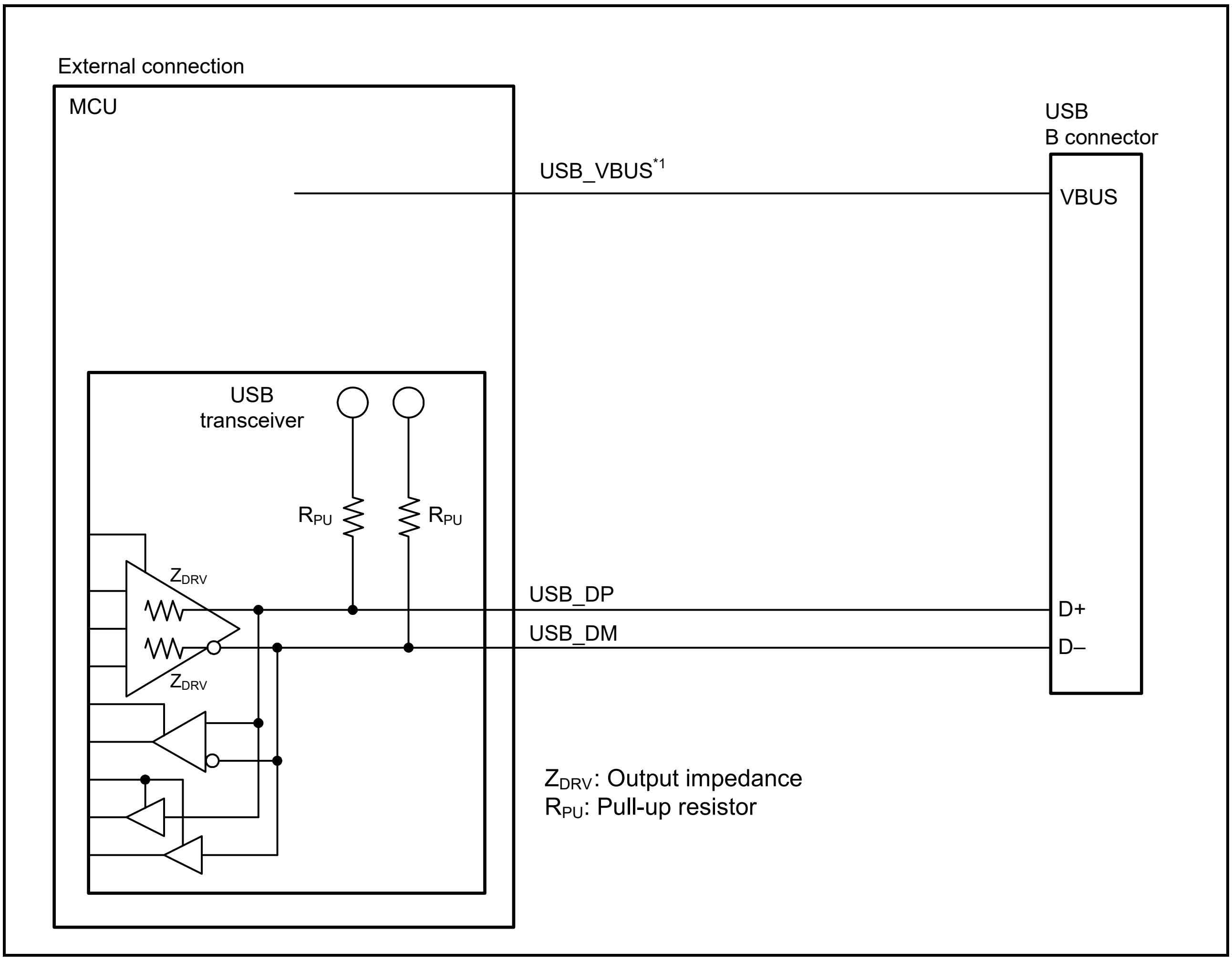
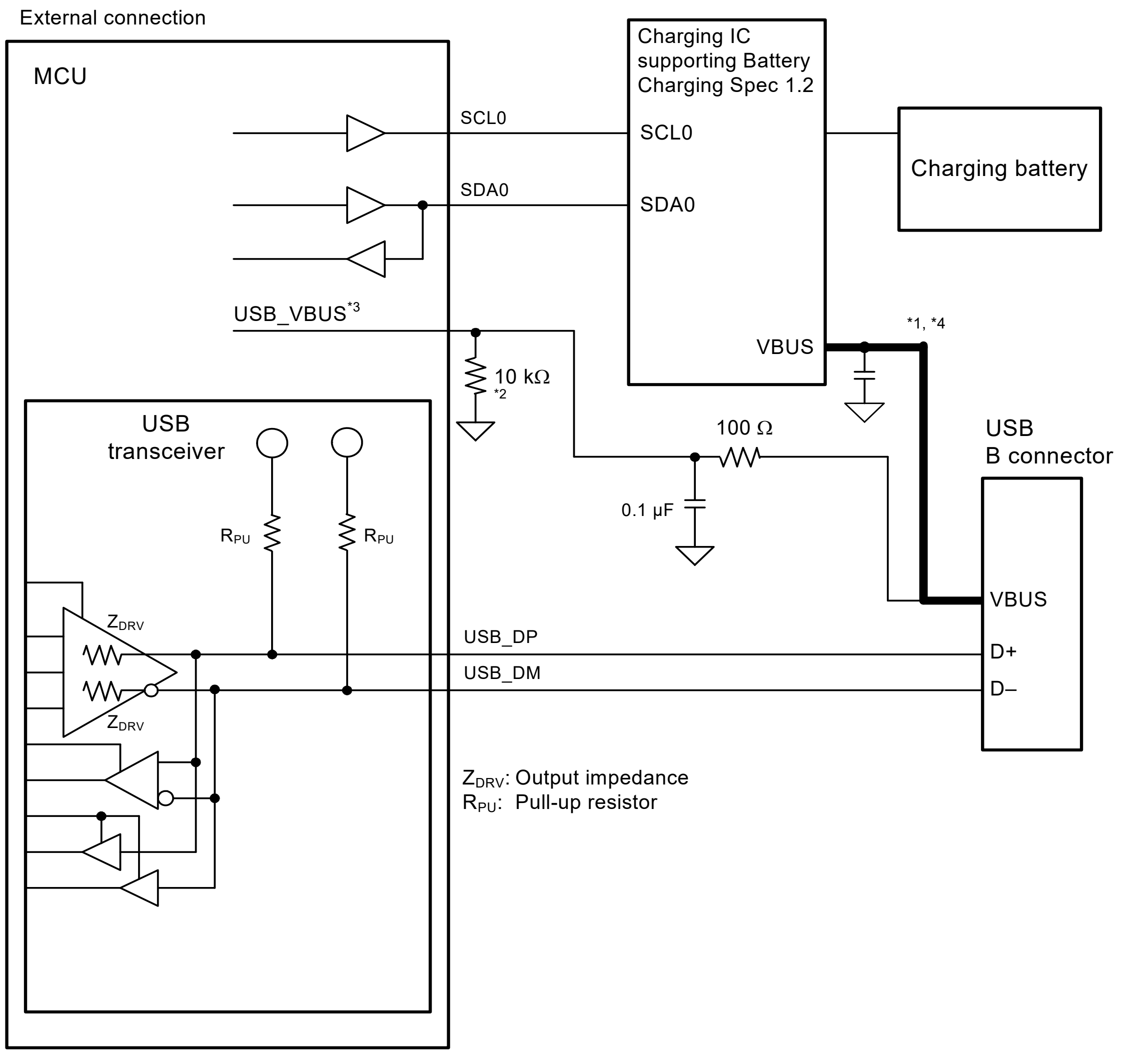

External connection
MCU
USB transceiver
R_PU
R_PU
Z_DRV
Z_DRV
USB_VBUS*1
USB_DP
USB_DM
USB B connector
VBUS
D+
D–
Z_DRV: Output impedance
R_PU: Pull-up resistor

Figure 28.5 Example device connection in bus-powered state 2

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

Figure 28.6 shows an example of functional connection of the USB connector with Battery Charging Rev 1.2 supported.


External connection
MCU
SCL0
SDA0
Charging IC supporting Battery Charging Spec 1.2
SCL0
SDA0
VBUS
Charging battery
USB_VBUS*3
10 kΩ
*2
*1, *4
100 Ω
0.1 μF
USB B connector
VBUS
D+
D−
USB transceiver
RPU
RPU
ZDRV
ZDRV
USB_DP
USB_DM
ZDRV: Output impedance
RPU: Pull-up resistor

Note 1. When Battery Charging Specification Rev 1.2 is to be supported, ensure that the VBUS wiring width is enough for at least 1.5 A (shown in bold lines).
Note 2. Use a resistor value such that the discharging time of VBUS is within 500 ms.
Note 3. USB_VBUS is 5 V tolerant.
Note 4. Design the board so that the total VBUS capacitance ranges from 1.0 to 10 μF.

Figure 28.6 Example of functional connection with Battery Charging Specification Rev 1.2 supported

28.3.2 Interrupt Sources

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 and a USBFS interrupt is generated. See section 14, Interrupt Controller Unit (ICU).

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	—
DVST	Device state transition interrupt	One of the following device state transitions was detected: • USB bus reset detected • Suspended 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 (read or write 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	—

Table 28.13 Interrupt sources (2 of 2)

Bit to be set	Name	Interrupt source	Applicable controller function	Status flag
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 28.7 shows the circuits related to the USBFS interrupts.

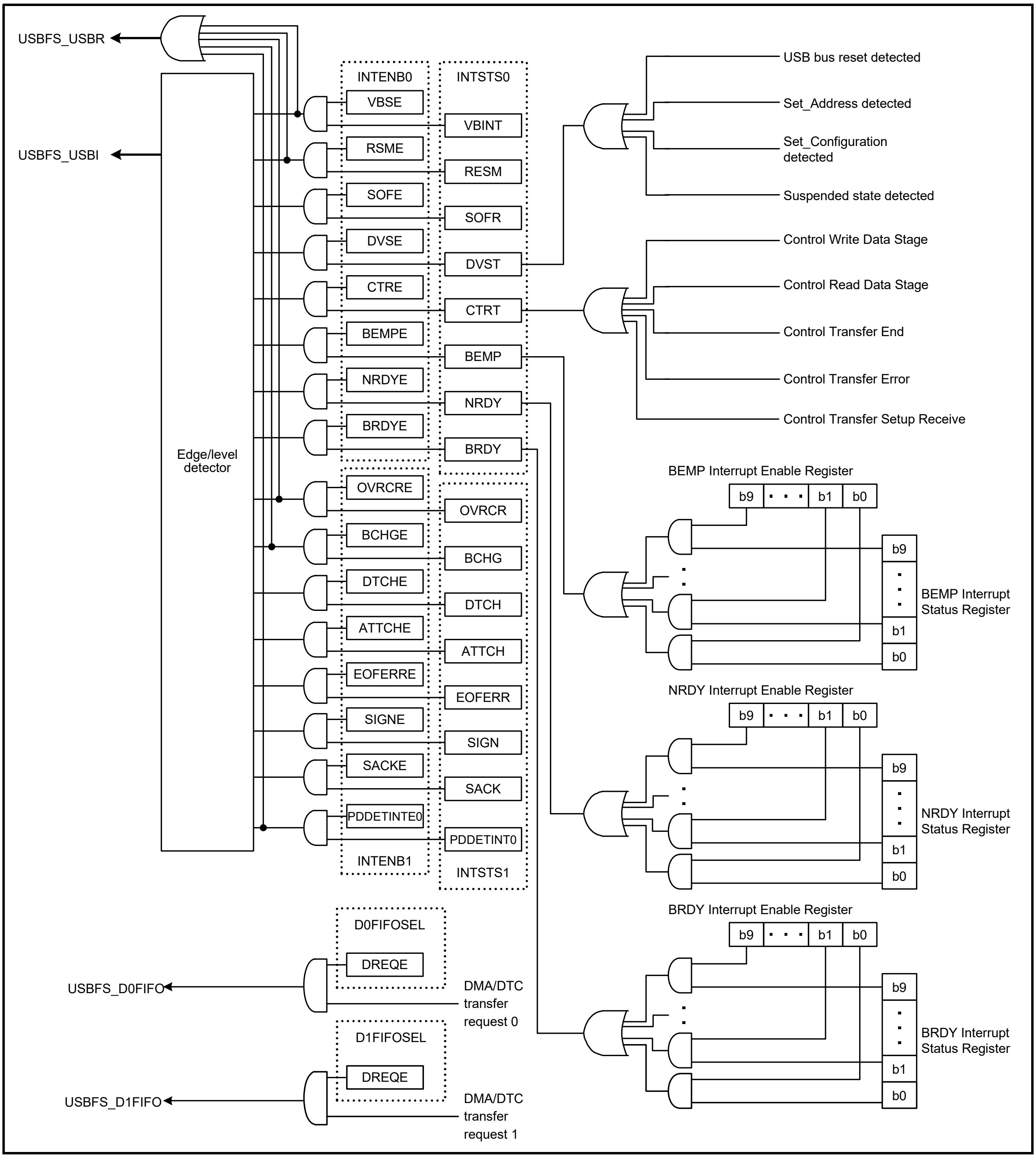

USBFS_USBR
USBFS_USBI
Edge/level detector
INTENB0
VBSE
RSME
SOFE
DVSE
CTRE
BEMPE
NRDYE
BRDYE
INTSTS0
VBINT
RESM
SOFR
DVST
CTRT
BEMP
NRDY
BRDY
OVRCRE
BCHGE
DTCHE
ATTCHE
EOFERRE
SIGNE
SACKE
PDDETINTE0
INTENB1
OVRCR
BCHG
DTCH
ATTCH
EOFERR
SIGN
SACK
PDDETINT0
INTSTS1
USB bus reset detected
Set_Address detected
Set_Configuration detected
Suspended state detected
Control Write Data Stage
Control Read Data Stage
Control Transfer End
Control Transfer Error
Control Transfer Setup Receive
BEMP Interrupt Enable Register
b9 · · · b1 b0
BEMP Interrupt Status Register
NRDY Interrupt Enable Register
NRDY Interrupt Status Register
BRDY Interrupt Enable Register
BRDY Interrupt Status Register
D0FIFOSEL
DREQE
USBFS_D0FIFO
DMA/DTC transfer request 0
D1FIFOSEL
DREQE
USBFS_D1FIFO
DMA/DTC transfer request 1

Figure 28.7 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, 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 software sets 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 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 during data transmission for control transfers.

(b) For receiving pipes

- When packet reception completes successfully therefore, enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the selected 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 completes.

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 software. In this case, write 1 to the PIPEnBRDY bits for the other pipes. 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 selected 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 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 software. In this case, the other PIPEnBRDY bits for the other pipes must 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 required 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

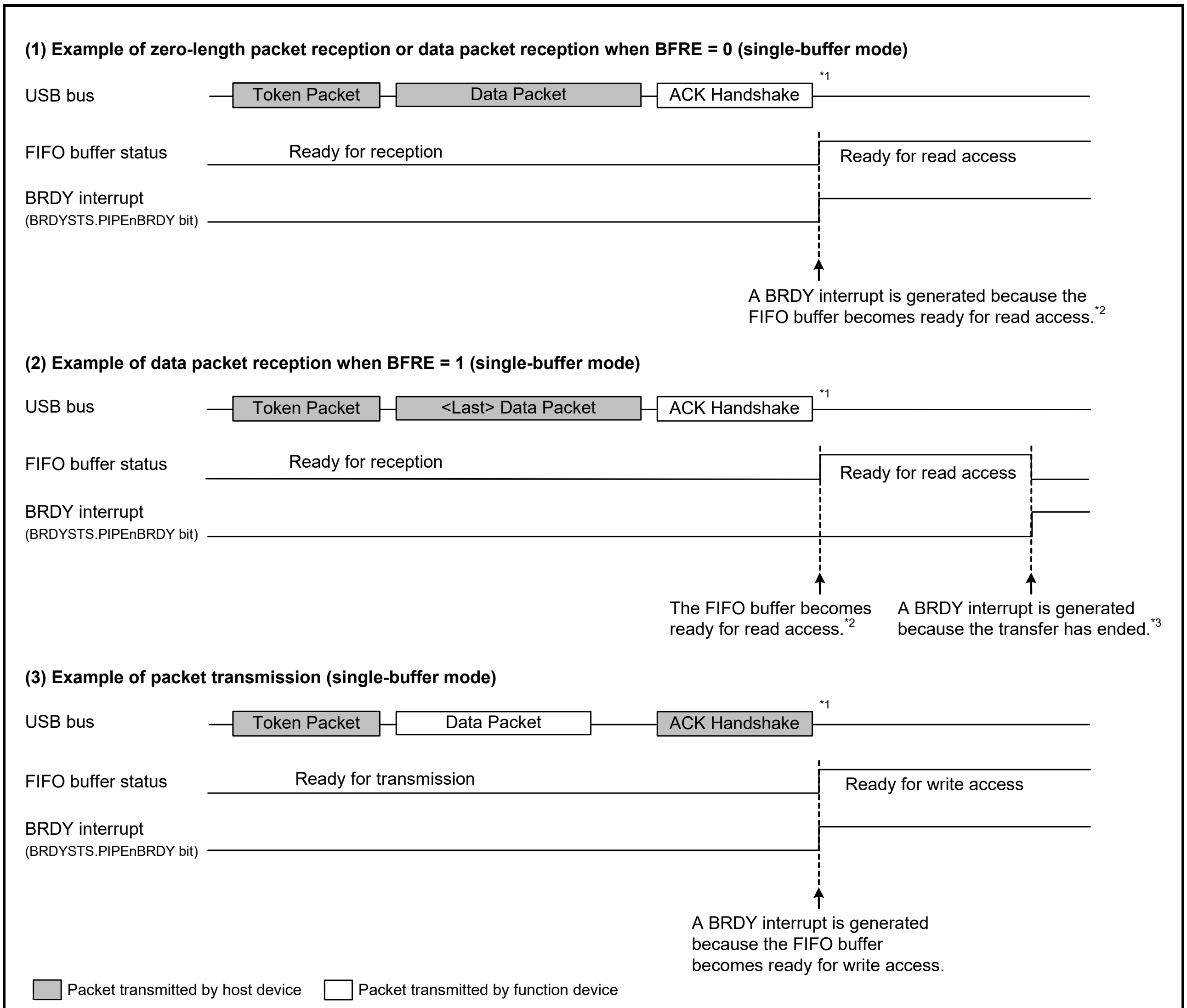
The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready. The BRDY interrupt is not generated for the DCP in the transmitting direction even when it is ready for write access.

(b) For receiving pipes

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data are 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 software writes 1 to BCLR. With this setting, the PIPEnBRDY bit cannot be set to 0 by software. When the SOFCFG.BRDYM bit is set to 1, set the PIPECFG.BFRE bit for all pipes to 0.

Figure 28.8 shows the timing of BRDY interrupt generation.


(1) Example of zero-length packet reception or data packet reception when BFRE = 0 (single-buffer mode)
USB bus
Token Packet
Data Packet
ACK Handshake
*1
FIFO buffer status
Ready for reception
Ready for read access
BRDY interrupt
(BRDYSTS.PIPEnBRDY bit)
A BRDY interrupt is generated because the FIFO buffer becomes ready for read access.*2
(2) Example of data packet reception when BFRE = 1 (single-buffer mode)
USB bus
Token Packet
<Last> Data Packet
ACK Handshake
*1
FIFO buffer status
Ready for reception
Ready for read access
BRDY interrupt
(BRDYSTS.PIPEnBRDY bit)
The FIFO buffer becomes ready for read access.*2
A BRDY interrupt is generated because the transfer has ended.*3
(3) Example of packet transmission (single-buffer mode)
USB bus
Token Packet
Data Packet
ACK Handshake
*1
FIFO buffer status
Ready for transmission
Ready for write access
BRDY interrupt
(BRDYSTS.PIPEnBRDY bit)
A BRDY interrupt is generated because the FIFO buffer becomes ready for write access.
Packet transmitted by host device
Packet transmitted by function device

Note 1. The ACK handshake is not used in isochronous transfers.

Note 2. The FIFO buffer becomes ready for read access under the following condition:
When a packet is received while no data remains unread in the FIFO buffer in the CPU.

Note 3. A transfer ends under either of the following conditions:
(1) When a short packet including a zero-length packet is received
(2) When the number of packets specified in the transaction counter are received

Figure 28.8 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 software, the USBFS sets the associated PIPEnNRDY bit in NRDYSTS to 1. If the associated bit in NRDYENB is set to 1 by 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

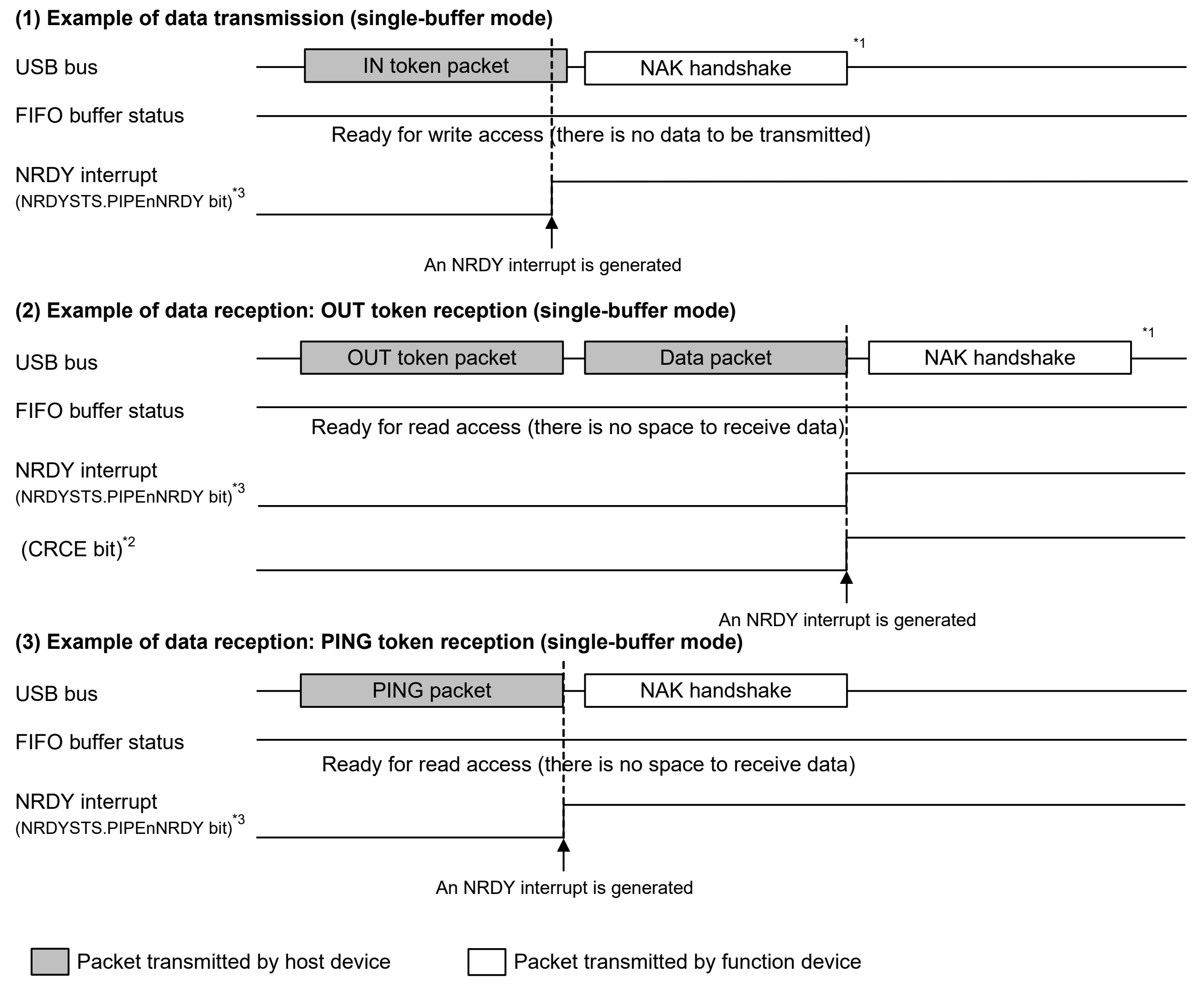
(a) For transmitting pipes

- 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 an NRDY interrupt request when the SOF is received, and sets the PIPEnNRDY bit to 1.

Figure 28.9 shows the timing of NRDY interrupt generation when the device controller is selected.


(1) Example of data transmission (single-buffer mode)
USB bus
IN token packet
NAK handshake
*1
FIFO buffer status
Ready for write access (there is no data to be transmitted)
NRDY interrupt
(NRDYSTS.PIPEnNRDY bit)*3
An NRDY interrupt is generated
(2) Example of data reception: OUT token reception (single-buffer mode)
USB bus
OUT token packet
Data packet
NAK handshake
*1
FIFO buffer status
Ready for read access (there is no space to receive data)
NRDY interrupt
(NRDYSTS.PIPEnNRDY bit)*3
(CRCE bit)*2
An NRDY interrupt is generated
(3) Example of data reception: PING token reception (single-buffer mode)
USB bus
PING packet
NAK handshake
FIFO buffer status
Ready for read access (there is no space to receive data)
NRDY interrupt
(NRDYSTS.PIPEnNRDY bit)*3
An NRDY interrupt is generated
▨ Packet transmitted by host device
☐ Packet transmitted by function device

Note 1. The handshake is not used in isochronous transfers.
Note 2. The CRCE and OVRN bits change only while the target pipe is set to isochronous transfers.
Note 3. The value of the PIPEnNRDY bit changes to 1 only when the PIPEnPID[1:0] bits are set to 01b (BUF response).

Figure 28.9 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 software, the USBFS sets the associated BEMPSTS.PIPEnBEMP bit to 1. If the associated bit in BEMPENB is set to 1 by 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 setting the PIPEnCTR.ACLRM or the BCLR bit in the port control register to 1
- When an IN transfer (zero-length packet transmission) is performed during the control transfer status stage in device

controller mode.

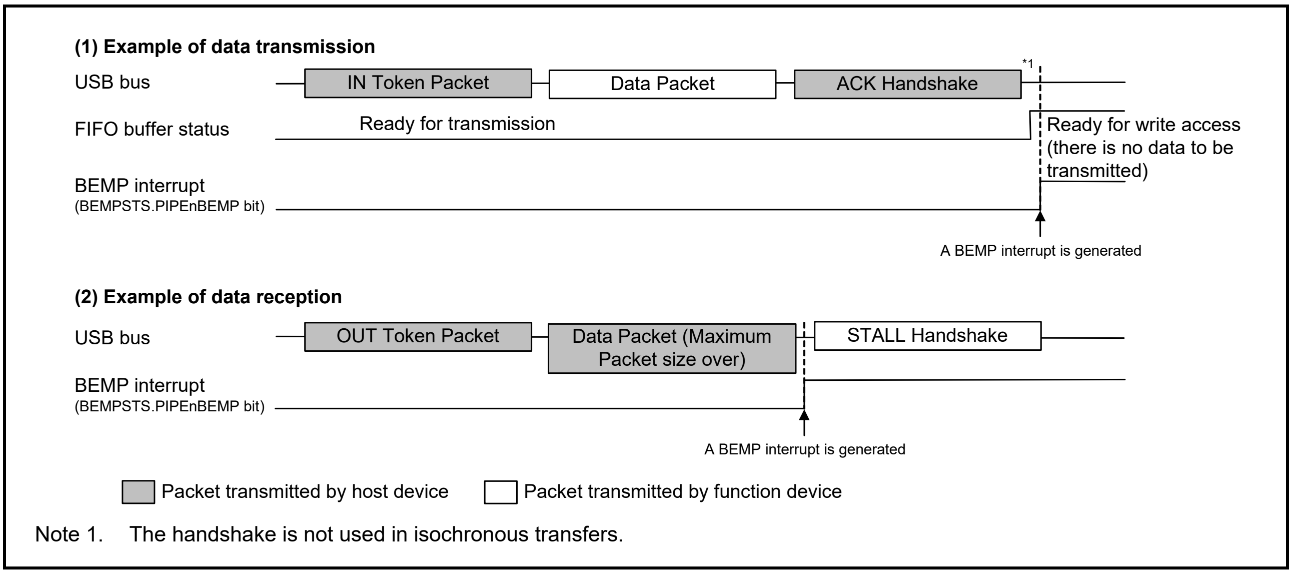
(2) For receiving pipes

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 28.10 shows the timing of BEMP interrupt generation in device controller mode.


(1) Example of data transmission
USB bus
IN Token Packet
Data Packet
ACK Handshake
*1
FIFO buffer status
Ready for transmission
Ready for write access (there is no data to be transmitted)
BEMP interrupt (BEMPSTS.PIPEnBEMP bit)
A BEMP interrupt is generated
(2) Example of data reception
USB bus
OUT Token Packet
Data Packet (Maximum Packet size over)
STALL Handshake
BEMP interrupt (BEMPSTS.PIPEnBEMP bit)
A BEMP interrupt is generated
Packet transmitted by host device
Packet transmitted by function device

Note 1. The handshake is not used in isochronous transfers.

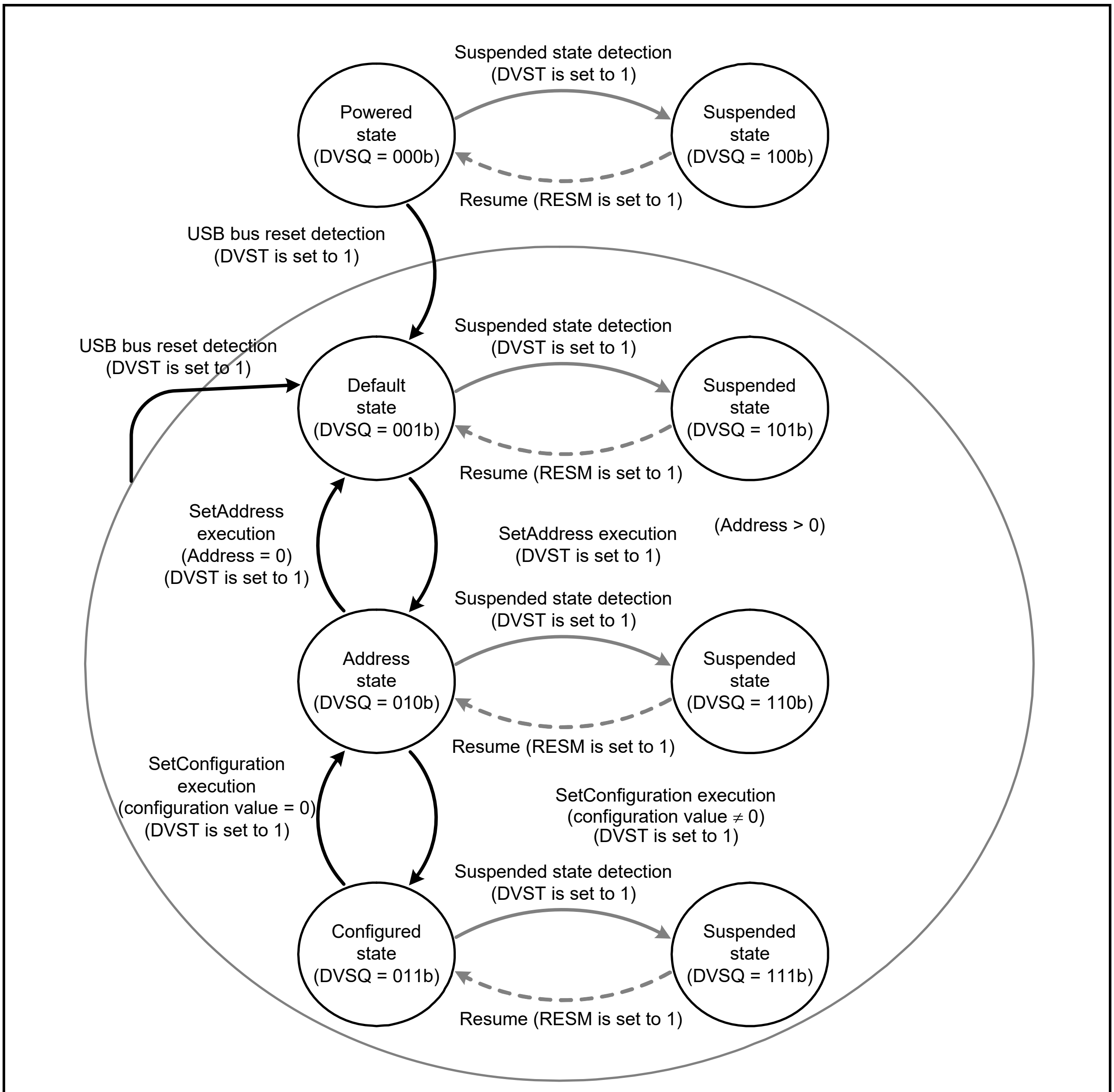
Figure 28.10 Timing of BEMP interrupt generation in device controller mode

28.3.3.4 Device state transition interrupt (device controller mode)

Figure 28.11 shows a diagram of device state transitions in the USBFS. 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 resumed 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.


Powered
state
(DVSQ = 000b)
Suspended state detection
(DVST is set to 1)
Suspended
state
(DVSQ = 100b)
Resume (RESM is set to 1)
USB bus reset detection
(DVST is set to 1)
USB bus reset detection
(DVST is set to 1)
Default
state
(DVSQ = 001b)
Suspended state detection
(DVST is set to 1)
Suspended
state
(DVSQ = 101b)
Resume (RESM is set to 1)
SetAddress
execution
(Address = 0)
(DVST is set to 1)
SetAddress execution
(DVST is set to 1)
(Address > 0)
Address
state
(DVSQ = 010b)
Suspended state detection
(DVST is set to 1)
Suspended
state
(DVSQ = 110b)
Resume (RESM is set to 1)
SetConfiguration
execution
(configuration value = 0)
(DVST is set to 1)
SetConfiguration execution
(configuration value ≠ 0)
(DVST is set to 1)
Configured
state
(DVSQ = 011b)
Suspended state detection
(DVST is set to 1)
Suspended
state
(DVSQ = 111b)
Resume (RESM is set to 1)

Note: For a transition indicated by a solid line, the DVST bit is set to 1.
For a resume indicated by a dashed line, the RESM bit is set to 1.

Figure 28.11 Device state transitions

28.3.3.5 Control transfer stage transition interrupt (device controller mode)

Figure 28.12 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. When 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 cannot be 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 when a new USB request is received. The USBFS saves the setup stage completion status, and generates a CTRT interrupt after software clears the interrupt status.

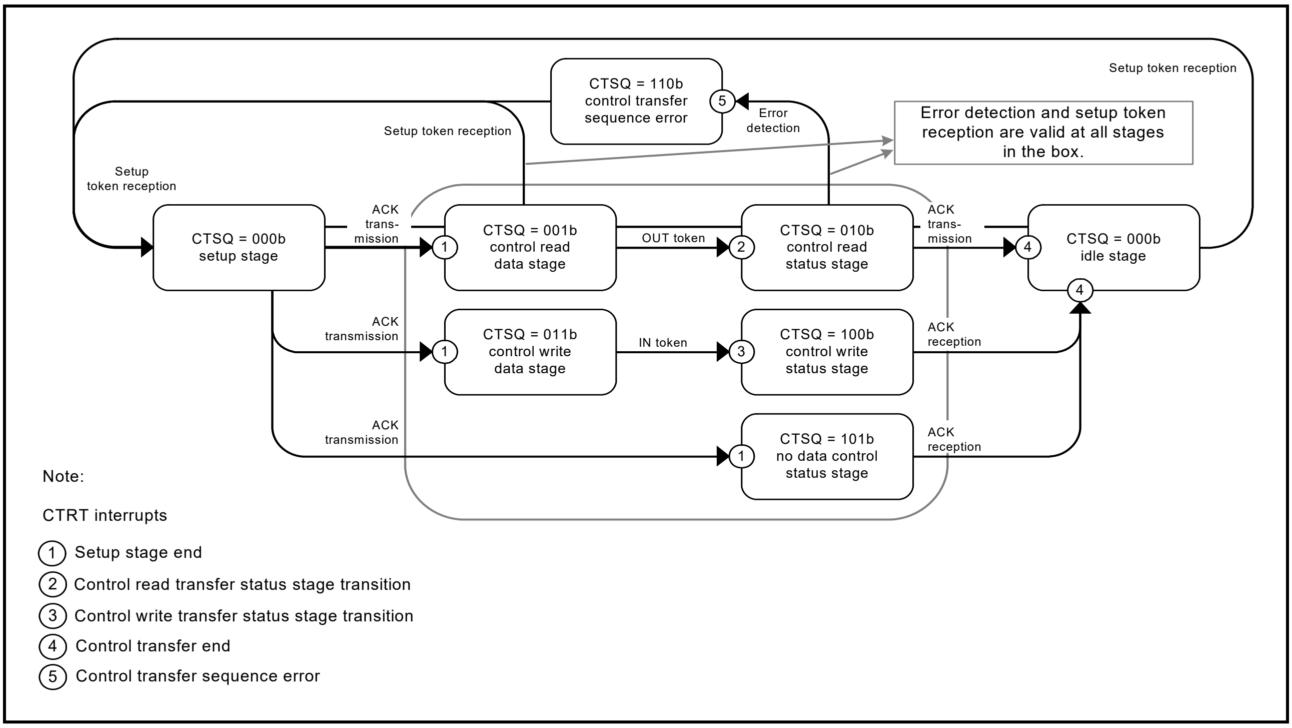

CTSQ = 110b control transfer sequence error
Setup token reception
Error detection
Error detection and setup token reception are valid at all stages in the box.
Setup token reception
Setup token reception
CTSQ = 000b setup stage
ACK trans-mission
CTSQ = 001b control read data stage
OUT token
CTSQ = 010b control read status stage
ACK trans-mission
CTSQ = 000b idle stage
ACK transmission
CTSQ = 011b control write data stage
IN token
CTSQ = 100b control write status stage
ACK reception
ACK transmission
CTSQ = 101b no data control status stage
ACK reception
Note:
CTRT interrupts
① Setup stage end
② Control read transfer status stage transition
③ Control write transfer status stage transition
④ Control transfer end
⑤ Control transfer sequence error

Figure 28.12 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 Suspend state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the Suspend 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.

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 in 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 software. The pipes enter the wait state for a bus connection to the port, waiting for an ATTCH interrupt to occur. 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 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 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 software associates with endpoints. The USBFS has 10 pipes for data transfer. Set up the pipes based on your system specifications.

Table 28.16 Pipe settings

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 when transfer ends	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
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 required (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 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 required (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

The DCPMAXP.MXPS[6:0] and PIPEMAXP.MXPS[8:0] bits specify the maximum packet size for each pipe. 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 to a value 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 ended. Two transaction counters are provided:

- The PIPEnTRN register that specifies the number of transactions to be executed
- 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 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

The PID[1:0] bits in DCPCTR and PIPEnCTR set the response PID for each pipe. 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 to transactions from the host as follows:

- 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] 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 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 software for the stage transitions.

For ClearFeature requests for transmission or reception, the data PID sequence bit must be set by 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 used for the FIFO buffer, this function enables reception of data packets in transfer units. If pipe operation is disabled, software must enable the pipe again (PID response = BUF).

The response PID = NAK function can only be used 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 in 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 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 to 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, 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 the BEMP interrupt does not show the buffer empty status because write access to the FIFO port by the CPU or DMA/DTC is slow, 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 is received. Reading from the FIFO port is allowed. Note: 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 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 to 1 and then to 0 clears the FIFO buffer of the selected pipe regardless of 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 use the BCLR bit to clear the buffer. The length of the received 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
	DTLN	Checks the length of received 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 with the CURPIPE[3:0] bits in the port select register. After the pipe is selected, 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, software checks that the FRDY bit in the Port Control Register is 1.

In addition, software must specify the bus width to be accessed with the MBW bit in the port select register. The FIFO buffer access direction conforms to the PIPECFG.DIR setting. Only in the DCP that 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
Pipe 1 to Pipe 9	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 the first pipe again. Use the REW bit in the port select register 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, 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 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 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 required	No clearing required	No clearing required

28.3.9 Control Transfers Using DCP

In the data stage of control transfers, data is transferred using the Default Control Pipe (DCP). The DCP FIFO buffer is a 64-byte single buffer with a fixed area for both control reads and control writes. The FIFO buffer can only be accessed 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, 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, software must issue setup transactions using this sequence with the assigned USB address set in the DEVSEL[3:0] bits and the bits in DEVADDn associated with 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 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, 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] bits 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 being processed when it receives 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. The USBFS also distinguishes between control read transfers, control write transfers, no-data control transfers, and control stage transitions. For an incorrect sequence, a sequence error occurs in the control transfer stage transition interrupt, and the interrupt is reported to software. For the stage control of the USBFS, see Figure 28.12.

(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, perform 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 the specified settings are made, the USBFS automatically executes the status stage according to the data transfer direction determined at the setup stage. The procedure is 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 software is required.

- 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) can be selected 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, 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.

(1) The USBFS issues interrupt transfer tokens based on this interval.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 Suspended 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 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 error	No interrupts are generated in either host or device controller mode (ignored as a corrupted packet)
2	CRC or bit stuffing error	No interrupts are generated in either host or device controller mode (ignored as a corrupted packet)
3	Overrun or underrun error	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 error	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 error	No interrupts are generated (ignored as a corrupted packet)
2	CRC or bit stuffing error	An NRDY interrupt is generated and the FRMNUM.CRCE bit is set to 1 in both host and device controller modes
3	Maximum packet size exceeded error	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	Transmit buffer flush	Failure to receive an IN token successfully in the interval frame during an isochronous IN transfer
OUT	Notification of no reception of token	Failure to receive an OUT token successfully in the interval frame during an isochronous OUT transfer

The interval count is performed when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is corrupt. The frame interval can be set to 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 software.

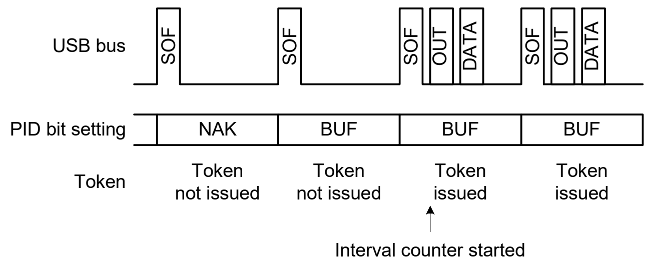


Figure 28.13 Token issuance when IITV = 0

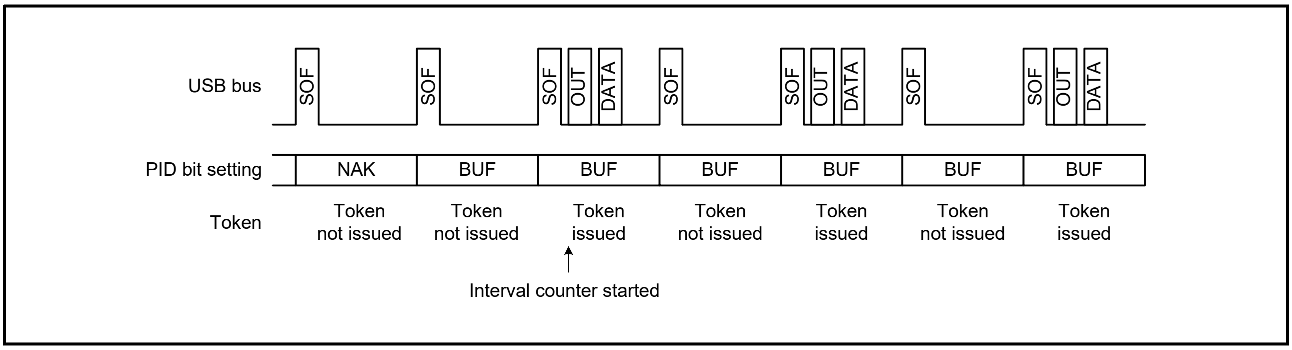


Figure 28.14 Token issuance when IITV = 1

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 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 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 software changes the PID[1:0] bits of the selected pipe to BUF.

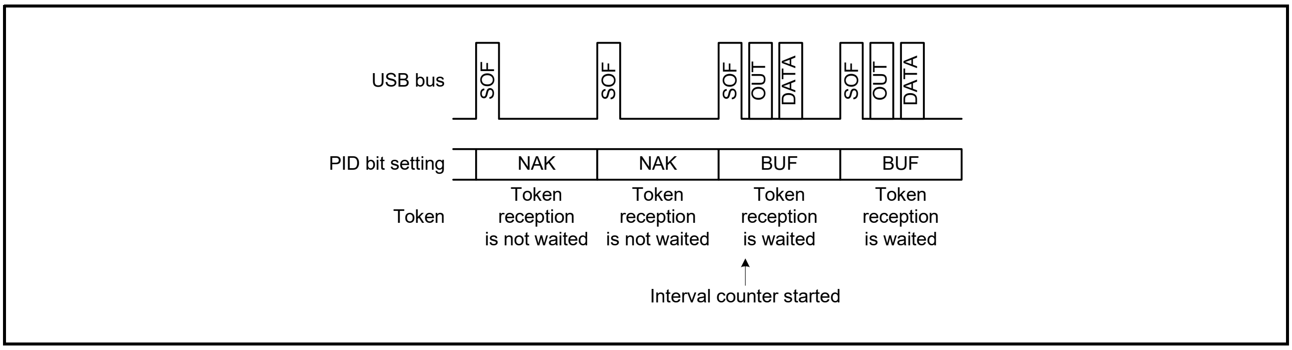

USB bus
SOF
SOF
SOF
OUT
DATA
SOF
OUT
DATA
PID bit setting
NAK
NAK
BUF
BUF
Token
Token reception is not waited
Token reception is not waited
Token reception is waited
Token reception is waited
Interval counter started

Figure 28.15 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.

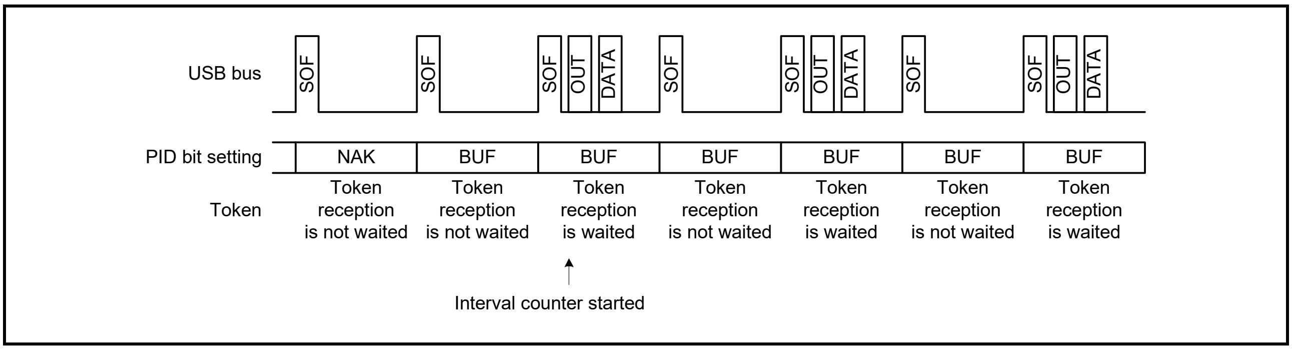

USB bus
SOF
SOF
SOF
OUT
DATA
SOF
SOF
OUT
DATA
SOF
SOF
OUT
DATA
PID bit setting
NAK
BUF
BUF
BUF
BUF
BUF
BUF
Token
Token reception is not waited
Token reception is not waited
Token reception is waited
Token reception is not waited
Token reception is waited
Token reception is not waited
Token reception is waited
Interval counter started

Figure 28.16 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 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 28.17 shows an example transmission using the isochronous transfer transmission data setup function when IITV = 0 (every frame) is set.

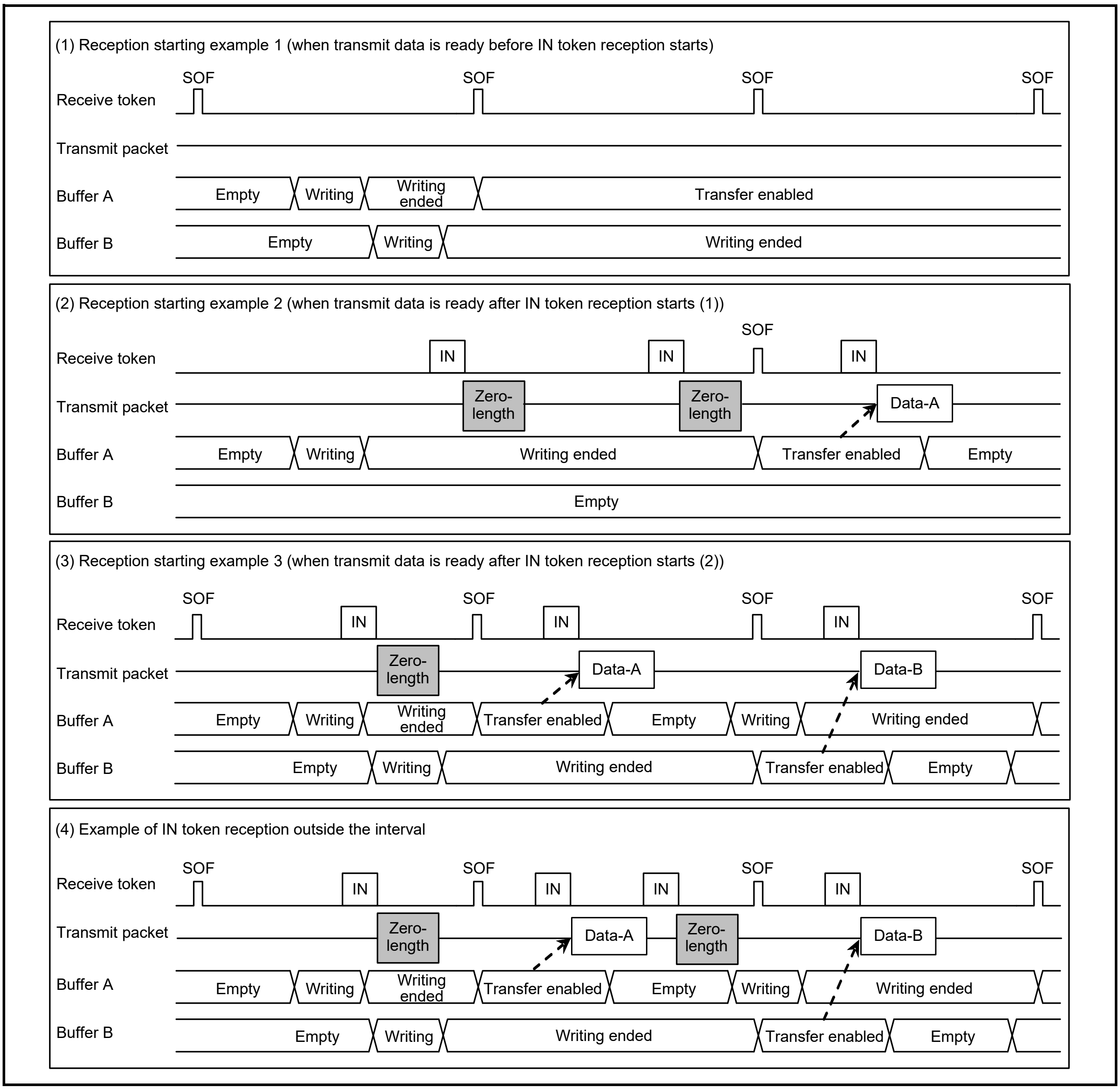

(1) Reception starting example 1 (when transmit data is ready before IN token reception starts)
SOF
Receive token
Transmit packet
Buffer A
Buffer B
Empty
Writing
Writing ended
Transfer enabled
(2) Reception starting example 2 (when transmit data is ready after IN token reception starts (1))
IN
Zero-length
Data-A
(3) Reception starting example 3 (when transmit data is ready after IN token reception starts (2))
Data-B
(4) Example of IN token reception outside the interval

Figure 28.17 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 28.18 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.

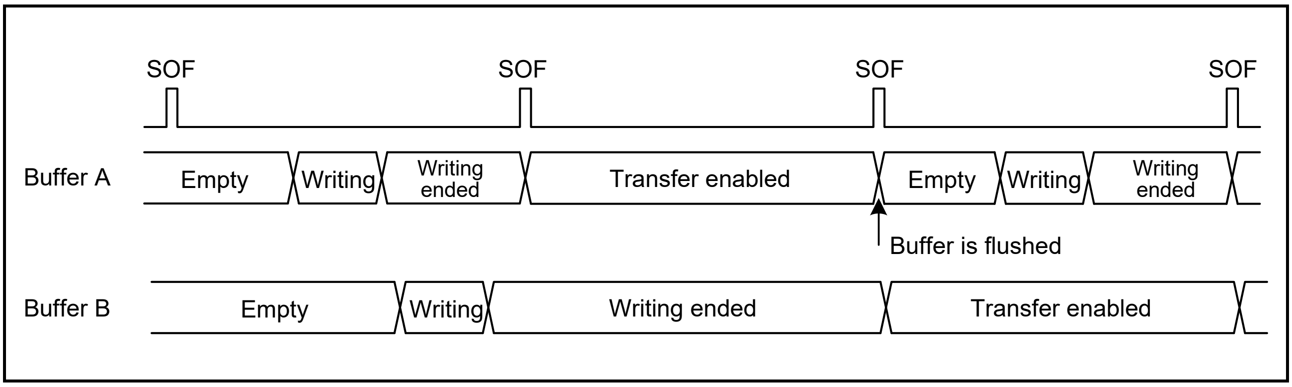

SOF
SOF
SOF
SOF
Buffer A
Empty
Writing
Writing ended
Transfer enabled
Empty
Writing
Writing ended
Buffer is flushed
Buffer B
Empty
Writing
Writing ended
Transfer enabled

Figure 28.18 Example buffer flush operation

Figure 28.19 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.

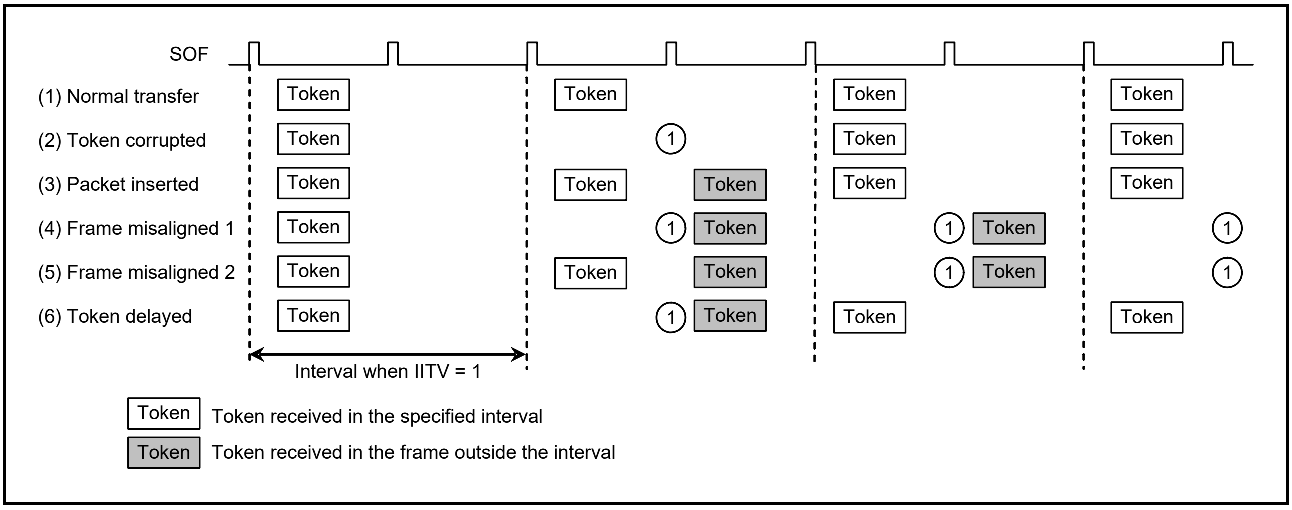

SOF
(1) Normal transfer
(2) Token corrupted
(3) Packet inserted
(4) Frame misaligned 1
(5) Frame misaligned 2
(6) Token delayed
Token
①
Interval when IITV = 1
Token — Token received in the specified interval
Token — Token received in the frame outside the interval

Figure 28.19 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
- Suspended 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 performed 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 if there is a pipe for which a transaction is for a bulk transfer, a control transfer data stage, or a control transfer status stage, the transaction is generated.
   When a transaction is generated, processing moves to the next transaction pipe regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, this step is repeated.

28.3.14.3 Enabling USB communication

Setting the DVSTCTR0.UACT bit to 1 initiates an 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. This 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[1:0] 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 28.20 shows the process flow.

Note 1. The Battery Charging specification describes two implementation methods for data contact detection (D+/D- line contact check). One method 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 USB transceiver in which the CHGDETSTS0 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[1:0] bits, as required.

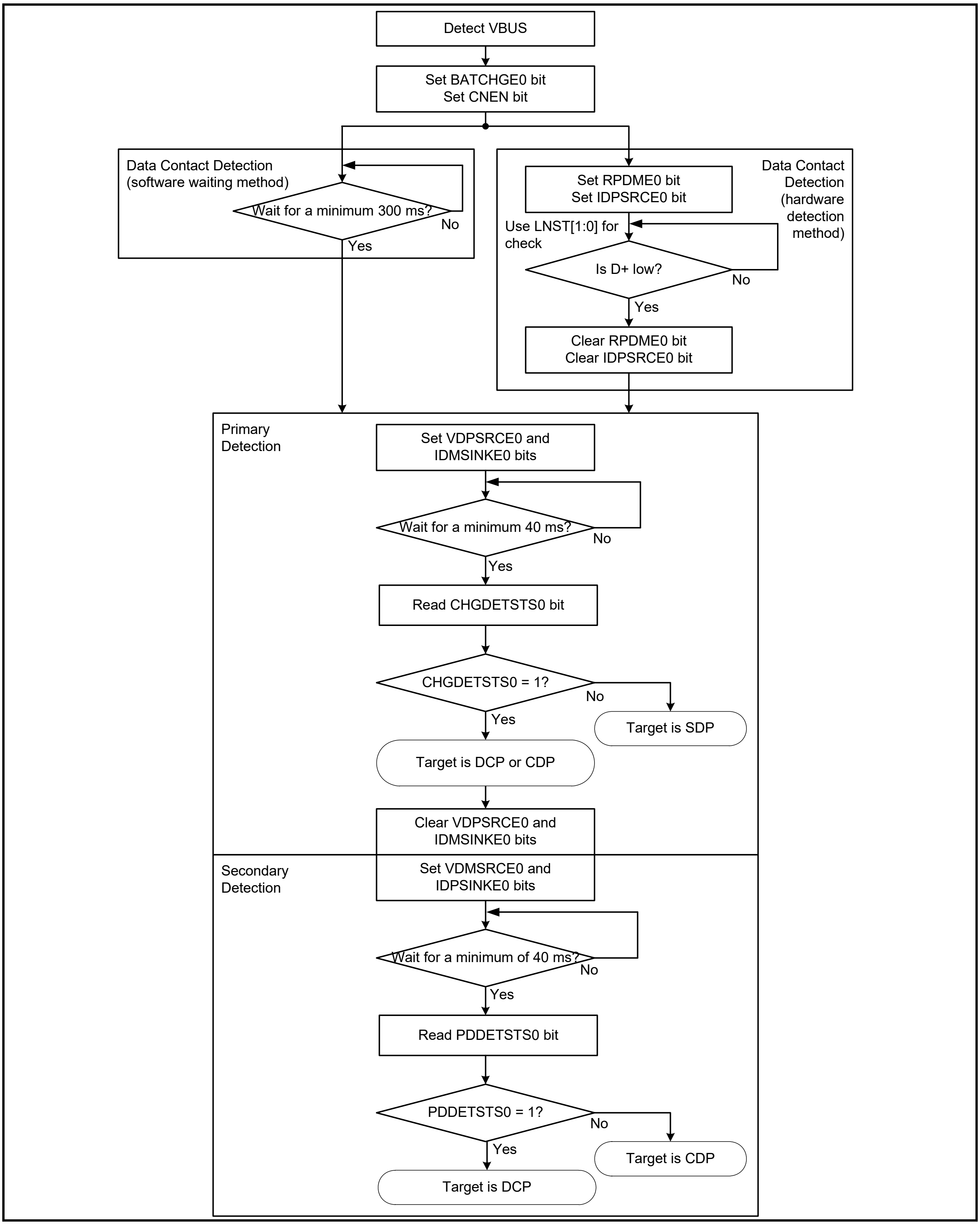

Detect VBUS
Set BATCHGE0 bit
Set CNEN bit
Data Contact Detection
(software waiting method)
Wait for a minimum 300 ms?
No
Yes
Set RPDME0 bit
Set IDPSRCE0 bit
Data Contact Detection (hardware detection method)
Use LNST[1:0] for check
Is D+ low?
No
Yes
Clear RPDME0 bit
Clear IDPSRCE0 bit
Primary Detection
Set VDPSRCE0 and IDMSINKE0 bits
Wait for a minimum 40 ms?
No
Yes
Read CHGDETSTS0 bit
CHGDETSTS0 = 1?
No
Yes
Target is SDP
Target is DCP or CDP
Clear VDPSRCE0 and IDMSINKE0 bits
Secondary Detection
Set VDMSRCE0 and IDPSINKE0 bits
Wait for a minimum of 40 ms?
No
Yes
Read PDDETSTS0 bit
PDDETSTS0 = 1?
No
Yes
Target is CDP
Target is DCP

Figure 28.20 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 28.21 show the process flow for steps 1 to 4 and the process flow for steps a to b, respectively.

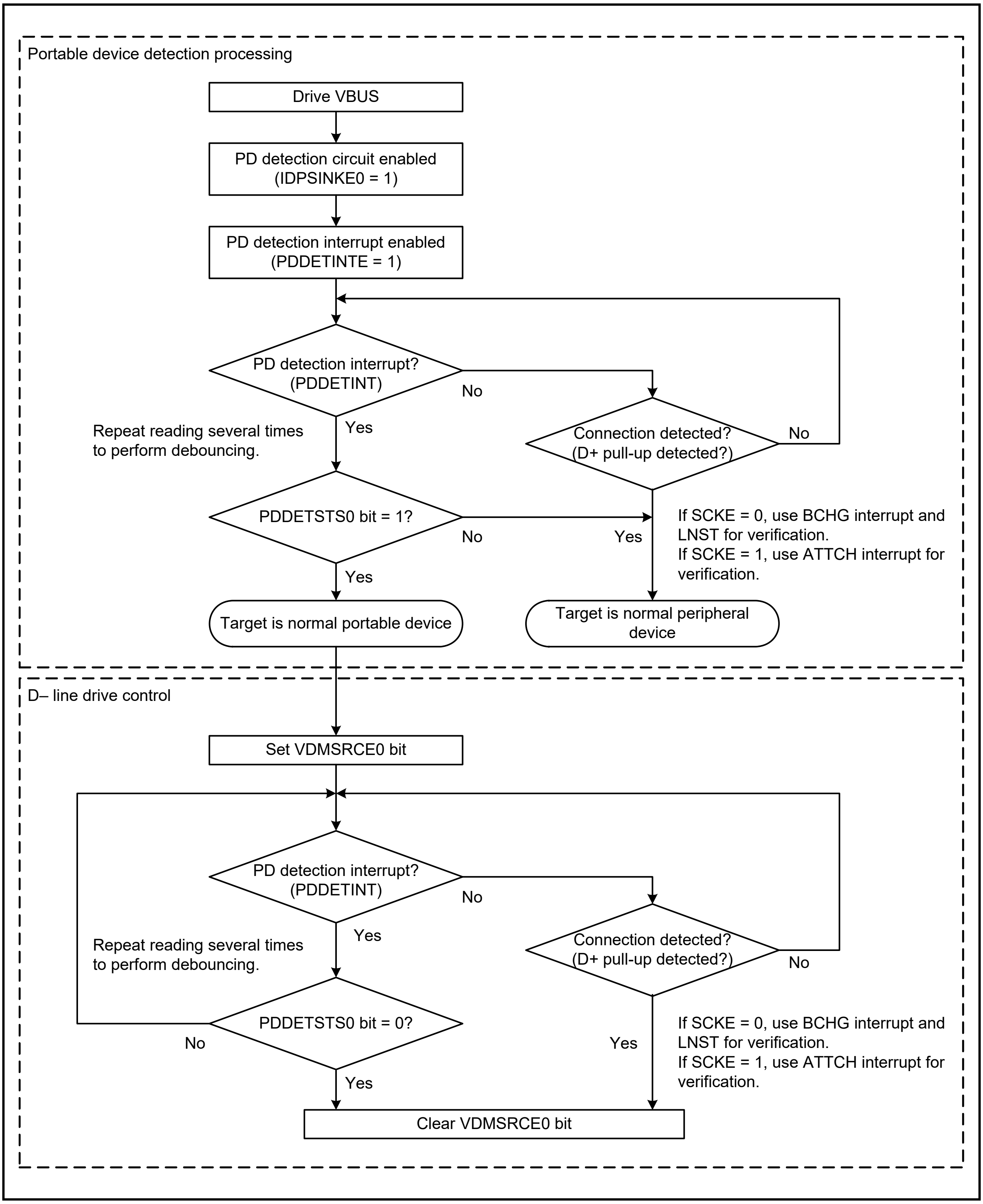

Portable device detection processing
Drive VBUS
PD detection circuit enabled (IDPSINKE0 = 1)
PD detection interrupt enabled (PDDETINTE = 1)
PD detection interrupt? (PDDETINT)
No
Yes
Repeat reading several times to perform debouncing.
Connection detected? (D+ pull-up detected?)
No
PDDETSTS0 bit = 1?
No
Yes
If SCKE = 0, use BCHG interrupt and LNST for verification.
If SCKE = 1, use ATTCH interrupt for verification.
Yes
Target is normal portable device
Target is normal peripheral device
D– line drive control
Set VDMSRCE0 bit
PD detection interrupt? (PDDETINT)
No
Yes
Repeat reading several times to perform debouncing.
Connection detected? (D+ pull-up detected?)
No
PDDETSTS0 bit = 0?
No
Yes
If SCKE = 0, use BCHG interrupt and LNST for verification.
If SCKE = 1, use ATTCH interrupt for verification.
Yes
Clear VDMSRCE0 bit

Figure 28.21 Process flow for operating as charging downstream port (steps 1 to 4)

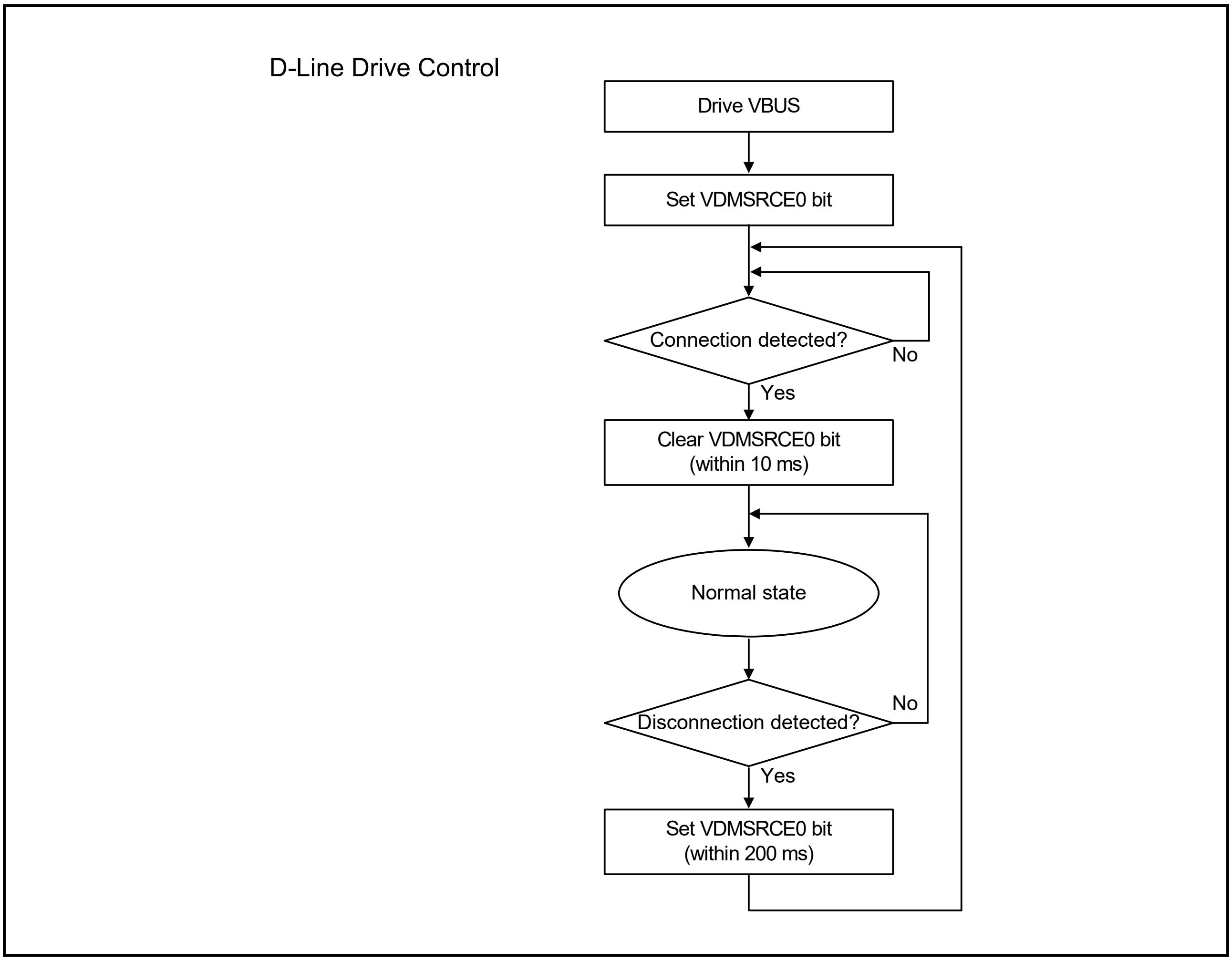

D-Line Drive Control
Drive VBUS
Set VDMSRCE0 bit
Connection detected?
No
Yes
Clear VDMSRCE0 bit (within 10 ms)
Normal state
Disconnection detected?
No
Yes
Set VDMSRCE0 bit (within 200 ms)

Figure 28.22 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

The Module Stop Control Register B (MSTPCRB) can enable or disable USBFS operation. The USBFS is initially stopped after a reset. Releasing the module-stop state enables access to the registers. 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 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 setting the ports 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 ports.

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 a block diagram, and Table 29.2 lists the I/O pins by mode.

Note: In this section, PCLK refers to PCLKA.

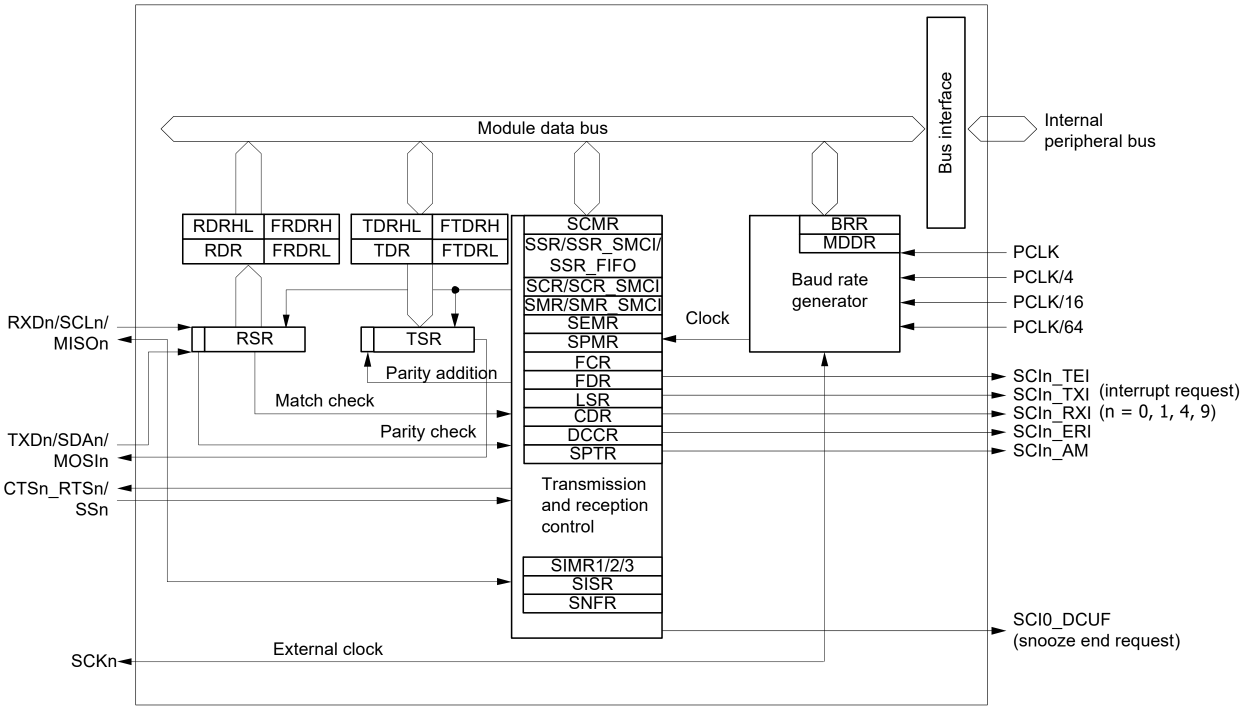
Table 29.1 SCI specifications (1 of 2)

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 structure Receiver: Continuous reception possible using double-buffering structure
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)

Table 29.1 SCI specifications (2 of 2)

Parameter		Description
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	Transmission and reception controllable with CTSn_RTSn pins
	Transmission/Reception	Selectable to 1-stage register or 16-stage FIFO (only SCI0 and SCI1 support FIFO)
	Address match	Interrupt request/event output can be issued on detecting a match between the received data and the value in the compare match register
	Address mismatch (SCI0 only) receive data	Snooze end request can be issued on detecting a mismatch between the received data and the value in the compare match register
	Start-bit detection	Selectable to low level or falling edge detection
	Break detection	Breaks from framing errors detectable by reading from SPTR register
	Clock source	Selectable to internal or external clock
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Multi-processor communications function	Serial communication enabled between multiple processors
	Noise cancellation	Digital noise filters included on the signal paths from RXDn pin inputs
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pins
	Transmission/Reception	Selectable to 1-stage register or 16-stage FIFO
Smart card interface mode	Error processing	Error signal can be automatically transmitted on detecting a parity error during reception
		Data can be automatically retransmitted on receiving an error signal during transmission
	Data type	Both direct and inverse convention are supported
Simple IIC mode	Transfer format	I^{2}C 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	Selectable to internal clock (master mode) or external clock (slave mode)
	SS input pin function	High impedance state can be invoked on the output pins by driving the SSn pin high
	Clock settings	Configurable between four clock phase and clock polarity settings
Bit rate modulation function		Error reduction through correction of outputs from the on-chip baud rate generator
Event link function		Error event output (SCIn_ERI*1) for receive error or error signal detection
		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, 1, 4, 9).
Note 2. Using this event link function is prohibited when the FIFO operation is selected in asynchronous mode.


Module data bus
Bus interface
Internal peripheral bus
RDRHL
FRDRH
RDR
FRDRL
TDRHL
FTDRH
TDR
FTDRL
SCMR
SSR/SSR_SMCI/ SSR_FIFO
SCR/SCR_SMCI
SMR/SMR_SMCI
SEMR
SPMR
FCR
FDR
LSR
CDR
DCCR
SPTR
Transmission and reception control
SIMR1/2/3
SISR
SNFR
BRR
MDDR
Baud rate generator
Clock
PCLK
PCLK/4
PCLK/16
PCLK/64
RXDn/SCLn/ MISOn
RSR
TSR
Parity addition
Match check
Parity check
TXDn/SDAn/ MOSIn
CTSn_RTSn/ SSn
SCIn_TEI
SCIn_TXI
SCIn_RXI
SCIn_ERI
SCIn_AM
(interrupt request) (n = 0, 1, 4, 9)
SCI0_DCUF (snooze end request)
External clock
SCKn

RSR:	Receive Shift Register
RDR:	Receive Data Register
TSR:	Transmit Shift Register
TDR:	Transmit Data Register
SMR/SMR_SMCI:	Serial Mode Register
SCR/SCR_SMCI:	Serial Control Register
SSR/SSR_SMCI/SSR_FIFO*1:	Serial Status Register
SCMR:	Smart Card Mode Register
BRR:	Bit Rate Register
MDDR:	Modulation Duty Register
SEMR:	Serial Extended Mode Register
SPMR:	SPI Mode Register
SNFR:	Noise Filter Setting Register
SIMR1/2/3:	I²C Mode Register 1/2/3
SISR:	I²C Status Register

RDRHL:	Receive 9-bit Data Register
FRDRH/L*1:	Receive FIFO Data Register
TDRHL:	Transmit 9-bit Data Register
FTDRH/L*1:	Transmit FIFO Data Register
FCR*1:	FIFO Control Register
FDR*1:	FIFO Data Count Register
LSR*1:	Line Status Register
CDR:	Compare Match Data Register
DCCR:	Data Compare Match Control Register
SPTR:	Serial Port Register

Note 1. SCI0 and SCI1 only

Figure 29.1 SCI block diagram

Table 29.2 SCI I/O 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 I^{2}C clock input/output SCI0 slave transmit data input/output
	TXD0/SDA0/ MOSI0	Input/Output	SCI0 transmit data output SCI0 I^{2}C 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

Table 29.2 SCI I/O pins (2 of 2)

Channel	Pin name	Input/Output	Function
SCI1	SCK1	Input/Output	SCI1 clock input/output
	RXD1/SCL1/ MISO1	Input/Output	SCI1 receive data input SCI1 I^2C clock input/output SCI1 slave transmit data input/output
	TXD1/SDA1/ MOSI1	Input/Output	SCI1 transmit data output SCI1 I^2C 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
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 the RDR, RDRHL, or the receive FIFO register. 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, SCI4.RDR 4007 0085h, SCI9.RDR 4007 0125h

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

RDR is an 8-bit register that stores receive data. When one frame of serial data is received, it is transferred from RSR to RDR, and the RSR register can receive more data. Because RSR and RDR function as a double buffer, continuous received operations can be performed.

Read the RDR register only once after a receive data full interrupt (SCIn_RXI) occurs.

Note: If the next frame of data is received before reading the received data from the RDR register, an overrun error occurs. The CPU cannot write to the RDR.

29.2.3 Receive 9-bit Data Register (RDRHL)

Address(es): SCI0.RDRHL 4007 0010h, SCI1.RDRHL 4007 0030h, SCI4.RDRHL 4007 0090h, SCI9.RDRHL 4007 0130h

	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

RDRHL is a 16-bit register that stores received data. Use this register when asynchronous mode and 9-bit data length are selected.

The lower 8 bits of RDRHL are the shadow register of RDR, so access to RDRHL affects 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 [15:9] 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

Receive FIFO Data Register L (FRDRL)

Address(es): SCI0.FRDRL 4007 0011h, SCI1.FRDRL 4007 0031h

Receive FIFO Data Register HL (FRDRHL)

Address(es): SCI0.FRDRHL 4007 0010h, SCI1.FRDRHL 4007 0030h

	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, and with FIFO selected	R
b9	MPB	Multi-Processor Bit Flag	Multi-processor bit associated with 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 a normal completion of received data 1: Next receive data is not received for a period after a normal completion of received data.	R*1

Bit	Symbol	Bit name	Description	R/W
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 received data written in FRDRH and FRDRL is below the specified received triggering number 1: The amount of received data written in FRDRH and FRDRL is equal to or greater than the specified received 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 RSR register into FRDRH and FRDRL for storage. Continuous reception is executed until 16 stages are stored. If data is read when there is no received data in FRDRH and FRDRL, the value is undefined. When FRDRH and FRDRL are full, subsequent serial receive data is lost. The CPU can read from 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 order from FRDRH to FRDRL. The FRDRHL register can be accessed in 16-bit units.

29.2.5 Transmit Data Register (TDR)

Address(es): SCI0.TDR 4007 0003h, SCI1.TDR 4007 0023h, 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.

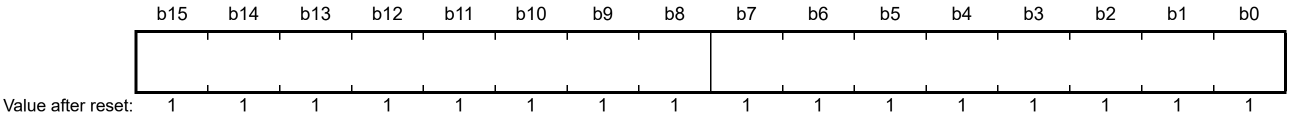
When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission.

The double-buffered structure of TDR and TSR enables continuous serial transmission. If the next transmit data is already written to TDR when one frame of data is transmitted, the SCI transfers the written data to 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, SCI4.TDRHL 4007 008Eh, SCI9.TDRHL 4007 012Eh



TDRHL is a 16-bit register that stores transmit data. Use this register when asynchronous mode and 9-bit data length are selected.

The lower 8 bits of TDRHL are the shadow register of TDR, so access to TDRHL affects 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 starts.

TSR and TDRHL have a double-buffered structure to support continuous transmission. When the next data to be transmitted is stored in TDRHL after one frame of data is transmitted, the transmitting operation continues by transferring the data to TSR.

The CPU can read and write to TDRHL. Bits [15:9] in TDRHL are fixed to 1. These bits are read as 1. The write value should be 1.

Write transmit data to 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)

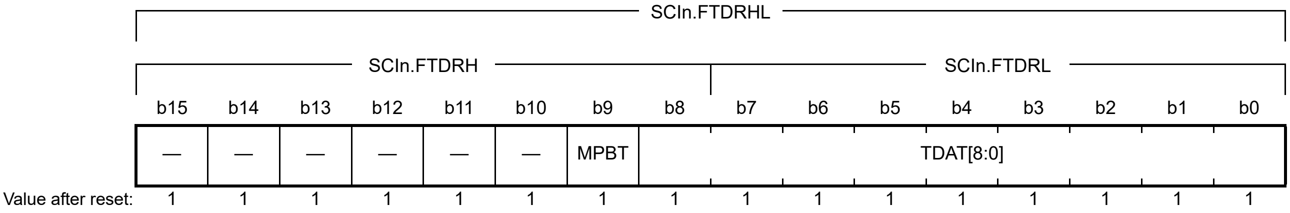
Address(es): SCI0.FTDRH 4007 000Eh, SCI1.FTDRH 4007 002Eh

Transmit FIFO Data Register L (FTDRL)

Address(es): SCI0.FTDRL 4007 000Fh, SCI1.FTDRL 4007 002Fh

Transmit FIFO Data Register HL (FTDRHL)

Address(es): SCI0.FTDRHL 4007 000Eh, SCI1.FTDRHL 4007 002Eh



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, and clock synchronous mode, and 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. This register is valid only in asynchronous mode, including multi-processor mode and clock synchronous mode.

When the SCI detects that the TSR register 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 order from FTDRH to FTDRL.

MPBT flag (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, 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 is not added 1: Parity bit is added. • When receiving: 0: Parity bit is not checked 1: Parity bit is checked.	R/W*4
b6	CHR	Character Length	Valid only in asynchronous mode*2. 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.	R/W*4

Bit	Symbol	Bit name	Description	R/W
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)

The CKS[1:0] bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of these bits and the baud rate, see section 29.2.17, Bit Rate Register (BRR).

MP bit (Multi-Processor Mode)

The MP bit disables or enables the multi-processor communications function. The settings of the PE and PM bits are invalid in multi-processor mode.

STOP bit (Stop Bit Length)

The STOP bit selects the stop bit length in transmission.

In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

PM bit (Parity Mode)

The PM bit selects the parity mode (even or odd) for transmission and reception.

The setting of the PM bit is invalid in multi-processor mode.

PE bit (Parity Enable)

When the PE bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception.

Regardless of the setting of the PE bit, the parity bit is not added or checked in multi-processor format.

CHR bit (Character Length)

The CHR bit selects the data length for transmission and reception in combination with the CHR1 bit in SCMR.

In modes other than asynchronous mode, a fixed data length of 8 bits is used.

CM bit (Communication Mode)

The CM bit selects the communication mode:

- Asynchronous mode or simple IIC mode
- Clock synchronous mode or simple SPI mode.

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, 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 SCR_SMCI.TE = 0 and SCR_SMCI.RE = 0 (both serial transmission and reception are disabled).

The SMR_SMCI register sets the communication format and clock source for the on-chip baud rate generator.

CKS[1:0] bits (Clock Select)

The CKS[1:0] bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of these bits and the baud rate, see section 29.2.17, Bit Rate Register (BRR).

BCP[1:0] bits (Base Clock Pulse)

The BCP[1:0] bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

Set these bits in combination with the SCMR.BCP2 bit.

For details, see section 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 (1 of 2)

SCMR.BCP2 bit	SMR_SMCI.BCP[1:0] bits	Number of base clock cycles for 1-bit transfer period
0	00	93 clock cycles (S = 93)*1
0	01	128 clock cycles (S = 128)*1
0	10	186 clock cycles (S = 186)*1
0	11	512 clock cycles (S = 512)*1
1	00	32 clock cycles (S = 32)*1 (initial value)
1	01	64 clock cycles (S = 64)*1

Table 29.3 Combinations of SCMR.BCP2 bit and SMR_SMCI.BCP[1:0] bits (2 of 2)

SCMR.BCP2 bit	SMR_SMCI.BCP[1:0] bits	Number of base clock cycles for 1-bit transfer period
1	10	372 clock cycles (S = 372)*1
1	11	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)

The PM bit selects the parity mode for transmission and reception (even or odd).

For details on the usage of this bit in smart card interface mode, see section 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 the BLK bit to 1 enables block transfer mode operation.

For details, see section 29.6.3, Block Transfer Mode.

GM bit (GSM Mode)

Setting the GM bit to 1 enables GSM mode operation.

In GSM mode, the SSR_SMCI.TEND flag set timing is moved forward to 11.0 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, 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

Bit	Symbol	Bit name	Description	R/W
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 in SSR 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 new value to a bit other than the MPIE bit of this register during multi-processor mode (SMR.MP bit = 1), write 0 to 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.

The SCR register controls operation and the clock source selection for transmission and reception.

CKE[1:0] bits (Clock Enable)

The CKE[1:0] 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. Set the TEIE bit to 0 to disable the 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 Communications Function.

When the receive data that includes the MPB bit 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 that 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.

Set MPIE to 0 if the multi-processor communications function is not used.

RE bit (Receive Enable)

The RE bit enables or disables serial reception.

When the RE bit is set to 1, serial reception starts by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Set the reception format in the SMR before setting the RE bit to 1.

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 values are 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 values are saved.

TE bit (Transmit Enable)

The TE bit enables or disables serial transmission.

When the TE bit is set to 1, serial transmission starts by writing transmit data to TDR. Set the transmission format in the SMR register before setting the TE bit to 1.

RIE bit (Receive Interrupt Enable)

The RIE bit enables or disables SCIn_RXI and SCIn_ERI interrupt requests.

Setting the RIE bit to 0 disables SCIn_RXI and SCIn_ERI interrupt requests.

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)

The TIE bit enables or disables SCIn_TXI interrupt request.

Setting the TIE bit to 0 disables an SCIn_TXI interrupt request. TIE should 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 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, 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

Bit	Symbol	Bit name	Description	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)

The CKE[1:0] bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, see section 29.6.8, Clock Output Control.

TEIE bit (Transmit End Interrupt Enable)

Set the TEIE bit to 0 in smart card interface mode.

RE bit (Receive Enable)

The RE bit enables or disables serial reception.

When this bit is set to 1, serial reception starts by detecting the start bit. Set the reception format in the SMR_SMCI register before setting the RE bit to 1.

When reception is halted by setting the RE bit to 0, the ORER, FER, and PER flags in SSR_SMCI are not affected and the previous values are saved.

TE bit (Transmit Enable)

The TE bit enables or disables serial transmission.

When this bit is set to 1, serial transmission starts by writing transmit data to TDR. Set the transmission format in the SMR_SMCI register before setting the TE bit to 1.

RIE bit (Receive Interrupt Enable)

The RIE bit enables or disables SCIn_RXI and SCIn_ERI interrupt requests.

Setting the RIE bit to 0 disables SCIn_RXI and SCIn_ERI interrupt requests.

To cancel an SCIn_ERI interrupt request, read 1 from the ORER, FER, or PER flag in SSR_SMCI, and then set the flag to 0, or set the RIE bit to 0.

TIE bit (Transmit Interrupt Enable)

The TIE bit enables or disables an SCIn_TXI interrupt request.

Setting the TIE bit to 0 disables an SCIn_TXI interrupt request. TIE should 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 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, 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 in 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: Parity error occurred.	R/(W)*1
b4	FER	Framing Error Flag	0: No framing error occurred 1: Framing error occurred.	R/(W)*1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: 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 flags and transmission and reception multi-processor bits.

MPBT bit (Multi-Processor Bit Transfer)

The MPBT bit selects the multi-processor bit in the transmit frame.

MPB bit (Multi-Processor)

The MPB bit holds the value of the multi-processor bit in the reception frame. This bit does not change when the SCR.RE bit is 0.

TEND flag (Transmit End Flag)

The TEND flag indicates completion of transmission.

[Setting conditions]

- When the SCR.TE bit is set to 0 to disable serial transmission 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 keeps 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)

The PER flag indicates that a parity error occurs 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 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 PER, read the PER bit to check that it was actually set to 0.

When the SCR.RE is set to 0 to disable serial reception, the PER flag is not affected and keeps its previous value.

FER flag (Framing Error Flag)

The FER 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 mode, only the first stop bit is checked but the second stop bit is not checked. Although received data is transferred to RDR when the framing error occurs, no SCIn_RXI interrupt request occurs. Also, when the FER flag is set to 1, the subsequent received 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 its value is 0.

When the SCR.RE bit is set to 0, the FER flag is not affected and keeps its previous value.

ORER flag (Overrun Error Flag)

The ORER flag indicates that an overrun error occurs during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before the receive data that does not have a parity error and a framing error is read from RDR.

  In RDR, data received prior to an overrun error occurrence is kept, 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 keeps its previous value.

RDRF flag (Receive Data Full Flag)

The RDRF flag indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing conditions]

- When RDRF is set to 0 after 1 is read

- When data is read from the RDR register.

Note: Do not clear RDRF flag by accessing RDRF bit in SSR register unless communication is aborted.

TDRE flag (Transmit Data Empty Flag)

The TDRE flag indicates the presence of transmit data in the TDR register.

[Setting conditions]

- When the SCR.TE bit is 0
- When data is transmitted from the TDR register to the TSR register.

[Clearing conditions]

- When TDRE is set to 0 after 1 is read
- When SCR.TE is 1, data is written to the TDR register.

Note: Do not clear 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

	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 a normal completion of received data (receive FIFO is empty) 1: The next received data is not received for a period after a normal completion of received data, and when the amount of data stored in the FIFO is equal to or less than the received 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 this bit to clear the flag after reading 1.

The SSR_FIFO register provides the SCI with FIFO mode status flags.

DR flag (Receive Data Ready Flag)

The DR flag indicates that the amount of data stored in the Receive FIFO Data Register (FRDRHL) falls below the specified receive triggering number, and that no subsequent data is received after the elapse of 15 ETUs (Elementary Time Units) from the last stop bit in asynchronous mode. This flag is valid only in asynchronous mode, including multi-processor mode, and when FIFO is selected.

In clock synchronous mode, this flag is not set to 1.

[Setting condition]

- When FRDRHL contains less data than the specified receive triggering number, and no subsequent data is received after the elapse of 15 ETUs*1 from the last stop bit, and the SSR_FIFO.FER and SSR_FIFO.PER flags are 0.

[Clearing conditions]

- When 1 is read from DR, and after all the received data are read
- When the FCR.FM bit changes from 0 to 1.

Note 1. This is equivalent to 1.5 frames in the 8-bit format with one stop bit.

TEND flag (Transmit End Flag)

The TEND flag indicates that FTDRHL does not contain valid data when transmitting the last bit of a serial character, so 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 changes from 0 to 1.

PER flag (Parity Error Flag)

The PER flag indicates whether there is a parity error in the data read from the FRDRHL register in asynchronous mode when the address match function is disabled (DCCR.DCME = 0).

[Setting condition]

- When data is received and a parity error is detected, 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 while data is received.

When the SCR.RE bit is set to 0, the PER flag is not affected and the previous state is kept.

FER flag (Framing Error Flag)

The FER flag indicates whether there is a framing error in the data read from the FRDRHL register in asynchronous mode when the address match function is disabled (DCCR.DCME = 0).

[Setting condition]

- When 0 is sampled as the stop bit during reception 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, and the receive data is stored to the FRDRHL register even when a framing error occurs while data is received.

When the SCR.RE bit is set to 0, the FER flag is not affected and the previous state is kept.

ORER flag (Overrun Error Flag)

The ORER flag indicates that the receive operation stopped abnormally because an overrun error occurred.

[Setting condition]

- When the next serial reception completes while the receive FIFO is full with 16-byte receive data.

[Clearing condition]

- When 0 is written after 1 is read from ORER.

When the SCR.RE bit is set to 0, the ORER flag is not affected and the previous state is kept.

RDF flag (Receive FIFO Data Full Flag)

The RDF flag indicates that receive data is transferred to the FRDRHL register, and the amount of data in FRDRHL is equal to or exceeds the specified receive triggering number. When RTRG is set to 0, the RDF flag is not set even when the amount of data in the receive FIFO is equal to 0.

[Setting condition]

- When the amount of receive data equal to or greater than the specified receive triggering number is stored in FRDRHL,*1 and the FIFO is not empty.

[Clearing conditions]

- When 0 is written after 1 is read from RDF
- When FRDRHL is read by the DMAC or DTC but only when block transfer is the last transmission
- When the setting condition and clearing condition occur at the same time. 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: Do not clear RDF flags by accessing RDF bit in the SSR register before reading receive data unless communication is aborted.

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.

TDFE flag (Transmit FIFO Data Empty Flag)

The TDFE flag indicates that when data is transferred from FTDRHL into TSR, the amount of data in FTDRHL has fallen below the specified transmit triggering number, and writing of transmit data to FTDRHL is enabled.

[Setting conditions]

- When the TE bit in SCR is 0
- When the amount of transmit data written in FTDRHL is equal to or less than the specified transmit triggering number.*1

[Clearing conditions]

- When writing to FTDRHL is executed on the last transmission while the DTC or DMAC is activated
- When 0 is written to the TDFE flag after reading TDFE = 1.
  When the setting condition and the clearing condition occur at the same time, the TDFE flag is 0. Thereafter, 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: Do not clear TDFE flags by accessing TDFE bit in the SSR register before writing transmit data unless communication is aborted.

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 set to 1 is 16 minus FDR.T[4:0]. If more data is written, data is discarded.

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, 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	Set this bit to 0 in smart card interface mode	R/W
b1	MPB	Multi-Processor	Set this bit to 0 in smart card interface mode	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	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, the TEND flag is set to 1 when more data is ready to be transferred to the TDR register.

[Setting conditions]

- When the SCR_SMCI.TE bit = 0 to disable serial transmission. When the SCR_SMCI.TE bit changes from 0 to 1, the TEND flag is not affected and keeps 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 while the SCR_SMCI.TE bit is 1.

PER flag (Parity Error Flag)

The PER flag indicates that a parity error occurs during reception in asynchronous mode and the reception ends 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 the PER bit after reading PER = 1. After writing 0 to the PER bit, read the PER bit to check that it is actually set to 0.

When the RE bit in SCR_SMCI is set to 0 to disable serial reception, 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)

The ORER flag indicates that an overrun error occurs 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 is read from the RDR register. In RDR, the data received before an overrun error occurred is saved, but data received after the overrun error is lost. When the ORER flag is set to 1, the 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 flag, read it to verify that its value is 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)

The RDRF 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 0 is written to RDRF after 1 is read
- When data is read from the RDR register.

Note: Do not clear RDRF flags by accessing RDRF bit in the SSR register unless communication is aborted.

TDRE flag (Transmit Data Empty Flag)

The TDRE flag indicates the presence of transmit data in the TDR register.

[Setting conditions]

- When the SCR_SMCI.TE bit is 0
- When data is transmitted from the TDR register to the TSR register.

[Clearing conditions]

- When 0 is written to the TDRE after 1 is read
- When the SCR_SMCI.TE bit is 1 and data is forwarded to the TDR register.

Note: Do not clear TDRE flags by accessing 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, 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 is. Receive data is stored as received in the RDR 1: TDR contents are inverted before 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 only 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 the smart card interface mode. Setting it to 0 selects all the other modes as follows:

- Asynchronous mode, including multi-processor mode
- Clock synchronous mode
- Simple SPI mode

- Simple IIC mode.

SINV bit (Transmitted/Received Data Invert)

The SINV bit inverts the transmit/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)

The CHR1 bit 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)

The BCP2 bit selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR_SMCI.BCP[1:0] bits.

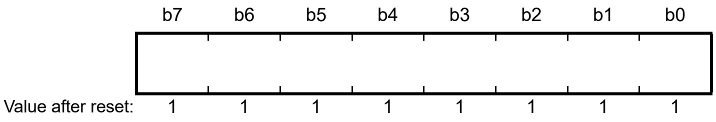
Table 29.4 Combinations of SCMR.BCP2 bit and SMR_SMCI.BCP[1:0] bits

SCMR.BCP2 bit	SMR_SMCI.BCP[1:0] bits	Number of base clock cycles for 1-bit transfer period
0	00	93 clock cycles (S = 93)*1
0	01	128 clock cycles (S = 128)*1
0	10	186 clock cycles (S = 186)*1
0	11	512 clock cycles (S = 512)*1
1	00	32 clock cycles (S = 32)*1 (initial value)
1	01	64 clock cycles (S = 64)*1
1	10	372 clock cycles (S = 372)*1
1	11	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, SCI4.BRR 4007 0081h, SCI9.BRR 4007 0121h



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) and the bit rate (B) in the BRR 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 can only be written to 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		
Asynchronous, multi-processor transfer	0	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N + 1)} - 1 \} \times 100$
	1	0	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N + 1)} - 1 \} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N + 1)} - 1 \} \times 100$
	Don't care	Don't care	1	$N = \frac{PCLK \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \{ \frac{PCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (N + 1)} - 1 \} \times 100$
Clock synchronous, simple SPI				$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface				$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$\text{Error (\%)} = \{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N + 1)} -1 \} \times 100$
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 widths at high and low level of the SCLn output in simple IIC mode satisfy the I²C 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] 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 (1 of 2)

SCMR.BCP2 bit	SMR_SMCI.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

Table 29.8 Base clock settings in smart card interface mode (2 of 2)

SCMR.BCP2 bit	SMR_SMCI.BCP[1:0] bits	Base clock cycles for 1-bit period	S
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 the examples of BRR (N) settings in smart card interface mode.

Table 29.17 lists the 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 (ABCS) bit or the Baud Rate Generator Double-Speed Mode Select (BGDM) bit in the Serial Extended Mode Register (SEMR) is set to 1 in asynchronous mode, the bit rate becomes twice the value listed in Table 29.16. When both of those bits 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
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 and BGDM are set to 1, the bit rate quadruples.

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 bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS and BGDM are set to 1, the bit rate quadruples.

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	250000	17.2032	0	0	0	0	0	537600
		1	0	0	0	500000			1	0	0	0	1075200
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1000000			1	0	0	0	2150400
	Don’t care	Don’t care	1	0	0	1333333		Don’t care	Don’t care	1	0	0	2867200
9.8304	0	0	0	0	0	307200	18	0	0	0	0	0	562500
		1	0	0	0	614400			1	0	0	0	1125000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1228800			1	0	0	0	2250000
	Don’t care	Don’t care	1	0	0	1638400		Don’t care	Don’t care	1	0	0	3000000
10	0	0	0	0	0	312500	19.6608	0	0	0	0	0	614400
		1	0	0	0	625000			1	0	0	0	1228800
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1250000			1	0	0	0	2457600
	Don’t care	Don’t care	1	0	0	1666666		Don’t care	Don’t care	1	0	0	3276800

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	
12	0	0	0	0	0	375000	20	0	0	0	0	0	625000
		1	0	0	0	750000			1	0	0	0	1250000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1500000			1	0	0	0	2500000
	Don't care	Don't care	1	0	0	2000000		Don't care	Don't care	1	0	0	3333333
12.288	0	0	0	0	0	384000	25	0	0	0	0	0	781250
		1	0	0	0	768000			1	0	0	0	1562500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1536000			1	0	0	0	3125000
	Don't care	Don't care	1	0	0	2048000		Don't care	Don't care	1	0	0	4166666
14	0	0	0	0	0	437500	30	0	0	0	0	0	937500
		1	0	0	0	875000			1	0	0	0	1875000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1750000			1	0	0	0	3750000
	Don't care	Don't care	1	0	0	2333333		Don't care	Don't care	1	0	0	5000000
16	0	0	0	0	0	500000	33	0	0	0	0	0	1031250
		1	0	0	0	1000000			1	0	0	0	2062500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2000000			1	0	0	0	4125000
	Don't care	Don't care	1	0	0	2666666		Don't care	Don't care	1	0	0	5500000
40	0	0	0	0	0	1250000							
		1	0	0	0	2500000							
	1	0	0	0	0								
		1	0	0	0	5000000							
	Don't care	Don't care	1	0	0	6666666							

Table 29.12 Maximum bit rate with external clock input in asynchronous mode (1 of 2)

PCLK (MHz)	External input clock (MHz)	Maximum bit rate (bps)	
		SEMR.ABCS bit = 0	SEMR.ABCS bit = 1
8	2.0000	125000	250000
9.8304	2.4576	153600	307200
10	2.5000	156250	312500
12	3.0000	187500	375000
12.288	3.0720	192000	384000
14	3.5000	218750	437500
16	4.0000	250000	500000
17.2032	4.3008	268800	537600
18	4.5000	281250	562500
19.6608	4.9152	307200	614400
20	5.0000	312500	625000

Table 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
25	6.2500	390625	781250
30	7.5000	468750	937500
33	8.2500	515625	1031250
40	10.0000	625000	1250000

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 to transmit or receive the next frame of data. The output of the synchronization clock is stopped for a 1-bit period. Therefore, it takes 9 bits worth of time to transfer one frame (8 bits) of data, and the average transfer rate is 8/9 times the bit rate.

Table 29.14 Maximum bit rate with external clock input in clock synchronous and simple SPI modes

PCLK (MHz)	External input clock (MHz)	Maximum bit rate (Mbps)
8	1.3333	1.3333333
10	1.6667	1.6666667
12	2.0000	2.0000000
14	2.3333	2.3333333
16	2.6667	2.6666667
18	3.0000	3.0000000
20	3.3333	3.3333333
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

Bit rate (bps)	Operating frequency PCLK (MHz)											
	7.1424			10.00			10.7136			13.00		
	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

Bit rate (bps)	Operating frequency PCLK (MHz)											
	14.2848			16.00			18.00			20.00		
	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

Bit rate (bps)	Operating frequency PCLK (MHz)											
	25.00			30.00			33.00			40.00		
	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	156250	0	0
10.7136	167400	0	0
13.00	203125	0	0
16.00	250000	0	0
18.00	281250	0	0
20.00	312500	0	0
25.00	390625	0	0
30.00	468750	0	0
33.00	515625	0	0
40.00	625000	0	0

Table 29.17 BRR settings for different bit rates in simple IIC mode

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
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

Bit rate (bps)	Operating frequency PCLK (MHz)								
	30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
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 minus 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 fast mode. The setting values are the same as in Table 29.17.

29.2.18 Modulation Duty Register (MDDR)

Address(es): SCI0.MDDR 4007 0012h, SCI1.MDDR 4007 0032h, 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 in 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	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (256/M) \times (N + 1)} - 1 \} \times 100$
	1	0	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (256/M) \times (N + 1)} - 1 \} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (256/M) \times (N + 1)} - 1 \} \times 100$
	Don't care	Don't care	1	$N = \frac{PCLK \times 10^6}{12 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \{ \frac{PCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (256/M) \times (N + 1)} - 1 \} \times 100$
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				$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (256/M) \times (N + 1)} - 1 \} \times 100$
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) for details.
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 I^2C 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 bits 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, 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
b1, b0	—	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

Bit	Symbol	Bit name	Description	R/W
b3	ABCSE	Asynchronous Mode Extended Base Clock Select 1	Valid only in asynchronous mode with SCR.CKE[1] = 0: 0: Clock cycle number for 1-bit period is determined 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. The NFEN bit must be 0 in all other modes.	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)

The BRME bit 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 the ABCSE 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)

The ABCS bit selects the clock cycles for a 1-bit period. Set this bit to 0 except in asynchronous mode.

NFEN bit (Digital Noise Filter Function Enable)

The NFEN bit enables or disables the digital noise filter function.

When the digital noise filter function is enabled:

- Noise cancellation is applied to the RXDn input signal in asynchronous mode
- Noise cancellation is applied to the SDAn and SCLn input signals in simple 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)

The BGDM bit selects the cycle of output clock for the baud rate generator to be either single or double frequency.

The BGDM bit is valid when the on-chip baud rate generator is selected as the clock source (SCR.CKE[1] = 0) in asynchronous mode (SMR.CM = 0). 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)

The RXDESEL bit selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, set this bit to 1 to stop reception, or to start reception without retaining 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, 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 I^{2}C Mode Register 1 (SIMR1)

Address(es): SCI0.SIMR1 4007 0009h, SCI1.SIMR1 4007 0029h, 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

Bit	Symbol	Bit name	Description	R/W
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, the IICM 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 I^{2}C Mode Register 2 (SIMR2)

Address(es): SCI0.SIMR2 4007 000Ah, SCI1.SIMR2 4007 002Ah, 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	I^{2}C Interrupt Mode Select	0: Use ACK/NACK interrupts 1: Use reception and transmission interrupts.	R/W*1
b1	IICCSC	Clock Synchronization	0: No synchronization with the clock signal 1: Synchronization with the 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 (I^{2}C 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 to synchronize the internally generated SCLn clock signal when the SCLn pin is driven low because of a wait inserted by another device, for example.

The SCLn clock signal is not synchronized if this bit is 0. The SCLn clock signal is generated according to the rate selected in the BRR regardless of the level input on the SCLn pin.

Set this bit to 1 except during debugging.

IICACKT bit (ACK Transmission Data)

The IICACKT bit transmits data that 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, SCI4.SIMR3 4007 008Bh, SCI9.SIMR3 4007 012Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	IICSCLS[1:0]		IICSDAS[1:0]		IICSTIF	IICSTPREQ	IICRSTAREQ	IICSTAREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	IICSTAREQ	Start Condition Generation	0: A start condition is not generated 1: A start condition is generated.*1, *3, *5, *6	R/W
b1	IICRSTAREQ	Restart Condition Generation	0: A restart condition is not generated 1: A restart condition is generated.*2, *3, *5, *6	R/W
b2	IICSTPREQ	Stop Condition Generation	0: A stop condition is not generated 1: A stop condition is generated.*2, *3, *5, *6	R/W
b3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag	0: There are no requests for generating conditions or a condition is being generated 1: A start, restart, or stop condition is completely generated. 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 level 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 level 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]

- Writing 1 to the bit.

[Clearing condition]

- When generation of a start condition is complete.

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]

- Writing 1 to the bit.

[Clearing condition]

- When generation of a restart condition is complete.

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]

- Writing 1 to the bit.

[Clearing condition]

- When generation of a stop condition is complete.

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 when an interrupt request is enabled by setting the SCR.TEIE bit, an STI request is output.

[Setting condition]

- When generation of a start, restart, or stop condition completes. If this conflicts with any of the clearing conditions for the flag, the clearing condition takes precedence.

[Clearing conditions]

- Writing 0 to the bit then, confirm that the IICSTIF flag is 0
- Writing 0 to the SIMR1.IICM bit when operation is not in simple IIC mode
- 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, 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

Bit	Symbol	Bit name	Description	R/W
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 the 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, 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: SSn pin function is disabled 1: SSn pin function is enabled.	R/W*1
b1	CTSE	CTS Enable	0: CTS function is disabled (RTS output function is enabled) 1: CTS function is enabled.	R/W*1
b2	MSS	Master Slave Select	0: Transmission is through the TXDn pin and reception is through the RXDn pin (master mode) 1: Reception is through the TXDn pin and transmission is through the 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: Clock polarity is not inverted 1: Clock polarity is inverted.	R/W*1
b7	CKPH	Clock Phase Select	0: Clock is not delayed 1: Clock is delayed.	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 to use the SSn pin to control transmission and reception in simple SPI mode. Set this bit to 0 in all other modes. When master mode (SCR.CKE[1:0] = 00b and MSS = 0) is selected and there is a single master, the SSn pin on the master side is not required to control reception and transmission. In such a case, set the SSE bit to 0. Do not enable both the SSE and CTSE bits as the operation is the same as that when these bits are set to 0.

CTSE bit (CTS Enable)

Set the CTSE bit to 1 to use the SSn pin to input the CTS control signal to control transmission and reception. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, and simple IIC mode. Do not set both the CTSE and SSE bits to 1 as the operation is the same as that when these bits are set to 0.

MSS bit (Master Slave Select)

The MSS bit selects between master and 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]

- Writing 0 to the flag 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 and clock synchronous modes.

29.2.26 FIFO Control Register (FCR)

Address(es): SCI0.FCR 4007 0014h, SCI1.FCR 4007 0034h

	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 Bit	When detecting a receive data ready, the interrupt request is selected: 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

Bit	Symbol	Bit name	Description	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, resets FTDRHL/FRDRHL, selects the FIFO data trigger number of 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 receive data count 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 Bit)

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

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

Bit	Symbol	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

This 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. A value of 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. A value of 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

	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, with FIFO 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 among 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, 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 value in the FNUM[4:0] bits indicates the amount of data stored in the FRDRHL register with a framing error.

PNUM[4:0] bits (Parity Error Count)

The value in the PNUM[4:0] bits 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, 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 the address match function is enabled (DCCR.DCME = 1).

Three bit lengths are available:

- CMPD[6:0] with 7-bit length
- CMPD[7:0] with 8-bit length
- CMPD[8:0] with 9-bit length.

29.2.30 Data Compare Match Control Register (DCCR)

Address(es): SCI0.DCCR 4007 0013h, SCI1.DCCR 4007 0033h, 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

Bit	Symbol	Bit name	Description	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 is 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.

DCCR sets control of the address match function.

DCMF flag (Data Compare Match Flag)

The DCMF flag indicates that the SCI detects a match of the comparison data (CDR.CMPD) with receive data.

[Setting condition]

- When comparison data (CDR.CMPD) matches 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 keeps its previous state.

DPER flag (Data Compare Match Parity Error Flag)

The DPER flag indicates that a parity error occurred at address match detection (reception data match detection).

[Setting condition]

- When a parity error is detected in the frame in which an address match was detected.

[Clearing conditions]

- When 0 is written after 1 is read from DPER.

Clearing the RE bit in SCR to 0 (serial reception is disabled) does not affect the DPER flag, which keeps its previous value.

DFER flag (Data Compare Match Framing Error Flag)

The DFER flag indicates that a framing error occurred at address match detection (reception data match detection).

[Setting conditions]

- When a stop bit is 0 in the frame in which an address match is detected.
- When in 2-stop mode, only the first stop bit is checked for a value of 1 while 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 keeps its previous value.

IDSEL bit (ID Frame Select)

The IDSEL bit selects whether to compare data regardless of the value of the MPB bit 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 used or not.

If SCI detects a match between the comparison data (CDR.CMPD) and receive data, DCME is cleared automatically and the SCI operates in receive mode without 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, 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: The value of SPB2DT bit is not output to TXDn pin 1: The value of SPB2DT bit is output to 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 bit 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 transmission data is output

x: Don't care.

Note: Use the SPTR register 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 the data can be read or written during transmission or reception, enabling continuous data transmission and reception.

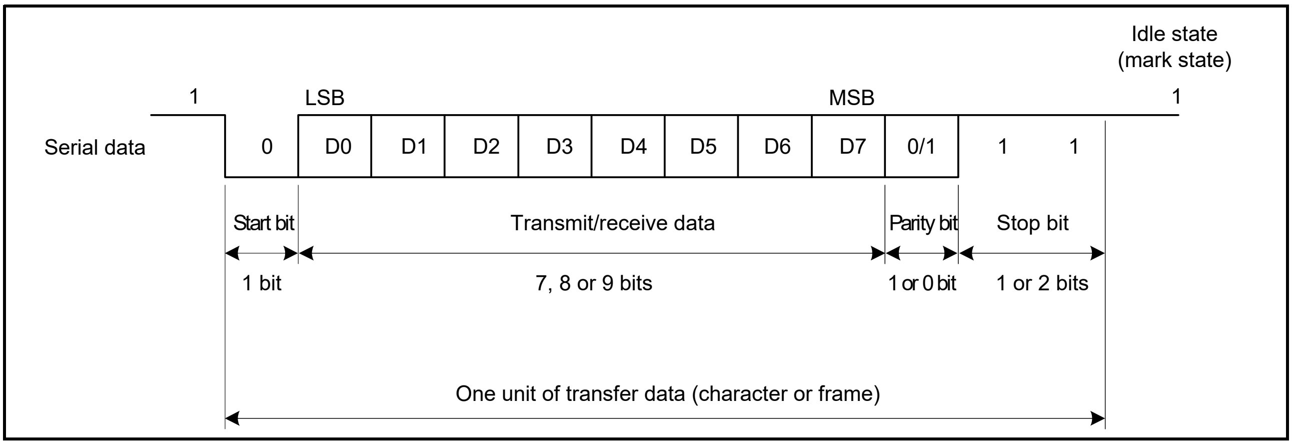

Idle state
(mark state)
1
LSB
MSB
1
Serial data
0
D0
D1
D2
D3
D4
D5
D6
D7
0/1
1
1
Start bit
Transmit/receive data
Parity bit
Stop bit
1 bit
7, 8 or 9 bits
1 or 0 bit
1 or 2 bits
One unit of transfer data (character or frame)

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 the 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 Communications Function.

Table 29.22 Serial transfer formats (asynchronous mode) (1 of 2)

SCMR setting	SMR setting				Serial transfer format and frame length
CHR1	CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	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												
CHR1	**CHR**	**PE**	**MP**	**STOP**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
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):

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\ F - \frac{\left| D - 0.5 \right|}{N}\ (1 + F) \right| \times 100\ [\%] \cdots \text{ Formula (1)}$$

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 frequency of 6 times the bit rate is used as a base clock and the receive data is sampled on the rising edge of the 3rd pulse of the base clock.

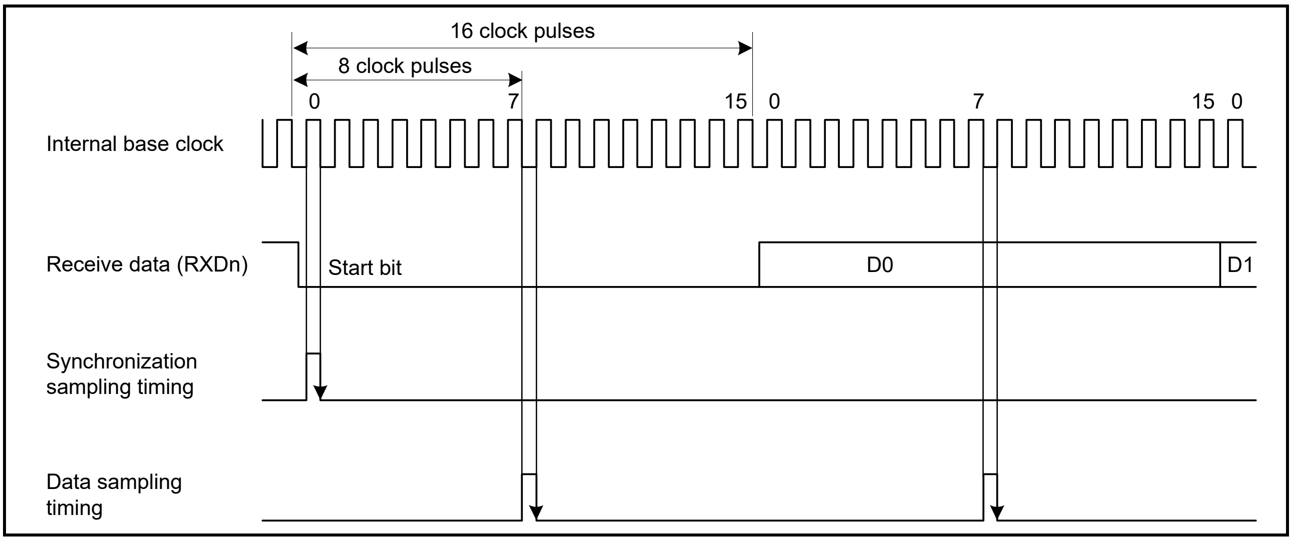


Figure 29.3 Receive data sampling timing in asynchronous mode

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.

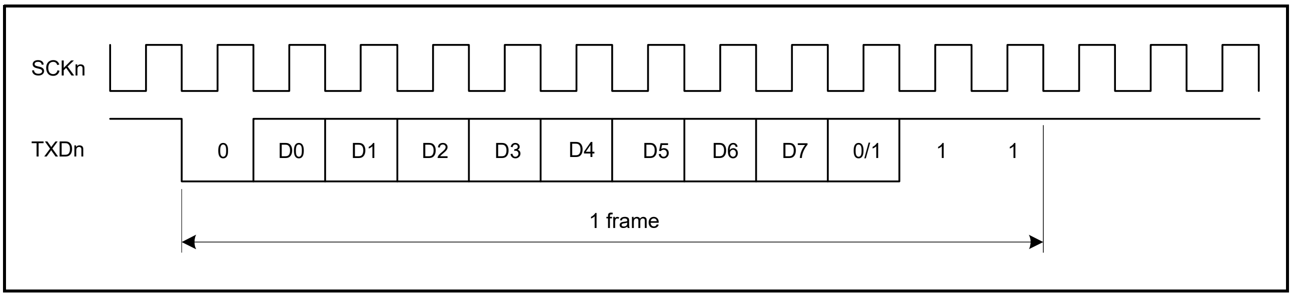


Figure 29.4 Phase relationship between output clock and transmit data in asynchronous mode when SMR.CHR = 0, PE = 1, MP = 0, and STOP = 1

29.3.4 Double-Speed Operation and Frequency of 6 Times the Bit Rate

When the ABCS bit in SEMR is set to 1 and 8 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 BGDM bit in SEMR is set to 1, the cycle of the base clock is halved 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 on 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 or ABCSE 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 ABCS and ABCSE 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 CTSE bit in SPMR 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 low level and high level output 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

- After reception is complete, if it is terminated with SCR.RE = 0 without reading the RDR register, then RTS remains high. 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 mismatch.

If DCCR.IDSEL is set to 0, the SCI performs address match or mismatch regardless of the MPB bit value of the received data. Until the SCI detects a match between the comparison data (CDR.CMPD*3) and receive data, the 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.

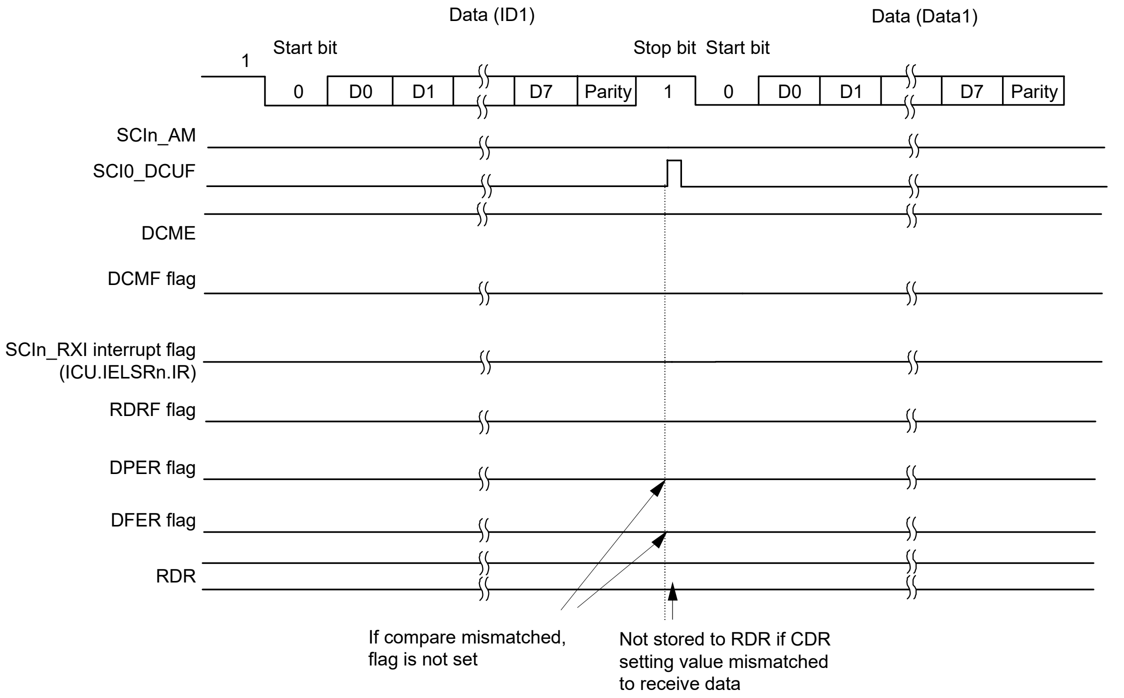
If DCCR.IDSEL is set to 1, the SCR.MPIE bit is automatically cleared. If DCCR.IDSEL is set to 0, the value of SCR.MPIE bit is retained. If SCR.RIE is set to 1, the SCI issues an SCIn_RXI interrupt request. If the SCI detects a framing error in the receive data for which a match is detected, DCCR.DFER 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 at 0.*2

After the SCI detects a match, and DCCR.DCME is automatically cleared, the SCI receives the next data continuously based on the current register setting.

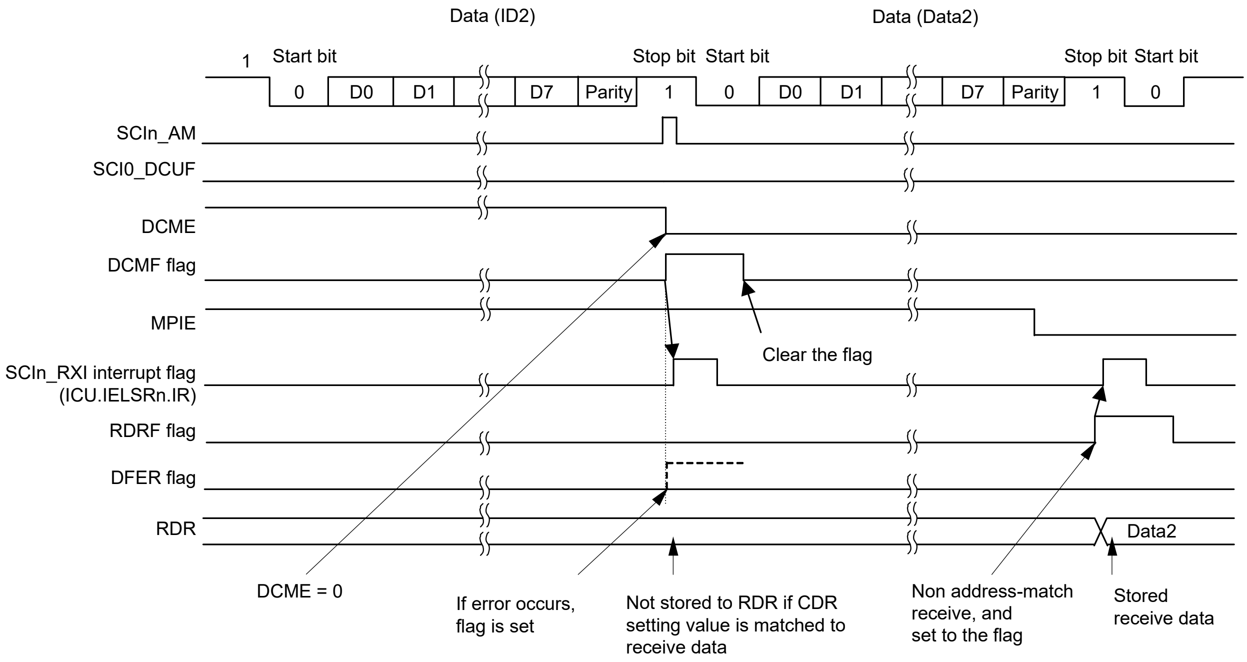
When the DCCR.DFER or DCCR.DPER flag is set, the address match is not performed. Before enabling the address match function, set the DCCR.DFER and DCCR.DPER flags to 0.

An example of the address match function is 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, or 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.


Data (ID1)
Data (Data1)
1
Start bit
Stop bit
Start bit
0
D0
D1
D7
Parity
1
0
D0
D1
D7
Parity
SCIn_AM
SCI0_DCUF
DCME
DCMF flag
SCIn_RXI interrupt flag
(ICU.IELSRn.IR)
RDRF flag
DPER flag
DFER flag
RDR
If compare mismatched,
flag is not set
Not stored to RDR if CDR
setting value mismatched
to receive data

(a) Example of compare mismatched between receive data and CDR (8-bit length/parity/non multi-processor mode)


Data (ID2)
Data (Data2)
1
Start bit
Stop bit
Start bit
Stop bit
Start bit
0
D0
D1
D7
Parity
1
0
D0
D1
D7
Parity
1
0
SCIn_AM
SCI0_DCUF
DCME
DCMF flag
MPIE
SCIn_RXI interrupt flag
(ICU.IELSRn.IR)
RDRF flag
DFER flag
RDR
Clear the flag
Data2
DCME = 0
If error occurs,
flag is set
Not stored to RDR if CDR
setting value is matched to
receive data
Non address-match
receive, and
set to the flag
Stored
receive data

(b) Example of compare matched between receive data and CDR (8-bit length/parity/non multi-processor mode)

Figure 29.5 **Example of address match (1) in non Multi-processor mode**

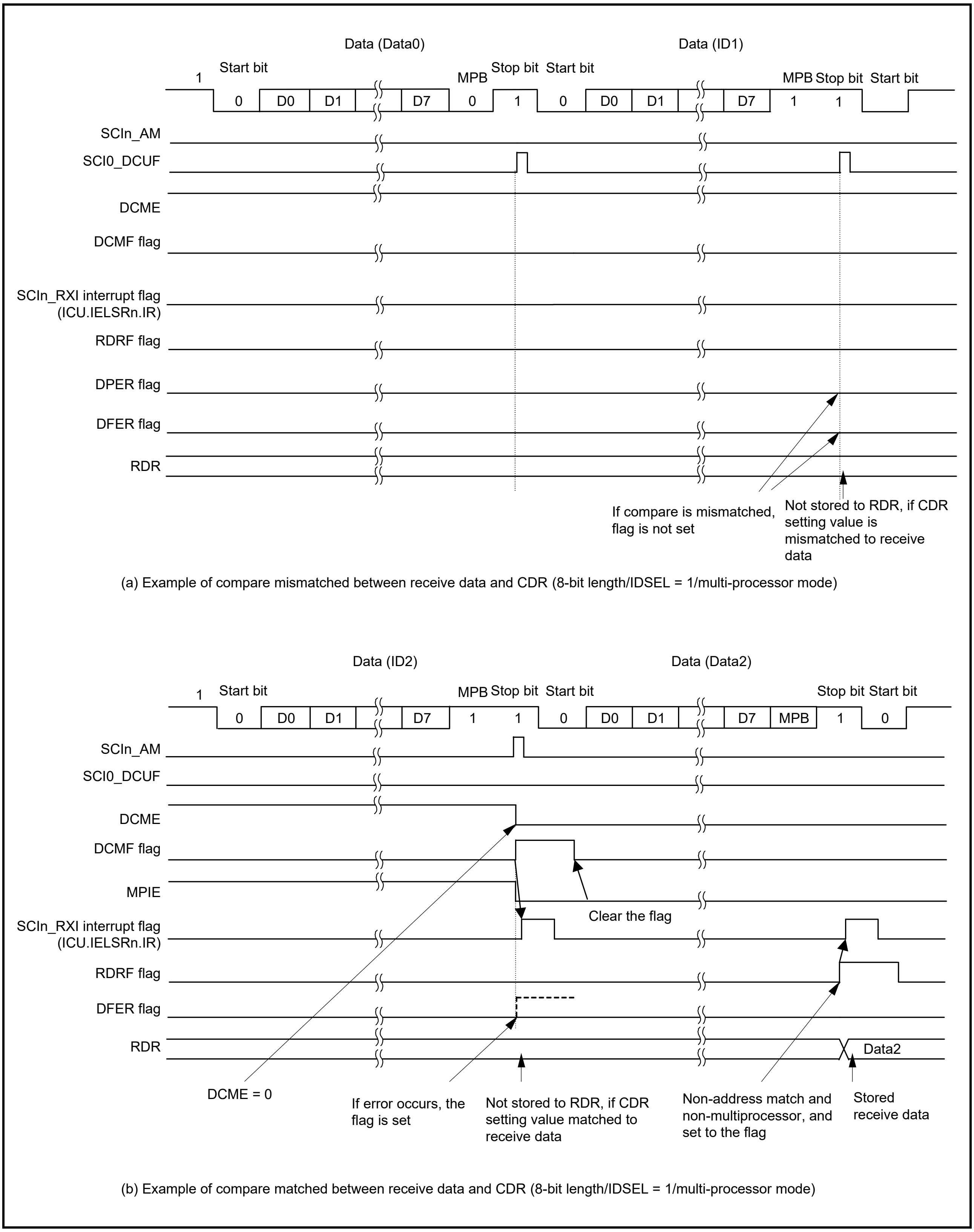

Data (Data0)
Data (ID1)
1
Start bit
MPB
Stop bit
Start bit
MPB
Stop bit
Start bit
0
D0
D1
D7
0
1
0
D0
D1
D7
1
1
SCIn_AM
SCI0_DCUF
DCME
DCMF flag
SCIn_RXI interrupt flag (ICU.IELSRn.IR)
RDRF flag
DPER flag
DFER flag
RDR
If compare is mismatched, flag is not set
Not stored to RDR, if CDR setting value is mismatched to receive data
(a) Example of compare mismatched between receive data and CDR (8-bit length/IDSEL = 1/multi-processor mode)
Data (ID2)
Data (Data2)
1
Start bit
MPB
Stop bit
Start bit
Stop bit
Start bit
0
D0
D1
D7
1
1
0
D0
D1
D7
MPB
1
0
SCIn_AM
SCI0_DCUF
DCME
DCMF flag
MPIE
SCIn_RXI interrupt flag (ICU.IELSRn.IR)
RDRF flag
DFER flag
RDR
Clear the flag
Data2
DCME = 0
If error occurs, the flag is set
Not stored to RDR, if CDR setting value matched to receive data
Non-address match and non-multiprocessor, and set to the flag
Stored receive data
(b) Example of compare matched between receive data and CDR (8-bit length/IDSEL = 1/multi-processor mode)

Figure 29.6 Example of address match (2) in multi-processor mode

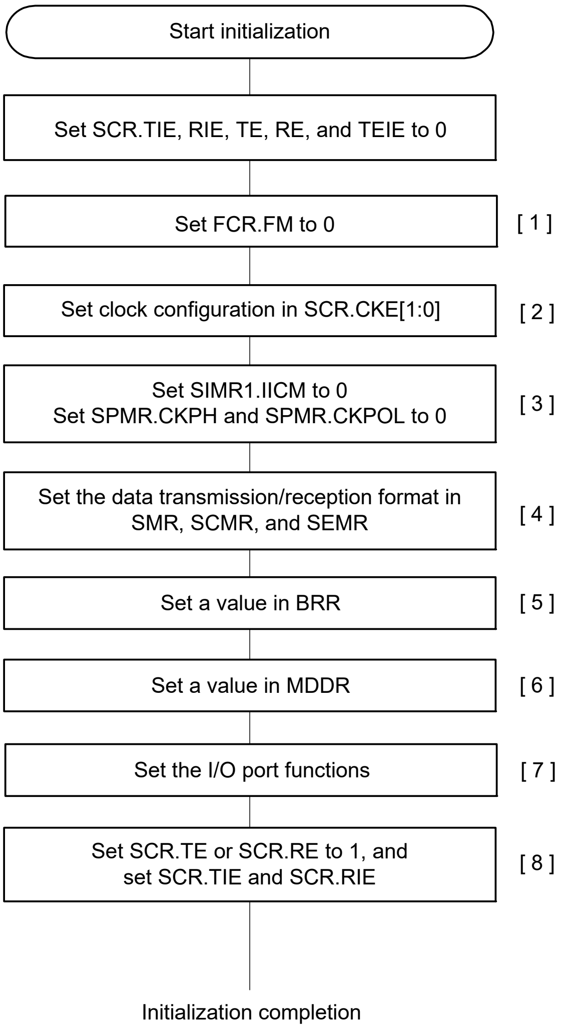
29.3.7 SCI Initialization in Asynchronous Mode

Before transmitting and receiving data, start by writing the initial value 00h to SCR, and 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, the SCR must be initialized before a change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied during initialization.

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 while the SCR.TIE bit is 1 leads to the generation of an SCIn_TXI interrupt request.


Start initialization
Set SCR.TIE, RIE, TE, RE, and TEIE to 0
Set FCR.FM to 0
[1]
Set clock configuration in SCR.CKE[1:0]
[2]
Set SIMR1.IICM to 0
Set SPMR.CKPH and SPMR.CKPOL to 0
[3]
Set the data transmission/reception format in SMR, SCMR, and SEMR
[4]
Set a value in BRR
[5]
Set a value in MDDR
[6]
Set the I/O port functions
[7]
Set SCR.TE or SCR.RE to 1, and set SCR.TIE and SCR.RIE
[8]
Initialization completion

[1] Set FCR.FM to 0.

[2] Set the clock selection in SCR.
When clock output is selected in asynchronous mode, the clock is output immediately after SCR settings are made.

[3] Set SIMR1.IICM to 0.
Set SPMR.CKPH and SPMR.CKPOL to 0.
Step [3] can be skipped if the values have not changed from the initial values.

[4] Set data transmission/reception format in SMR, SCMR, and SEMR.

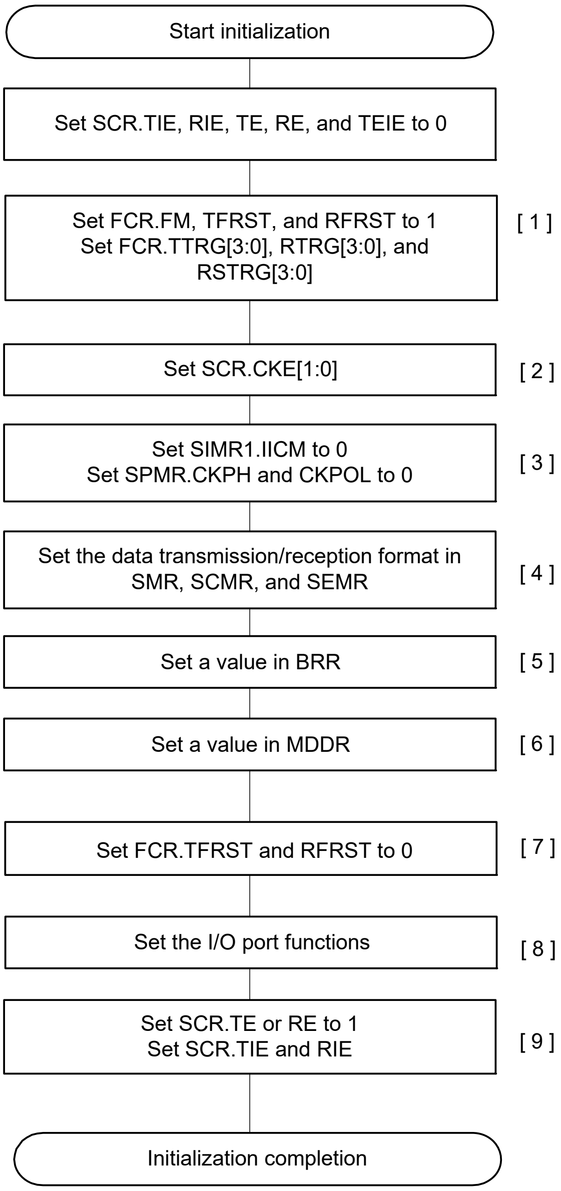
[5] Write a value associated with the bit rate to BRR.
This step is not required if an external clock is used.

[6] Write the value obtained by correcting a bit rate error in MDDR. This step is not required if the BRME bit in SEMR is set to 0 or an external clock is used.

[7] Specify the I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[8] Set SCR.TE or SCR.RE to 1. Also set SCR.TIE and SCR.RIE.
Setting SCR.TE and SCR.RE allows TXDn and RXDn to be used.

Figure 29.7 Example SCI initialization flow in asynchronous mode with non-FIFO selected


Start initialization
Set SCR.TIE, RIE, TE, RE, and TEIE to 0
Set FCR.FM, TFRST, and RFRST to 1
Set FCR.TTRG[3:0], RTRG[3:0], and RSTRG[3:0]
[1]
Set SCR.CKE[1:0]
[2]
Set SIMR1.IICM to 0
Set SPMR.CKPH and CKPOL to 0
[3]
Set the data transmission/reception format in SMR, SCMR, and SEMR
[4]
Set a value in BRR
[5]
Set a value in MDDR
[6]
Set FCR.TFRST and RFRST to 0
[7]
Set the I/O port functions
[8]
Set SCR.TE or RE to 1
Set SCR.TIE and RIE
[9]
Initialization completion

[1] Set FCR.FM, TFRST, and RFRST to 1.
This enables FIFO mode and clears the FIFOs.
Set the trigger values in FCR.TTRG[3:0], RTRG[3:0], and RSTRG[3:0] bits.

[2] Set the clock selection in SCR.
When the clock output is selected in asynchronous mode, the clock is output immediately after SCR settings are made.

[3] Set SIMR1.IICM to 0.
Set SPMR.CKPH and SPMR.CKPOL to 0.
Step [3] can be skipped if the values have not changed from the initial values.

[4] Set data transmission/reception format in SMR, SCMR, and SEMR.

[5] Write a value associated with the bit rate to BRR.
This step is not required if an external clock is used.

[6] Write the value obtained by correcting a bit rate error in MDDR. This step is not required if the BRME bit in SEMR is set to 0 or an external clock is used.

[7] Set FCR.TFRST and FCR.RFRST to 0.

[8] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[9] Set SCR.TE or SCR.RE to 1. Also set SCR.TIE and RIE.
Setting SCR.TE and SCR.RE allows TXDn and RXDn to be used.

Figure 29.8 Example SCI initialization flow in synchronous 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.

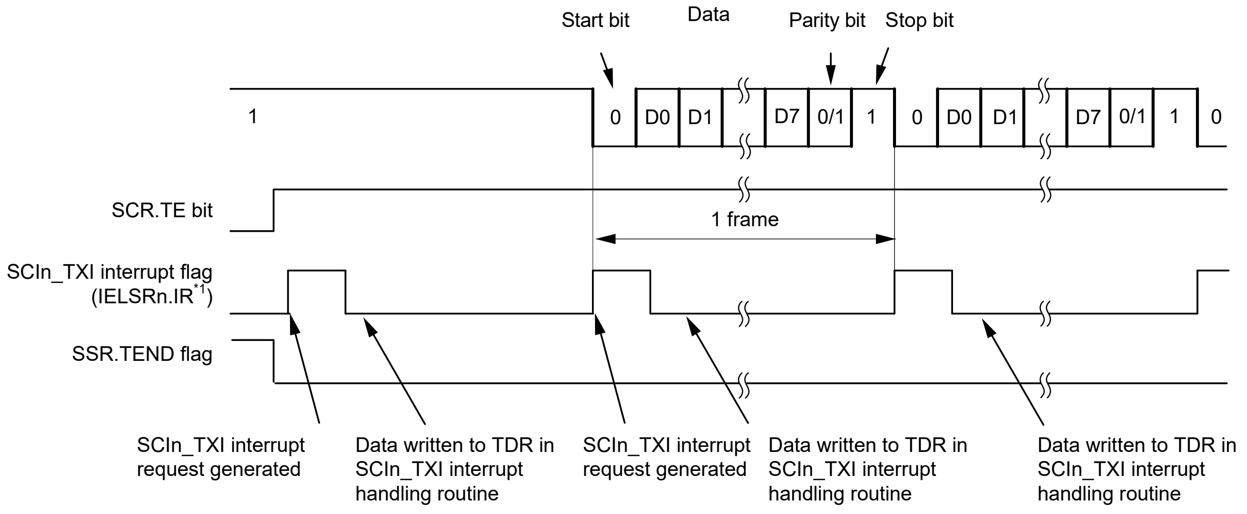
1. The SCI transfers data from TDR*1 to TSR when data is written to TDR*1 in the SCIn_TXI interrupt handling routine.
   The SCIn_TXI interrupt request at the beginning of transmission is generated when the TE and TIE bits in SCR are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CTSE bit in SPMR is set to 0 (CTS function is disabled) 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 TIE to 0 (an SCIn_TXI interrupt request is disabled) and TEIE to 1 (an SCIn_TEI interrupt request is enabled) in the SCR register 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 CTSE bit in SPMR 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.

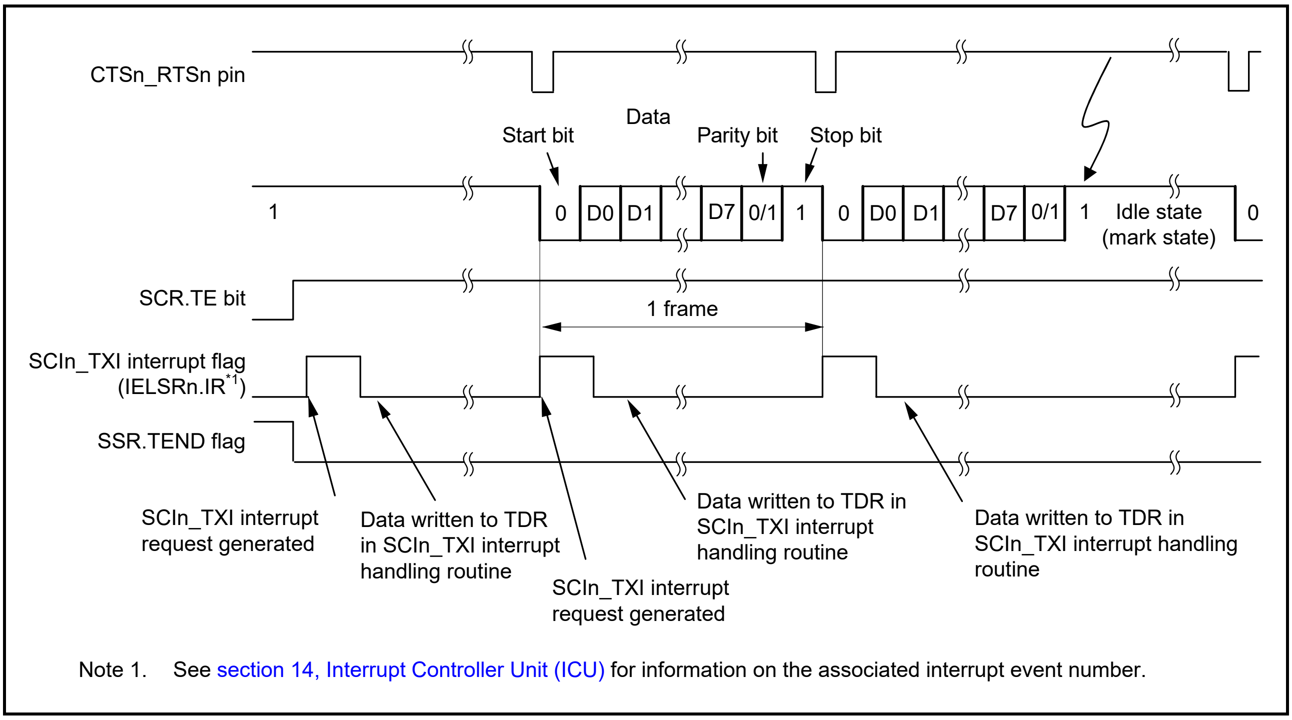
Note 1. Only write data to TDRHL when 9-bit data length is selected.

Figure 29.9, Figure 29.10, and Figure 29.11 show an example flow of serial transmission in asynchronous mode.


Start bit
Data
Parity bit
Stop bit
1
0
D0
D1
D7
0/1
1
0
D0
D1
D7
0/1
1
0
SCR.TE bit
1 frame
SCIn_TXI interrupt flag (IELSRn.IR*1)
SSR.TEND flag
SCIn_TXI interrupt request generated
Data written to TDR in SCIn_TXI interrupt handling routine
SCIn_TXI interrupt request generated
Data written to TDR in SCIn_TXI interrupt handling routine
Data written to TDR in SCIn_TXI interrupt handling routine

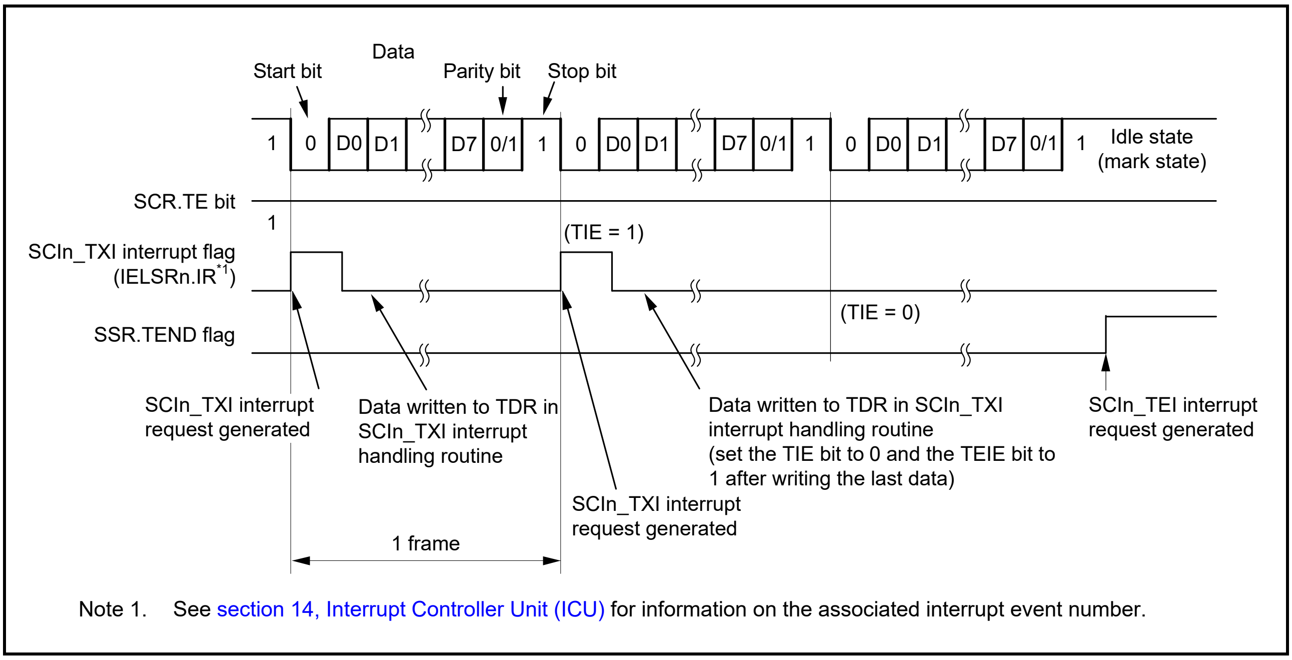
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 29.9 Example operation of 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


CTSn_RTSn pin
Data
Start bit
Parity bit
Stop bit
1
0
D0
D1
D7
0/1
1
0
D0
D1
D7
0/1
1
Idle state
(mark state)
0
SCR.TE bit
1 frame
SCIn_TXI interrupt flag
(IELSRn.IR*1)
SSR.TEND flag
SCIn_TXI interrupt
request generated
Data written to TDR
in SCIn_TXI interrupt
handling routine
SCIn_TXI interrupt
request generated
Data written to TDR in
SCIn_TXI interrupt
handling routine
Data written to TDR in
SCIn_TXI interrupt handling
routine

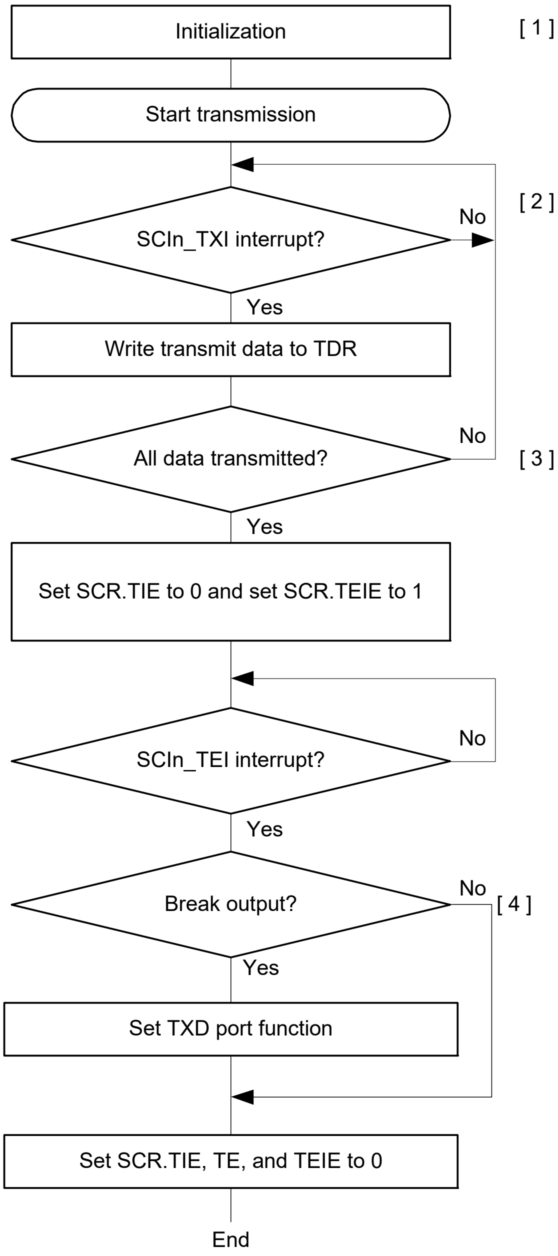
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 29.10 Example operation of serial transmission in asynchronous mode (2) with 8-bit data, parity bit, 1 stop bit, CTS function used, and at the beginning of transmission


Data
Start bit
Parity bit
Stop bit
1
0
D0
D1
D7
0/1
1
0
D0
D1
D7
0/1
1
0
D0
D1
D7
0/1
1
Idle state
(mark state)
SCR.TE bit
1
SCIn_TXI interrupt flag
(IELSRn.IR*1)
(TIE = 1)
(TIE = 0)
SSR.TEND flag
SCIn_TXI interrupt
request generated
Data written to TDR in
SCIn_TXI interrupt
handling routine
Data written to TDR in SCIn_TXI
interrupt handling routine
(set the TIE bit to 0 and the TEIE bit to
1 after writing the last data)
SCIn_TEI interrupt
request generated
SCIn_TXI interrupt
request generated
1 frame

Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 29.11 Example operation of 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


Initialization
[1]
Start transmission
SCIn_TXI interrupt?
No
[2]
Yes
Write transmit data to TDR
All data transmitted?
No
[3]
Yes
Set SCR.TIE to 0 and set SCR.TEIE to 1
SCIn_TEI interrupt?
No
Yes
Break output?
No
[4]
Yes
Set TXD port function
Set SCR.TIE, TE, and TEIE to 0
End

[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 while 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 using 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 of serial transmission flow 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 that corresponds to the correct data length is set to FTDRH and FTDRL. Write 0 for unused bits. Write in order from FTDRH to FTDRL.

Data Length	Register Setting		Transmit data in FTDRH, FTDRL															
			FTDRHL															
			FTDRH								FTDRL							
	SCMR. CHR1	SMR. CHR	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	0	—	—	—	—	—	—	—	—	—	7-bit transmit data						
8 bits	1	1	—	—	—	—	—	—	—	—	8-bit transmit data							
9 bits	0	Don't care	—	—	—	—	—	—	—	9-bit transmit data								

—: Invalid. The write value should be 0.

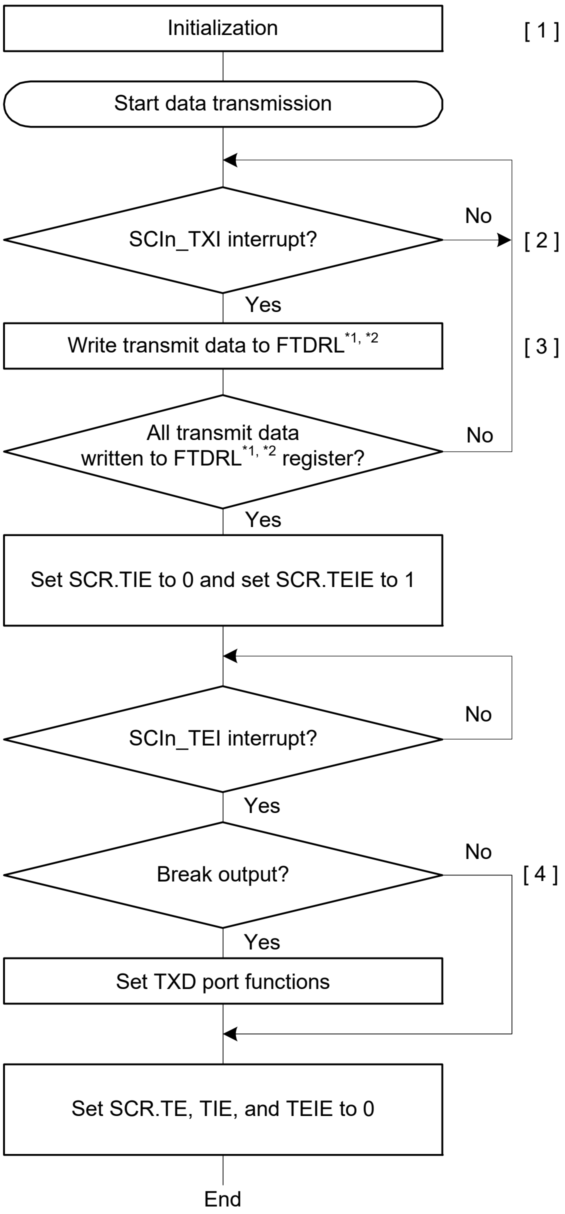
Figure 29.13 Data format written to FTDRH and FTDRL with FIFO selected

In serial transmission, the SCI operates as described in this section. When the TE bit in SCR 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, TDFE bit in SSR_FIFO 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 TIE to 0 (an SCIn_TXI interrupt request is disabled) and TEIE to 1 (an SCIn_TEI interrupt request is enabled) in the SCR register 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 the non-transmitted data remains in FTDRL*3 on the output of the stop bit.
5. When data is set to FTDRL*3, setting of CTSE to 0 (CTS function is disabled) in the SPMR register 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 where 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.


Initialization
[1]
Start data transmission
SCIn_TXI interrupt?
No
[2]
Yes
Write transmit data to FTDRL*1, *2
[3]
All transmit data written to FTDRL*1, *2 register?
No
Yes
Set SCR.TIE to 0 and set SCR.TEIE to 1
SCIn_TEI interrupt?
No
Yes
Break output?
No
[4]
Yes
Set TXD port functions
Set SCR.TE, TIE, and TEIE to 0
End

[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:
For data transmission 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 set the SSR_FIFO.TDFE flag to 0. It is possible to write 16 bits of transmit data, which is also the amount of data stored in the transmit FIFO.
Transmit data can also be written to FTDRL by activating the DMAC or DTC. When data is written to FTDRL by the DMAC or DTC, the TDFE flag is cleared automatically. Therefore, 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 using SPTR.SPB2IO and SPTR.SPB2DT, set SCR.TE to 0.

Note 1. When data length is 9 bits, FTDRH and FTDRL registers are used.
Note 2. When data length is 9 bits, write in 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.

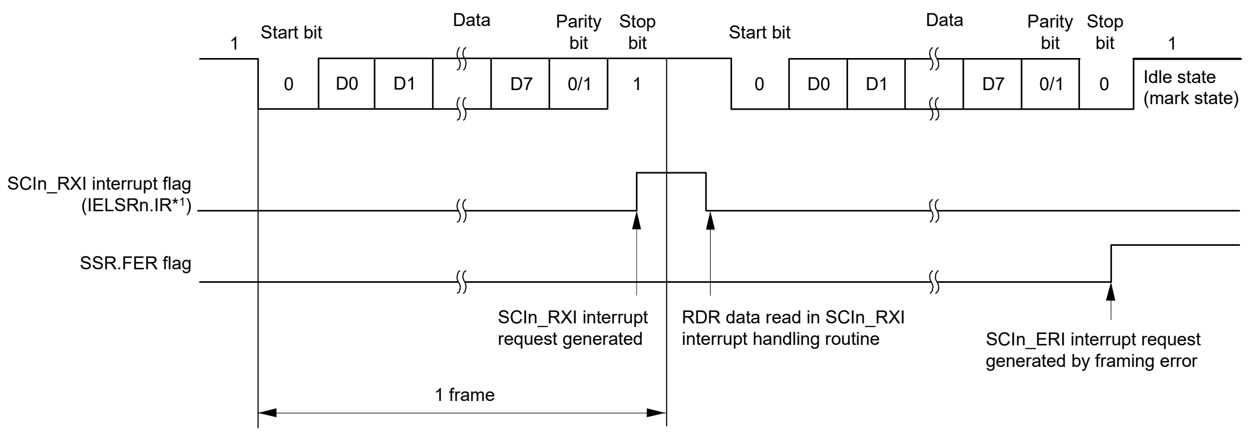
In serial data reception, the SCI operates as follows:

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.

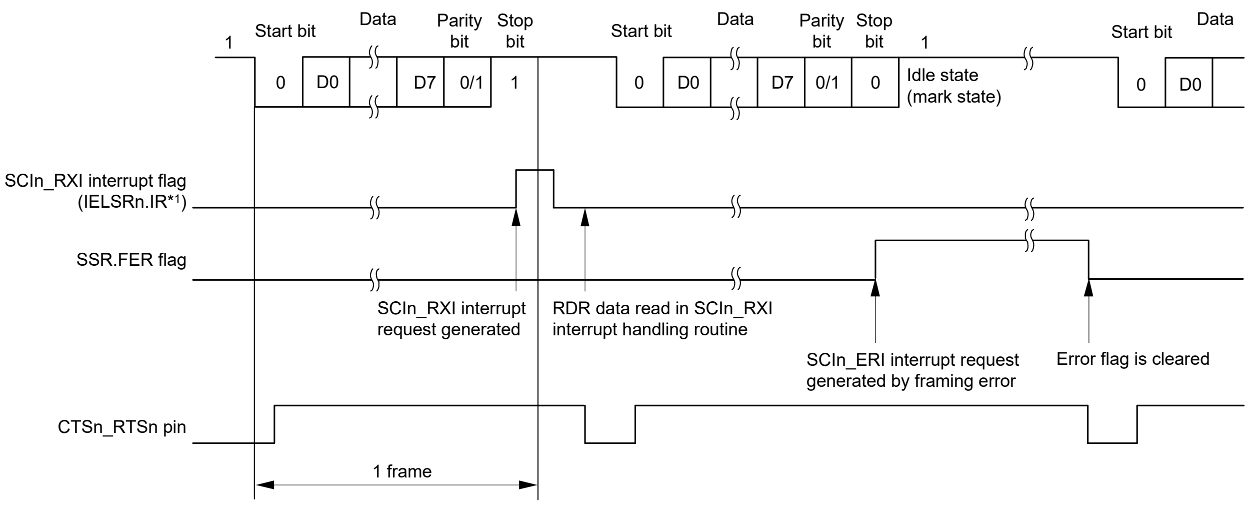
5. If a frame error is detected, the FER flag in the SSR is set to 1 and receive data is transferred to RDR*1. If the RIE bit in the SCR is 1, an SCIn_ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to RDR*1. If the RIE bit in the SCR is 1, an 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 completes. Reading the received data that was transferred to RDR causes the CTSn_RTSn pin to output low.

Note 1. Only read data in RDRHL when 9-bit data length is selected.


1
Start bit
Data
Parity bit
Stop bit
0
D0
D1
D7
0/1
1
Start bit
Data
Parity bit
Stop bit
1
0
D0
D1
D7
0/1
0
Idle state (mark state)
SCIn_RXI interrupt flag (IELSRn.IR*1)
SSR.FER flag
SCIn_RXI interrupt request generated
RDR data read in SCIn_RXI interrupt handling routine
SCIn_ERI interrupt request generated by framing error
1 frame

Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 29.15 Example of SCI operation for serial reception in asynchronous mode (1) when RTS function is not used, and with 8-bit data, parity bit, and 1 stop bit


1
Start bit
Data
Parity bit
Stop bit
0
D0
D7
0/1
1
Start bit
Data
Parity bit
Stop bit
1
0
D0
D7
0/1
0
Idle state (mark state)
Start bit
Data
0
D0
SCIn_RXI interrupt flag (IELSRn.IR*1)
SSR.FER flag
SCIn_RXI interrupt request generated
RDR data read in SCIn_RXI interrupt handling routine
SCIn_ERI interrupt request generated by framing error
Error flag is cleared
CTSn_RTSn pin
1 frame

Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 29.16 Example of SCI operation for serial reception in asynchronous mode (2) when RTS function is used, and with 8-bit data, parity bit, and 1 stop bit

Table 29.23 lists the states of the flags in the SSR register and the 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 when the receive error flag is 1. Also, set the ORER, FER, and PER flags to

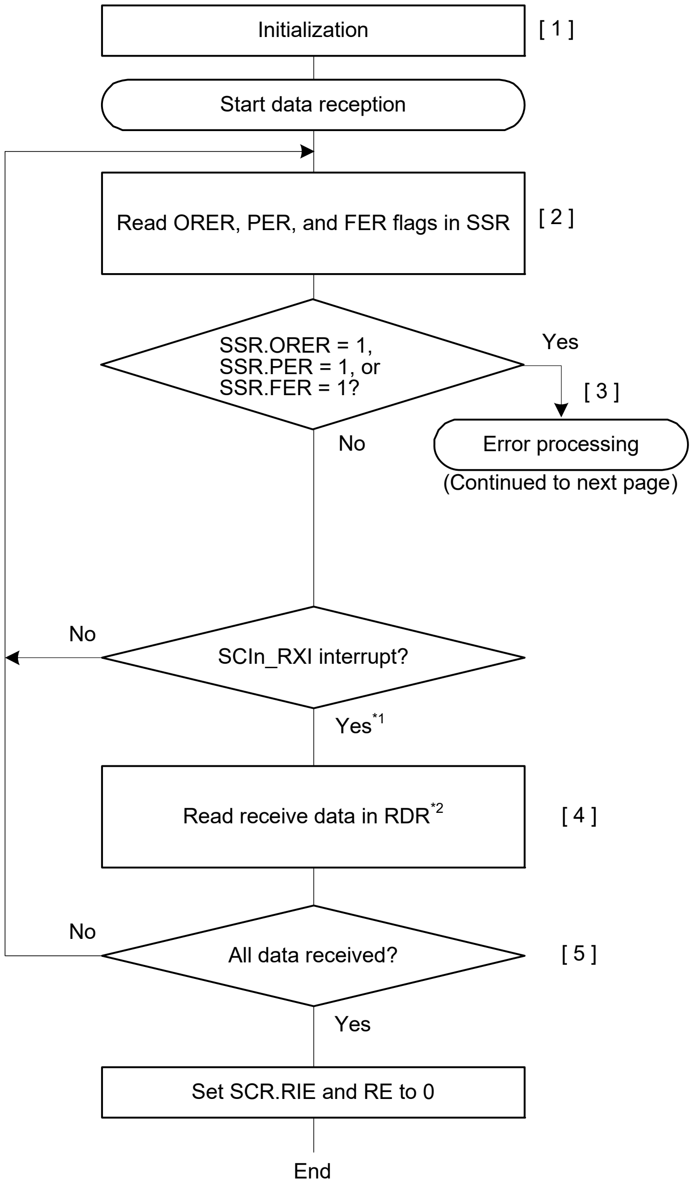
0 before resuming reception. In addition, be sure to read RDR or RDRHL during overrun error processing. When reception is forcibly terminated by setting the RE bit in SCR to 0 during operation, read RDR or RDRHL because the received data that is not read might be left in RDR or RDRHL.

Figure 29.17 and Figure 29.18 show example flows for serial data reception.

Table 29.23 Flags in SSR Status Register and receive data handling

Flags in the SSR Status Register				
ORER	**FER**	**PER**	**Received 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: Only read data in the RDRHL register when 9-bit data length is selected.


Initialization
[1]
Start data reception
Read ORER, PER, and FER flags in SSR
[2]
SSR.ORER = 1,
SSR.PER = 1, or
SSR.FER = 1?
Yes
[3]
Error processing
(Continued to next page)
No
No
SCIn_RXI interrupt?
Yes*1
Read receive data in RDR*2
[4]
No
All data received?
[5]
Yes
Set SCR.RIE and RE to 0
End

[1] SCI initialization:
Set data reception.

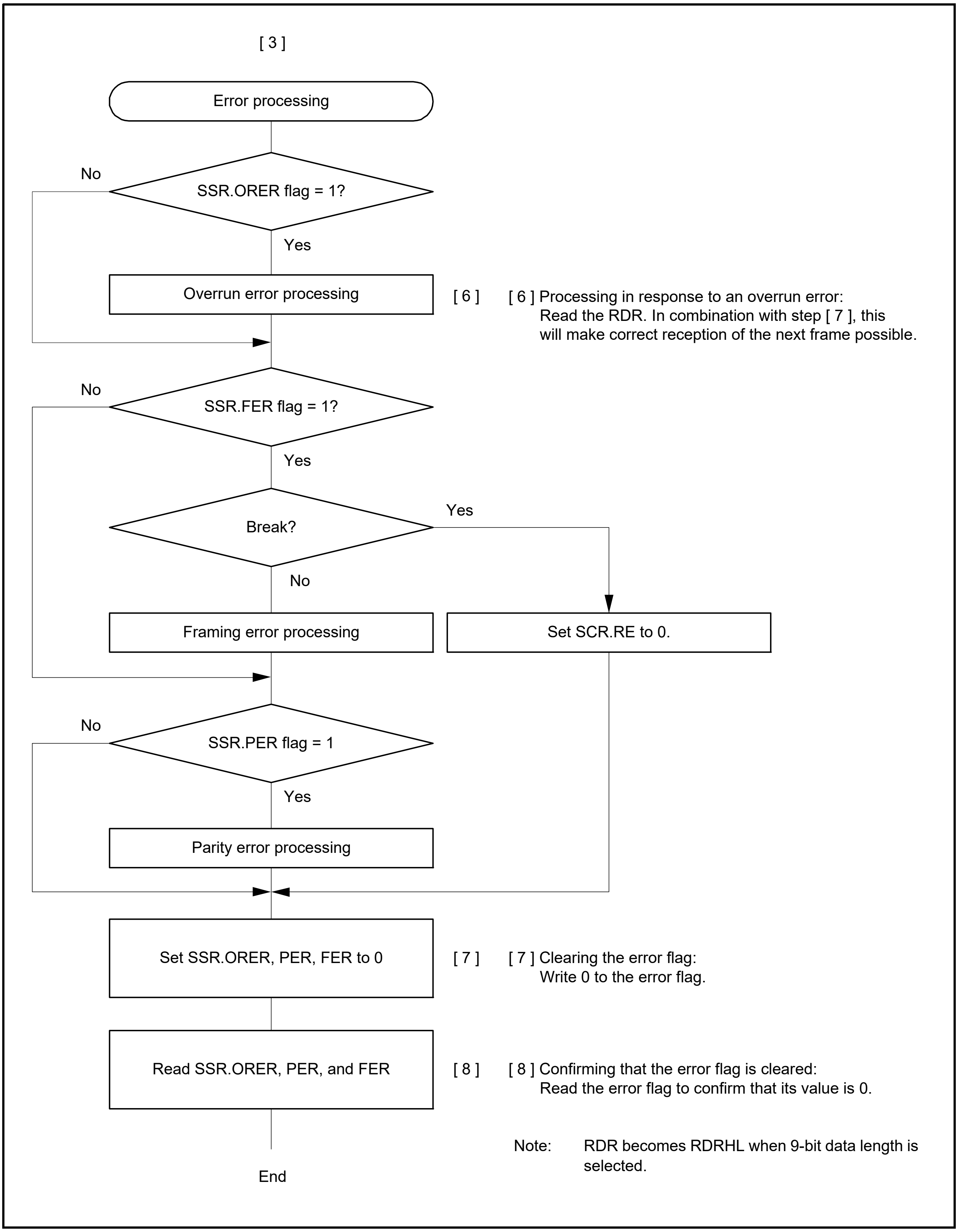
[2] [3] Receive error processing and break detection:
If a receive error occurs, an SCIn_ERI interrupt is generated. The error type is identified by reading the ORER, PER, and FER flags in SSR. After performing the appropriate error processing, be sure to set the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags are set to 1. For a framing error, a break can be detected by reading the value of the input port associated with the RXDn pin.

[4] Read the receive data in RDR once in the SCIn_RXI interrupt handling routine.

[5] 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 RE to 0 before reading RDR.
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)


[3]
Error processing
SSR.ORER flag = 1?
No
Yes
Overrun error processing
[6]
SSR.FER flag = 1?
No
Yes
Break?
Yes
No
Framing error processing
Set SCR.RE to 0.
SSR.PER flag = 1
No
Yes
Parity error processing
Set SSR.ORER, PER, FER to 0
[7]
Read SSR.ORER, PER, and FER
[8]
End

[6] Processing in response to an overrun error:
Read the RDR. In combination with step [7], this will make correct reception of the next frame possible.

[7] Clearing the error flag:
Write 0 to the error flag.

[8] Confirming that the error flag is cleared:
Read the error flag to confirm that its value is 0.

Note: RDR becomes RDRHL when 9-bit data length is selected.

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 bit in the FRDRH register. Data that corresponds to the data length is written to FRDRH and FRDRL. Unused bits are written as 0. Read in order from FRDRH to FRDRL. If software reads FRDRL, SCI updates FER, PER and receive data (RDAT[8:0]) in the FRDRL register with the next data. The RDF, ORER, and DR flags 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								

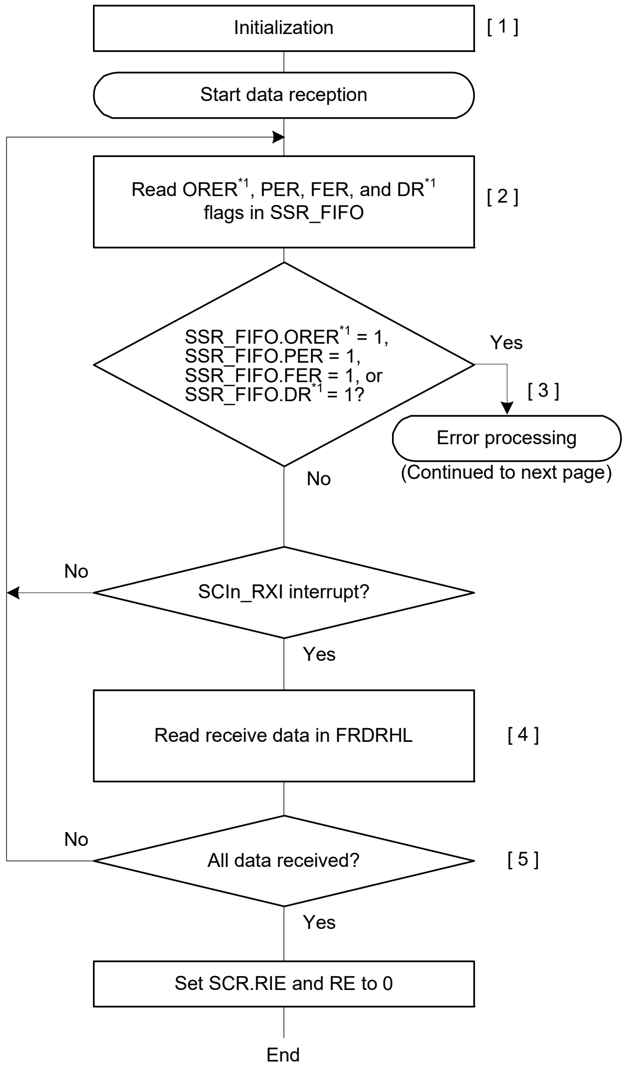
Note: 0 is always read for MPB flag (FRDRH[1]).
When data length is 7 bits, 0 is always read for FRDRH[0] and FRDRL[7].
When data length is 8 bits, 0 is always read for FRDRH[0].
FRDRH[7] bit is read as an indefinite value.

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. When the FRDRL register is full, an overrun error occurs. If an overrun error occurs, the ORER flag in SSR_FIFO is set to 1. When the RIE bit 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 RIE bit 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 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, the DR bit in SSR_FIFO is set to 1. When RIE is 1 and the DRES bit in FCR register is 0, SCI generates an SCIn_RXI interrupt request. When the DRES bit 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 the 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 the data in the FRDRH and FRDRL registers when 9-bit data length is selected.
Note 2. Read the data in order from FRDRH to FRDRL when 9-bit data length is selected.
Note 3. The SCI only checks for updates to the FRDRL register and not to the FRDRH register when 9-bit data length is selected.


Initialization
[1]
Start data reception
Read ORER*1, PER, FER, and DR*1 flags in SSR_FIFO
[2]
SSR_FIFO.ORER*1 = 1, SSR_FIFO.PER = 1, SSR_FIFO.FER = 1, or SSR_FIFO.DR*1 = 1?
Yes
[3]
Error processing
(Continued to next page)
No
No
SCIn_RXI interrupt?
Yes
Read receive data in FRDRHL
[4]
No
All data received?
[5]
Yes
Set SCR.RIE and RE to 0
End

[1] SCI initialization:
Set data reception.

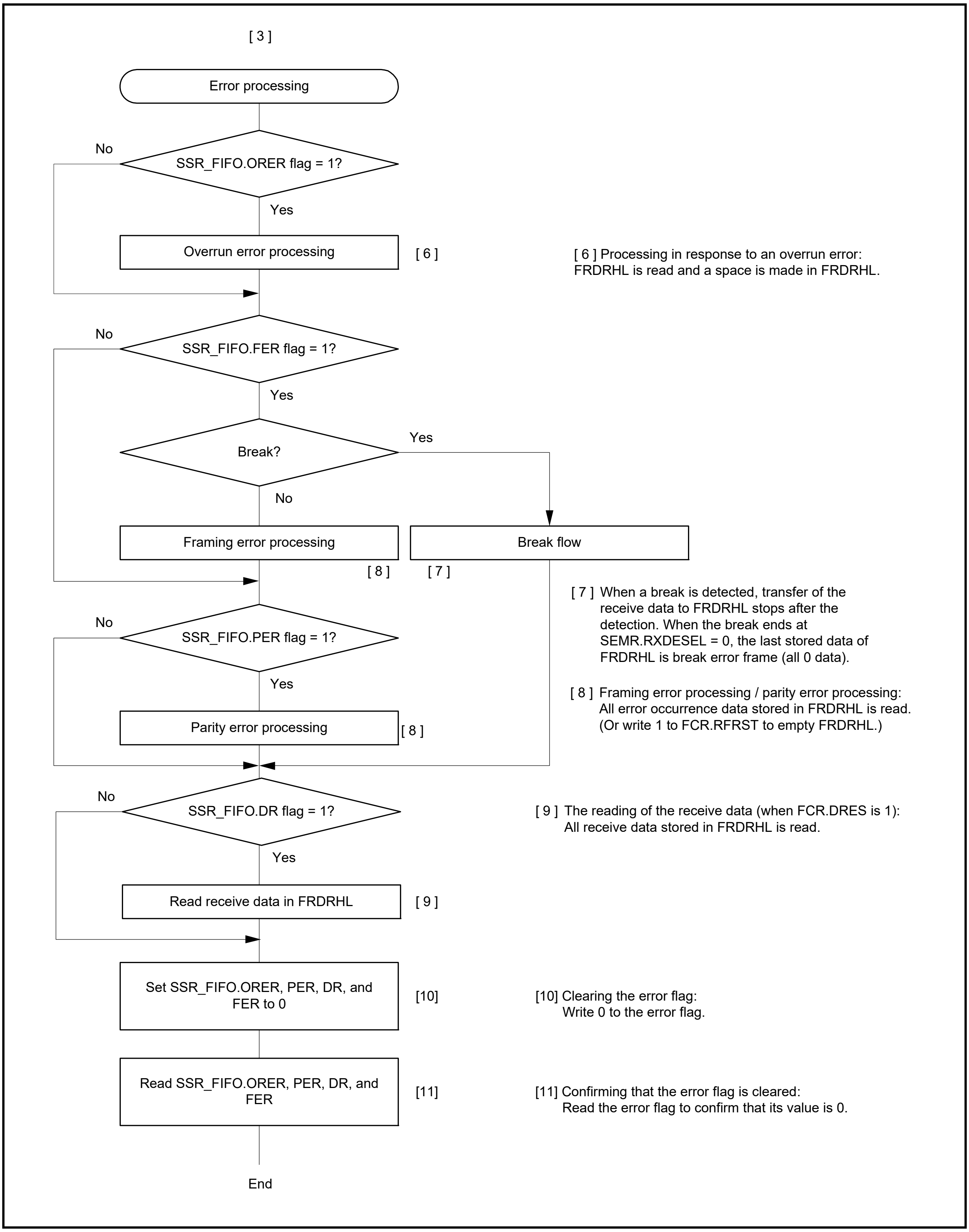
[2] [3] Receive error processing and break detection:
If a receive error occurs, an SCIn_ERI interrupt is generated. A break can be detected by reading the SPTR.RXDMON bit. An error is identified by reading the ORER*1, PER, DR*1, and FER flags in SSR_FIFO. After performing the appropriate error processing, 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 amount of stored data is below the FCR.RTRG value. Confirm the number of the receive data bits 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 and DR flags to 0.
The FRDRHL data can also be read by activating the DMAC or DTC. The RDF flag is cleared automatically in this case. Therefore, 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)


[3]
Error processing
No
SSR_FIFO.ORER flag = 1?
Yes
Overrun error processing
[6]
No
SSR_FIFO.FER flag = 1?
Yes
Break?
Yes
No
Framing error processing
Break flow
[8]
[7]
No
SSR_FIFO.PER flag = 1?
Yes
Parity error processing
[8]
No
SSR_FIFO.DR flag = 1?
Yes
Read receive data in FRDRHL
[9]
Set SSR_FIFO.ORER, PER, DR, and FER to 0
[10]
Read SSR_FIFO.ORER, PER, DR, and FER
[11]
End

[6] Processing in response to an overrun error:
FRDRHL is read and a space is made in FRDRHL.

[7] When a break is detected, transfer of the receive data to FRDRHL stops after the detection. When the break ends at SEMR.RXDESEL = 0, the last stored data of FRDRHL is break error frame (all 0 data).

[8] Framing error processing / parity error processing:
All error occurrence data stored in FRDRHL is read.
(Or write 1 to FCR.RFRST to empty FRDRHL.)

[9] The reading of the receive data (when FCR.DRES is 1):
All receive data stored in FRDRHL is read.

[10] Clearing the error flag:
Write 0 to the error flag.

[11] Confirming that the error flag is cleared:
Read the error flag to confirm that its value is 0.

Figure 29.21 Example flow of serial reception in asynchronous mode with FIFO selected (2)

29.4 Multi-Processor Communications Function

The multi-processor communication function enables the SCI to transmit and receive data by sharing a communication line between multiple processors, using asynchronous serial communication in which the multi-processor bit is added. 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 the respective status flags RDRF, ORER, and FER in SSR

On receiving a 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. 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. The clock used for the multi-processor communication is the same as the clock used in non multi-processor asynchronous mode.

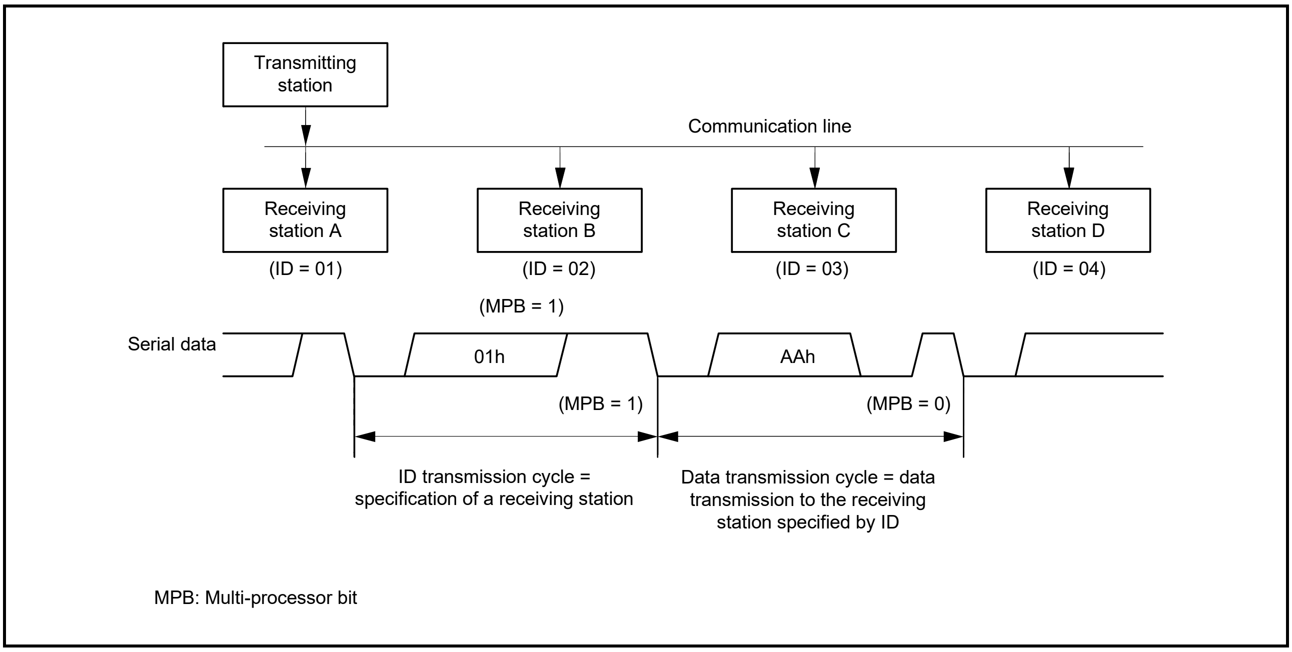

Transmitting station
Communication line
Receiving station A
Receiving station B
Receiving station C
Receiving station D
(ID = 01)
(ID = 02)
(ID = 03)
(ID = 04)
(MPB = 1)
Serial data
01h
AAh
(MPB = 1)
(MPB = 0)
ID transmission cycle = specification of a receiving station
Data transmission cycle = data transmission to the receiving station specified by ID
MPB: Multi-processor bit

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 the MPBT bit in FTDRHL register that corresponds to transmit data in the TDAT bit in FTDRHL register. For data reception, the multi-processor bit that is part of the receive data is written to the MPB bit in FRDRHL register and receive data is written to FRDRL register.

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 register
- Detection of a receive error
- Break
- Setting of the respective status flags RDF, ORER, and FER in SSR_FIFO register.

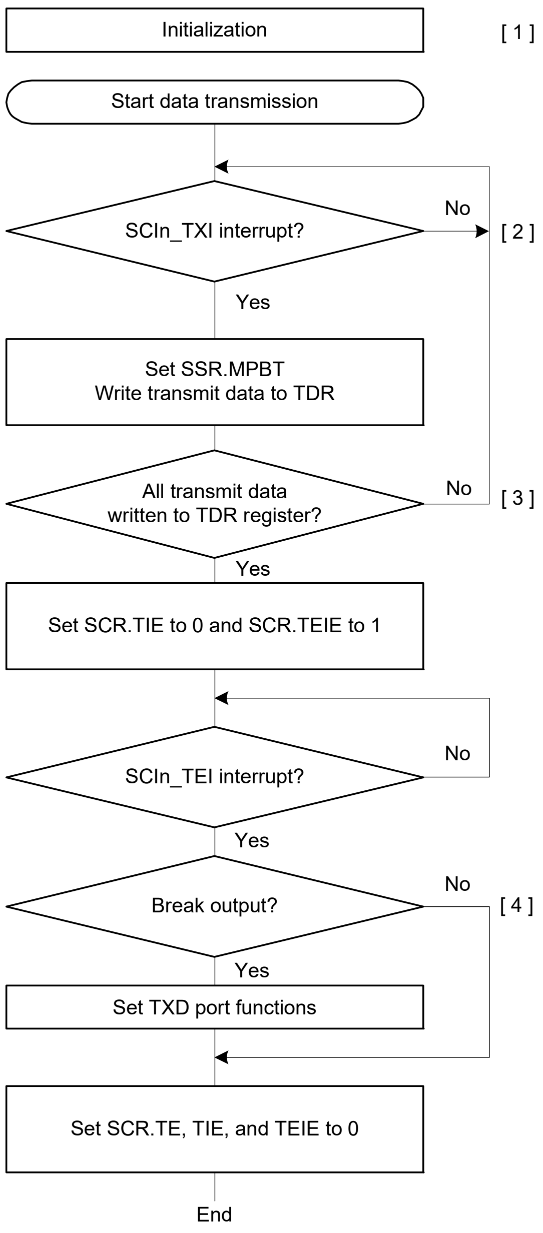
On receiving an 8-bit character in which the multi-processor bit is set to 1, the MPB bit in FRDRHL is set to 1 and receive data is written to the RDAT bit in FRDRHL. The MPIE bit in SCR is automatically cleared, therefore returning the SCI to non multi-processor reception operation. An SCIn_RXI interrupt 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 in operation from 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 in asynchronous mode.


Initialization
[1]
Start data transmission
SCIn_TXI interrupt?
No
[2]
Yes
Set SSR.MPBT
Write transmit data to TDR
All transmit data
written to TDR register?
No
[3]
Yes
Set SCR.TIE to 0 and SCR.TEIE to 1
SCIn_TEI interrupt?
No
Yes
Break output?
No
[4]
Yes
Set TXD port functions
Set SCR.TE, TIE, and TEIE to 0
End

[1] SCI initialization:
Set data transmission.
After SCR.TE is set to 1, 1 is output for a frame, and transmission is enabled.

[2] SCIn_TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (SCIn_TXI) request is generated.
Set SSR.MPBT to 0 or 1, and write transmit data to TDR 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 is 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.

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.

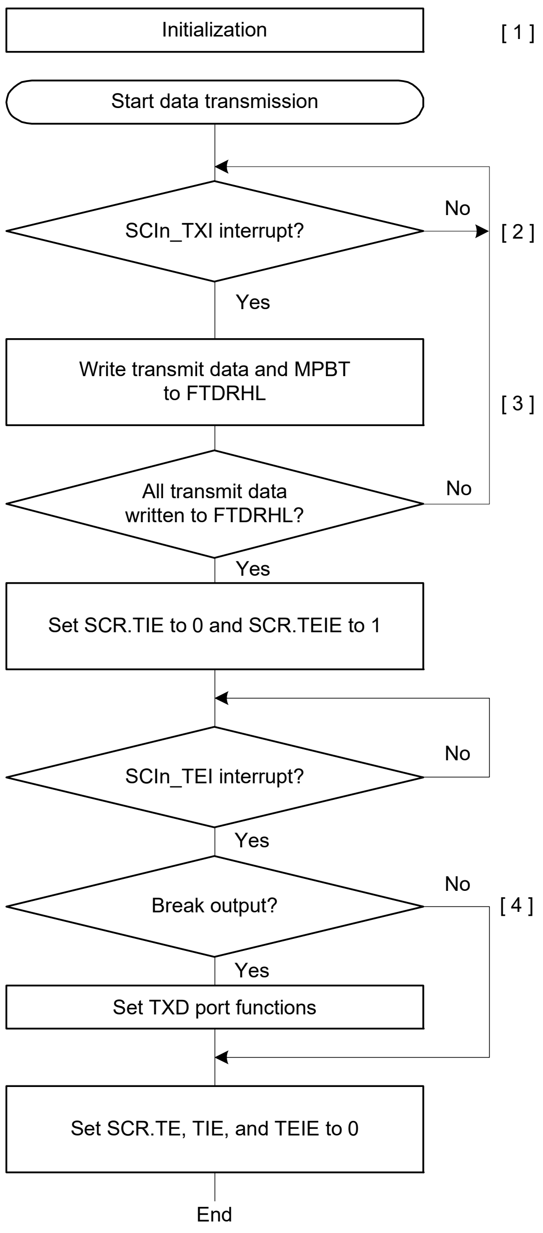
MPBT is set to 1 in FTDRH register. Data is set to FTDRH and FTDRL with the correct data length. Write 0 for unused bits. Write in order from FTDRH to FTDRL.

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 FIFO selected.


Initialization
[1]
Start data transmission
SCIn_TXI interrupt?
No
[2]
Yes
Write transmit data and MPBT to FTDRHL
[3]
All transmit data written to FTDRHL?
No
Yes
Set SCR.TIE to 0 and SCR.TEIE to 1
SCIn_TEI interrupt?
No
Yes
Break output?
No
[4]
Yes
Set TXD port functions
Set SCR.TE, TIE, and TEIE to 0
End

[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:
For data transmission 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.

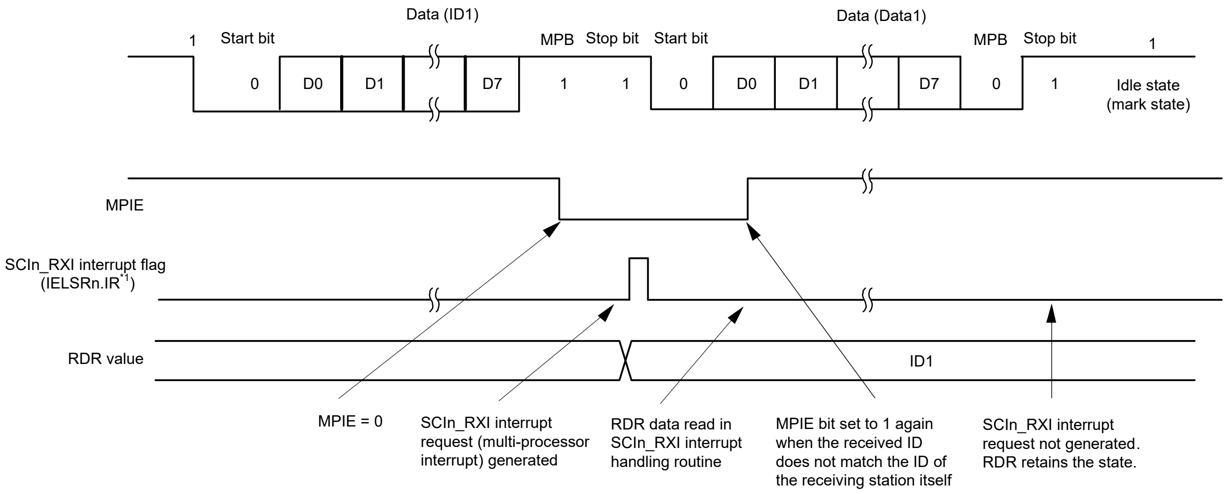
Figure 29.25 Example flow of serial transmission in multi-processor mode with FIFO selected

29.4.2 Multi-Processor Serial Data Reception

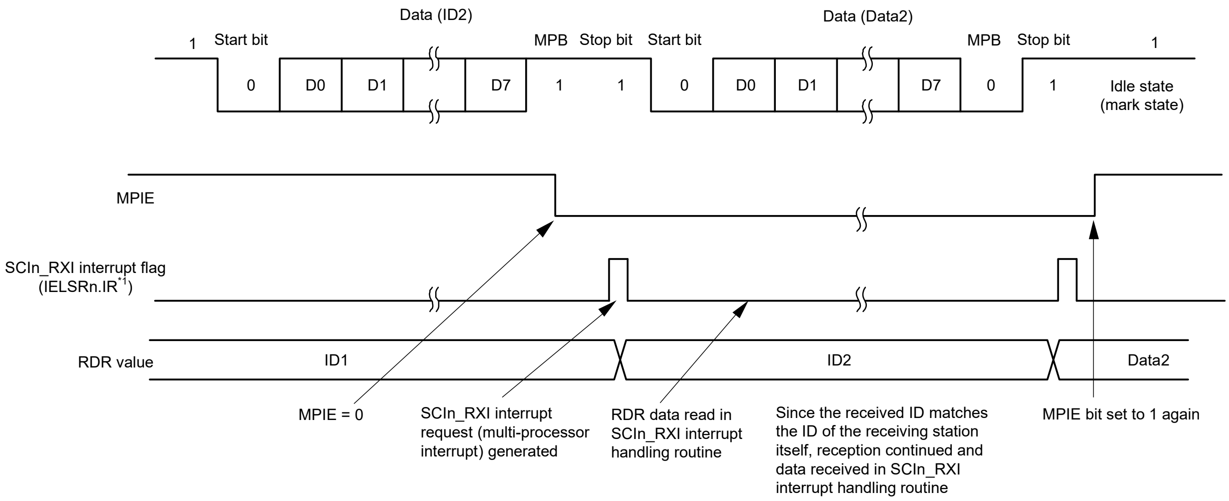
(1) Non-FIFO selected

Figure 29.26 and Figure 29.27 show 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, or 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 in asynchronous mode.

Figure 29.26 shows an example operation for data reception.


Data (ID1)
Data (Data1)
1
Start bit
MPB
Stop bit
Start bit
MPB
Stop bit
1
0
D0
D1
D7
1
1
0
D0
D1
D7
0
1
Idle state
(mark state)
MPIE
SCIn_RXI interrupt flag
(IELSRn.IR*1)
RDR value
ID1
MPIE = 0
SCIn_RXI interrupt request (multi-processor interrupt) generated
RDR data read in SCIn_RXI interrupt handling routine
MPIE bit set to 1 again when the received ID does not match the ID of the receiving station itself
SCIn_RXI interrupt request not generated. RDR retains the state.

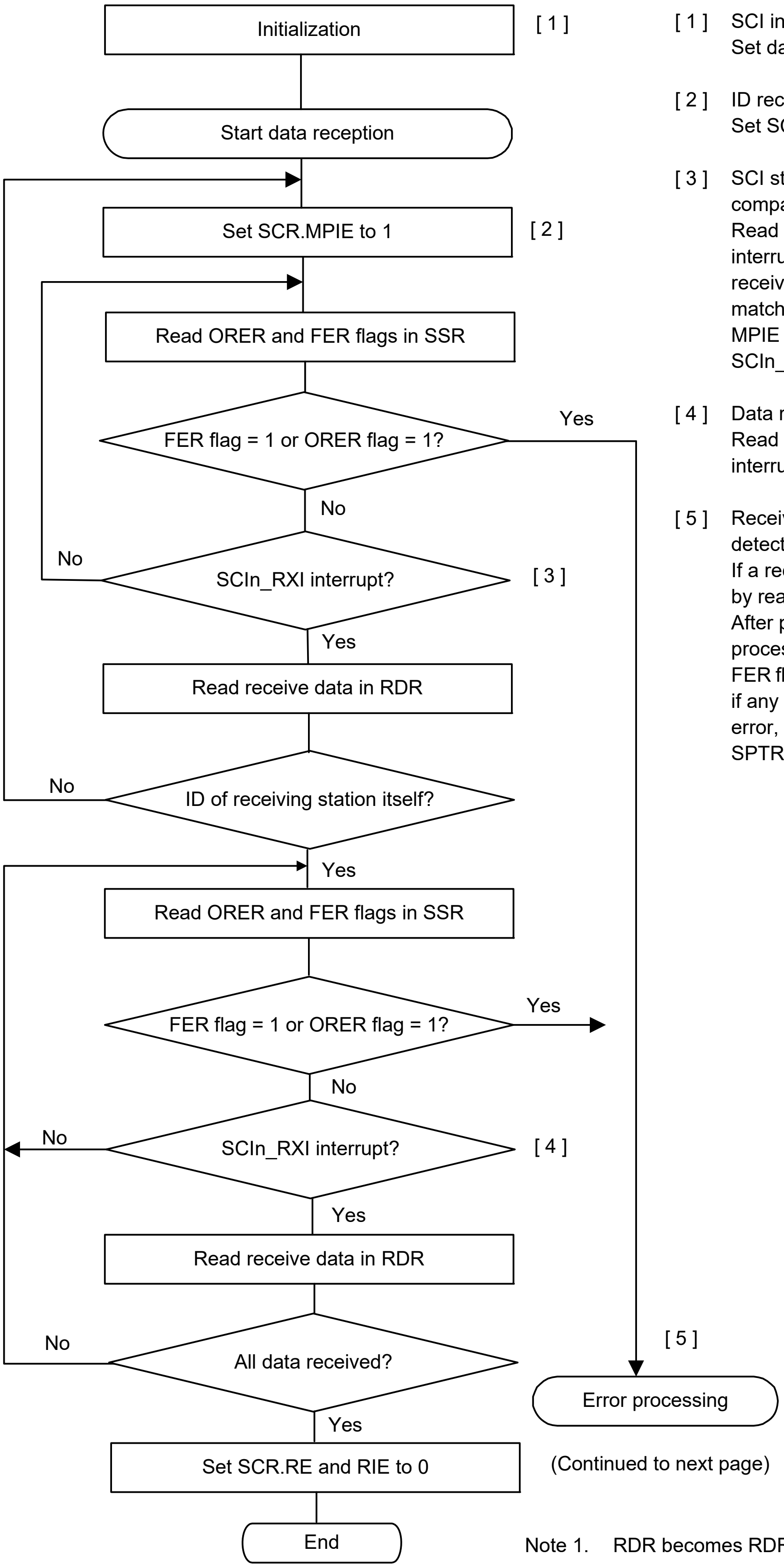
(a) When the received ID does not match the ID of the receiving station itself


Data (ID2)
Data (Data2)
1
Start bit
MPB
Stop bit
Start bit
MPB
Stop bit
1
0
D0
D1
D7
1
1
0
D0
D1
D7
0
1
Idle state
(mark state)
MPIE
SCIn_RXI interrupt flag
(IELSRn.IR*1)
RDR value
ID1
ID2
Data2
MPIE = 0
SCIn_RXI interrupt request (multi-processor interrupt) generated
RDR data read in SCIn_RXI interrupt handling routine
Since the received ID matches the ID of the receiving station itself, reception continued and data received in SCIn_RXI interrupt handling routine
MPIE bit set to 1 again

(b) When the received ID matches the ID of the receiving station itself

Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 29.26 Example of SCI reception with 8-bit data, multi-processor bit, and 1 stop bit


Initialization
[1]
Start data reception
Set SCR.MPIE to 1
[2]
Read ORER and FER flags in SSR
FER flag = 1 or ORER flag = 1?
Yes
No
No
SCIn_RXI interrupt?
[3]
Yes
Read receive data in RDR
No
ID of receiving station itself?
Yes
Read ORER and FER flags in SSR
FER flag = 1 or ORER flag = 1?
Yes
No
No
SCIn_RXI interrupt?
[4]
Yes
Read receive data in RDR
No
All data received?
[5]
Yes
Error processing
Set SCR.RE and RIE to 0
End

[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:
Read data in RDR*1 on the first SCIn_RXI interrupt and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, set MPIE to 1 again, and wait for another SCIn_RXI interrupt request.

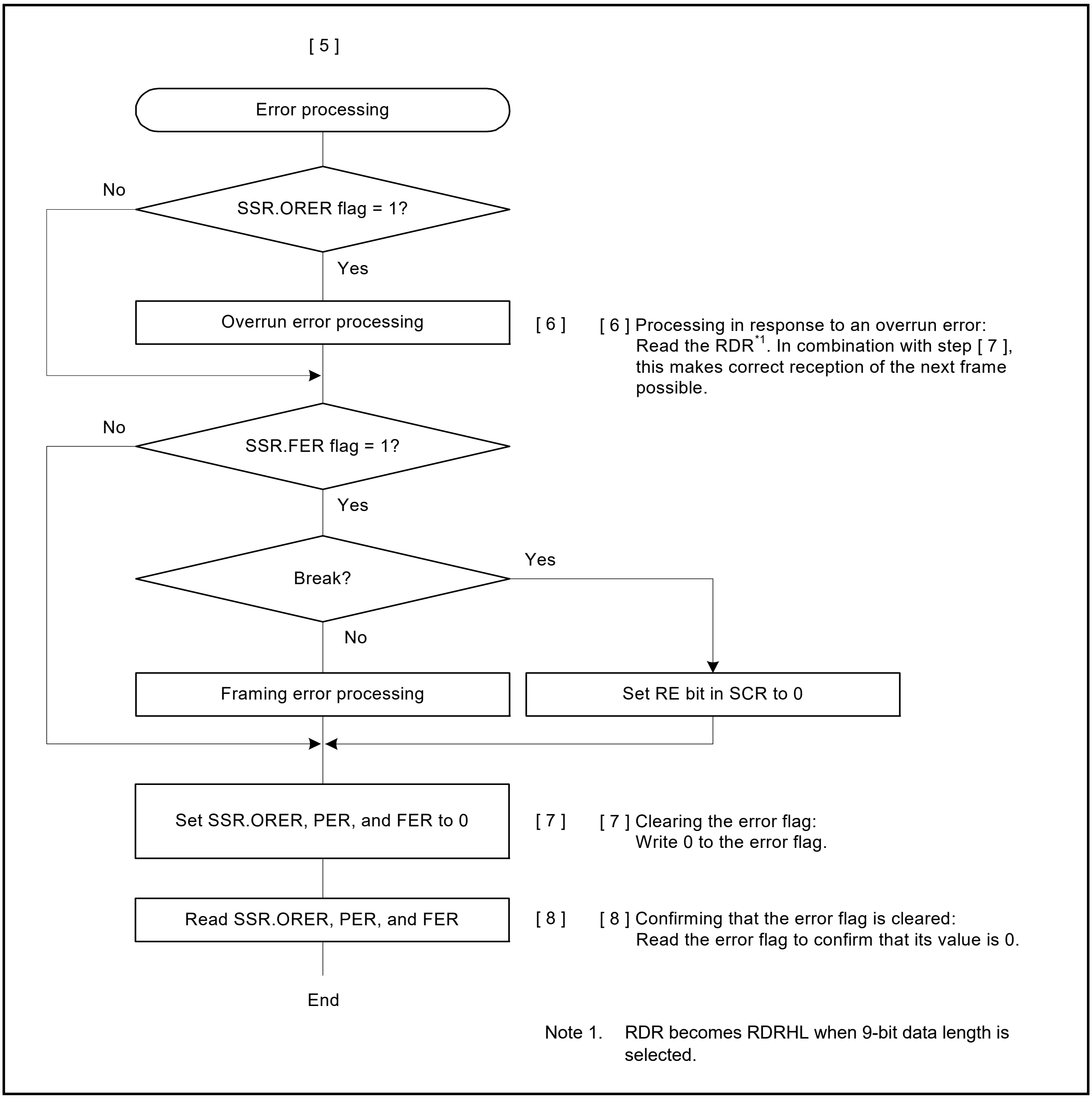
[4] Data reception at an SCIn_RXI interrupt:
Read data in RDR*1 in the SCIn_RXI interrupt routine.

[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. After performing the appropriate error processing, be sure to set the ORER and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. For a framing error, a break can be detected by reading SPTR.RXDMON bit.

(Continued to next page)

Note 1. RDR becomes RDRHL when 9-bit data length is selected.

Figure 29.27 Example flow of multi-processor serial reception(1) with non-FIFO selected


[5]
Error processing
No
SSR.ORER flag = 1?
Yes
Overrun error processing
[6]
No
SSR.FER flag = 1?
Yes
Break?
Yes
No
Framing error processing
Set RE bit in SCR to 0
Set SSR.ORER, PER, and FER to 0
[7]
Read SSR.ORER, PER, and FER
[8]
End

[6] Processing in response to an overrun error:
Read the RDR[*1]. 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.

Note 1. RDR becomes RDRHL when 9-bit data length is selected.

Figure 29.28 Example flow of multi-processor serial reception (2) with non-FIFO selected

(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 RDF, ORER, and DR flags in FRDRH always reflect the associated flags in the SSR_FIFO register.

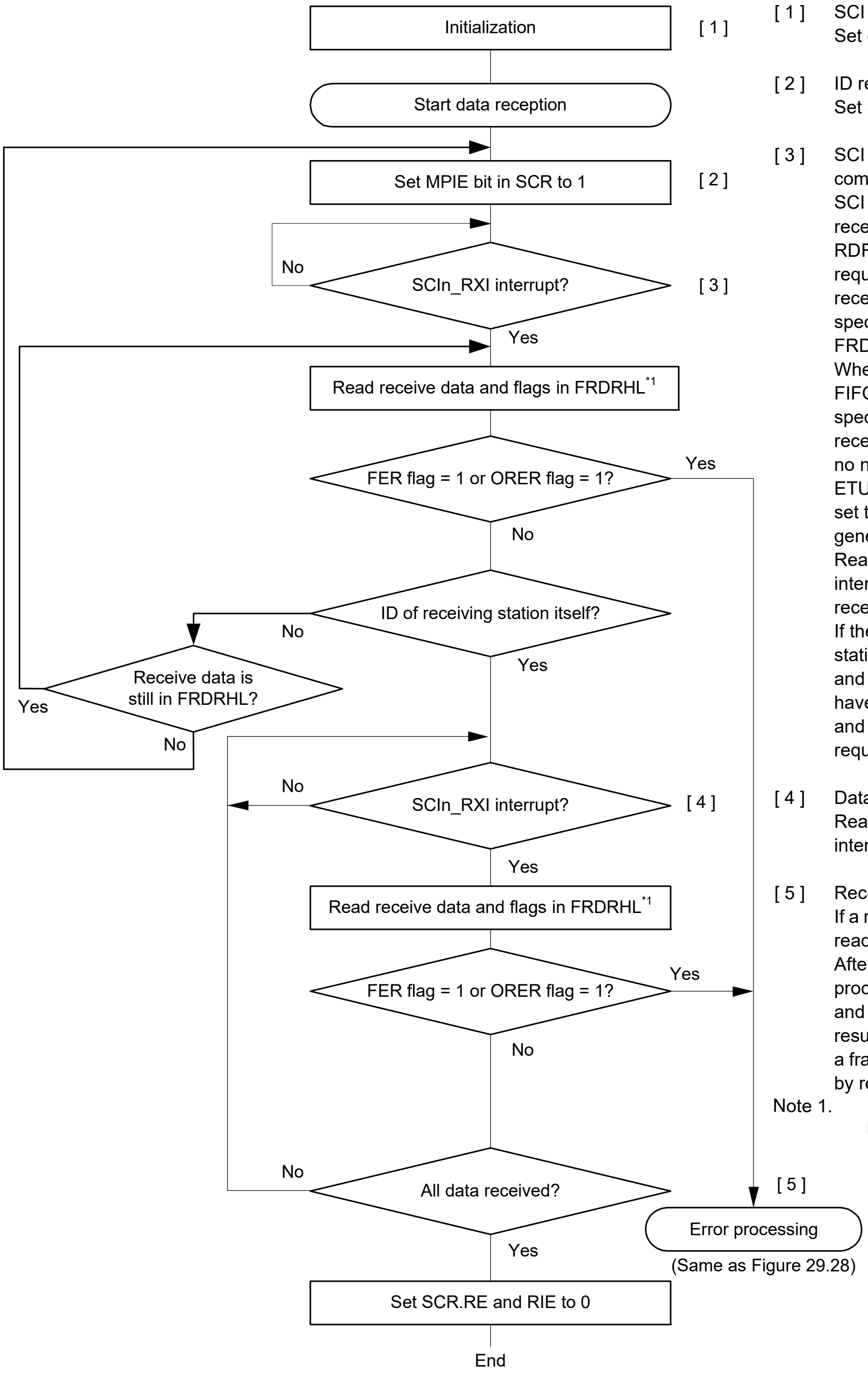
Data Length	Register Setting		Receive data in FRDRH, FRDRL															
			FRDRHL															
			FRDRH								FRDRL							
	SCMR. CHR1	SMR. CHR	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	0	—	RDF	ORER	FER	0	DR	MPB	0	0	7-bit receive data						
8 bits	1	1	—	RDF	ORER	FER	0	DR	MPB	0	8-bit 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 an example flow of multi-processor data reception with FIFO selected. When the MPIE bit in SCR is set to 1, reading communication data is skipped until the 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 the associated errors are transferred to FRDRHL. The MPIE bit in SCR register is automatically cleared and non multi-processor reception continues.

If a frame error occurs and the FER flag in SSR_FIFO is set to 1, the SCI continues data reception. The rest of the operations are the same as in asynchronous mode with FIFO selected.


Initialization
[1]
Start data reception
Set MPIE bit in SCR to 1
[2]
No
SCIn_RXI interrupt?
[3]
Yes
Read receive data and flags in FRDRHL*1
FER flag = 1 or ORER flag = 1?
Yes
No
ID of receiving station itself?
No
Yes
Receive data is still in FRDRHL?
Yes
No
No
SCIn_RXI interrupt?
[4]
Yes
Read receive data and flags in FRDRHL*1
FER flag = 1 or ORER flag = 1?
Yes
No
No
All data received?
Yes
Set SCR.RE and RIE to 0
End
[5]
Error processing
(Same as Figure 29.28)

[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 is equal to or greater than the specified receive triggering number 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 is 1, and compare the next ID. If the data does not have 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 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. When a framing error occurs, 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.

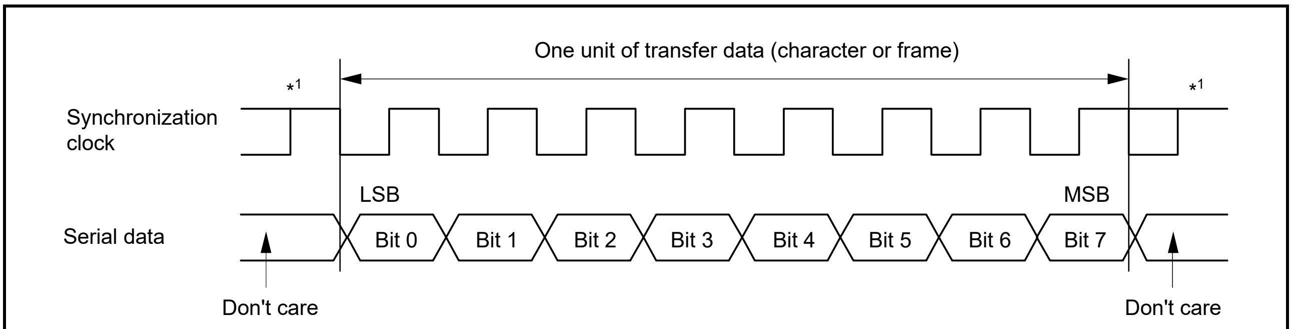
In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In clock synchronous mode, no parity bit can be added.

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 the output state. When the CKPH bit in SPMR register 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 perform 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.



Note 1. Holds a high level except during continuous transfer.

Figure 29.31 Data format in clock synchronous serial communications with 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 setting of the CKE[1:0] bits in the SCR register.

When the SCI operates on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character. When no transfer is performed, the clock is held high. However, when only data reception is performed while the CTS function is disabled, the synchronization clock output starts when the RE bit in SCR is set to 1. The synchronization clock stops when it is held high*1 and when an overrun error occurs, or when the RE bit is set to 0.

When only data reception is performed and the CTS function is enabled, the clock output does not start when the RE is set to 1 and the CTSn_RTSn input is high. The synchronization clock output starts when the RE bit 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 RE bit is set to 0.

Note 1. The signal is held high when SPMR.CKPH bit is 0 and SPMR.CKPOL bit is 0, or when SPMR.CKPH bit is 1 and SPMR.CKPOL bit is 1.
It is held low when SPMR.CKPH bit is 0 and SPMR.CKPOL bit is 1, or when SPMR.CKPH bit is 1 and SPMR.CKPOL bit is 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 the CTSE bit in SPMR register 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 TSR (when SCR.TE is 1)
- The SSR.ORER flag 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 SSR_FIFO.ORER flag is 0.

[Condition for high output]

(a) Non-FIFO selected

- The conditions for low output are not satisfied
- After reception is complete, if it is terminated with SCR.RE = 0 without reading the RDR register, then RTS remains high. Read the SCR register for dummy after writing SCR.RE = 0.

(b) FIFO selected

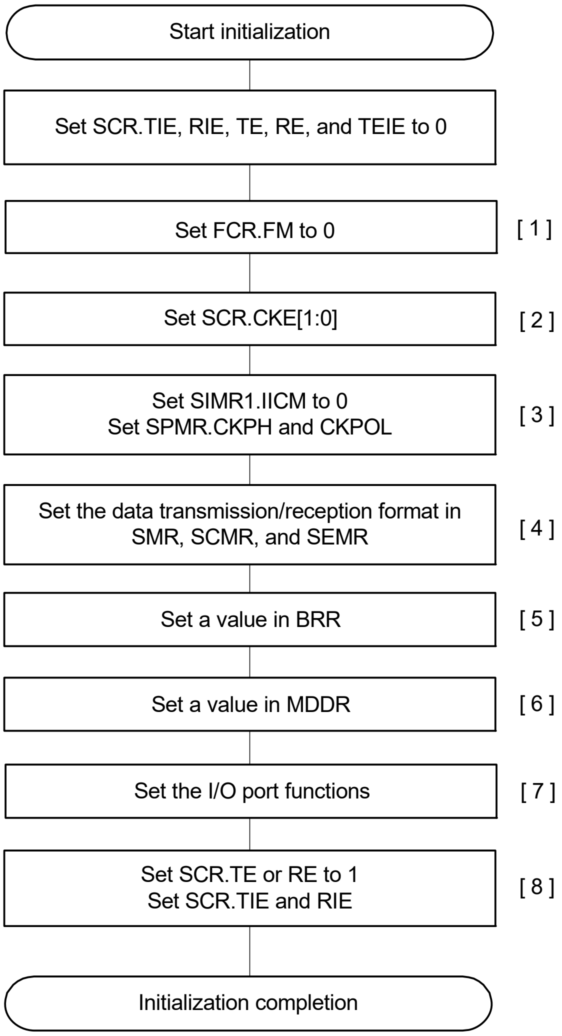
The conditions for low output are not satisfied.

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 TE bit in the SCR register 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 TIE bit is 1 generates an SCIn_TXI interrupt request.


Start initialization
Set SCR.TIE, RIE, TE, RE, and TEIE to 0
Set FCR.FM to 0
[1]
Set SCR.CKE[1:0]
[2]
Set SIMR1.IICM to 0
Set SPMR.CKPH and CKPOL
[3]
Set the data transmission/reception format in SMR, SCMR, and SEMR
[4]
Set a value in BRR
[5]
Set a value in MDDR
[6]
Set the I/O port functions
[7]
Set SCR.TE or RE to 1
Set SCR.TIE and RIE
[8]
Initialization completion

[1] Set FCR.FM to 0.

[2] Set the clock selection in SCR.

[3] Set SIMR1.IICM to 0.
Set SPMR.CKPH and CKPOL.
Step [3] can be skipped if the values have not changed from the initial values.

[4] Set data transmission/reception format in SMR, SCMR, and SEMR.

[5] Write a value associated with the bit rate to BRR.
This step is not required if an external clock is used.

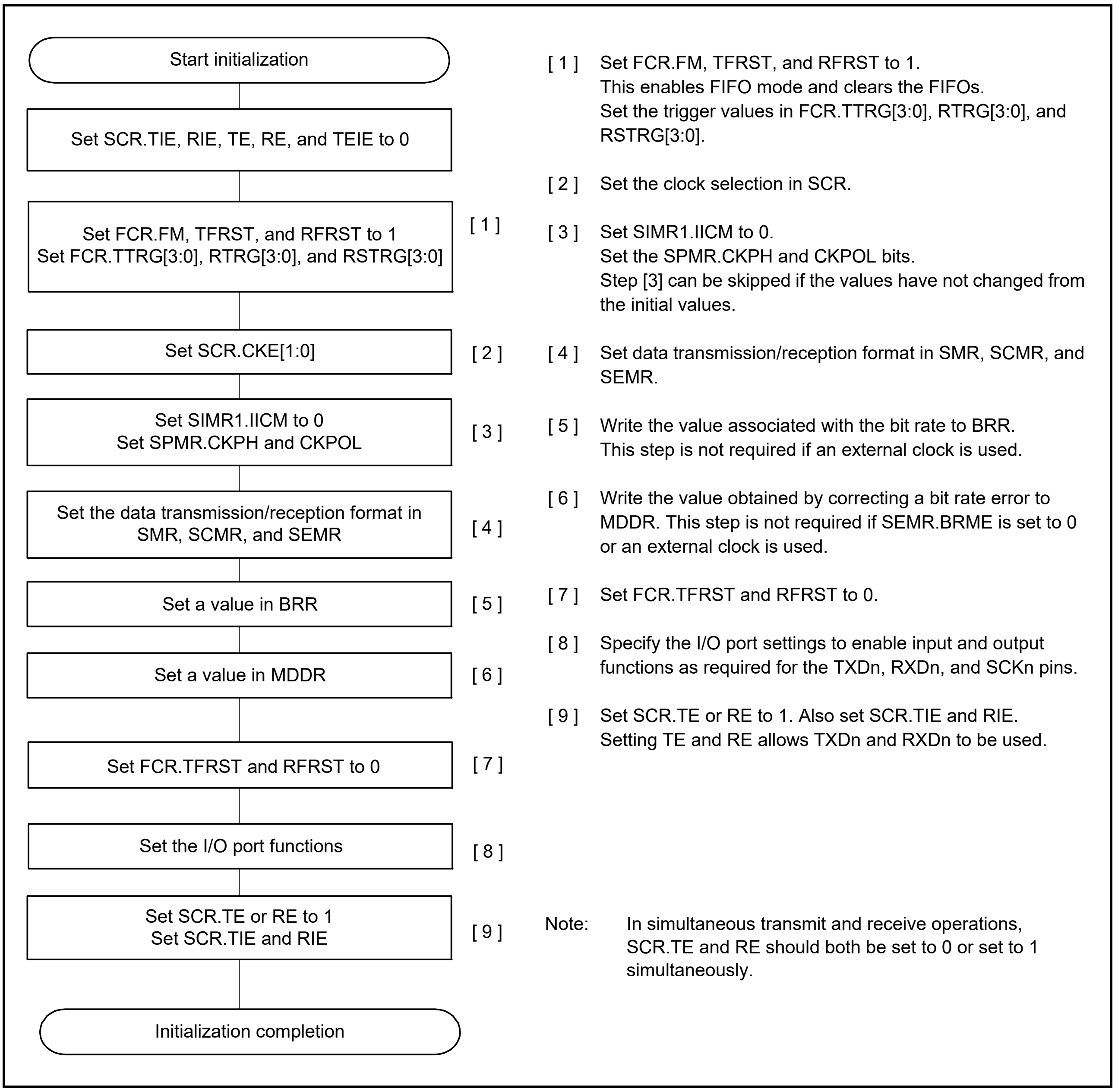
[6] Write the value obtained by correcting a bit rate error in MDDR. This step is not required if SEMR.BRME is set to 0 or an external clock is used.

[7] Specify the I/O port settings to enable input and output functions as required for the TXDn, RXDn, and SCKn pins.

[8] Set SCR.TE or RE to 1. Also set SCR.TIE and RIE.
Setting TE and RE allows TXDn and RXDn to be used.

Note: In simultaneous transmit and receive operations, the SCR.TE and SCR.RE must both be set to 0 or set to 1 simultaneously.

Figure 29.32 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected


Start initialization
Set SCR.TIE, RIE, TE, RE, and TEIE to 0
Set FCR.FM, TFRST, and RFRST to 1
Set FCR.TTRG[3:0], RTRG[3:0], and RSTRG[3:0]
[1]
Set SCR.CKE[1:0]
[2]
Set SIMR1.IICM to 0
Set SPMR.CKPH and CKPOL
[3]
Set the data transmission/reception format in SMR, SCMR, and SEMR
[4]
Set a value in BRR
[5]
Set a value in MDDR
[6]
Set FCR.TFRST and RFRST to 0
[7]
Set the I/O port functions
[8]
Set SCR.TE or RE to 1
Set SCR.TIE and RIE
[9]
Initialization completion

[1] Set FCR.FM, TFRST, and RFRST to 1.
This enables FIFO mode and clears the FIFOs.
Set the trigger values in FCR.TTRG[3:0], RTRG[3:0], and RSTRG[3:0].

[2] Set the clock selection in SCR.

[3] Set SIMR1.IICM to 0.
Set the SPMR.CKPH and CKPOL bits.
Step [3] can be skipped if the values have not changed from the initial values.

[4] Set data transmission/reception format in SMR, SCMR, and SEMR.

[5] Write the value associated with the bit rate to BRR.
This step is not required if an external clock is used.

[6] Write the value obtained by correcting a bit rate error to MDDR. This step is not required if SEMR.BRME is set to 0 or an external clock is used.

[7] Set FCR.TFRST and RFRST to 0.

[8] Specify the I/O port settings to enable input and output functions as required for the TXDn, RXDn, and SCKn pins.

[9] Set SCR.TE or RE to 1. Also set SCR.TIE and RIE.
Setting TE and RE allows TXDn and RXDn to be used.

Note: In simultaneous transmit and receive operations, SCR.TE and RE should both be set to 0 or set to 1 simultaneously.

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 the TE bit in SCR 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 the TIE bit in SCR is set to 1, an SCIn_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to TDR in the SCIn_TXI interrupt handling routine before transmission of the current transmit data finishes. When SCIn_TEI interrupt requests are in use, set the TIE bit in SCR to 0 and the TEIE bit in SCR to 1 after the last of the

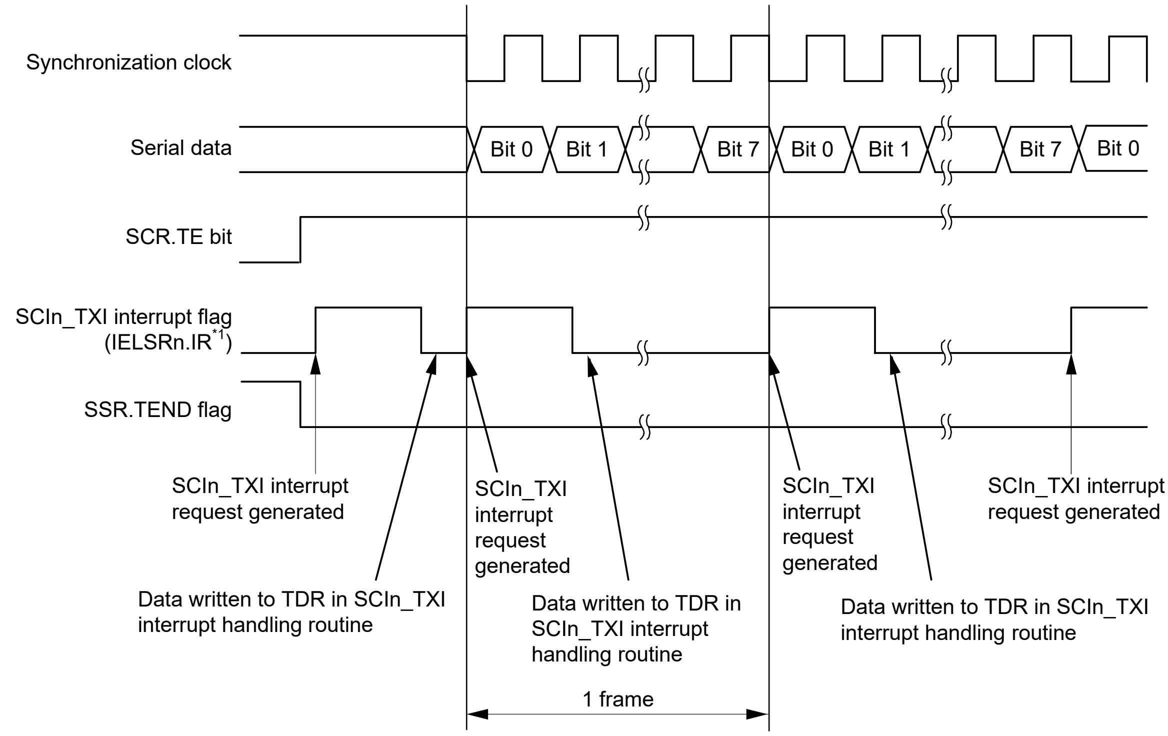
data to be transmitted is written to TDR.

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 signal is low while the CTSE bit in SPMR register is 1.
4. The SCI checks for updates 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 TEND flag in SSR register is set to 1. The TXDn pin keeps the output state of the last bit. If the TEIE bit in SCR register 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.

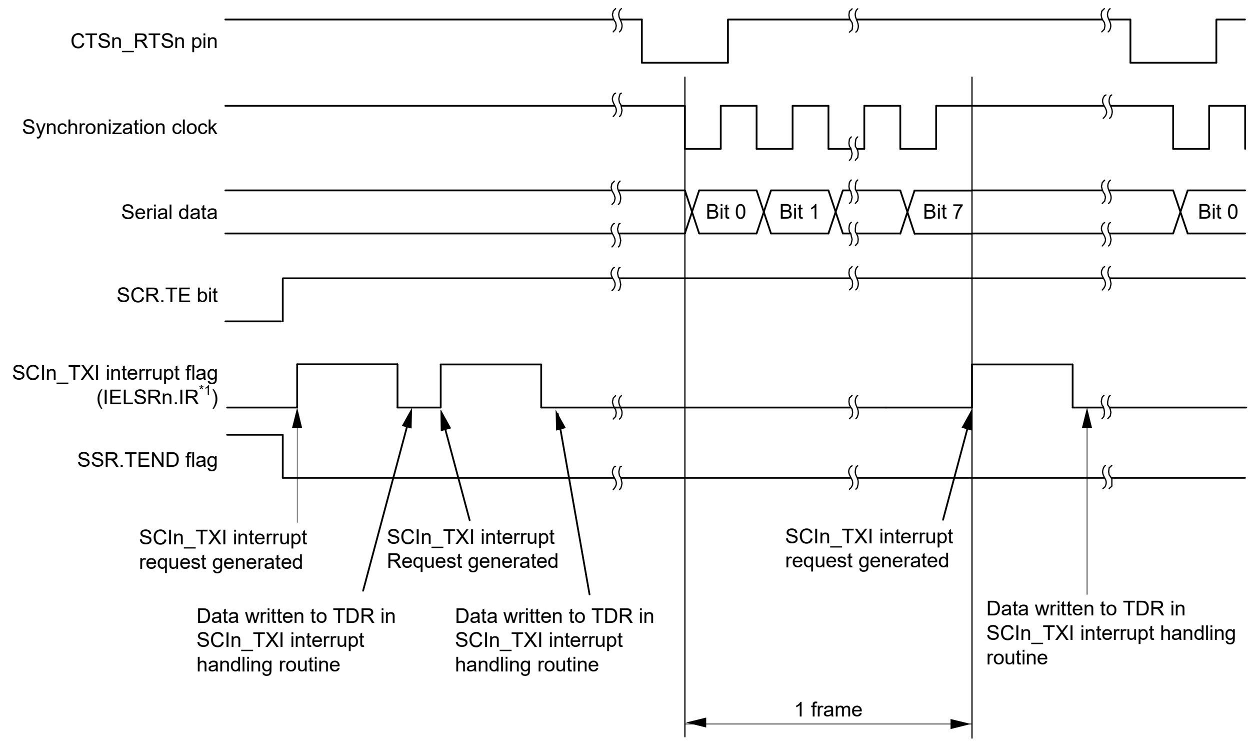
Transmission does not start while a receive error flag (ORER, FER, or PER in SSR) is set to 1. Be sure to set the receive error flags to 0 before starting transmission.

Note: Setting the RE bit in SCR register to 0 does not clear the receive error flags.


Synchronization clock
Serial data
Bit 0
Bit 1
Bit 7
Bit 0
Bit 1
Bit 7
Bit 0
SCR.TE bit
SCIn_TXI interrupt flag (IELSRn.IR*1)
SSR.TEND flag
SCIn_TXI interrupt request generated
Data written to TDR in SCIn_TXI interrupt handling routine
SCIn_TXI interrupt request generated
Data written to TDR in SCIn_TXI interrupt handling routine
SCIn_TXI interrupt request generated
Data written to TDR in SCIn_TXI interrupt handling routine
SCIn_TXI interrupt request generated
1 frame

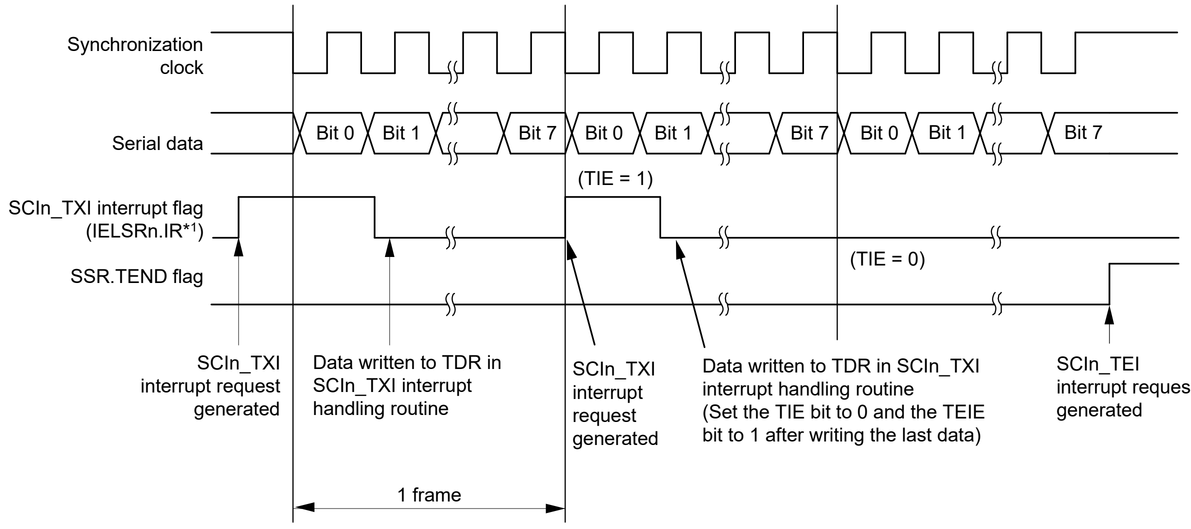
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 29.34 Example of serial data transmission in clock synchronous mode when the CTS function is not used at the beginning of transmission


CTSn_RTSn pin
Synchronization clock
Serial data
Bit 0
Bit 1
Bit 7
Bit 0
SCR.TE bit
SCIn_TXI interrupt flag (IELSRn.IR*1)
SSR.TEND flag
SCIn_TXI interrupt request generated
SCIn_TXI interrupt Request generated
SCIn_TXI interrupt request generated
Data written to TDR in SCIn_TXI interrupt handling routine
Data written to TDR in SCIn_TXI interrupt handling routine
Data written to TDR in SCIn_TXI interrupt handling routine
1 frame

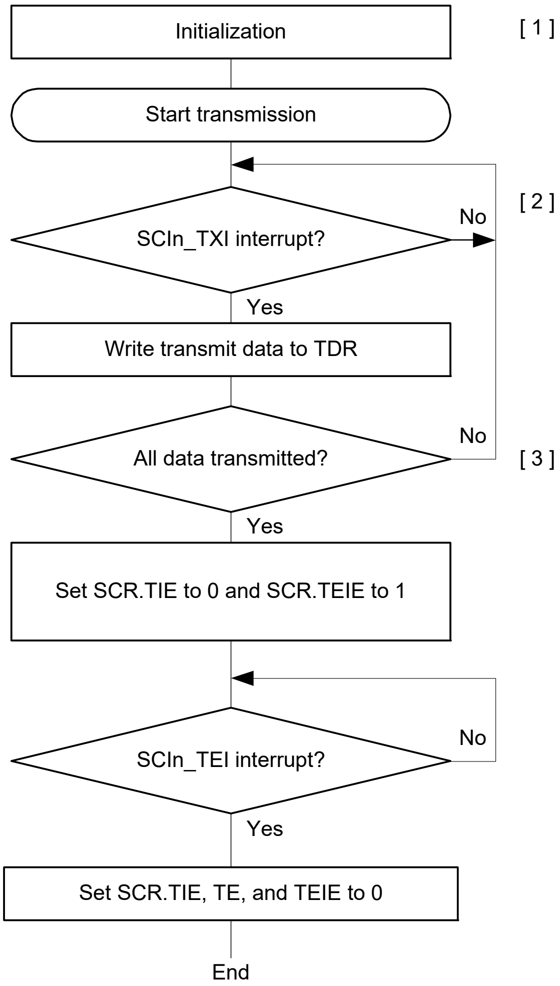
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 29.35 Example of serial data transmission in clock synchronous mode when the CTS function is used at the beginning of transmission


Synchronization clock
Serial data
Bit 0
Bit 1
Bit 7
Bit 0
Bit 1
Bit 7
Bit 0
Bit 1
Bit 7
(TIE = 1)
(TIE = 0)
SCIn_TXI interrupt flag (IELSRn.IR*1)
SSR.TEND flag
SCIn_TXI interrupt request generated
Data written to TDR in SCIn_TXI interrupt handling routine
SCIn_TXI interrupt request generated
Data written to TDR in SCIn_TXI interrupt handling routine (Set the TIE bit to 0 and the TEIE bit to 1 after writing the last data)
SCIn_TEI interrupt request generated
1 frame

Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the associated 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 might 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.

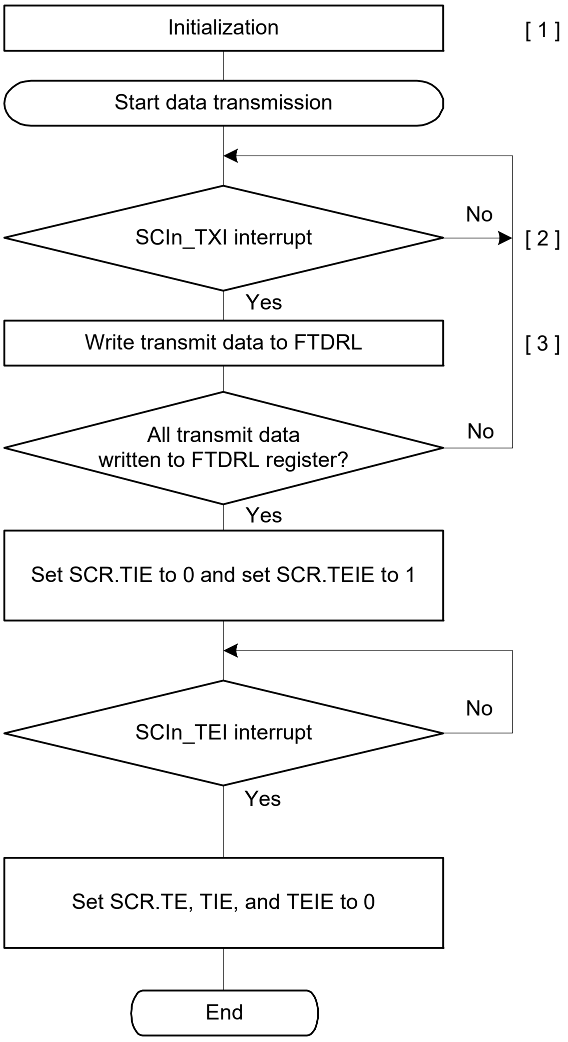
In serial data transmission, the SCI operates as follows:

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 TDFE flag in SSR_FIFO is set to 1. When the TIE bit in SCR 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 the TIE bit in SCR to 0 and the TEIE bit in SCR 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 while the CTSE bit in SPMR is 1.

4. The SCI checks whether non-transmitted data remains in FTDRL on 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 TEND flag in SSR_FIFO is set to 1. The TXDn pin keeps the output state of the last bit. If the EIE bit in SCR 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.


Initialization
[1]
Start data transmission
SCIn_TXI interrupt
No
[2]
Yes
Write transmit data to FTDRL
[3]
All transmit data written to FTDRL register?
No
Yes
Set SCR.TIE to 0 and set SCR.TEIE to 1
SCIn_TEI interrupt
No
Yes
Set SCR.TE, TIE, and TEIE to 0
End

[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:
For transmission data from FTDRL to TSR, when the amount of transmit data written in FTDRL is equal to or less than the specified transmit triggering number, a transmit FIFO data 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 might 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

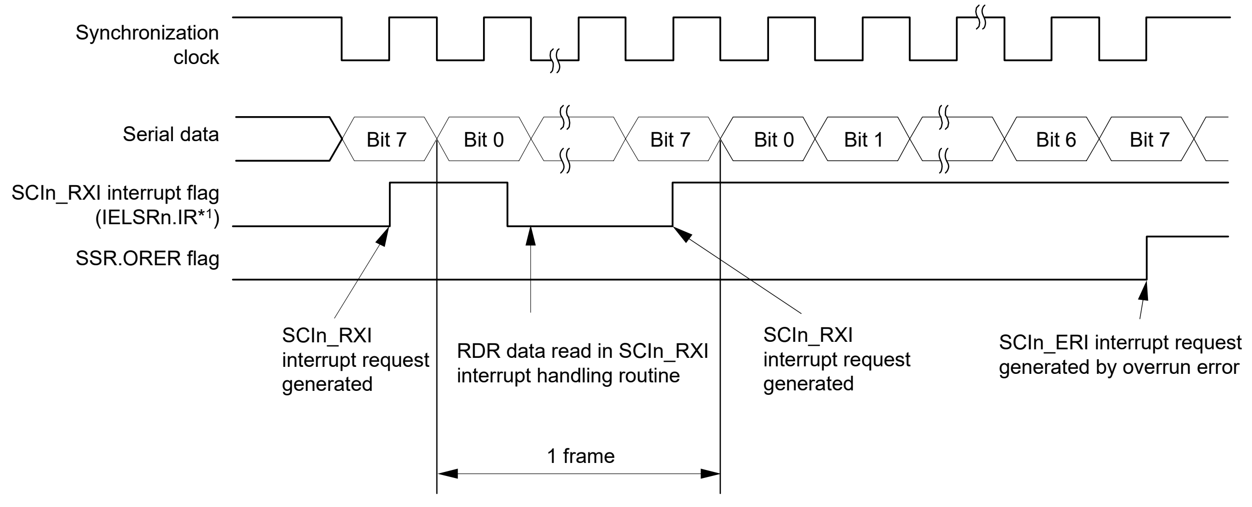
29.5.5 Serial Data Reception in Clock Synchronous Mode

(1) Non-FIFO selected

Figure 29.39 and Figure 29.40 show examples of SCI operation for serial reception in clock synchronous mode.

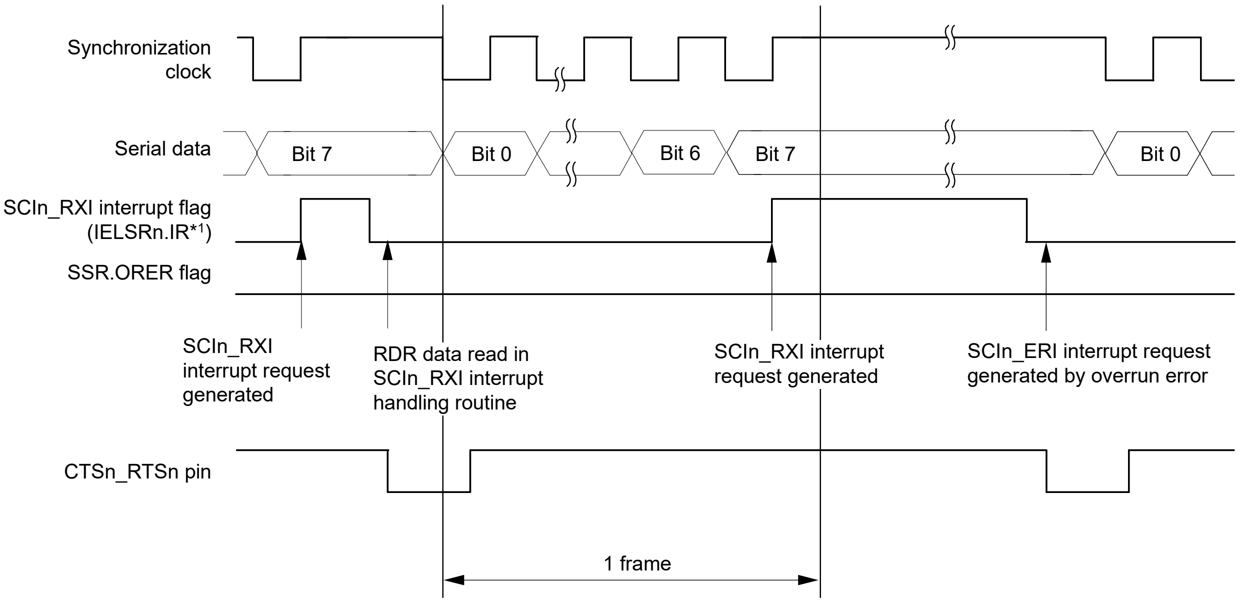
In serial data reception, the SCI operates as follows:

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 receive 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.


Synchronization clock
Serial data
Bit 7
Bit 0
Bit 7
Bit 0
Bit 1
Bit 6
Bit 7
SCIn_RXI interrupt flag (IELSRn.IR*1)
SSR.ORER flag
SCIn_RXI interrupt request generated
RDR data read in SCIn_RXI interrupt handling routine
SCIn_RXI interrupt request generated
SCIn_ERI interrupt request generated by overrun error
1 frame

Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 29.39 Example operation of serial reception in clock synchronous mode (1) when RTS function is not used

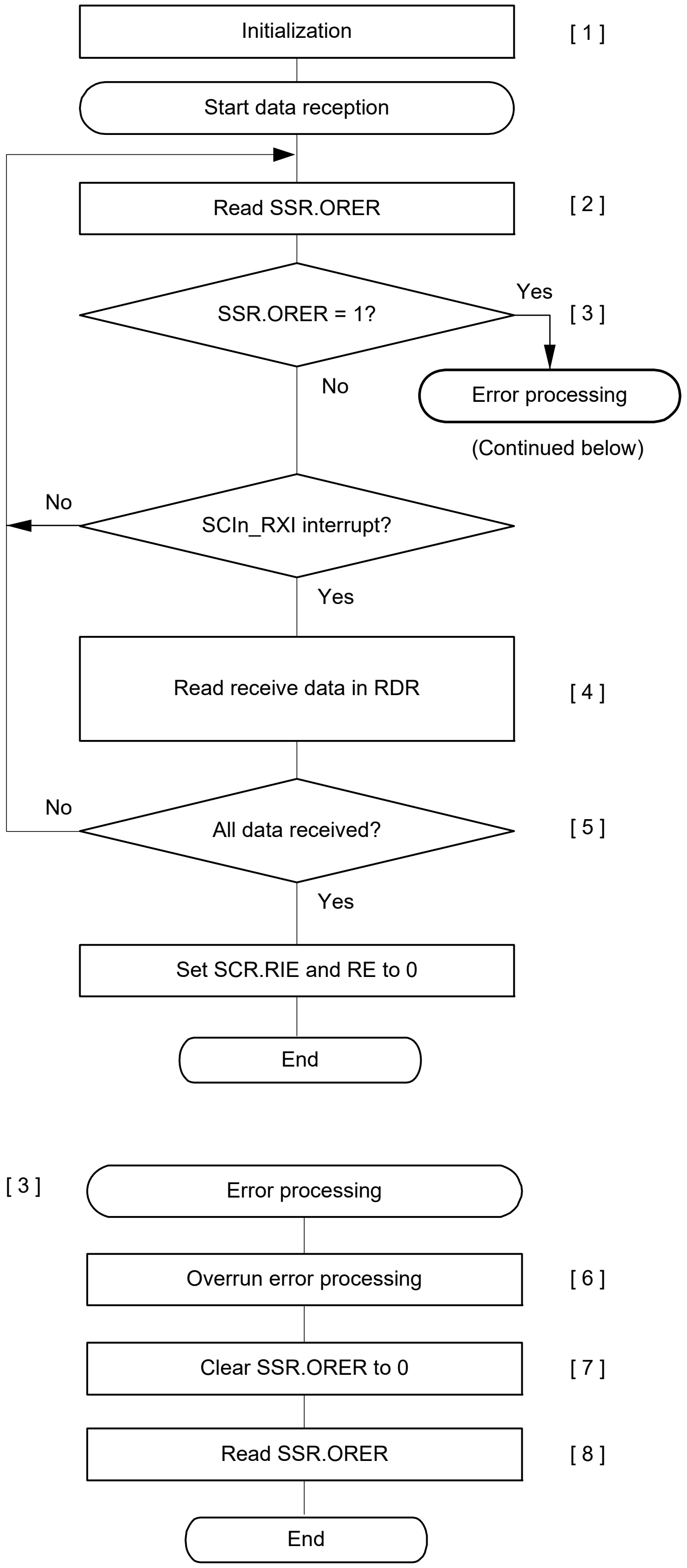

Synchronization clock
Serial data
Bit 7
Bit 0
Bit 6
Bit 7
Bit 0
SCIn_RXI interrupt flag (IELSRn.IR*1)
SSR.ORER flag
SCIn_RXI interrupt request generated
RDR data read in SCIn_RXI interrupt handling routine
SCIn_RXI interrupt request generated
SCIn_ERI interrupt request generated by overrun error
CTSn_RTSn pin
1 frame

Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 29.40 Example operation of serial reception in clock synchronous mode (2) when RTS function is used

Data transfer cannot resume while the receive error flag is 1. Therefore, clear the ORER, FER, and PER flags in SSR to 0 before resuming data reception. Additionally, be sure to read the RDR during overrun error processing. When a data reception is forcibly terminated by setting the RE bit in SCR to 0 during operation, read the RDR because the received data that is not read yet might be left in the RDR.

Figure 29.41 shows an example flow of serial data reception.


Initialization
[1]
Start data reception
Read SSR.ORER
[2]
SSR.ORER = 1?
Yes
[3]
No
Error processing
(Continued below)
No
SCIn_RXI interrupt?
Yes
Read receive data in RDR
[4]
No
All data received?
[5]
Yes
Set SCR.RIE and RE to 0
End
[3]
Error processing
Overrun error processing
[6]
Clear SSR.ORER to 0
[7]
Read SSR.ORER
[8]
End

[1]	SCI initialization: Set the input port for pins to be used as RXDn pins.
[2]	[3] Receive error processing: If a receive error occurs, read SSR.ORER, perform the relevant error processing, and set SSR.ORER to 0. Data reception cannot resume while SSR.ORER is 1.
[4]	Read the receive data in RDR 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 enables correct reception of the next frame possible.
[7]	Clearing the error flag: Write 0 to the error flag.
[8]	Confirming that the error flag is cleared: Read the error flag to confirm that its value is 0.

Figure 29.41 Example flow of serial reception in clock synchronous mode with non-FIFO selected

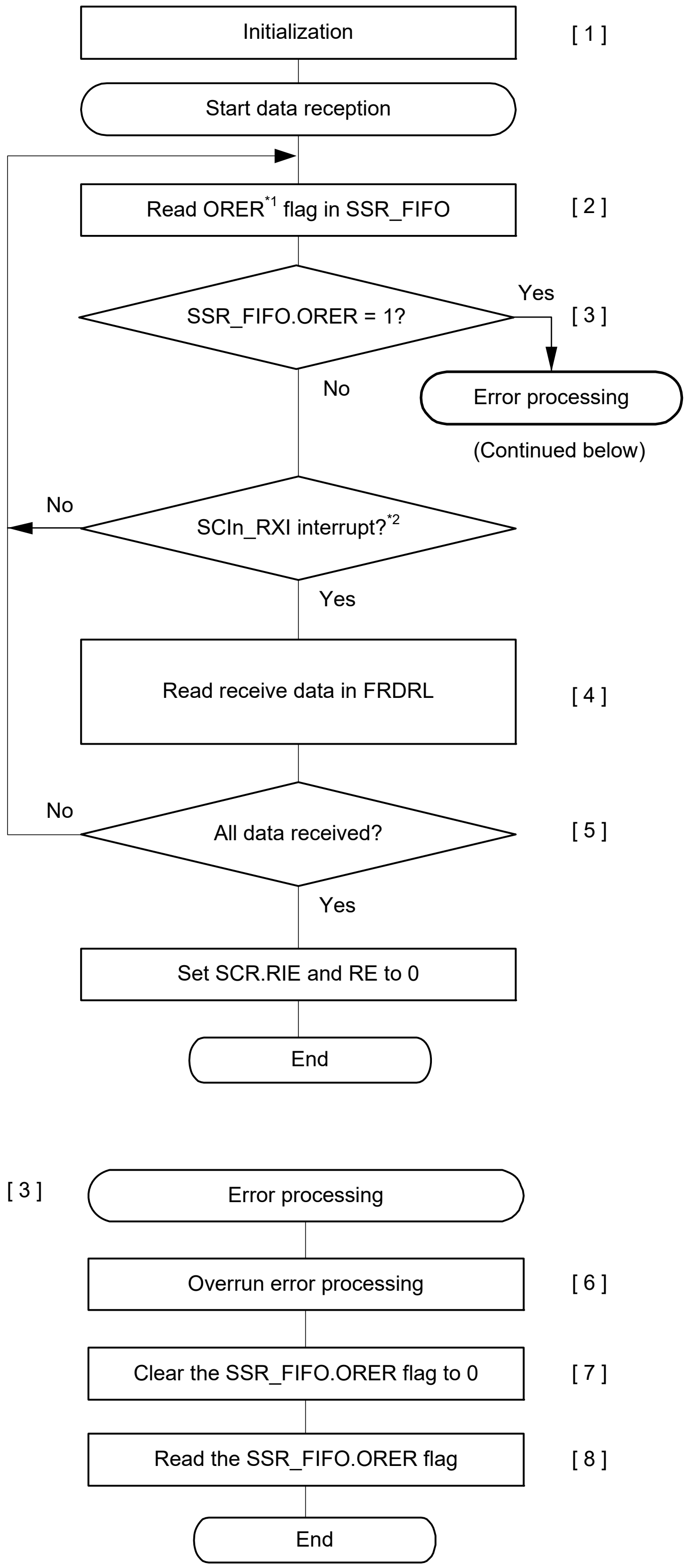
(2) FIFO selected

Figure 29.42 shows an example of serial reception in clock synchronous mode with FIFO 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 the RIE bit in SCR is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to FRDRL*1.
4. When data reception completes successfully, the receive data is transferred to FRDRL*1. The RDF in SSR_FIFO 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 the RIE bit in SCR register is 1, an SCIn_RXI interrupt request is generated. Continuous data reception is enabled by reading the receive data transferred to FRDRL*2 in the SCIn_RXI interrupt handling routine before an overrun error occurs. If the amount of received data that is transferred to FRDRL is less than the RTS trigger number, the CTSn_RTSn pin goes low.

Note 1. In clock synchronous mode, FRDRH is not used.
Note 2. Read data in order from FRDRH to FRDRL when RDF and ORER are read with receive data.


Initialization
[1]
Start data reception
Read ORER*1 flag in SSR_FIFO
[2]
SSR_FIFO.ORER = 1?
Yes
[3]
No
Error processing
(Continued below)
No
SCIn_RXI interrupt?*2
Yes
Read receive data in FRDRL
[4]
No
All data received?
[5]
Yes
Set SCR.RIE and RE to 0
End
[3]
Error processing
Overrun error processing
[6]
Clear the SSR_FIFO.ORER flag to 0
[7]
Read the SSR_FIFO.ORER flag
[8]
End

[1] SCI initialization:
Set the input port for pins to be used as RXDn pins.

[2] [3] Receive error processing:
If a receive error occurs, read the 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] The receive data stored in FRDRL is read until the amount of stored data is less than the FCR.RTRG value. Software can check readable data in FDR.R[4:0].

[5] Serial reception continuation procedure:
To continue serial reception before overrun error occurs, finish reading the receive data in FRDRL and clear SSR_FIFO.RDF to 0. The FRDRL data can also be read by activating the DMAC or DTC by an SCIn_RXI interrupt request. The RDF flag is cleared automatically in this case. Do not write to the RDF flag.

[6] Processing in response to an overrun error:
Read FRDRL. In combination with step [7], this enables correct reception of the next frame possible.

[7] Clearing the error flag:
Write 0 to the error flag.

[8] Confirming that the error flag is cleared:
Read the error flag to confirm that its value is 0.

Note 1. ORER can also be read from FRDRH.ORER. However, to clear the ORER flag, write 0 to SSR_FIFO.ORER.

Note 2. All receive data is an integer multiple of the FIFO triggering number.

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

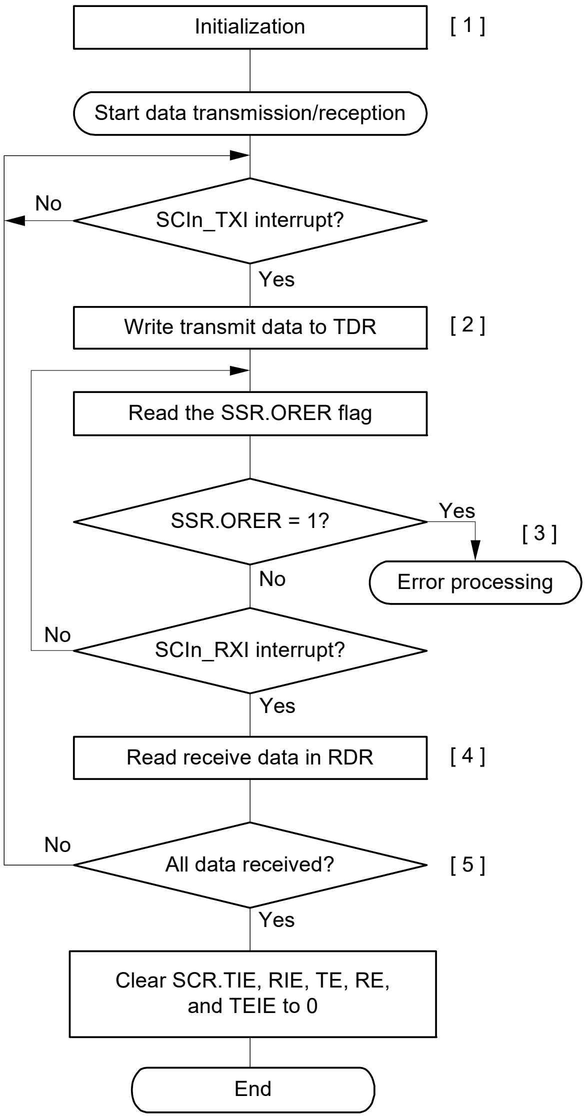
Figure 29.43 shows an example flow for simultaneous serial transmission and reception operations in clock synchronous mode. After initializing the SCI, use the following procedure for simultaneous serial data transmission and reception operations.

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 the SSR register is set to 1.
2. Initialize the SCR register and then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously using 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 in the SCR register, then check that the receive error flag ORER in SSR is 0.
3. Set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously using a single instruction.


Initialization
[1]
Start data transmission/reception
SCIn_TXI interrupt?
No
Yes
Write transmit data to TDR
[2]
Read the SSR.ORER flag
SSR.ORER = 1?
Yes
[3]
Error processing
No
SCIn_RXI interrupt?
No
Yes
Read receive data in RDR
[4]
All data received?
No
[5]
Yes
Clear SCR.TIE, RIE, TE, RE, and TEIE to 0
End

[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 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 resume while SSR.ORER is 1.

[4] Reading receive data:
Read the receive data in RDR 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 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.

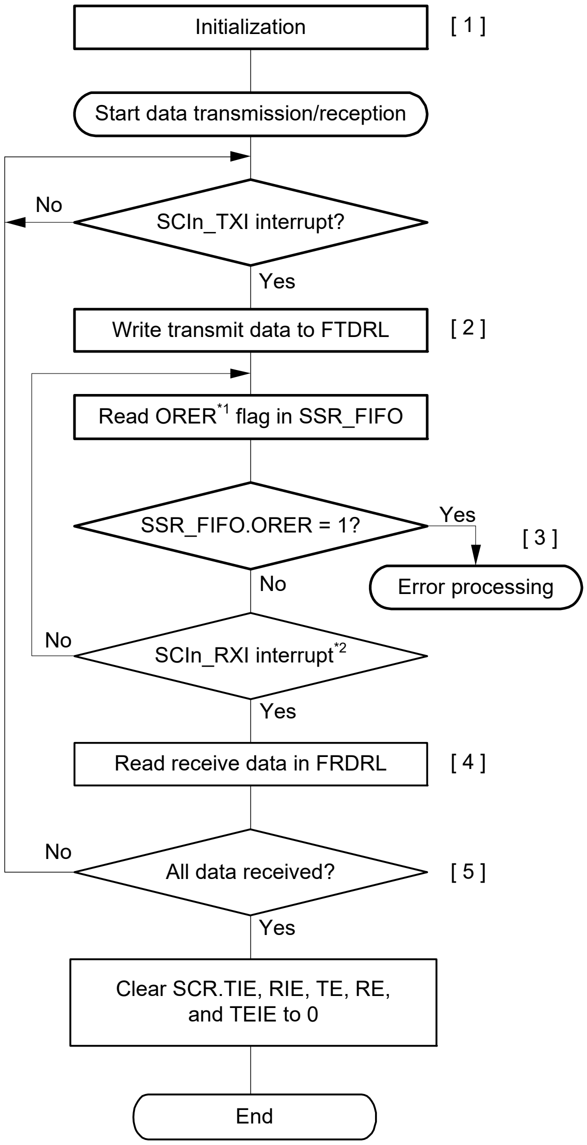
To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the transmission by verifying that the TEND flag in SSR_FIFO is set to 1.
2. Initialize SCR, then set the TIE, RIE, TE, and RE bits in SCR to 1 simultaneously using 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.


Initialization
[1]
Start data transmission/reception
No
SCIn_TXI interrupt?
Yes
Write transmit data to FTDRL
[2]
Read ORER*1 flag in SSR_FIFO
SSR_FIFO.ORER = 1?
Yes
[3]
Error processing
No
No
SCIn_RXI interrupt*2
Yes
Read receive data in FRDRL
[4]
No
All data received?
[5]
Yes
Clear SCR.TIE, RIE, TE, RE, and TEIE to 0
End

[1] SCI initialization:
The TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to FTDRL in the SCIn_TXI interrupt request handling routine. The number of transmission data that can be written is 16 bits, which is the number of stored transmit data in the FIFO.

[3] Receive error processing:
If a receive error occurs, read the 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 received data stored in FRDRL register is read until the amount 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.

Note 1. ORER can also read from FRDRH.ORER. To clear the ORER flag, write 0 to SSR_FIFO.ORER.

Note 2. All receive data must be an integer multiple of the FIFO triggering number.

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.

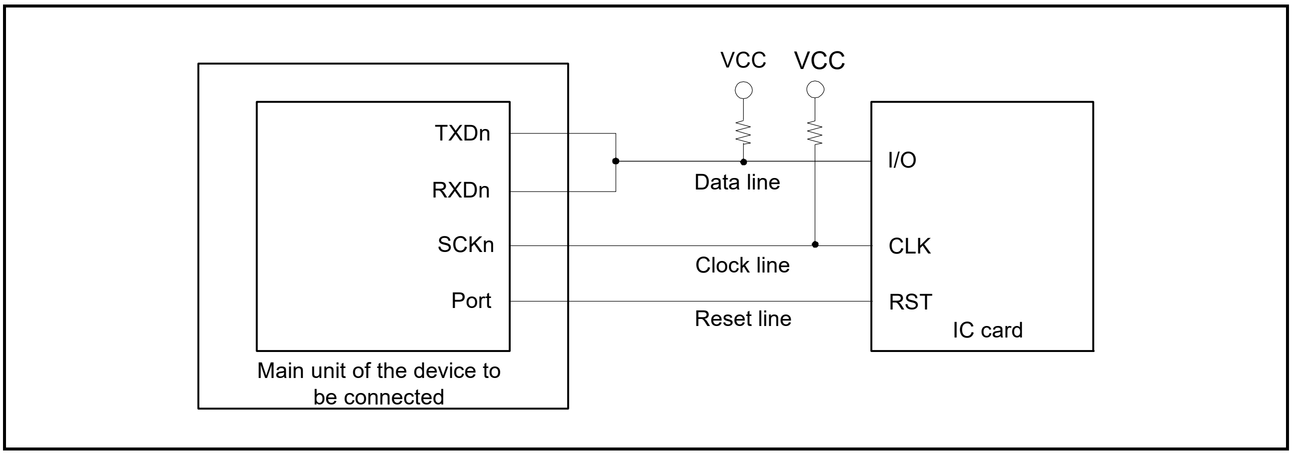


Figure 29.45 Example connection with a smart card (IC card)

29.6.2 Data Format (Except in Block Transfer Mode)

Figure 29.46 shows the data transfer formats in smart card interface mode.

The data transfer format is as follows:

- One frame consists of 8-bit data and a parity bit in asynchronous mode
- During transmission, a guard time of at least 2 ETUs (elementary time unit – the time required to transfer 1 bit) is set 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.

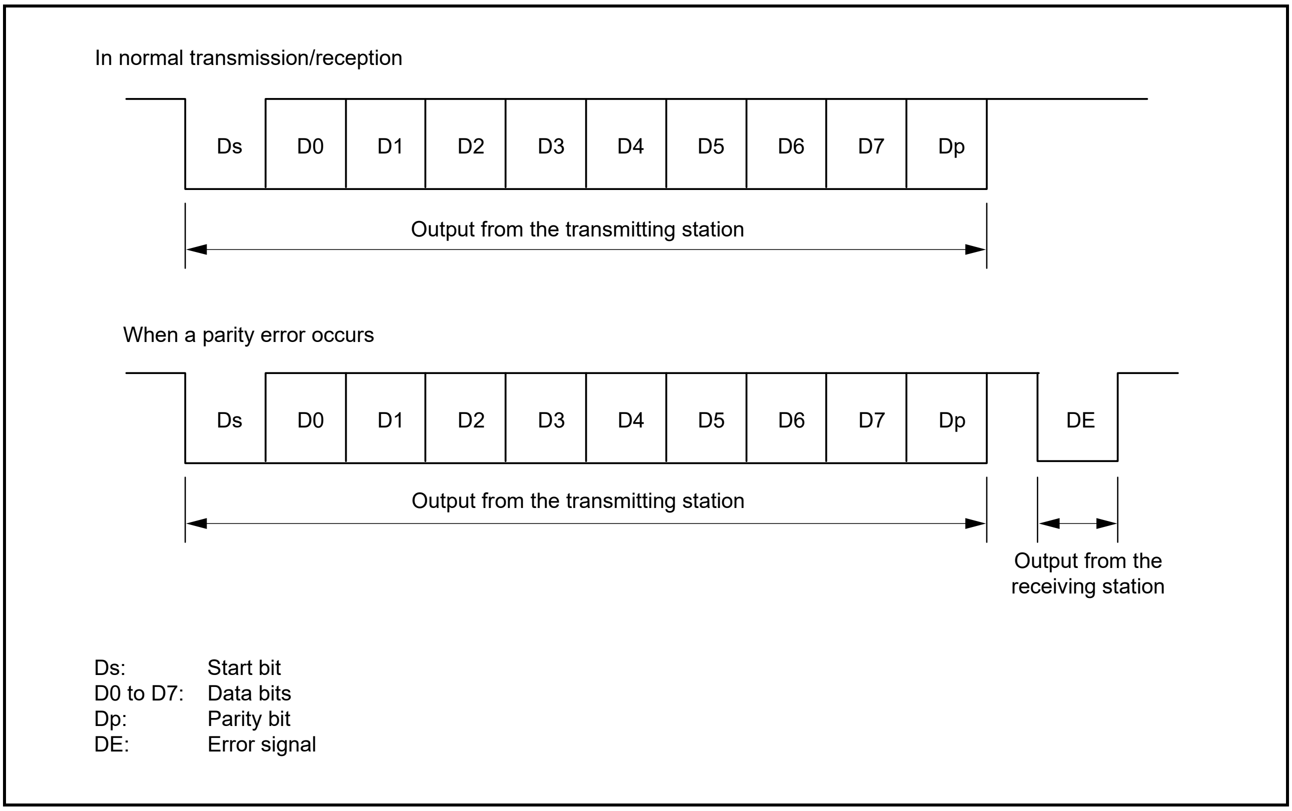

In normal transmission/reception
Ds
D0
D1
D2
D3
D4
D5
D6
D7
Dp
Output from the transmitting station
When a parity error occurs
Ds
D0
D1
D2
D3
D4
D5
D6
D7
Dp
DE
Output from the transmitting station
Output from the receiving station
Ds: Start bit
D0 to D7: Data bits
Dp: Parity bit
DE: Error signal

Figure 29.46 Data formats in smart card interface mode

For communication 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 correspond to the Z and A states, respectively, and data is transferred with LSB-first as the start character, as shown in Figure 29.47. Therefore, data in the start character in Figure 29.47 is 3Bh.

When using the direct convention type, write 0 to both the SDIR and SINV bits in SCMR. Write 0 to the PM bit in SMR_SMCI to use even parity, which is described by the smart card standard.

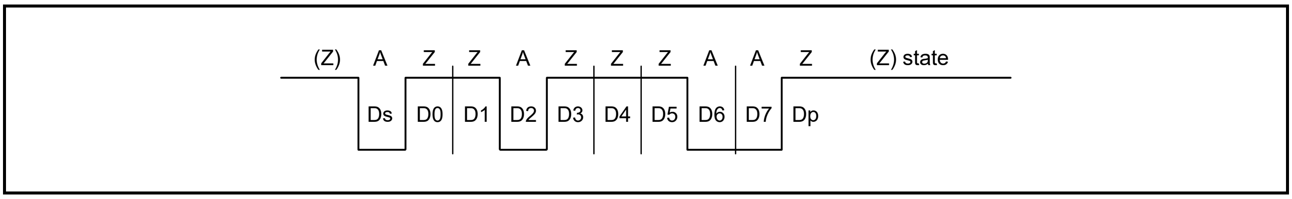

(Z)
A
Z
Z
A
Z
Z
Z
A
A
Z
(Z) state
Ds
D0
D1
D2
D3
D4
D5
D6
D7
Dp

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 correspond to the A and Z states, respectively and data is transferred with MSB-first as the start character, as shown in Figure 29.48. Therefore, data in the start character in Figure 29.48 is 3Fh.

When using the inverse convention type, write 1 to both the SDIR and SINV bits in SCMR. The parity bit is at 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 PM bit in SMR_SMCI to invert the parity bit for both transmission and reception.

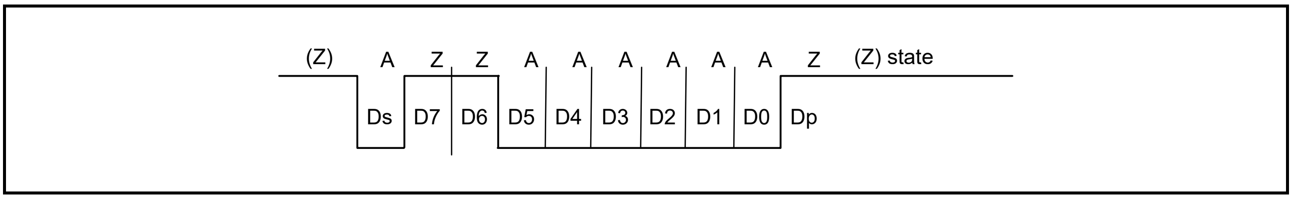


Figure 29.48 Inverse convention with SDIR in SCMR = 1, SINV in SCMR = 1, and PM in SMR_SMCI = 1

29.6.3 Block Transfer Mode

Block transfer mode differs from non-block transfer mode of the smart card interface mode in the following respects:

- If a parity error is detected during reception, no error signal is output. Because the PER bit in SSR_SMCI is set by error detection, clear the PER bit before receiving the parity bit of the next frame.
- During transmission, at least 1 ETU is set as a guard time from the end of the parity bit until the start of the next frame
- Because the same data is not retransmitted, the TEND flag in SSR_SMCI is set to 11.5 ETUs after transmission starts
- In block transfer mode, the ERS flag in SSR_SMCI indicates the error signal status as in 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 based on the BCP2 settings in SCMR and the BCP[1:0] bits in SMR_SMCI.

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:

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ [\%]$$

M: Reception margin (%)

N: Ratio of bit rate to clock (N = 32, 64, 372, 256)

D: Duty cycle of clock (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the specified formula, the reception margin is determined by the following formula:

$M = \{0.5 - 1/(2 \times 372)\} \times 100\ [\%] = 49.866\%$

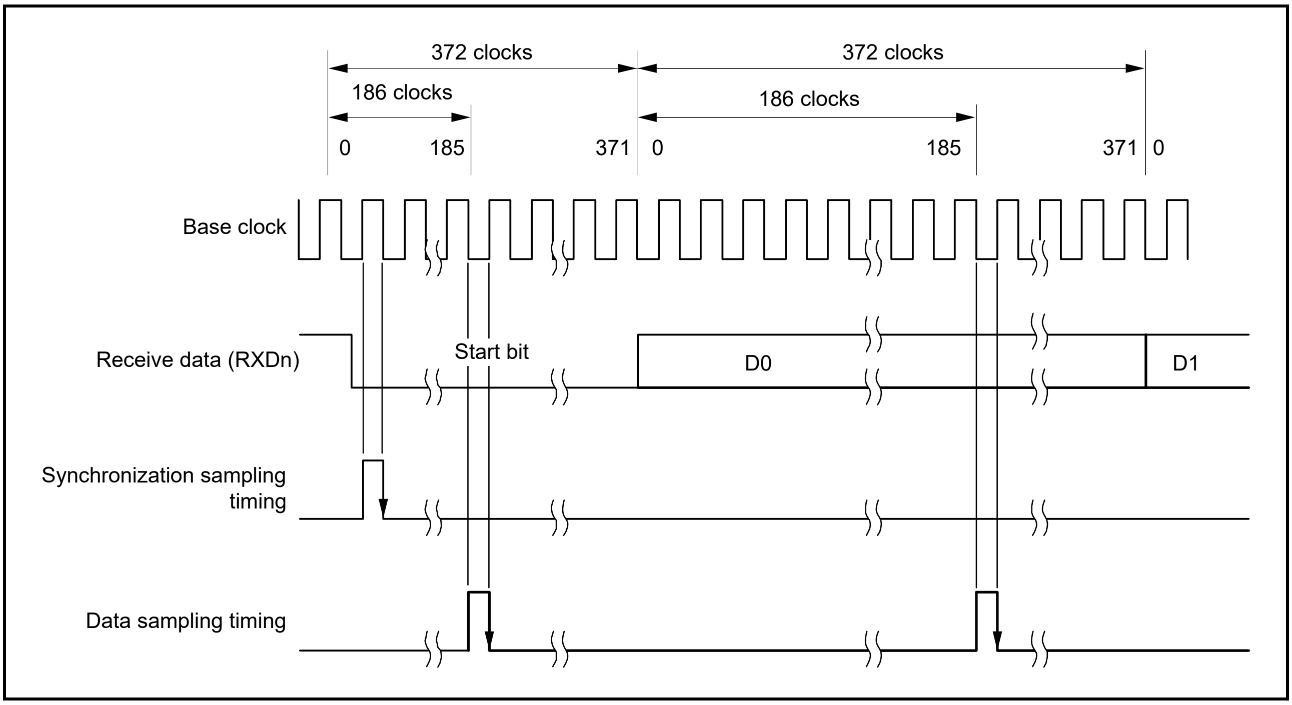

372 clocks
372 clocks
186 clocks
186 clocks
0
185
371
0
185
371
0
Base clock
Receive data (RXDn)
Start bit
D0
D1
Synchronization sampling timing
Data sampling timing

Figure 29.49 **Receive data sampling timing in smart card interface mode for clock frequency 372 times the bit rate**

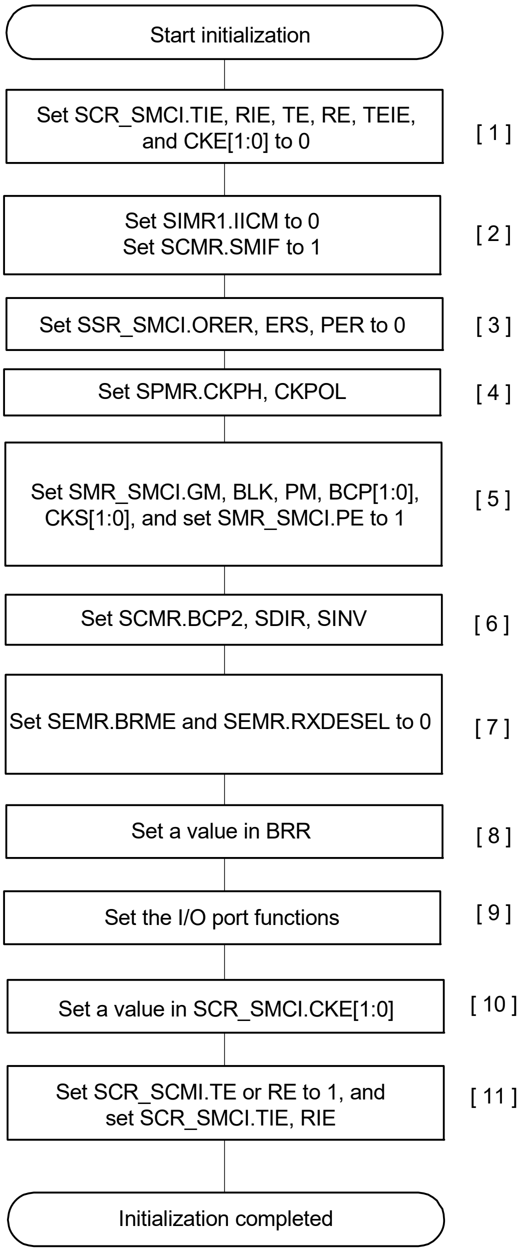
29.6.5 SCI Initialization

Before transmitting and receiving data, write the initial value 00h to SCR_SMCI and initialize the SCI as shown in the example 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 the RE bit in SCR_SMCI is set to 0, the RDR register is not initialized.

To change from reception to transmission mode, first check that reception is complete, and then initialize the SCI. At the end of initialization, set the TE bit to 1 and the RE bit to 0 in the SCR_SMCI register. Reception completion can be verified by reading the SCIn_RXI request, or ORER or PER flag in SSR_SMCI.

To change from transmission to reception mode, first check that transmission is complete, and then initialize the SCI. At the end of initialization, set the TE bit to 0 and the RE bit to 1 in the SCR_SMCI register. Transmission completion can be verified by reading the TEND flag in SSR_SMCI.


Start initialization
Set SCR_SMCI.TIE, RIE, TE, RE, TEIE, and CKE[1:0] to 0
[1]
Set SIMR1.IICM to 0
Set SCMR.SMIF to 1
[2]
Set SSR_SMCI.ORER, ERS, PER to 0
[3]
Set SPMR.CKPH, CKPOL
[4]
Set SMR_SMCI.GM, BLK, PM, BCP[1:0], CKS[1:0], and set SMR_SMCI.PE to 1
[5]
Set SCMR.BCP2, SDIR, SINV
[6]
Set SEMR.BRME and SEMR.RXDESEL to 0
[7]
Set a value in BRR
[8]
Set the I/O port functions
[9]
Set a value in SCR_SMCI.CKE[1:0]
[10]
Set SCR_SCMI.TE or RE to 1, and set SCR_SMCI.TIE, RIE
[11]
Initialization completed

[1] Stop the communication and initialize SKE[1:0].

[2] Set to smart card interface mode.

[3] Write to SSR_SMCI after reading SSR_SMCI.

[4] Set the transmission or reception format in SPMR.

[5] Set the operation mode and the transmission or reception format in SMR_SMCI.

[6] Set the transmission or reception format in SCMR.

[7] Set SEMR.BRME and SEMR.RXDESEL to 0.

[8] Write the value for the bit rate in BRR.

[9] Set the I/O port functions for TXDn, RXDn, and SCKn.

[10] Set the SCR_SMCI.CKE[1:0]. Even though the function depends on SMR_SMCI.GM, when the CKE[0] bit is set to 1, the clock is output from the SCKn pin.

[11] Set the TE or RE bit in SCR_SMCI to 1, then set the TIE and RIE bits in SCR_SMCI. Do not simultaneously set the TE and RE bits to 1 if self-diagnosis is not used.

Figure 29.50 Example flow of SCI initialization in smart card interface mode

Figure 29.51 shows a timing diagram when data transmission is performed by transitioning to smart card interface mode according to the flow in Figure 29.50. Figure 29.51 shows when the GM bit in SMR_SMCI is set to 0. The timing in Figure 29.51 shows when the port is connected as SCKn and TXDn, the pins are Hi-Z because CKE[0] bit in SCR_SMCI is 0.

Start the clock output to the SCK pin by setting CKE[0] bit in SCR_SMCI to 1, then start data transmission by writing transmit data after setting TE bit in SCR_SMCI to 1. When TE bit in SCR_SMCI changes from 0 to 1, there is a preamble period for one frame before data transmission starts. In smart card interface mode, the TXDn pin is Hi-Z during a preamble period. Pull-up or pull-down for the SCKn and TXDn pins is required outside the MCU.

In smart card interface mode, even when the TE and RE bits in SCR_SMCI are 0, the clock is continuously output if the clock output setting is used.

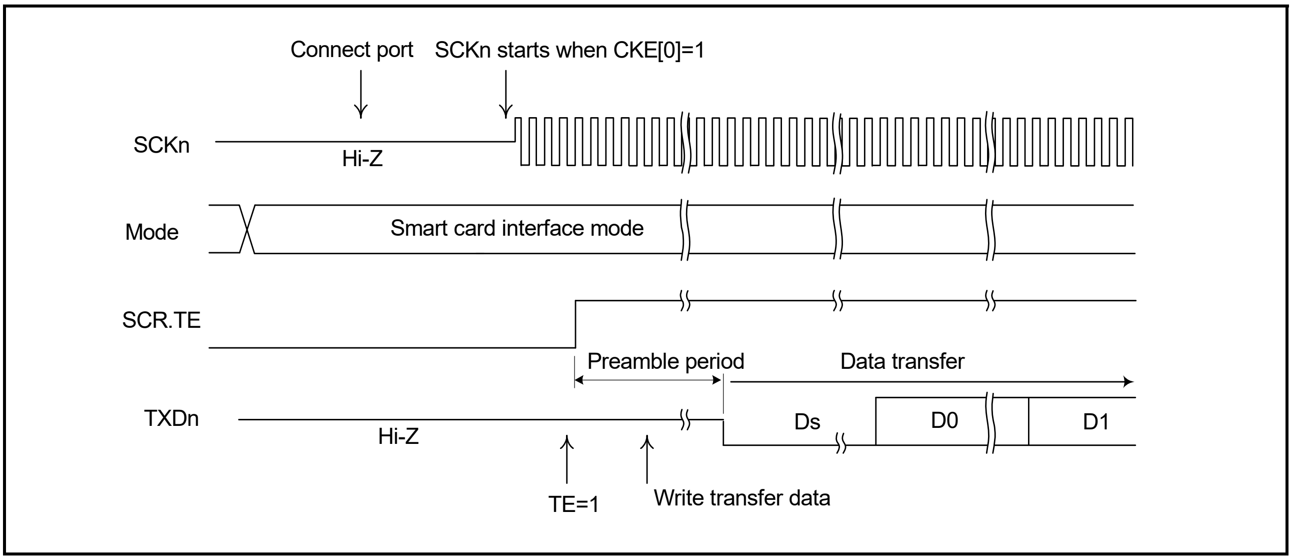


Figure 29.51 Example timing of data transmission in smart card interface mode

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.

In Figure 29.52:

- [1] indicates 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 the RIE bit is 1 in SCR_SMCI, an SCIn_ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.
- [2] indicates 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] indicates if no error signal is returned from the receiver, the ERS flag is not set to 1.
- [4] indicates the SCI determines that the transmission of 1-frame data including the retransfer is complete, and the TEND flag is set. If the TIE bit in SCR_SMCI 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 TEND flag in SSR_SMCI is set to 1 in transmission and when the TIE bit in SCR_SMCI 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 at 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 to enable an SCIn_ERI interrupt request generation when 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).

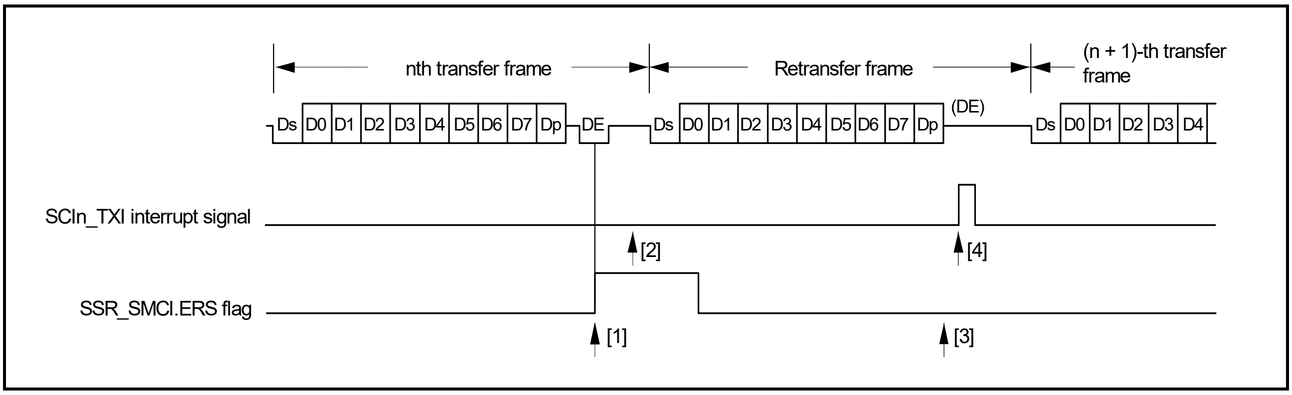

nth transfer frame
Retransfer frame
(n + 1)-th transfer frame
Ds D0 D1 D2 D3 D4 D5 D6 D7 Dp DE
Ds D0 D1 D2 D3 D4 D5 D6 D7 Dp (DE)
Ds D0 D1 D2 D3 D4
SCIn_TXI interrupt signal
SSR_SMCI.ERS flag
[1]
[2]
[3]
[4]

Figure 29.52 Data retransfer operation in SCI transmission mode

Note: The TEND flag in SSR_SMCI is set at different timings depending on the GM bit setting in SMR_SMCI.

Figure 29.53 shows the TEND flag generation timing.

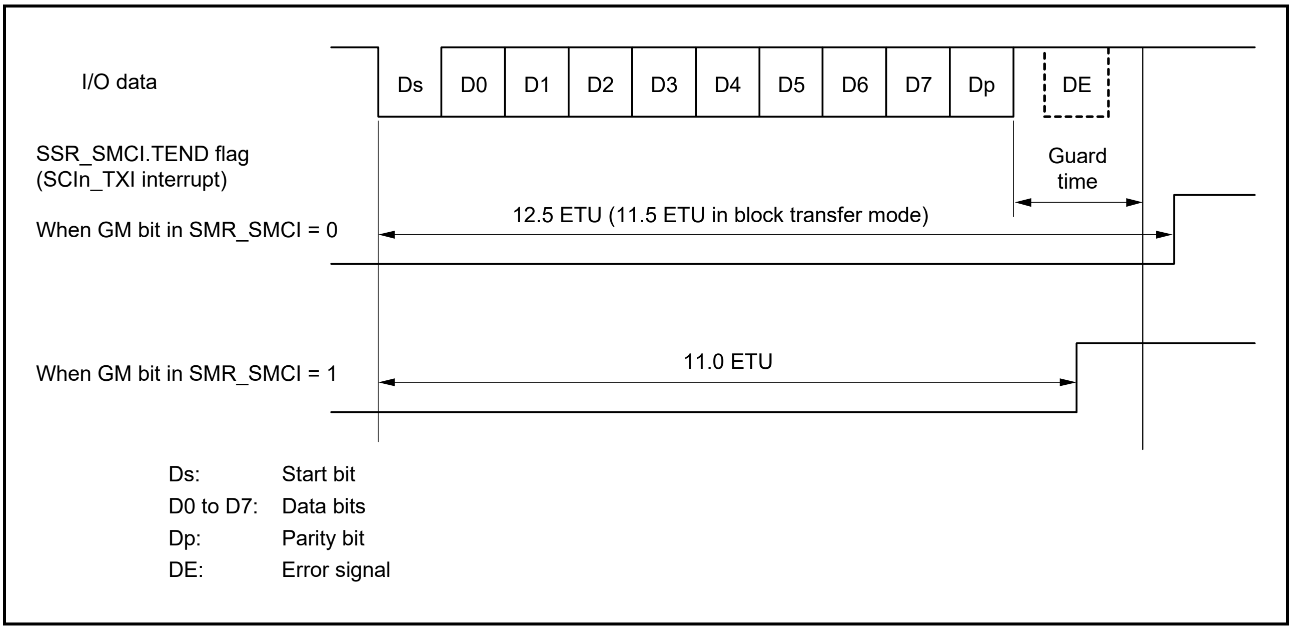

I/O data
Ds D0 D1 D2 D3 D4 D5 D6 D7 Dp DE
SSR_SMCI.TEND flag
(SCIn_TXI interrupt)
When GM bit in SMR_SMCI = 0
Guard time
12.5 ETU (11.5 ETU in block transfer mode)
When GM bit in SMR_SMCI = 1
11.0 ETU
Ds: Start bit
D0 to D7: Data bits
Dp: Parity bit
DE: Error signal

Figure 29.53 SSR.TEND flag generation timing during transmission

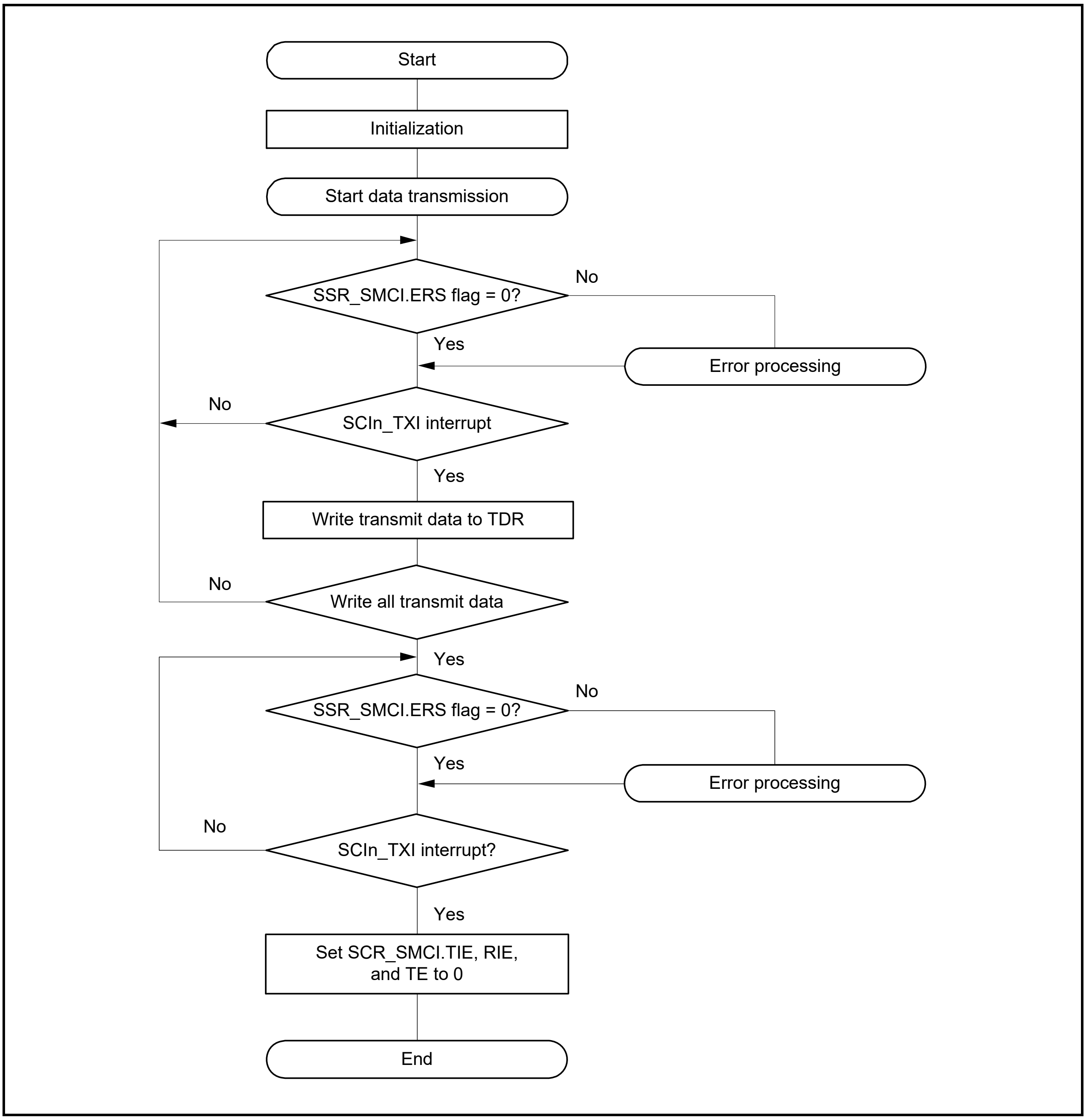

Start
Initialization
Start data transmission
SSR_SMCI.ERS flag = 0?
No
Yes
Error processing
No
SCIn_TXI interrupt
Yes
Write transmit data to TDR
No
Write all transmit data
Yes
SSR_SMCI.ERS flag = 0?
No
Yes
Error processing
No
SCIn_TXI interrupt?
Yes
Set SCR_SMCI.TIE, RIE, and TE to 0
End

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] indicates 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] indicates for a frame in which a parity error is detected, no SCIn_RXI interrupt is generated.
- [3] indicates when no parity error is detected, the PER flag in SSR_SMCI is not set to 1.
- [4] indicates the data is determined to be received successfully. When the RIE bit in SCR_SMCI is 1, an SCIn_RXI interrupt request is generated.

Figure 29.56 shows an example flow of serial data reception. All the processing steps are automatically performed using an SCIn_RXI interrupt request to activate the 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 are 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 the RE bit in SCR_SMCI 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.

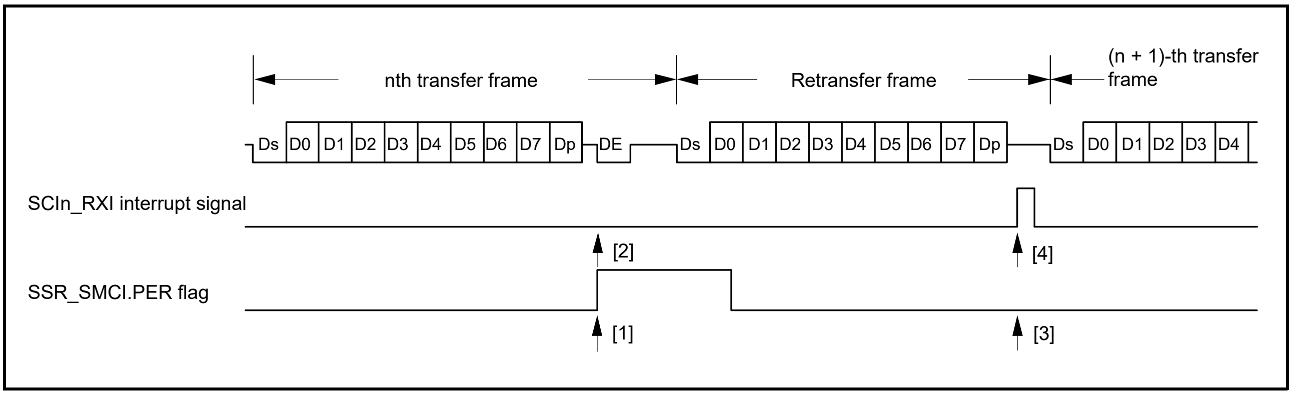


Figure 29.55 Data retransfer operation in SCI reception mode with data retransfer operation during reception

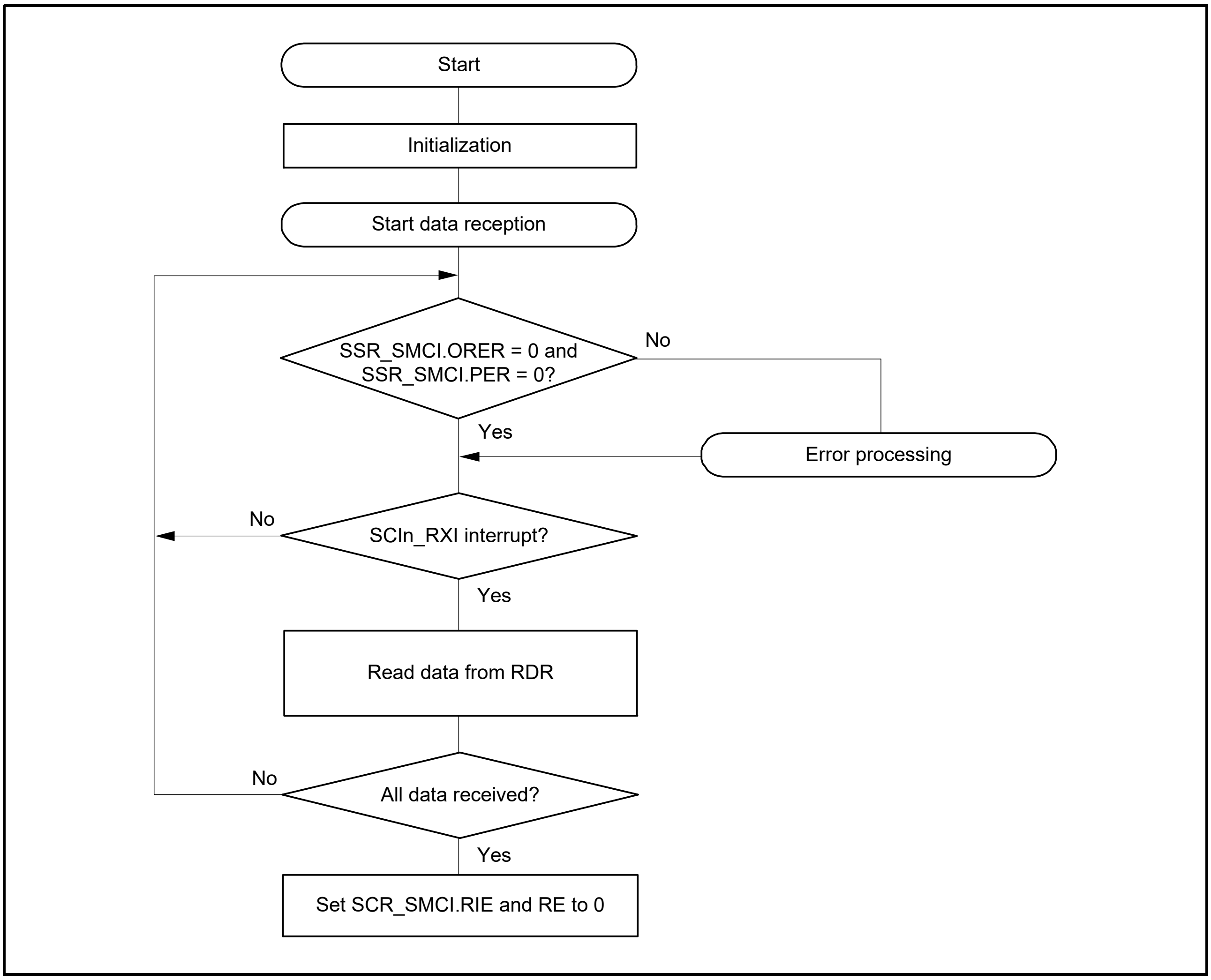

Start
Initialization
Start data reception
SSR_SMCI.ORER = 0 and SSR_SMCI.PER = 0?
No
Yes
Error processing
No
SCIn_RXI interrupt?
Yes
Read data from RDR
No
All data received?
Yes
Set SCR_SMCI.RIE and RE to 0

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.

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_SMCI is immediately reflected on the SCK pin, so there is a possibility that pulses with an unintended width might be output from the SCK pin.

When the GM bit in SMR_SMCI is 1, the clock with the same pulse width as the base clock is output even if the CKE[0] bit in SCR_SMCI is changed.

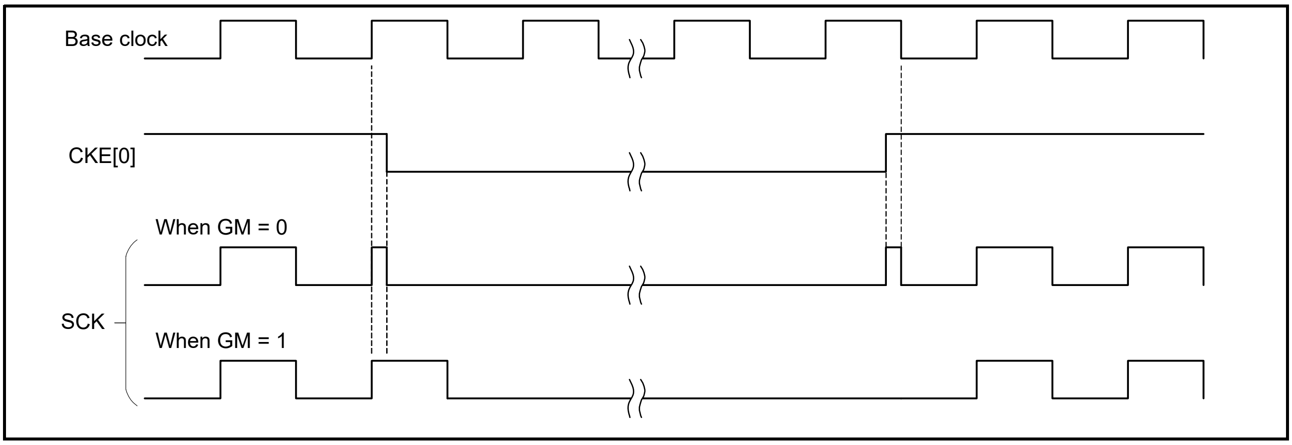

Base clock
CKE[0]
SCK
When GM = 0
When GM = 1

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 a 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 are shown in Figure 29.58 and Figure 29.59.

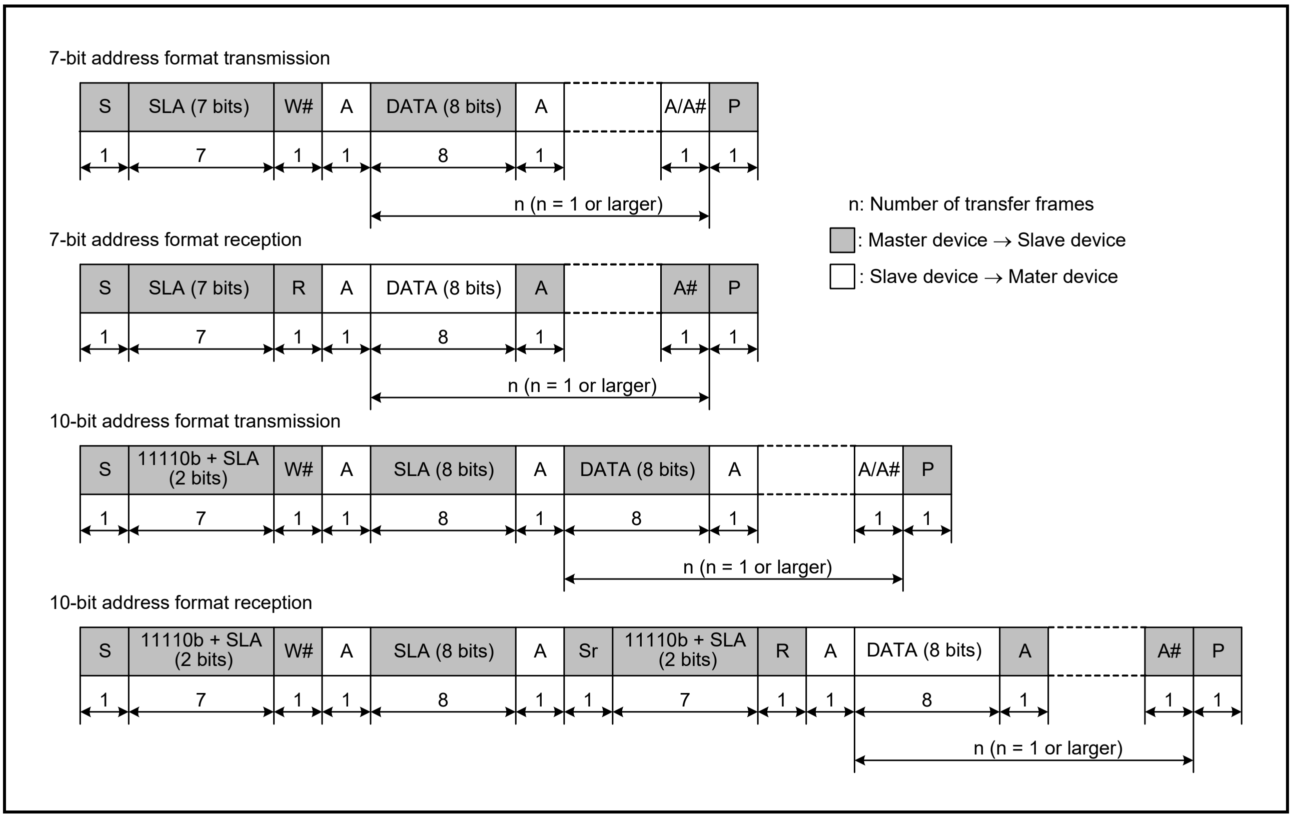

7-bit address format transmission
S
SLA (7 bits)
W#
A
DATA (8 bits)
A
A/A#
P
1 7 1 1 8 1 1 1
n (n = 1 or larger)
7-bit address format reception
S
SLA (7 bits)
R
A
DATA (8 bits)
A
A#
P
1 7 1 1 8 1 1 1
n (n = 1 or larger)
10-bit address format transmission
S
11110b + SLA (2 bits)
W#
A
SLA (8 bits)
A
DATA (8 bits)
A
A/A#
P
1 7 1 1 8 1 8 1 1 1
n (n = 1 or larger)
10-bit address format reception
S
11110b + SLA (2 bits)
W#
A
SLA (8 bits)
A
Sr
11110b + SLA (2 bits)
R
A
DATA (8 bits)
A
A#
P
1 7 1 1 8 1 1 7 1 1 8 1 1 1
n (n = 1 or larger)
n: Number of transfer frames
▨: Master device → Slave device
☐: Slave device → Mater device

Figure 29.58 I²C bus format

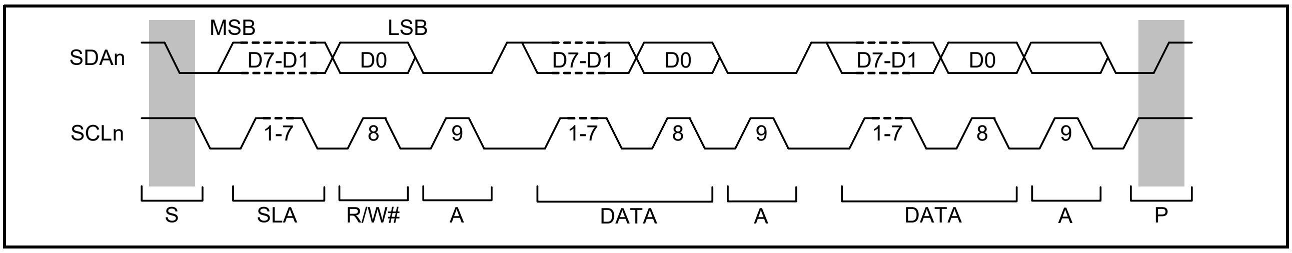


Figure 29.59 I^{2}C bus timing when SLA is 7 bits

S: Indicates a start condition, when the master device changes the level on the SDAn line from high to low while the SCLn line is high.

SLA: Indicates a slave address, by which the master device selects a slave device.

R/W#: Indicates the direction of transfer (reception or transmission). The value 1 indicates transfer from the slave device to the master device and 0 indicates transfer from the master device to the slave device.

A/A#: Indicates an acknowledge bit. This is returned by the slave device for master transmission and by the master device for master reception. Return to low indicates ACK and return to 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 level when 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 high level to 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 high level to low level), the IICSTAREQ bit 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 a 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 low to 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 high level to 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 high level to 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 high level to low level) and the SCLn line is kept at a 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 low to 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 low to 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.

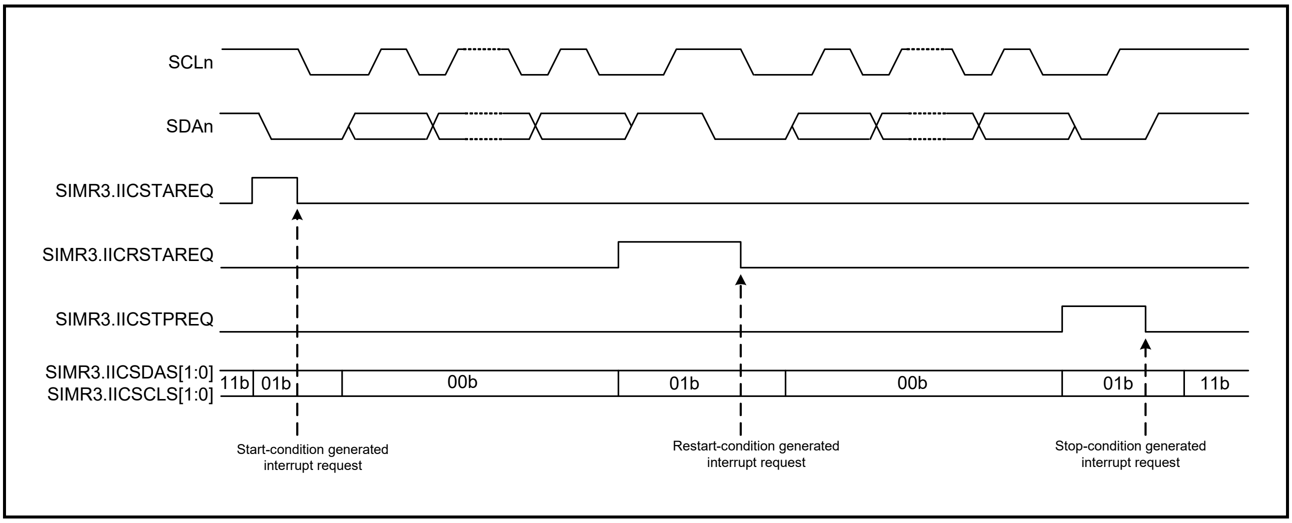

SCLn
SDAn
SIMR3.IICSTAREQ
SIMR3.IICRSTAREQ
SIMR3.IICSTPREQ
SIMR3.IICSDAS[1:0]
SIMR3.IICSCLS[1:0]
11b
01b
00b
01b
00b
01b
11b
Start-condition generated interrupt request
Restart-condition generated interrupt request
Stop-condition generated interrupt request

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 allows clock synchronization control 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 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 input on the SCLn pin transitions to the high level.

The interval from the time until counting, to determine the period at high level that 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 PCLK cycles). The period at high level of the internal SCLn clock is extended even when other devices are not placing the low level on the SCLn line.

If the ICCSC bit in SIMR2 register 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 operation to synchronize the clocks.

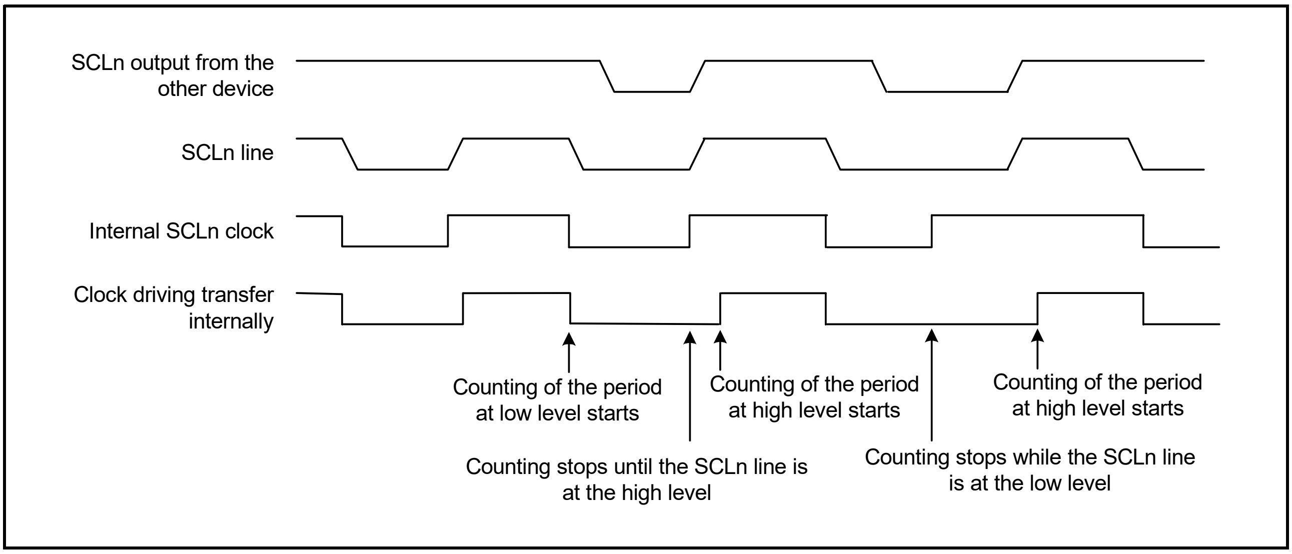

SCLn output from the other device
SCLn line
Internal SCLn clock
Clock driving transfer internally
Counting of the period at low level starts
Counting of the period at high level starts
Counting of the period at high level starts
Counting stops until the SCLn line is at the high level
Counting stops while the SCLn line is at the low level

Figure 29.61 Example operation 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. The delay settings represent periods of the associated 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 CKS[1:0] bit in the SMR register). 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 required 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 for erroneous operation of slave devices. Ensure that the settings for the output delay on the SDAn pin specify a time period greater than the time that the 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.

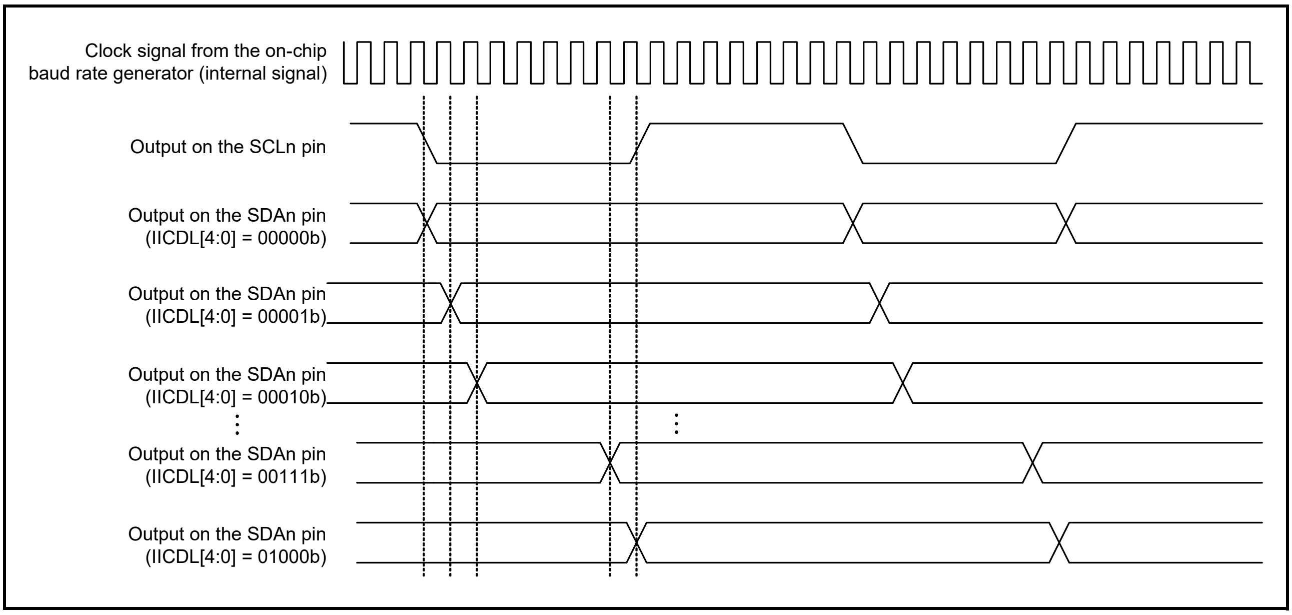
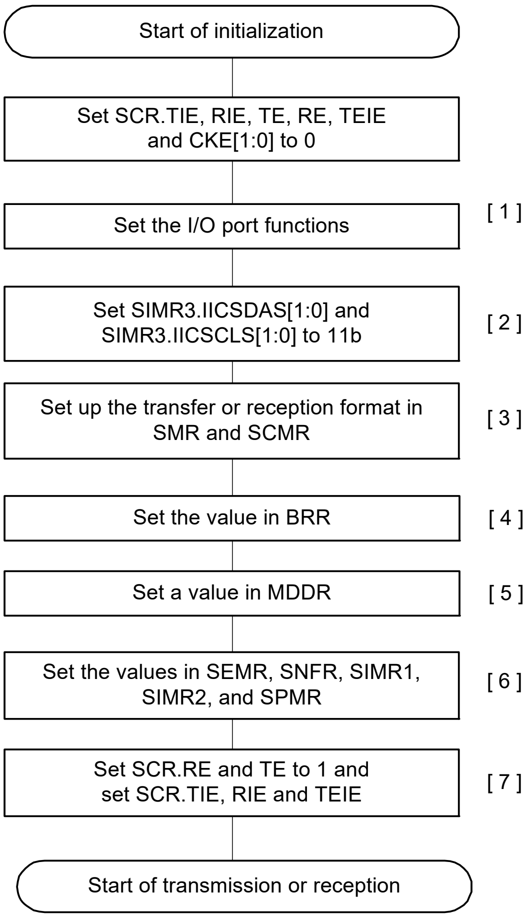

Clock signal from the on-chip baud rate generator (internal signal)
Output on the SCLn pin
Output on the SDAn pin (IICDL[4:0] = 00000b)
Output on the SDAn pin (IICDL[4:0] = 00001b)
Output on the SDAn pin (IICDL[4:0] = 00010b)
Output on the SDAn pin (IICDL[4:0] = 00111b)
Output on the SDAn pin (IICDL[4:0] = 01000b)

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 of 00h to SCR and initialize the interface as shown in the example 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.


Start of initialization
Set SCR.TIE, RIE, TE, RE, TEIE and CKE[1:0] to 0
Set the I/O port functions
[1]
Set SIMR3.IICSDAS[1:0] and SIMR3.IICSCLS[1:0] to 11b
[2]
Set up the transfer or reception format in SMR and SCMR
[3]
Set the value in BRR
[4]
Set a value in MDDR
[5]
Set the values in SEMR, SNFR, SIMR1, SIMR2, and SPMR
[6]
Set SCR.RE and TE to 1 and set SCR.TIE, RIE and TEIE
[7]
Start of transmission or reception

[1] Set the I/O ports to allow use (on N-channel open-drain output pins) of the SCLn and SDAn pin functions.

[2] Place the SCLn and SDAn pins in the high-impedance state until a start condition is to be generated.

[3] Set the format for transmission and reception in SMR and SCMR.
In SMR, set CKS[1:0] to the target value and set the other bits to 0.
In SCMR, set SDIR to 1 and set SINV and SMIF to 0.

[4] Write the value for the target bit rate to BRR.

[5] Write the value obtained by correcting a bit rate error in MDDR. This step is not required if SEMR.BRME is set to 0.

[6] Set the values in SEMR, SNFR, SIMR1, SIMR2, and SPMR.
In SEMR, set NFEN and BRME.
In SNFR, set NFCS[2:0].
In SIMR1, set IICM to 1 and set IICDL[4:0] as required.
In SIMR2, set IICACKT and IICCSC to 1 and set IICINTM as required.
In SPMR, set all the bits to 0.

[7] Set SCR.RE and TE to 1. Then, set SCR.TIE, RIE, and TEIE (for transmission and when SIMR2.IICINTM is 1, set RIE to 0).
Setting SCR.TE and RE to 1 enables the SCLn and SDAn pin functions.

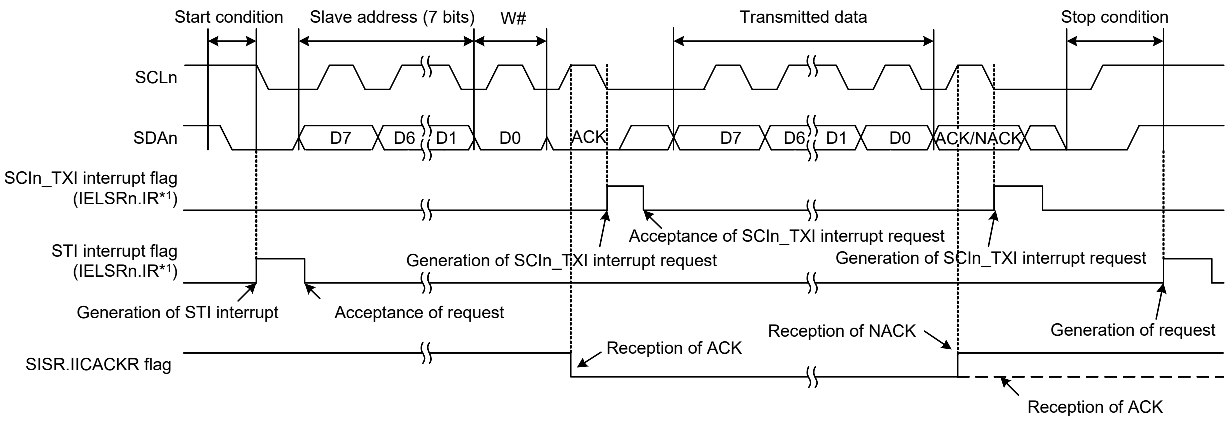
Figure 29.63 Example flow of SCI initialization in simple IIC mode

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 the IICINTM bit in the SIMR2 register is assumed to be 1 (use reception and transmission interrupts) and the value of the RIE bit in the SCR register 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 used, steps [3] and [4] in Figure 29.66 are repeated twice.

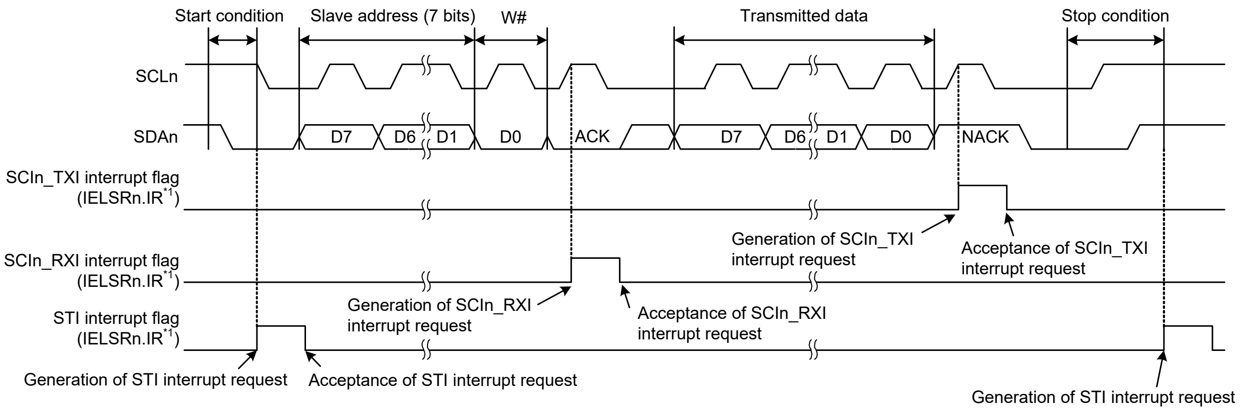
In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the timing of generation of the SCIn_TXI interrupt request during clock synchronous transmission.


Start condition
Slave address (7 bits)
W#
Transmitted data
Stop condition
SCLn
SDAn
D7
D6
D1
D0
ACK
ACK/NACK
SCIn_TXI interrupt flag (IELSRn.IR*1)
STI interrupt flag (IELSRn.IR*1)
SISR.IICACKR flag
Acceptance of SCIn_TXI interrupt request
Generation of SCIn_TXI interrupt request
Generation of SCIn_TXI interrupt request
Generation of STI interrupt
Acceptance of request
Reception of NACK
Generation of request
Reception of ACK
Reception of ACK

Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

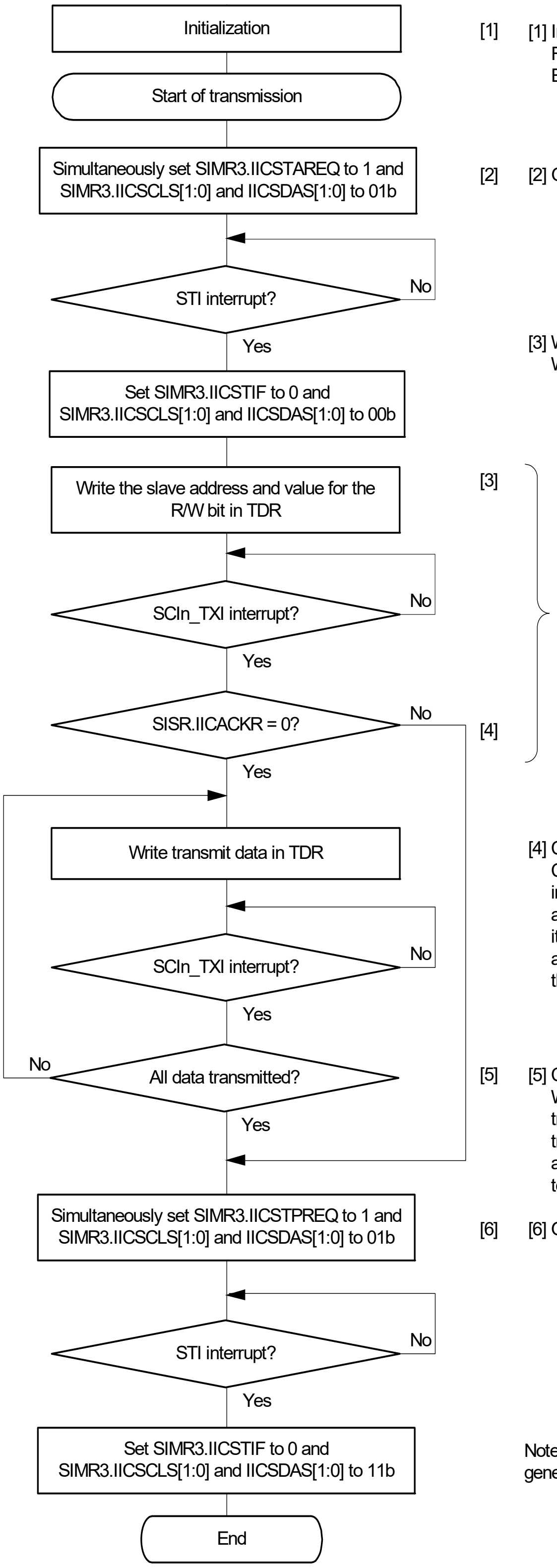
Figure 29.64 **Example 1 operation for master transmission in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts**

When the IICINTM bit in SIMR2 register is set to 0, using ACK/NACK interrupts during master transmission, the DMAC or DTC is activated by the ACK interrupt as the trigger and the 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.


Start condition
Slave address (7 bits)
W#
Transmitted data
Stop condition
SCLn
SDAn
D7
D6
D1
D0
ACK
NACK
SCIn_TXI interrupt flag (IELSRn.IR*1)
SCIn_RXI interrupt flag (IELSRn.IR*1)
STI interrupt flag (IELSRn.IR*1)
Generation of SCIn_TXI interrupt request
Acceptance of SCIn_TXI interrupt request
Generation of SCIn_RXI interrupt request
Acceptance of SCIn_RXI interrupt request
Generation of STI interrupt request
Acceptance of STI interrupt request
Generation of STI interrupt request

Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 29.65 **Example 2 operation for master transmission in simple IIC mode with 7-bit slave addresses, ACK interrupts, and NACK interrupts**


Initialization
[1]
Start of transmission
Simultaneously set SIMR3.IICSTAREQ to 1 and SIMR3.IICSCLS[1:0] and IICSDAS[1:0] to 01b
[2]
STI interrupt?
No
Yes
Set SIMR3.IICSTIF to 0 and SIMR3.IICSCLS[1:0] and IICSDAS[1:0] to 00b
Write the slave address and value for the R/W bit in TDR
[3]
SCIn_TXI interrupt?
No
Yes
SISR.IICACKR = 0?
No
[4]
Yes
Write transmit data in TDR
SCIn_TXI interrupt?
No
Yes
No
All data transmitted?
[5]
Yes
Simultaneously set SIMR3.IICSTPREQ to 1 and SIMR3.IICSCLS[1:0] and IICSDAS[1:0] to 01b
[6]
STI interrupt?
No
Yes
Set SIMR3.IICSTIF to 0 and SIMR3.IICSCLS[1:0] and IICSDAS[1:0] to 11b
End

[1] Initialization for simple IIC mode:
For transmission, set the SCR.RIE bit to 0 to disable RXI and ERI interrupts requests.

[2] Generate a start condition.

[3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

If 10-bit slave addresses are in use, processing of [3] and [4] is repeated twice.

[4] Confirming ACK response from the slave device:
Check the SISR.IICACKR bit. If its value is 0, it indicates that the slave device responded with ACK and operations proceed. If its value is 1, it indicates that there was no response from a slave device so the next transition is to generate the stop condition.

[5] Continuing with serial transmission:
When transmission is to continue, write additional transmit data to TDR. Except for the first data to be transmitted, an SCIn_TXI interrupt request can activate the DMAC or DTC to handle writing of data to TDR, but cannot confirm the status of ACK or NACK.

[6] Generation of a stop condition.

Note: In simple IIC mode, the SCIn_TXI interrupt request is generated when communication is complete.

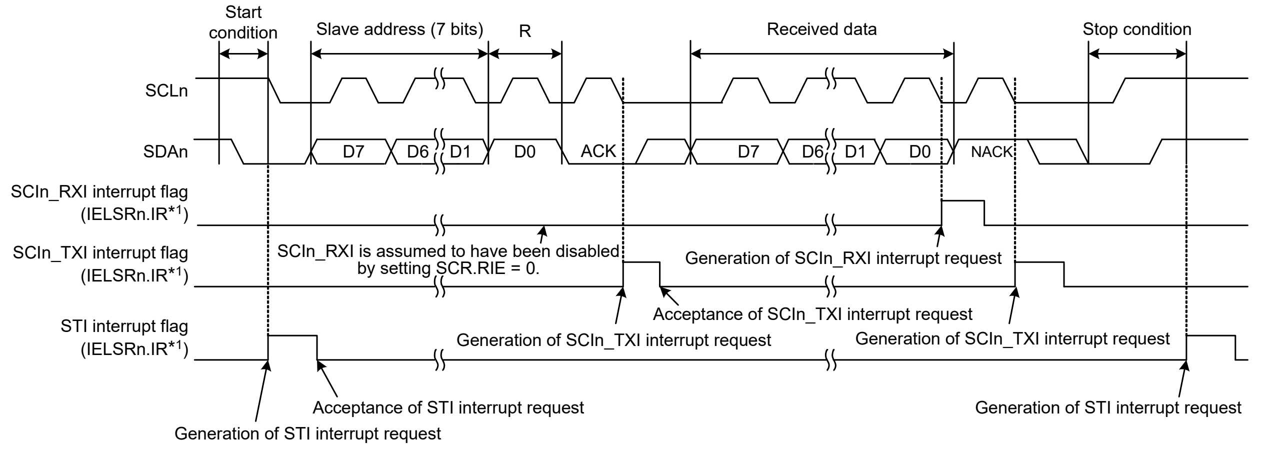
Figure 29.66 **Example procedure for master transmission in simple IIC mode with transmission interrupts and reception interrupts**

29.7.6 Master Reception in Simple IIC Mode

Figure 29.67 shows an example of master reception operation in simple IIC mode and Figure 29.68 shows an example flow of master reception.

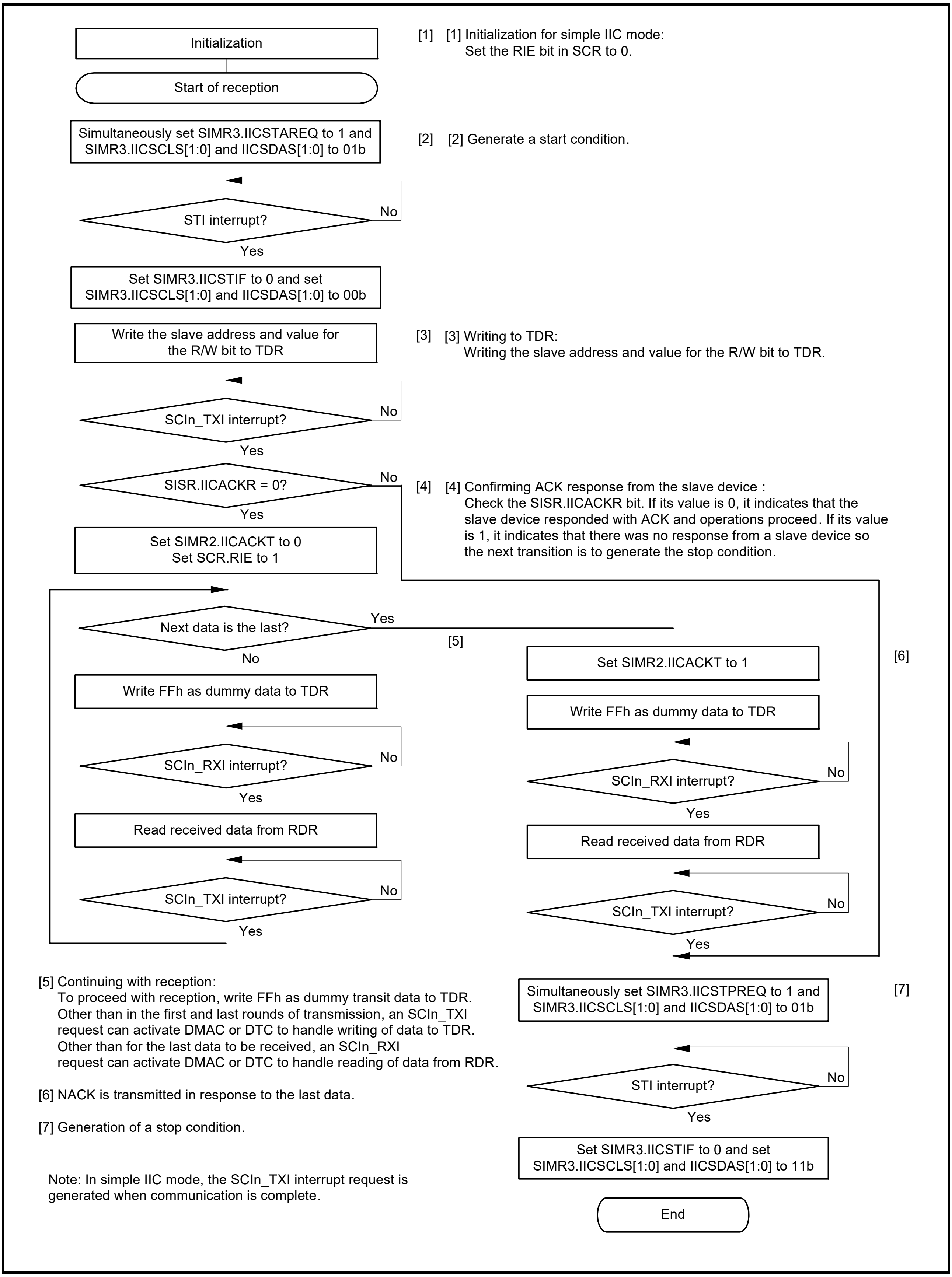
The value of the IICINTM bit in SIMR2 register is assumed to be 1 using reception and transmission interrupts.

In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame completes, unlike the timing of the SCIn_TXI interrupt request generation during clock synchronous transmission.


Start condition
Slave address (7 bits)
R
Received data
Stop condition
SCLn
SDAn
D7
D6
D1
D0
ACK
D7
D6
D1
D0
NACK
SCIn_RXI interrupt flag (IELSRn.IR*1)
SCIn_TXI interrupt flag (IELSRn.IR*1)
STI interrupt flag (IELSRn.IR*1)
SCIn_RXI is assumed to have been disabled by setting SCR.RIE = 0.
Generation of SCIn_RXI interrupt request
Acceptance of SCIn_TXI interrupt request
Generation of SCIn_TXI interrupt request
Generation of SCIn_TXI interrupt request
Acceptance of STI interrupt request
Generation of STI interrupt request
Generation of STI interrupt request

Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 29.67 Example operation for master reception in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts


Initialization
Start of reception
Simultaneously set SIMR3.IICSTAREQ to 1 and SIMR3.IICSCLS[1:0] and IICSDAS[1:0] to 01b
STI interrupt?
No
Yes
Set SIMR3.IICSTIF to 0 and set SIMR3.IICSCLS[1:0] and IICSDAS[1:0] to 00b
Write the slave address and value for the R/W bit to TDR
SCIn_TXI interrupt?
No
Yes
SISR.IICACKR = 0?
No
Yes
Set SIMR2.IICACKT to 0
Set SCR.RIE to 1
Next data is the last?
Yes
No
Write FFh as dummy data to TDR
SCIn_RXI interrupt?
No
Yes
Read received data from RDR
SCIn_TXI interrupt?
No
Yes
Set SIMR2.IICACKT to 1
Write FFh as dummy data to TDR
SCIn_RXI interrupt?
No
Yes
Read received data from RDR
SCIn_TXI interrupt?
No
Yes
Simultaneously set SIMR3.IICSTPREQ to 1 and SIMR3.IICSCLS[1:0] and IICSDAS[1:0] to 01b
STI interrupt?
No
Yes
Set SIMR3.IICSTIF to 0 and set SIMR3.IICSCLS[1:0] and IICSDAS[1:0] to 11b
End
[1]
[2]
[3]
[4]
[5]
[6]
[7]

[1] Initialization for simple IIC mode:
Set the RIE bit in SCR to 0.

[2] Generate a start condition.

[3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[4] Confirming ACK response from the slave device :
Check the SISR.IICACKR bit. If its value is 0, it indicates that the slave device responded with ACK and operations proceed. If its value is 1, it indicates that there was no response from a slave device so the next transition is to generate the stop condition.

[5] Continuing with reception:
To proceed with reception, write FFh as dummy transit data to TDR. Other than in the first and last rounds of transmission, an SCIn_TXI request can activate DMAC or DTC to handle writing of data to TDR. Other than for the last data to be received, an SCIn_RXI request can activate DMAC or DTC to handle reading of data from RDR.

[6] NACK is transmitted in response to the last data.

[7] Generation of a stop condition.

Note: In simple IIC mode, the SCIn_TXI interrupt request is generated when communication is complete.

Figure 29.68 Example flow for master reception in simple IIC mode with transmission interrupts and reception interrupts

29.8 Operation in Simple SPI Mode

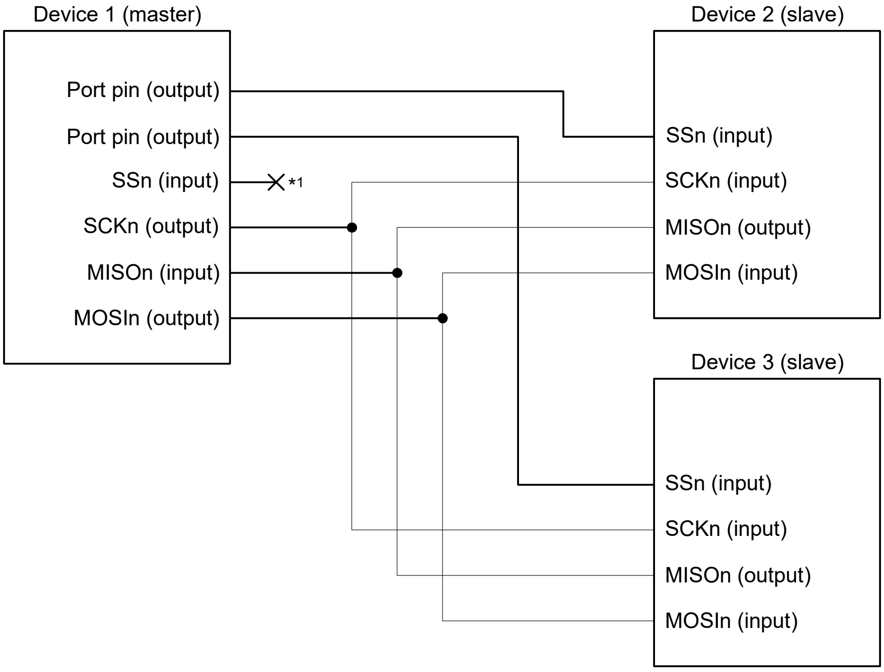
As an extended function, the SCI supports a simple SPI mode that handles transfer in one or multiple master devices and multiple slave devices.

To place the SCI in simple SPI mode, use the settings for clock synchronous mode (SCMR.SMIF = 0, SIMR1.IICM = 0, SMR.CM = 1) and set the SSE bit in SPMR register to 1. When the configuration only has a single master, the SSn pin function is not required to connect the device used as the master in simple SPI mode. Therefore, set the SSE bit in the SPMR register to 0.

Figure 29.69 shows an example of connections in simple SPI mode. Use 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 transfer data consists of 8 bits of data, and parity bits cannot be appended. The data can be inverted by setting the SINV bit in SCMR 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, it is possible to both write the next transmit data while transmission is in progress and read previously received data while reception is in progress. This enables continuous transfer.



Note 1. The SSn input is not required in a single-master system (the interface is used with the setting SPMR.SSE = 0).

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 the input level on the SSn pin.

Table 29.24 Pin states 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 (transfer can proceed)	Output for data transmission*2	Input for received data	Clock output*3
	Low (transfer cannot proceed)	High-impedance	Input for received data (but disabled)	High-impedance
Slave mode	High (transfer cannot proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low (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 multi-master 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 MFF bit in SPMR is 1, indicating a mode fault error. In a multi-master configuration, start error processing by reading the MFF flag in SPMR. If a mode fault error occurs while transmission or reception is in progress, transmission or reception does not stop, but the MOSIn and SCKn outputs are in the high-impedance state after the completion of the transfer. Use 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 a 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 valid 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 a 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 current transmitted or received character is complete, after which it stops, and the appropriate interrupt (SCIn_TXI, SCIn_RXI, or SCIn_TEI) is then 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.

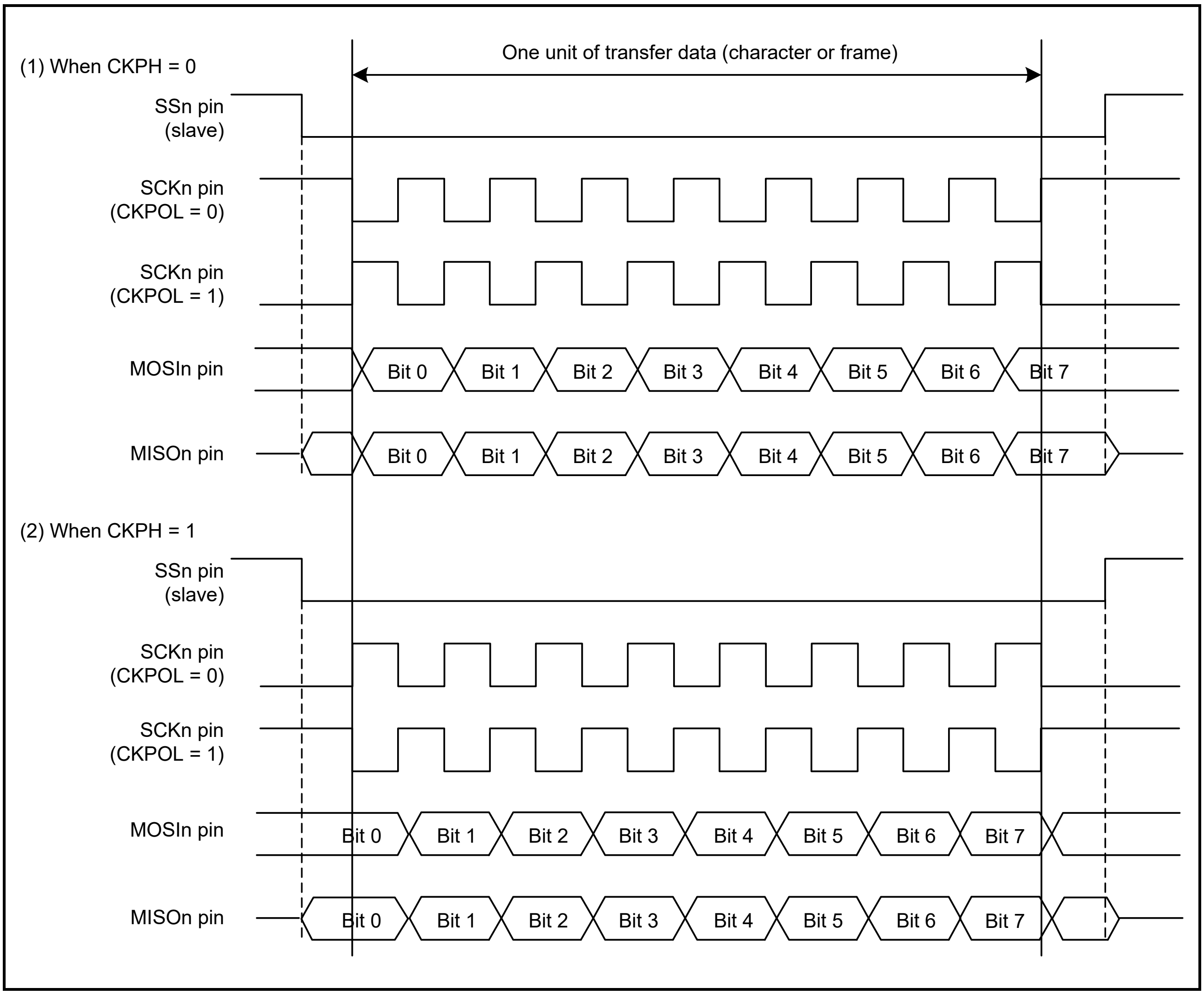

(1) When CKPH = 0
One unit of transfer data (character or frame)
SSn pin (slave)
SCKn pin (CKPOL = 0)
SCKn pin (CKPOL = 1)
MOSIn pin
MISOn pin
Bit 0
Bit 1
Bit 2
Bit 3
Bit 4
Bit 5
Bit 6
Bit 7
(2) When CKPH = 1
SSn pin (slave)
SCKn pin (CKPOL = 0)
SCKn pin (CKPOL = 1)
MOSIn pin
MISOn pin

Figure 29.70 Relation between clock signal and transmit or receive data in simple SPI mode

29.8.5 SCI Initialization in Simple SPI Mode

SCI initialization in simple SPI mode is the same as in clock synchronous mode. See Figure 29.32 for an example of initialization flow. The CKPOL and CKPH bits in SPMR must be set to ensure that the selected clock signal configuration is suitable for both master and slave devices.

Always initialize the SCR register before making any changes to the operating mode or transfer format.

Note: Only the SCR.RE bit is set to 0. The SSR.ORER, FER, PER, and RDR flags are not initialized.

Changing the value of the TE bit in the SCR register from 1 to 0 or from 0 to 1 when the TIE bit in the SCR register is 1 at the same time, leads to the generation of a transmit data empty interrupt (SCIn_TXI).

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 a low level before starting the transfer and at a 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 with the CKS[1:0] bits in SMR/SMR_SMCI.

Figure 29.71 shows an example where PCLK is selected in 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).

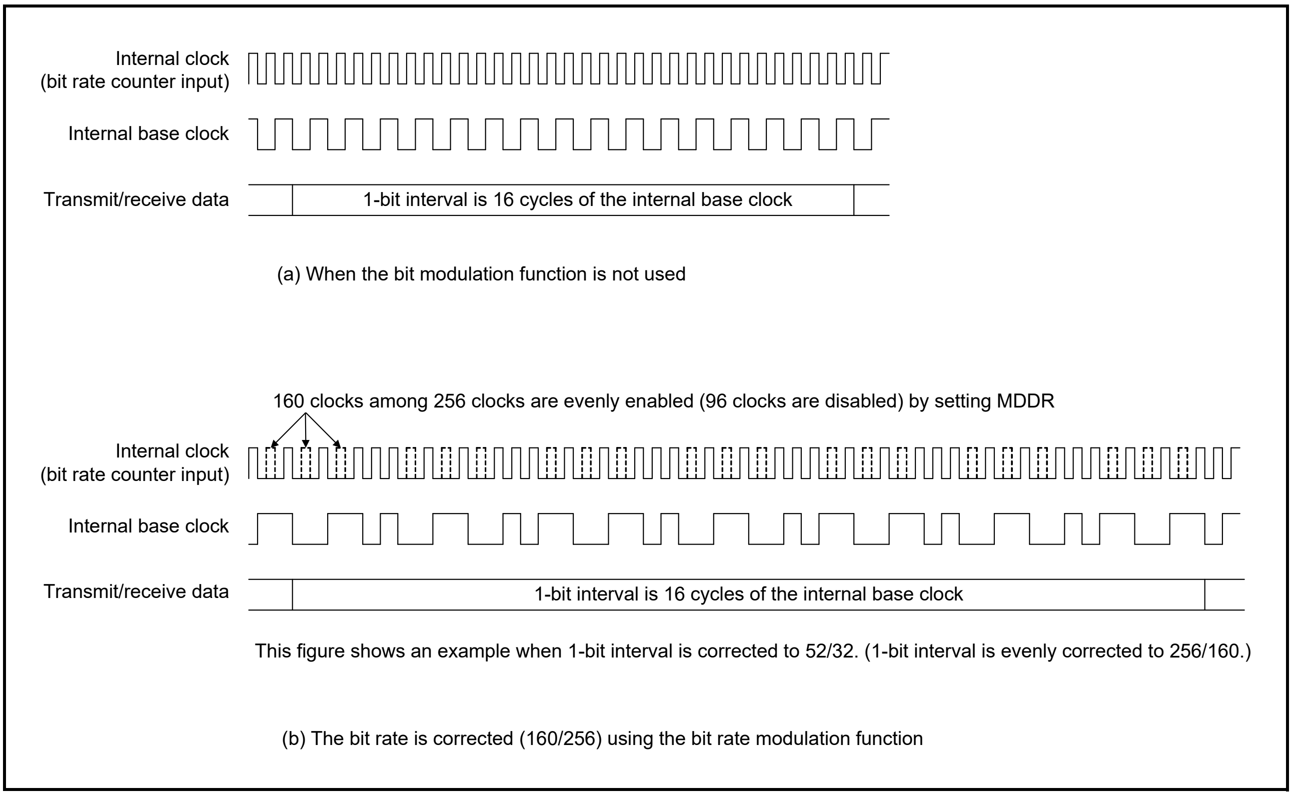

Internal clock (bit rate counter input)
Internal base clock
Transmit/receive data
1-bit interval is 16 cycles of the internal base clock
(a) When the bit modulation function is not used
160 clocks among 256 clocks are evenly enabled (96 clocks are disabled) by setting MDDR
Internal clock (bit rate counter input)
Internal base clock
Transmit/receive data
1-bit interval is 16 cycles of the internal base clock
This figure shows an example when 1-bit interval is corrected to 52/32. (1-bit interval is evenly corrected to 256/160.)
(b) The bit rate is corrected (160/256) using the bit rate modulation function

Figure 29.71 Example internal base clock using bit rate modulation function

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 ICU is 1, the ICU does not output the interrupt request but saves it internally with a capacity for retention of one request per source.

When the value of the interrupt status flag in the ICU becomes 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 ICU is set to 1, the SCIn_TXI and SCIn_RXI interrupts do not output interrupt requests to the ICU. When an interrupt status flag of the ICU is set to 0, and if the conditions for an SCIn_TXI and SCIn_RXI interrupts are satisfied, an interrupt request is generated.

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 TIE bit in SCR register 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 TE and TIE bits to 1 in the SCR 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 TE to 1 when the TIE bit is 0 or by setting TIE to 1 when TE is 1*2 in the SCR register.

When new data is not written by the time of transmission of the last bit of the current transmit data and the TEIE bit is 1 in SCR, the TEND flag in SSR becomes 1 and an SCIn_TEI interrupt request is generated. Additionally, when the TE bit is 1 in SCR, the TEND flag in SSR saves the value 1 until more transmit data are written to the TDR or TDRHL*1, and setting the TEIE bit in SCR to 1 leads to the generation of an SCIn_TEI interrupt request.

Writing data to the TDR or TDRHL*1 leads to clearing of the TEND flag in SSR and, after a certain time, discarding of the SCIn_TEI interrupt request.

If RIE is 1 in SCR, 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 the RIE bit in SCR 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 the TIE bit in the SCR register is 1, an SCIn_TXI interrupt request is generated when the amount of stored data in the FTDRL register is equal to or less than the threshold value indicated in the TTRG bit in FCR register. An SCIn_TXI interrupt request can also be generated by using a single instruction to set the TE and TIE bits in SCR to 1 simultaneously.

An SCIn_TXI interrupt request is not generated by setting the TE bit in the SCR register to 1 when TIE bit in the SCR register is 0 or by setting TIE to 1 when TE is 1.

If the TEIE bit in SCR register is 1 and if the next data is not written to the FTDRL register by the time the last bit of the transmission data is sent, the TEND flag in SSR_FIFO register is set to 1 and the SCIn_TEI interrupt request is generated.

If the RIE bit in the SCR register is 1, an SCIn_RXI interrupt request is generated when the amount of stored data in the FRDRL becomes equal to or greater than the threshold value indicated in the RTRG bit in the FCR register. When RTRG is 0, an SCIn_RXI interrupt does not occur even when the amount of data in the receive FIFO is equal to 0.

If the RIE bit in the SCR register is 1, when the ORER flag in the SSR_FIFO register 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 the ORER, FER, and PER flags in the SSR_FIFO register are all cleared.

Note 1. When asynchronous mode and 9-bit data length are selected.

Note 2. To temporarily prohibit SCIn_TXI interrupts on transmission of the last of the data when a new round of transmission is to be started, after handling the transmission-completed interrupt, control activation of the interrupt using the Interrupt Request Enable bit in the ICU rather than using the TIE bit in the SCR register. 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 (1 of 2)

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

Table 29.25 SCI interrupt sources with non-FIFO selected (2 of 2)

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation	DMAC activation
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 modes.

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 modes.

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 in 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 at 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 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 IICINTM bit in SIMR2 register 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 IICINTM bit in SIMR2 register 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 was set up as an activation source for the DMAC or DTC beforehand, 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 in 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 is selected and FCR.DRES is 1.

(2) Receive data full event output

- Indicates that ACK is detected if the IICINTM bit in SIMR2 register is 0 in simple IIC mode
- Indicates that the 8th-bit SCLn falling edge is detected if the IICINTM bit in SIMR2 register is 1 in simple IIC mode.
- When the IICINTM bit in SIMR2 register 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 the IICINTM bit in SIMR2 register is 0 in simple IIC mode
- Indicates that the 9th-bit SCLn falling edge is detected if the IICINTM bit in SIMR2 register 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, restart condition, or stop 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 mismatch event output (SCI0_DCUF)

SCI0_DCUF indicates a mismatch of the comparison data (CDR.CMPD) with one frame of receive data when DCCR.DCME is 1 in asynchronous mode, including multi-processor mode. This event can be used for snooze end request only.

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 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 TE and RE in SCR 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 considered as an internal signal. When the input level corresponds to 0, the initial output of the noise filter is retained until the level matches in 3 consecutive sampling cycles.

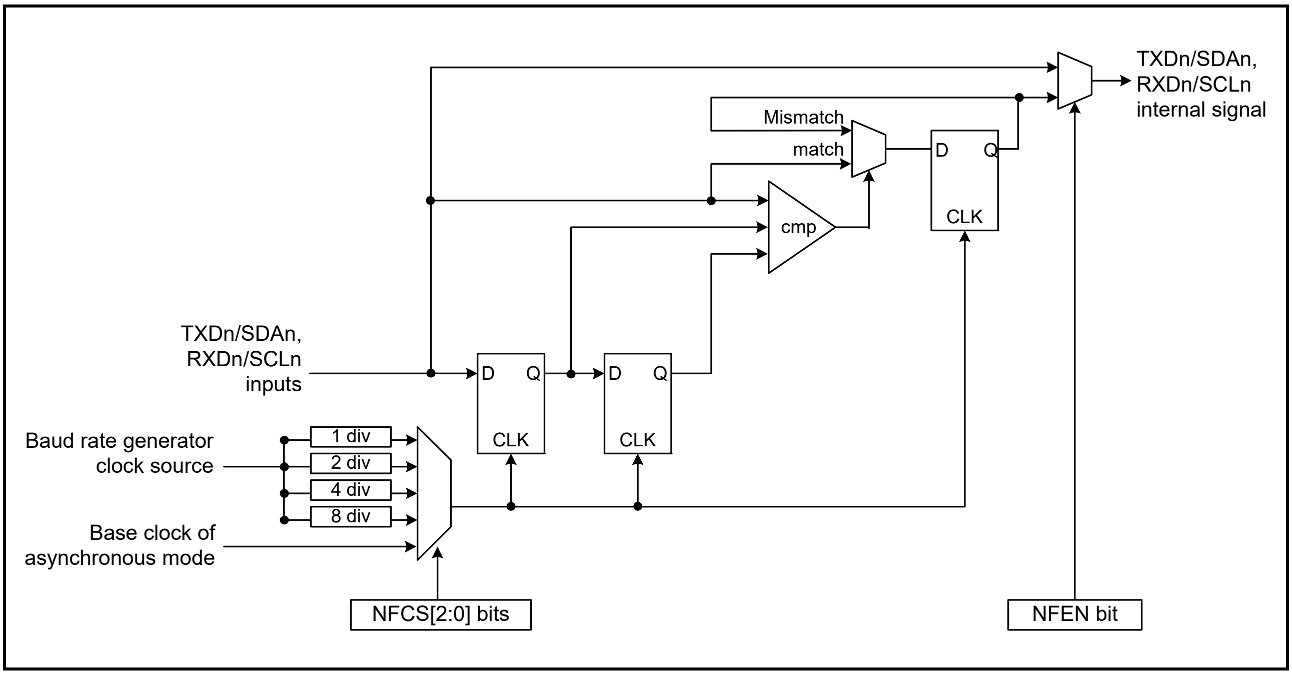


Figure 29.72 Digital noise filter circuit block diagram

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 a 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 transition to Software Standby mode, stop the operation (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 the SPTR register settings, output pins might output the level before a transition to the low power state is made after release from the module-stop state or Software Standby mode. When transitions to these states are made during transmission, the transmitted data becomes indeterminate.

To transmit data in the same transmission mode after cancellation of the low power state:

1. Set the TE bit to 1.
2. Read SSR/SSR_FIFO/SSR_SMCI.
3. Write data to TDR sequentially to start data transmission.

To transmit data with a different transmission mode, initialize the SCI first.

Figure 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 transitioning to Software Standby mode from the transmission mode using DTC transfer, stop the transmit operations (TE = 0). To start transmission after cancellation using the DTC, set the TE bit to 1. The SCIn_TXI interrupt flag is set to 1 and transmission starts using the DTC.

(2) Reception

(a) When address match function is not used as wakeup condition

Before specifying the module-stop state or transitioning to Software Standby mode, stop the receive operations (RE = 0 in SCR/SCR_SMCI). If transition is made during data reception, the data being received 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 condition of resumption (wakeup)

Before specifying the module-stop state or transitioning to Software Standby mode:

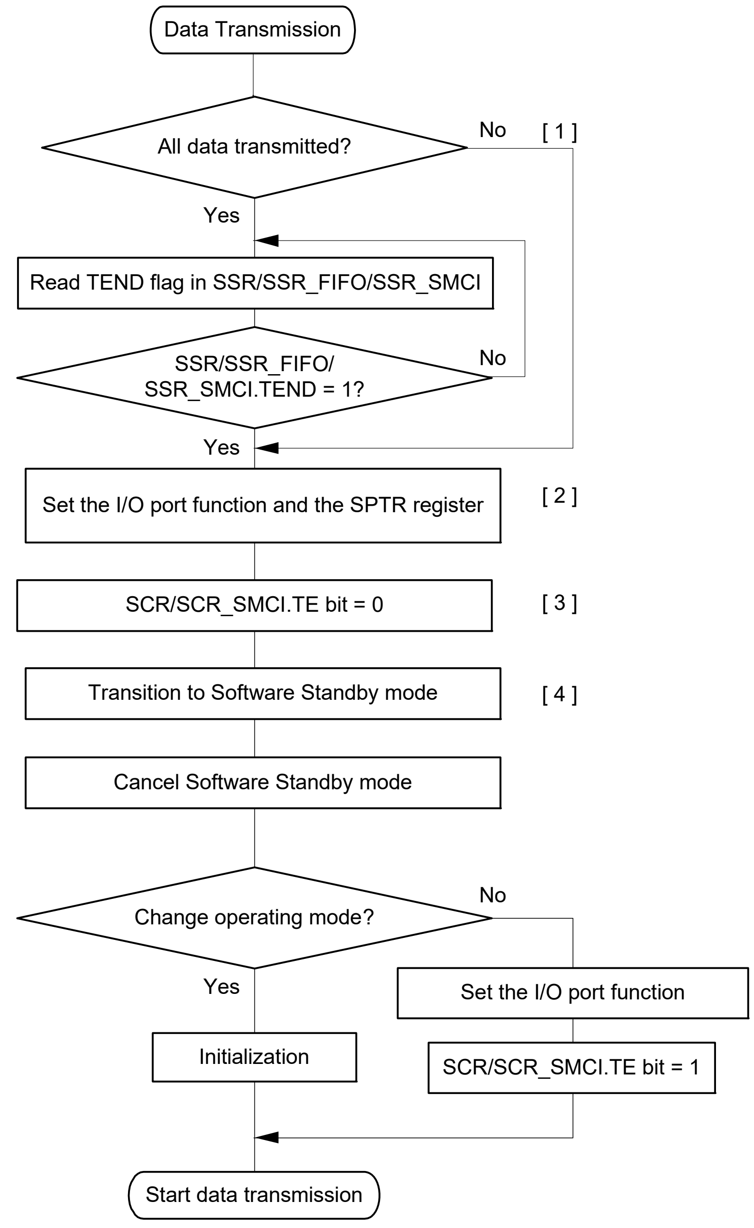
1. Set the operations after cancellation of the low power state.
2. Set CDR.CMPD and DCCR.DCME to 1.
3. Set the receive operations (RE = 1 in SCR/SCR_SMCI).
4. Set the module-stop state or Software Standby mode.

When the SCI transfers to the low power mode, if the receive data pin (RXD) is at the low level, set the RXDESEL bit in SEMR to 0. If RXDESEL is set to 1, there is a possibility that a start bit (falling 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.


Data Transmission
All data transmitted?
No
[1]
Yes
Read TEND flag in SSR/SSR_FIFO/SSR_SMCI
SSR/SSR_FIFO/ SSR_SMCI.TEND = 1?
No
Yes
Set the I/O port function and the SPTR register
[2]
SCR/SCR_SMCI.TE bit = 0
[3]
Transition to Software Standby mode
[4]
Cancel Software Standby mode
Change operating mode?
No
Yes
Set the I/O port function
Initialization
SCR/SCR_SMCI.TE bit = 1
Start data transmission

[1] Data being transmitted is lost. Data can be normally transmitted from the CPU by setting the TE bit in SCR/SCR_SMCI to 1, reading SSR/SSR_FIFO/SSR_SMCI, and writing in Software Standby mode. However, if the 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] Set the I/O port function and SPTR register settings to switch the TXDn pin to operate as a general I/O port.

[3] Set SCR/SCR_SMCI.TE to 0. If SCR/ SCR_SMCI.TIE = 1 and SCR/ SCR_SMCI.TEIE = 1, these are set to 0 simultaneously with the SCR.TE bit.

[4] This includes the setting for the module-stop state.

Figure 29.73 Example flow of transition to Software Standby mode during transmission

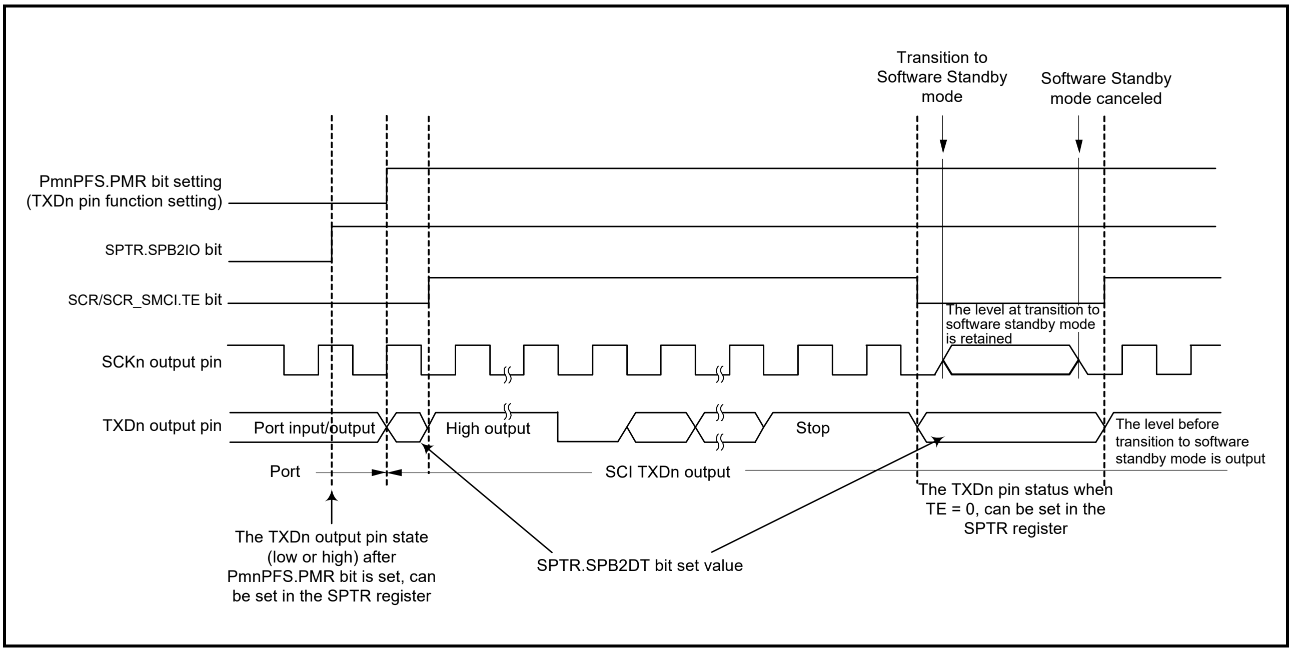

Transition to Software Standby mode
Software Standby mode canceled
PmnPFS.PMR bit setting (TXDn pin function setting)
SPTR.SPB2IO bit
SCR/SCR_SMCI.TE bit
The level at transition to software standby mode is retained
SCKn output pin
TXDn output pin
Port input/output
High output
Stop
The level before transition to software standby mode is output
Port
SCI TXDn output
The TXDn pin status when TE = 0, can be set in the SPTR register
The TXDn output pin state (low or high) after PmnPFS.PMR bit is set, can be set in the SPTR register
SPTR.SPB2DT bit set value

Figure 29.74 Port pin states during transition to Software Standby mode with internal clock and asynchronous transmission

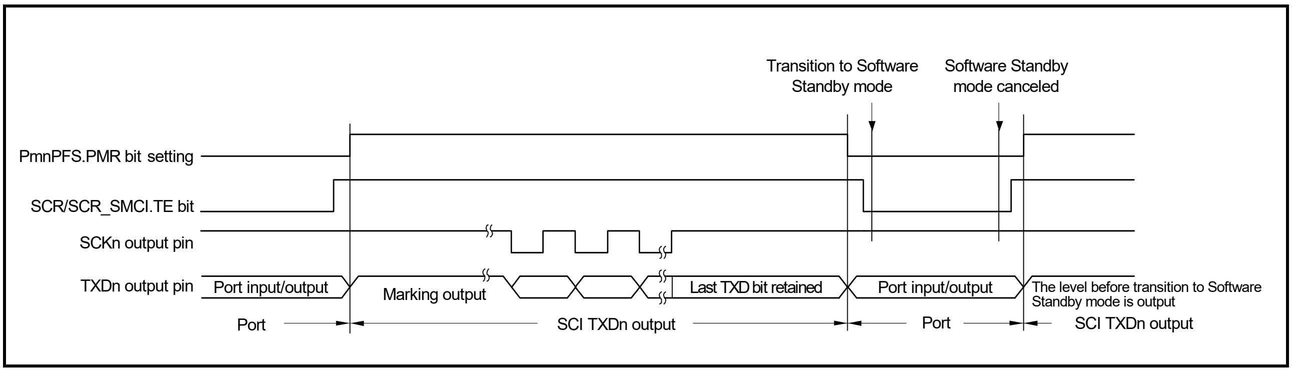
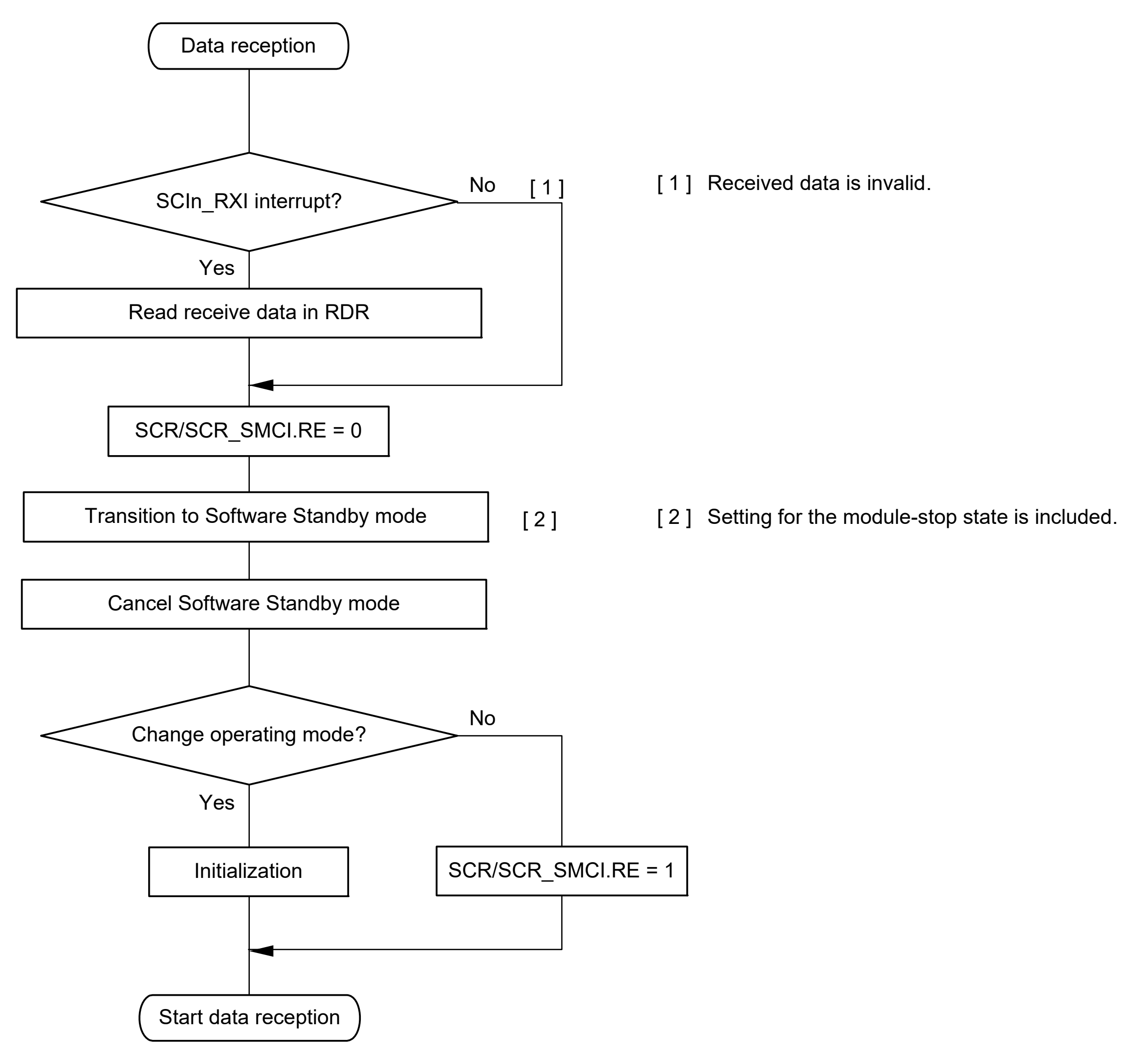

Transition to Software Standby mode
Software Standby mode canceled
PmnPFS.PMR bit setting
SCR/SCR_SMCI.TE bit
SCKn output pin
TXDn output pin
Port input/output
Marking output
Last TXD bit retained
Port input/output
The level before transition to Software Standby mode is output
Port
SCI TXDn output
Port
SCI TXDn output

Figure 29.75 Port pin states during transition to Software Standby mode with internal clock and clock synchronous transmission


Data reception
SCIn_RXI interrupt?
No
[1]
Yes
Read receive data in RDR
SCR/SCR_SMCI.RE = 0
Transition to Software Standby mode
[2]
Cancel Software Standby mode
Change operating mode?
No
Yes
Initialization
SCR/SCR_SMCI.RE = 1
Start data reception

[1] Received data is invalid.

[2] Setting for the module-stop state is included.

Figure 29.76 Example flow of transition to Software Standby mode during reception

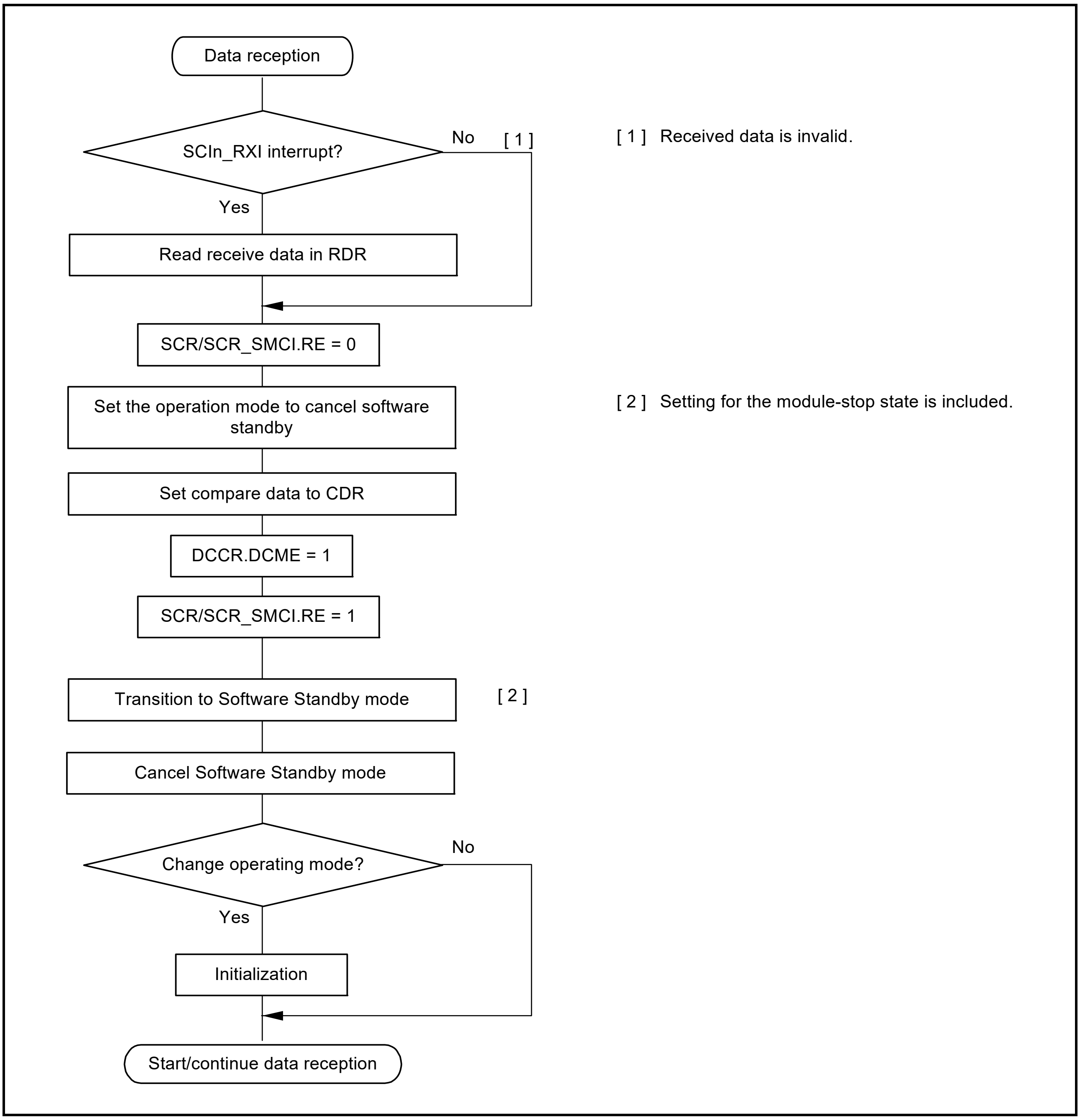

Data reception
SCIn_RXI interrupt?
No
[1]
Yes
Read receive data in RDR
SCR/SCR_SMCI.RE = 0
Set the operation mode to cancel software standby
Set compare data to CDR
DCCR.DCME = 1
SCR/SCR_SMCI.RE = 1
Transition to Software Standby mode
[2]
Cancel Software Standby mode
Change operating mode?
No
Yes
Initialization
Start/continue data reception
[1] Received data is invalid.
[2] Setting for the module-stop state is included.

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 that no framing error occurred, it is set to 1 again. When the RXDESEL bit in SEMR is 1, the SCI sets the FER flag in SSR to 1 and stops receiving operations until a start bit of the next data frame is detected. If the FER flag in SSR is 0, the FER flag in SSR 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 one frame, reception stops. When a framing error is detected, a break can be detected by reading the RXDMON bit value in SPTR. After the RXD signal is in the mark state and the break ends, reception of data to FRDRHL resumes.

29.14.4 Mark State and Production of Breaks

When the TE bit is 0 in SCR/SCR_SMCI, disabling serial transmission, the state of the TXDn pin can be set using the SPB2IO bit in SPTR and the SPB2DT bit in SPTR. With this approach, a TXDn pin can be placed in the mark state to transmit a break.

Before setting the TE bit in SCR/SCR_SMCI 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 TE bit in SCR/SCR_SMCI to 0. When the TE bit in SCR/SCR_SMCI 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 and Simple SPI Modes

Transmission cannot start when a receive error flag (ORER) in SSR/SSR_FIFO is set to 1, even when data is written to the TDR or FTDRL*1 registers. Be sure to set the receive error flags to 0 before starting transmission.

Note: The receive error flags cannot be set to 0 if serial reception is disabled by setting the RE bit in SCR/SCR_SMCI to 0.

Note 1. 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:

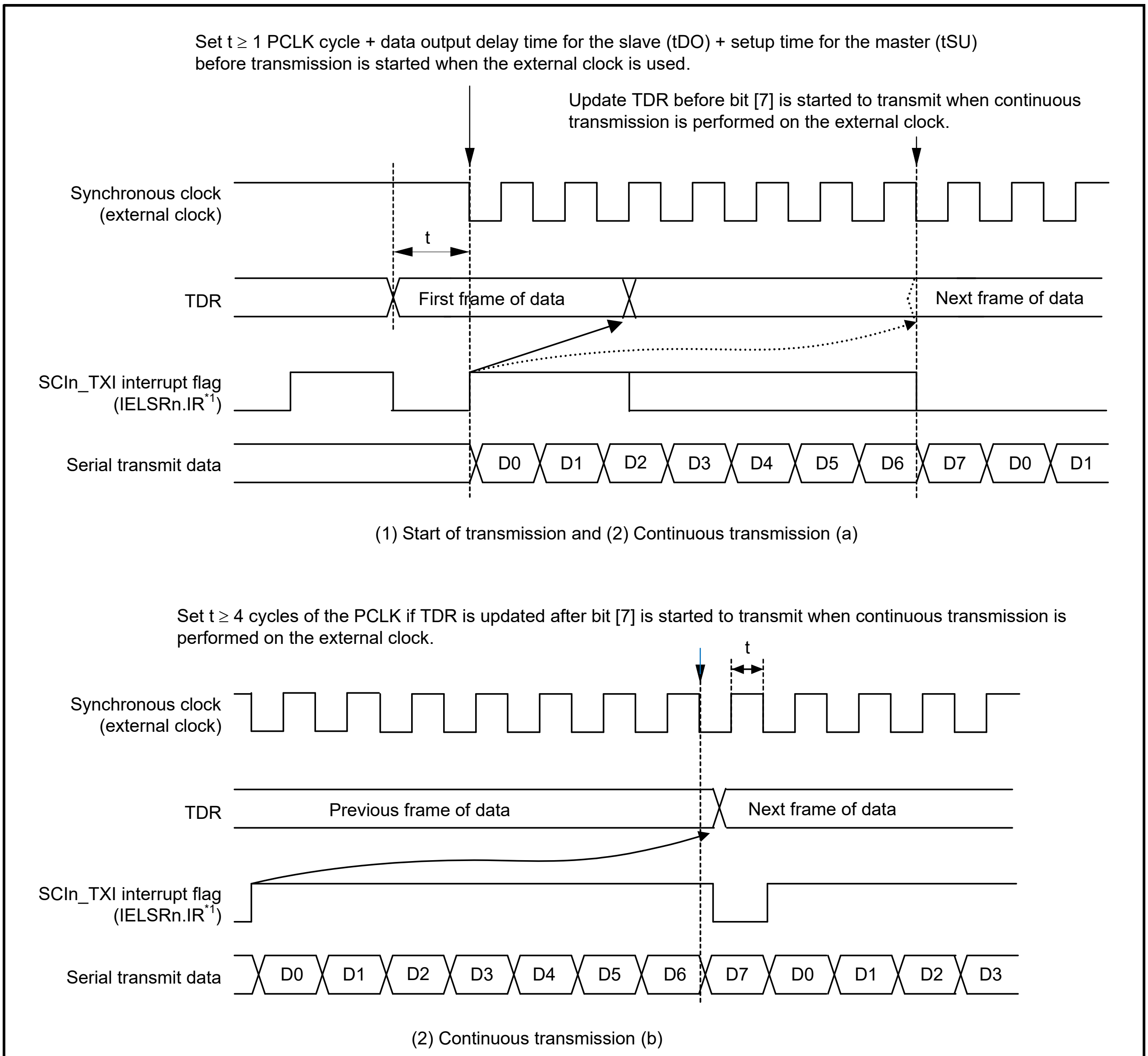
1 PCLK cycle + data output delay time for the slave + setup time for the master (tSU).

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 when 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.


Set t ≥ 1 PCLK cycle + data output delay time for the slave (tDO) + setup time for the master (tSU) before transmission is started when the external clock is used.
Update TDR before bit [7] is started to transmit when continuous transmission is performed on the external clock.
Synchronous clock (external clock)
t
TDR
First frame of data
Next frame of data
SCIn_TXI interrupt flag (IELSRn.IR*1)
Serial transmit data
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1
(1) Start of transmission and (2) Continuous transmission (a)
Set t ≥ 4 cycles of the PCLK if TDR is updated after bit [7] is started to transmit when continuous transmission is performed on the external clock.
t
Synchronous clock (external clock)
TDR
Previous frame of data
Next frame of data
SCIn_TXI interrupt flag (IELSRn.IR*1)
Serial transmit data
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3
(2) Continuous transmission (b)

Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

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 or reception operations using the DMAC or DTC, do not set the transfer information 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 the TE bit is 1 in SCR register. 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, 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 is 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 CKPH and CKPOL bits when the SSE bit is 1 in the SPMR register. This prevents the clock line from being placed in the high-impedance state when the TE bit is set to 0 or unexpected edges from being generated on the clock line when the TE bit changes from 0 to 1 in the SCR register. When the 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 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.

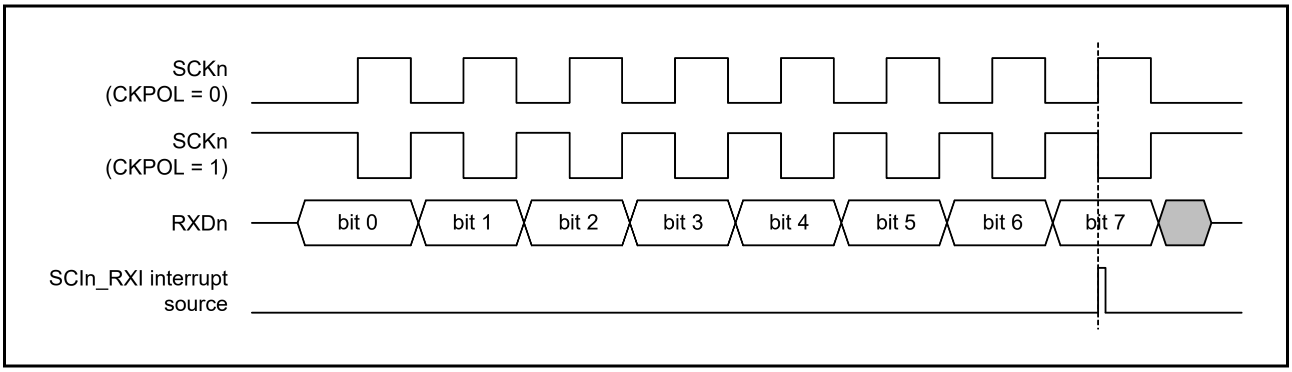


Figure 29.79 Timing of SCIn_RXI interrupt in simple SPI mode with clock delay

(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. I^{2}C Bus Interface (IIC)

30.1 Overview

The MCU has a 2-channel I^{2}C Bus Interface (IIC). The IIC module conforms with and provides a subset of the NXP I^{2}C (Inter-Integrated Circuit) bus interface functions.

Table 30.1 lists the IIC specifications, Figure 30.1 shows a block diagram, and Figure 30.2 shows an example of I/O pin connections to external circuits, with an I^{2}C bus configuration example. Table 30.2 lists the I/O pins.

Table 30.1 IIC specifications (1 of 2)

Parameter	Description
Communications format	• I^{2}C 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 can create conflict on the bus, loss of arbitration is detected by testing for a mismatch between the internal signal for the SDA line and the level on the SDA line - In master operation, loss of arbitration is detected by testing for a mismatch 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 mismatching 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 30.1 IIC specifications (2 of 2)

Parameter	Description
Event link function (output)	• Transfer error or event occurrences (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.

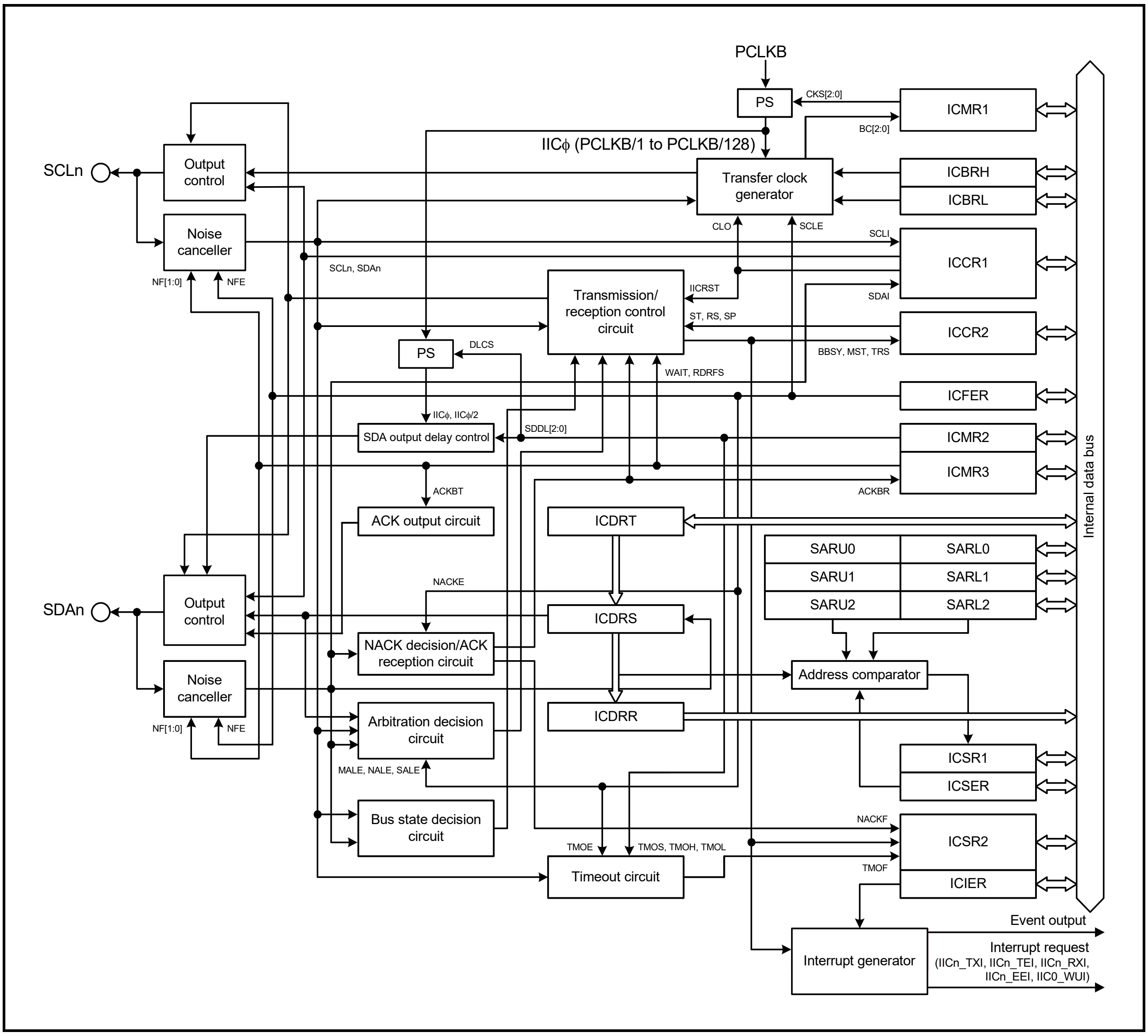

PCLKB
PS
CKS[2:0]
ICMR1
BC[2:0]
IICϕ (PCLKB/1 to PCLKB/128)
SCLn
Output control
Transfer clock generator
ICBRH
ICBRL
Noise canceller
CLO
SCLE
SCLI
ICCR1
SCLn, SDAn
NF[1:0]
NFE
IICRST
Transmission/ reception control circuit
SDAI
ST, RS, SP
ICCR2
PS
DLCS
BBSY, MST, TRS
WAIT, RDRFS
ICFER
IICϕ, IICϕ/2
SDDL[2:0]
SDA output delay control
ICMR2
ICMR3
ACKBT
ACKBR
ACK output circuit
ICDRT
Internal data bus
SARU0
SARL0
SARU1
SARL1
SARU2
SARL2
NACKE
SDAn
Output control
ICDRS
NACK decision/ACK reception circuit
Address comparator
Noise canceller
NF[1:0]
NFE
Arbitration decision circuit
ICDRR
ICSR1
MALE, NALE, SALE
ICSER
NACKF
Bus state decision circuit
ICSR2
TMOE
TMOS, TMOH, TMOL
TMOF
Timeout circuit
ICIER
Event output
Interrupt generator
Interrupt request (IICn_TXI, IICn_TEI, IICn_RXI, IICn_EEI, IIC0_WUI)

Figure 30.1 IIC block diagram

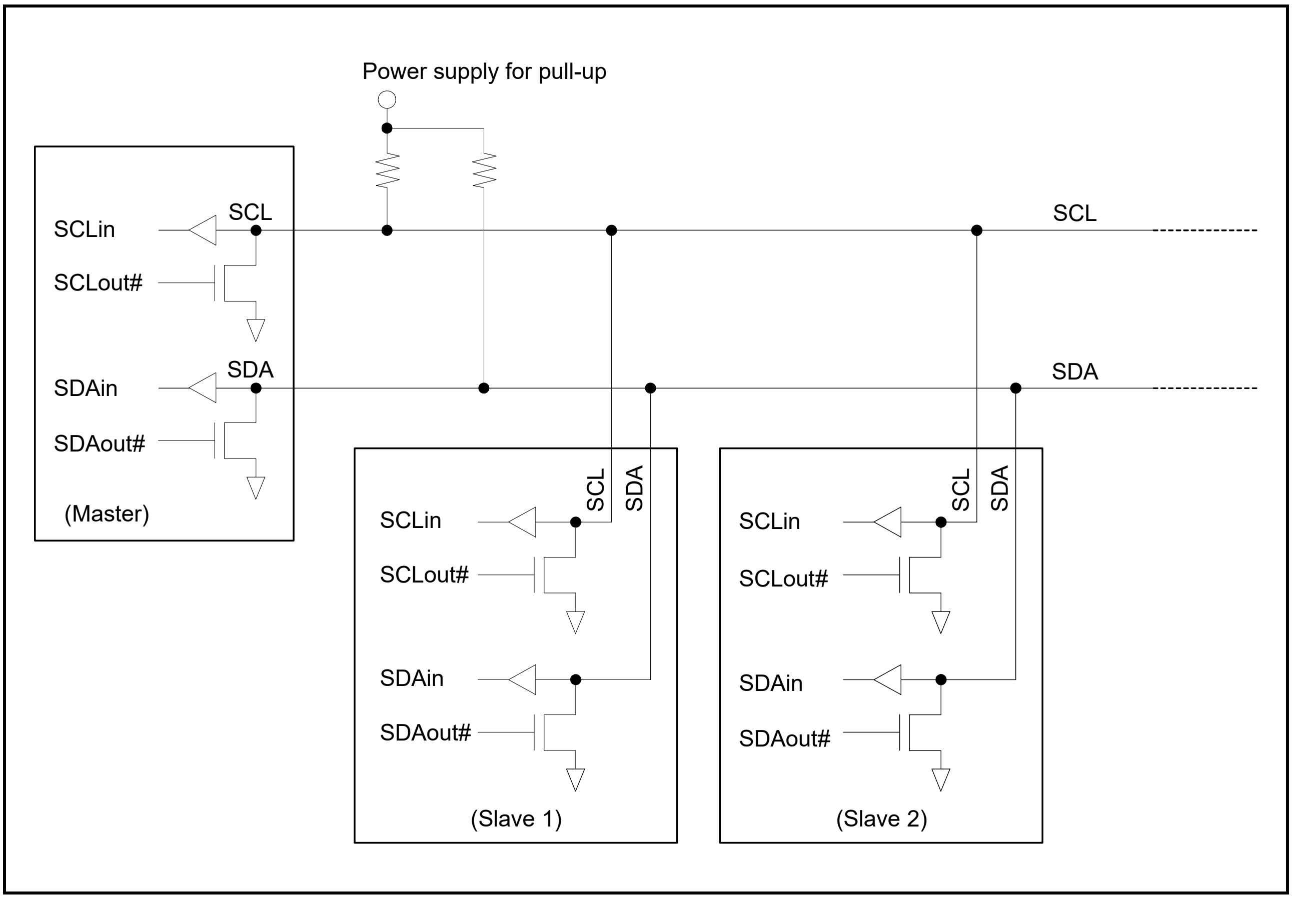

Power supply for pull-up
SCLin
SCL
SCLout#
SDAin
SDA
SDAout#
(Master)
SCL
SDA
SCLin
SCLout#
SDAin
SDAout#
SCL
SDA
(Slave 1)
SCLin
SCLout#
SDAin
SDAout#
SCL
SDA
(Slave 2)

Figure 30.2 I/O pin connection to the external circuit (I^2C bus configuration example)

The input level of the signals for IIC is CMOS when I^2C bus is selected (ICMR3.SMBS = 0), or TTL when SMBus is selected (ICMR3.SMBS = 1).

Table 30.2 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

30.2 Register Descriptions

30.2.1 I^{2}C Bus Control Register 1 (ICCR1)

Address(es): IIC0.ICCR1 4005 3000h, IIC1.ICCR1 4005 3100h

	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 drives the SDAn pin low 1: IIC releases the SDAn pin. • Write: 0: IIC drives SDAn pin low 1: IIC releases SDAn pin.	R/W
b3	SCLO	SCL Output Control/Monitor	• Read: 0: IIC drives the SCLn pin low 1: IIC releases the SCLn pin. • Write: 0: IIC drives SCLn pin low 1: IIC releases 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 1 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 specified 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 30.12.2, 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 30.3 lists the IIC resets.

The IIC reset initializes all registers except ICCR1.ICE and ICCR1.ICCRST bits and internal states of the IIC.

The internal reset initializes the following:

- Bit counter (ICMR1.BC[2:0] bits)
- I^{2}C Bus Shift Register (ICDRS)
- I^{2}C Bus Status Registers (ICSR1 and ICSR2)
- SDAO and SCLO Output Control/Monitor (ICCR1.SCLO and ICCR1.SDAO bits)
- I^{2}C Bus Control Register 2 (except ICCR2.BBSY bit).

For the reset conditions of each register, see section 30.15, State of Registers when Issuing each Condition.

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the IIC without initializing the port settings and the control and setting registers of the IIC. If the IIC hangs 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 and the master devices might enter different states, because the bit counter information differs. For this reason, do not initiate an internal reset in slave mode. Initiate recovery processing from the master device. If an internal reset is required because the IIC hangs with the 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 30.3 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 following: • ICMR1.BC[2:0] bits • ICSR1 register • ICSR2 register • ICDRS register • ICCR1.SCLO bit • ICCR1.SDAO bit • ICCR2 register (except ICCR2.BBSY bit) • 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 30.3 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.

30.2.2 I^{2}C Bus Control Register 2 (ICCR2)

Address(es): IIC0.ICCR2 4005 3001h, IIC1.ICCR2 4005 3101h

	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: I^{2}C bus released (bus free state) 1: I^{2}C 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 30.11, Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the ST bit.

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition is issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Only set the ST bit to 1 (start condition request) when the BBSY flag is set to 0 (bus free state). Arbitration might be lost if the ST bit is set to 1 (start condition request) when the BBSY flag is 1 (bus busy state).

RS bit (Restart Condition Issuance Request)

The RS bit requests that a restart condition be issued in master mode.

When this bit is set to 1 to request a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on issuing a restart condition, see section 30.11, 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 30.11, 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 IIC operating mode.

The value of the TRS bit automatically changes to 1 for transmit mode or 0 for receive mode when a start condition is issued or detected and the R/W# bit is set. Although writing to the TRS bit is possible when the MTWP bit in ICMR1 is set to 1, writing to 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 with the SCLn line high, assuming that a start condition was issued.

This flag is set to 0 when the SDAn line changes from low to high with the SCLn line high, 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).

30.2.3 I^{2}C Bus Mode Register 1 (ICMR1)

Address(es): IIC0.ICMR1 4005 3002h, IIC1.ICMR1 4005 3102h

	b7	b6	b5	b4	b3	b2	b1	b0
	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset:	0	0	0	0	1	0	0	0

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 required to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one for an additional acknowledge bit, between transferred frames when the 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.

30.2.4 I^{2}C Bus Mode Register 2 (ICMR2)

Address(es): IIC0.ICMR2 4005 3003h, IIC1.ICMR2 4005 3103h

	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 30.12.1, 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 SDDL[2:0] bits can be used to delay the SDA output. This counter works with the clock source selected by the DLCS bit. The SDDL[2:0] 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 30.5, 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)

30.2.5 I2C Bus Mode Register 3 (ICMR3)

Address(es): IIC0.ICMR3 4005 3004h, IIC1.ICMR3 4005 3104h

	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 1 IICϕ cycle filtered out (single-stage filter) 0 1: Noise of up to 2 IICϕ cycles filtered out (2-stage filter) 1 0: Noise of up to 3 IICϕ cycles filtered out (3-stage filter) 1 1: Noise of up to 4 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 software 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 30.6, Digital Noise Filter Circuits.

Note: Set the noise range to be filtered 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 by the noise filter function of the IIC, 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 controls the value of 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 issue 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 SCLn line is 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 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 a single-byte of data is received in receive mode.

When the WAIT is 0, the receive operation continues 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 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/I^{2}C Bus Select)

Setting the SMBS bit to 1 selects the SMBus and enables the HOAE bit in ICSER.

30.2.6 I^{2}C Bus Function Enable Register (ICFER)

Address(es): IIC0.ICFER 4005 3005h, IIC1.ICFER 4005 3105h

	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 disabled 1: NACK transmission arbitration-lost detection 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 30.12.1, Timeout Function.

MALE bit (Master Arbitration-Lost Detection Enable)

The MALE bit specifies whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE bit (NACK Transmission Arbitration-Lost Detection Enable)

The NALE bit specifies whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode, for 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. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended. When the NACKE bit is 0, the next transfer operation continues regardless of the received acknowledge content.

For details on the NACK reception transfer suspension function, see section 30.9.2, NACK Reception Transfer Suspension Function.

SCLE bit (SCL Synchronous Circuit Enable)

The SCLE bit specifies whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.

When the SCLE bit is set to 0 (no SCL synchronous circuit used), the IIC does not synchronize the SCL clock with the SCL input clock. 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 I^{2}C bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, a short-cycle SCL clock that does not meet the specification might be output. When no SCL synchronous circuit is used, it also affects the issuing of the start, restart, and stop conditions, and the continuous output of extra SCL clock cycles.

The SCLE bit must not be set to 0 except for checking the output of the set transfer rate.

30.2.7 I^{2}C Bus Status Enable Register (ICSER)

Address(es): IIC0.ICSER 4005 3006h, IIC1.ICSER 4005 3106h

	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 condition 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 30.7.3, 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.

30.2.8 I^{2}C Bus Interrupt Enable Register (ICIER)

Address(es): IIC0.ICIER 4005 3007h, IIC1.ICIER 4005 3107h

	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.

30.2.9 I^{2}C Bus Status Register 1 (ICSR1)

Address(es): IIC0.ICSR1 4005 3008h, IIC1.ICSR1 4005 3108h

	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

Bit	Symbol	Bit name	Description	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 condition 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 condition 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.

30.2.10 I^{2}C Bus Status Register 2 (ICSR2)

Address(es): IIC0.ICSR2 4005 3009h, IIC1.ICSR2 4005 3109h

	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

Bit	Symbol	Bit name	Description	R/W
b6	TEND	Transmit End Flag	0: Data being transmitted 1: Data transmission 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 TMOE bit in ICFER 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 is 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 30.4 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 and 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 or 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 condition or a 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 or 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.

30.2.11 I^{2}C 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 30.5.	R/W
b5	WUF	Wakeup Event Occurrence Flag	0: Slave address not matching during wakeup 1: Slave address matching during wakeup.	R/W

Bit	Symbol	Bit name	Description	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 30.5 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 recovery 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.

30.2.12 I^{2}C Bus Wakeup Unit Register 2 (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	0	1

Bit	Symbol	Bit name	Description	R/W
b0	WUSEN	Wakeup Function Synchronous Enable	0: IIC asynchronous operation enabled 1: IIC synchronous operation enabled.	R/W
b1	WUASYF	Wakeup Function Asynchronous Operation Status Flag	0: IIC synchronous operation enabled 1: IIC asynchronous operation enabled.	R
b2	WUSYF	Wakeup Function Synchronous Operation Status Flag	0: IIC asynchronous operation enabled 1: IIC synchronous operation enabled.	R
b7 to b3	—	Reserved	These bits are read as1. The write value should be 1.	R/W

WUSEN bit (Wakeup Function Synchronous Enable)

The WUSEN bit is used in combination with the WUASYF flag (or WUSYF flag) to switch between PCLKB synchronous and asynchronous operation, when the wakeup function is enabled (ICWUR.WUE = 1).

The PCLKB operation switches from synchronous to asynchronous operation:

- When the ICCR2.BBSY flag is 0 (bus free state) if 0 is written to the WUSEN bit while the WUASYF flag is 0.
- The reception occurs independently of the operation of PCLKB (with PCLKB stopped) after it switches to the PCLKB asynchronous operation, on wakeup event detection.

The PCLKB operation switches asynchronous to synchronous operation:

- When 1 is written to the WUSEN bit with the WUASYF flag at 1, when a wakeup event is detected. After writing 1, the WUASYF flag immediately becomes 0.

- When a stop condition is detected with a wakeup event undetected.

WUASYF flag (Wakeup Function Asynchronous Operation Status Flag)

This flag can place the IIC in PCLKB asynchronous operation when the wakeup function is enabled (ICWUR.WUE = 1).

[Setting condition]

- When the ICCR2.BBSY flag is 0, and WUSEN bit is set to 0, with the ICWUR.WUE bit set to 1.

[Clearing conditions]

- When 1 is written to the WUSEN bit, after detecting a wakeup event with ICWUR.WUE bit set to 1
- When a stop condition is detected with WUSEN bit set to 1 before detecting the wakeup event, with WUASY flag set to 1 and ICWUR.WUE bit set to 1
- When 1 is written to the WUSEN bit with the WUASYF flag set to 1, and a wakeup event is detected with ICWUR.WUE set to 1
- When ICCR1.ICE = 0 and ICCRST = 1 (ICC reset)
- When ICWUR.WUE = 0.

WUSYF flag (Wakeup Function Synchronous Operation Status Flag)

This flag can place the IIC in PCLKB synchronous operation when the wakeup function is enabled (ICWUR.WUE = 1). When this flag is used, the WUASYF flag is reserved.

[Setting conditions]

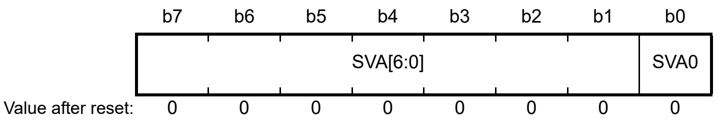
- When 1 is written to the WUSEN bit after detecting a wakeup event with ICWUR.WUE bit set to 1, and WUSYF flag set to 0
- When a stop condition is detected with WUSEN bit set to 1, before detecting a wakeup event with the WUSYF flag set to 0 and the ICWUR.WUE bit set to 1
- When ICCR1.ICE = 0 and ICCRST = 1 (ICC reset)
- When ICWUR.WUE = 0.

[Clearing condition]

- When the ICCR2.BBSY flag is 0 with the ICWUR.WUE bit set to 1, after writing 0 to the WUSEN bit.

30.2.13 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): IIC0.SARL0 4005 300Ah, IIC1.SARL0 4005 310Ah,
IIC0.SARL1 4005 300Ch, IIC1.SARL1 4005 310Ch,
IIC0.SARL2 4005 300Eh, IIC1.SARL2 4005 310Eh



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), the SVA0 bit functions as the LSB of a 10-bit address and is combined with the SVA[6:0] bits to form the lower 8 bits of a 10-bit address.

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), the SVA[6:0] 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. When the SARyE bit in ICSER is 0, the setting of these bits is ignored.

30.2.14 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): IIC0.SARU0 4005 300Bh, IIC1.SARU0 4005 310Bh,
IIC0.SARU1 4005 300Dh, IIC1.SARU1 4005 310Dh,
IIC0.SARU2 4005 300Fh, IIC1.SARU2 4005 310Fh

	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: Select 7-bit address format 1: Select 10-bit address format.	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), the SVA[1:0] 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.

30.2.15 I^{2}C Bus Bit Rate Low-Level Register (ICBRL)

Address(es): IIC0.ICBRL 4005 3010h, IIC1.ICBRL 4005 3110h

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

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

BRL[4:0] bits (Bit Rate Low-Level Period)

The BRL[4:0] bits set the low-level period of the SCL clock. ICBRL counts the low-level period with the internal reference clock source (IICϕ) specified in the CKS[2:0] bits in ICMR1. ICBRL also generates the data setup time for automatic SCL low-hold operation (see section 30.9, Automatic Low-Hold Function for SCL). When the IIC is used only in slave mode, the BRL[4:0] bits must be set to a value longer than the data setup time[*1].

If the digital noise filter is enabled (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 details on the number of stages, 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)

30.2.16 I^{2}C Bus Bit Rate High-Level Register (ICBRH)

Address(es): IIC0.ICBRH 4005 3011h, IIC1.ICBRH 4005 3111h

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

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

BRH[4:0] bits (Bit Rate High-Level Period)

The BRH[4:0] bits set the high-level period of the SCL clock. BRH[4:0] bits are valid in master mode. If the IIC is used only in slave mode, do not set the BRH[4:0] bits.

ICBRH counts the high-level period with the internal reference clock source (IICϕ) specified in the CKS[2:0] bits in ICMR1.

If the digital noise filter is enabled (the NFE bit in ICFER is 1), set these bits to a value at least one greater than the number of stages in the noise filter. For the number of stages in the noise filter, see the description of the 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 = $1/\{[(BRH + 1) + (BRL + 1)]/IIC\varphi^{*1} + tr^{*2} + tf^{*2}\}$
   Duty cycle = $\{tr + [(BRH + 1)/IIC\varphi]\}/\{tr + tf + [(BRH + 1) + (BRL + 1)]/IIC\varphi\}$
2) ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] = 000b (IICφ = PCLKB)
   Transfer rate = $1/\{[(BRH + 3) + (BRL+ 3)]/IIC\varphi + tr + tf\}$
   Duty cycle = $\{tr + [(BRH + 3)/IIC\varphi]\}/\{tr + tf + [(BRH + 3) + (BRL + 3)]/IIC\varphi\}$
3) ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] = 000b (IICφ = PCLKB)
   Transfer rate = $1/\{[(BRH + 3 + nf^{*3}) + (BRL + 3 + nf)]/IIC\varphi + tr + tf\}$
   Duty cycle = $\{tr + [(BRH + 3 + nf)/IIC\varphi]\}/\{tr + tf + [(BRH + 3 + nf) + (BRL + 3 + nf)]/IIC\varphi\}$
4) ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] ≠ 000b
   Transfer rate = $1/\{[(BRH + 2) + (BRL + 2)]/IIC\varphi + tr + tf\}$
   Duty cycle = $\{tr + [(BRH + 2)/IIC\varphi]\}/\{tr + tf + [(BRH + 2) + (BRL + 2)]/IIC\varphi\}$
5) ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] ≠ 000b
   Transfer rate = $1/\{[(BRH + 2 + nf) + (BRL + 2 + nf)]/IIC\varphi + tr + tf\}$
   Duty cycle = $\{tr + [(BRH + 2 + nf)/IIC\varphi]\}/\{tr + tf + [(BRH + 2 + nf) + (BRL + 2 + nf)]/IIC\varphi\}$

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 I^{2}C bus standard from NXP Semiconductors.
Note 3. nf = Number of digital noise filter stages selected in the ICMR3.NF bits.

Table 30.6 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 30.7 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 30.8 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

30.2.17 I^{2}C Bus Transmit Data Register (ICDRT)

Address(es): IIC0.ICDRT 4005 3012h, IIC1.ICDRT 4005 3112h

	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 I^{2}C 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.

30.2.18 I^{2}C Bus Receive Data Register (ICDRR)

Address(es): IIC0.ICDRR 4005 3013h, IIC1.ICDRR 4005 3113h

	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 I^{2}C 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 for 1 cycle before the RDRF flag is set to 1 again.

30.2.19 I2C Bus Shift Register (ICDRS)

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

ICDRS is an 8-bit shift register to transmit and receive data. 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.

30.3 Operation

30.3.1 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 30.3 shows the I2C bus format and Figure 30.4 shows the I2C bus timing.

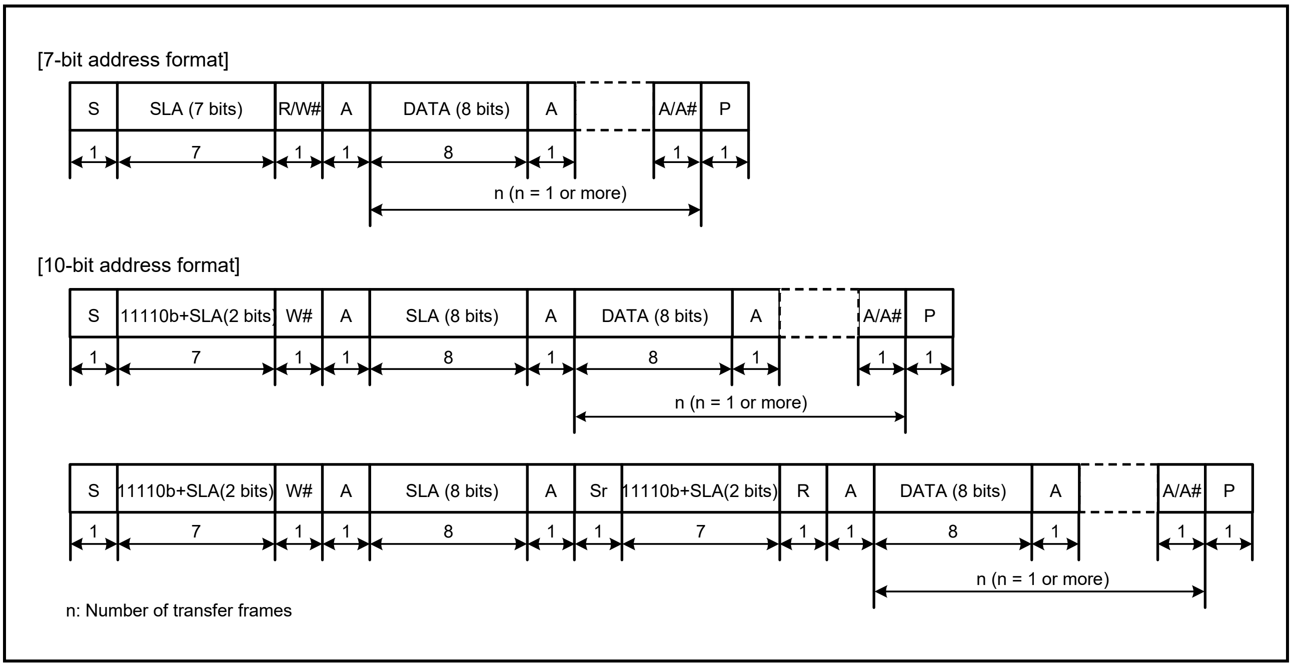

[7-bit address format]
S
SLA (7 bits)
R/W#
A
DATA (8 bits)
A
A/A#
P
1
7
1
1
8
1
1
1
n (n = 1 or more)
[10-bit address format]
S
11110b+SLA(2 bits)
W#
A
SLA (8 bits)
A
DATA (8 bits)
A
A/A#
P
1
7
1
1
8
1
8
1
1
1
n (n = 1 or more)
S
11110b+SLA(2 bits)
W#
A
SLA (8 bits)
A
Sr
11110b+SLA(2 bits)
R
A
DATA (8 bits)
A
A/A#
P
1
7
1
1
8
1
1
7
1
1
8
1
1
1
n (n = 1 or more)
n: Number of transfer frames

Figure 30.3 I2C bus format

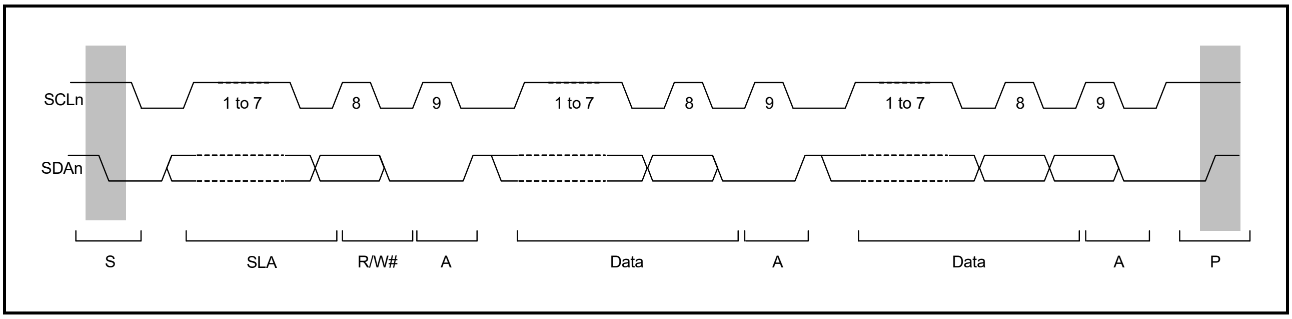

SCLn
SDAn
1 to 7
8
9
1 to 7
8
9
1 to 7
8
9
S
SLA
R/W#
A
Data
A
Data
A
P

Figure 30.4 I2C bus timing (SLA = 7 bits)

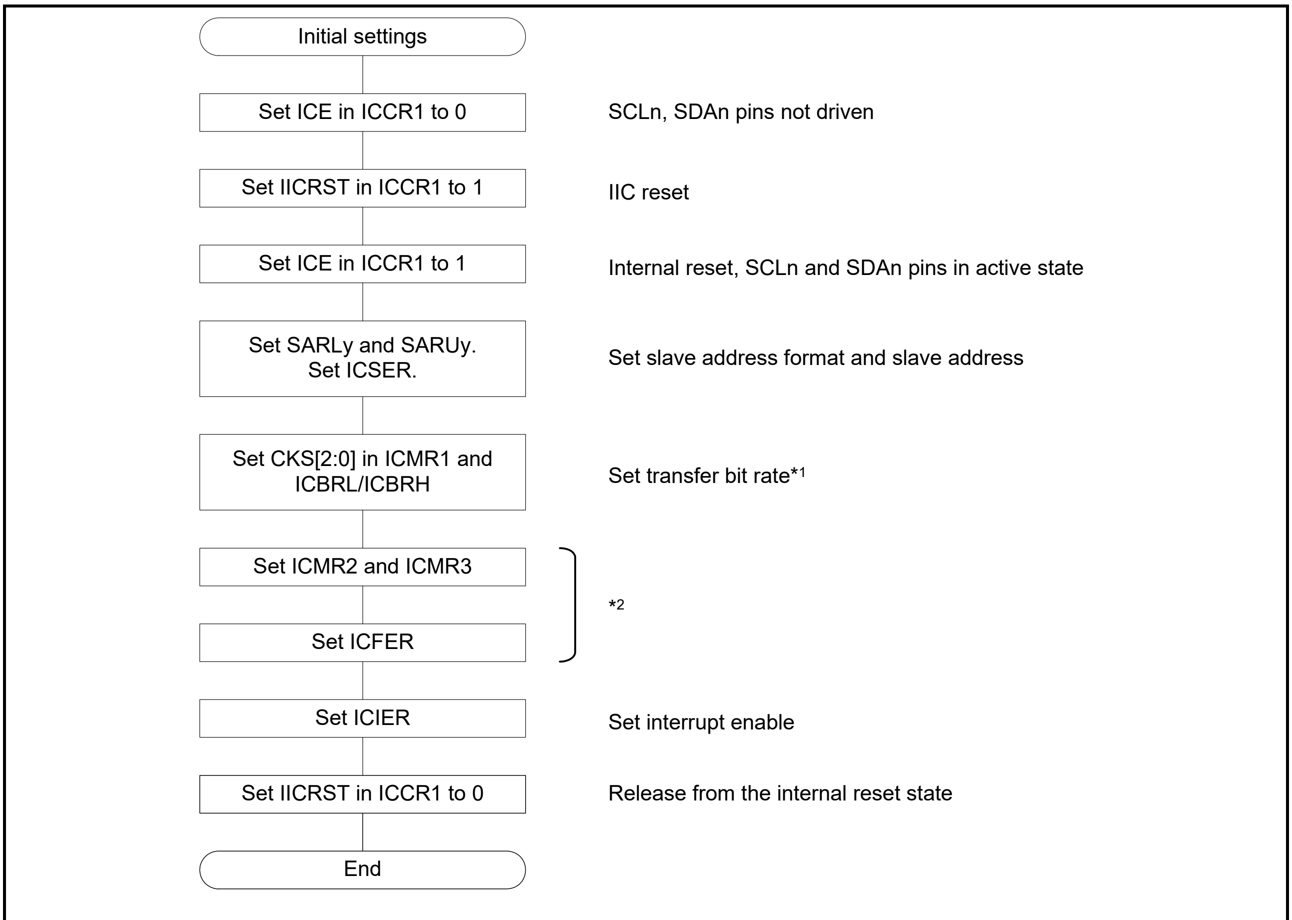
S:	Start condition. The master device drives the SDAn line low from high while the SCLn line is high.
SLA:	Slave address, by which the master device selects a slave device
R/W#:	Indicates the direction of data transfer: from the slave device to the master device when R/W# is 1, or from the master device 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.

30.3.2 Initial Settings

Before starting data transmission or reception, initialize the IIC using the procedure shown in Figure 30.5.

1. Set the ICCR1.ICE bit to 0 to set the SCLn and SDAn pins to the inactive state.
2. Set the ICCR1.IICRST bit to 1 to initiate IIC reset.
3. Set the ICCR1.ICE bit to 1 to initiate internal reset.
4. Set the SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL registers (y = 0 to 2), and set the other registers as required. For initial settings of the IIC, see Figure 30.5.
5. When the required register settings are complete, set the ICCR1.IICRST bit to 0 to release the IIC reset.

Note: This procedure is not required if the IIC initialization is already complete.


Initial settings
Set ICE in ICCR1 to 0
SCLn, SDAn pins not driven
Set IICRST in ICCR1 to 1
IIC reset
Set ICE in ICCR1 to 1
Internal reset, SCLn and SDAn pins in active state
Set SARLy and SARUy.
Set ICSER.
Set slave address format and slave address
Set CKS[2:0] in ICMR1 and ICBRL/ICBRH
Set transfer bit rate*1
Set ICMR2 and ICMR3
Set ICFER
*2
Set ICIER
Set interrupt enable
Set IICRST in ICCR1 to 0
Release from the internal reset state
End

y = 0 to 2

Note 1. When the IIC is used only in slave mode, set the ICBRL register to a value longer than the data setup time.
Note 2. Set these registers as required.

Figure 30.5 Example of IIC initialization flow

30.3.3 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 30.6 shows an example of master transmission. Figure 30.7 to Figure 30.9 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 30.3.2, 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 and START flags in ICSR2 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 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 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 is set to 1. After the byte containing the slave address and R/W# bit is transmitted, the value of the TRS bit automatically updates to select master transmit or master receive mode 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, see section 30.11.3, 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.

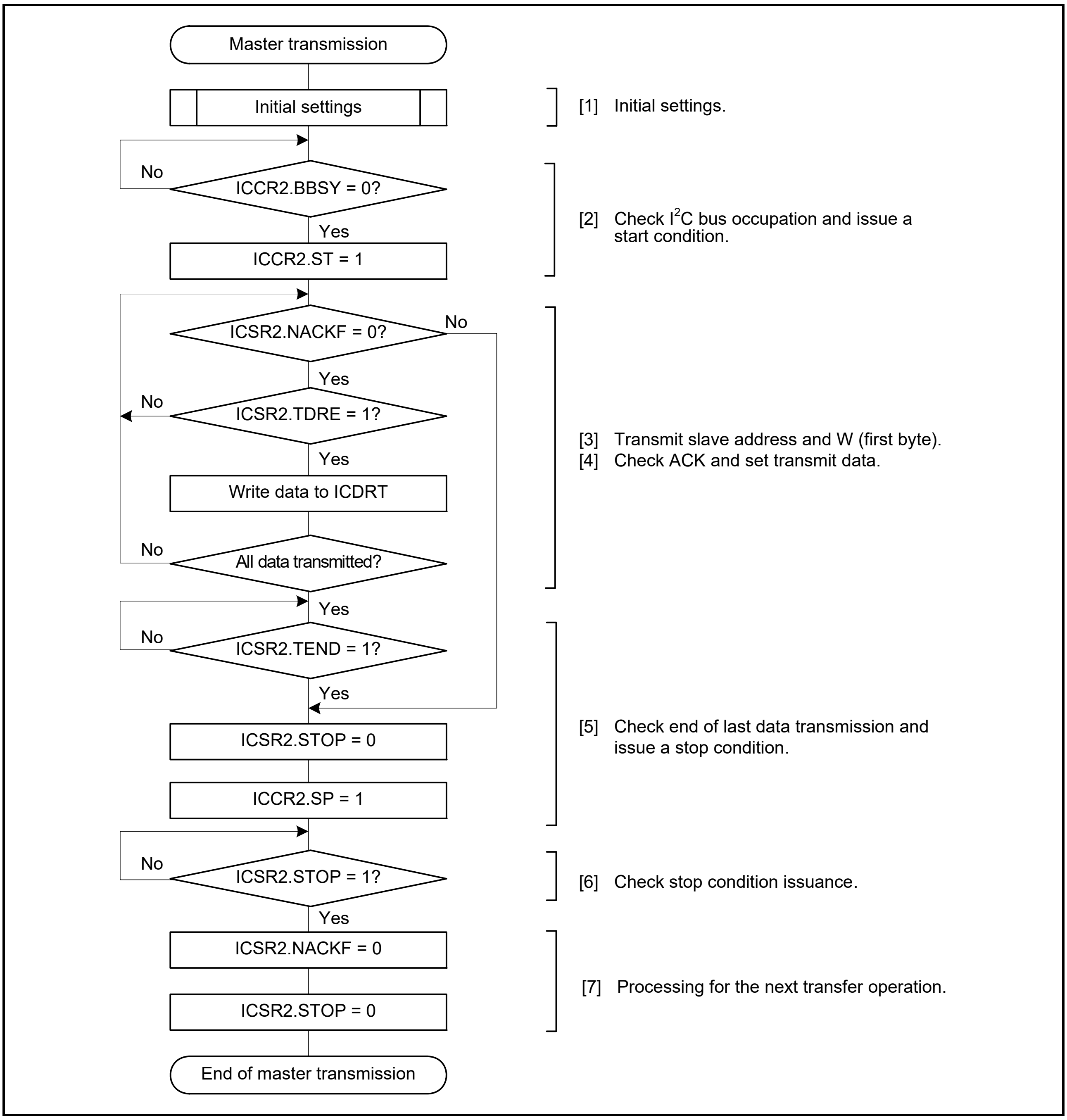

Master transmission
Initial settings
No
ICCR2.BBSY = 0?
Yes
ICCR2.ST = 1
ICSR2.NACKF = 0?
No
Yes
No
ICSR2.TDRE = 1?
Yes
Write data to ICDRT
No
All data transmitted?
Yes
No
ICSR2.TEND = 1?
Yes
ICSR2.STOP = 0
ICCR2.SP = 1
No
ICSR2.STOP = 1?
Yes
ICSR2.NACKF = 0
ICSR2.STOP = 0
End of master transmission
[1] Initial settings.
[2] Check I2C bus occupation and issue a start condition.
[3] Transmit slave address and W (first byte).
[4] Check ACK and set transmit data.
[5] Check end of last data transmission and issue a stop condition.
[6] Check stop condition issuance.
[7] Processing for the next transfer operation.

Figure 30.6 Example of master transmission flow

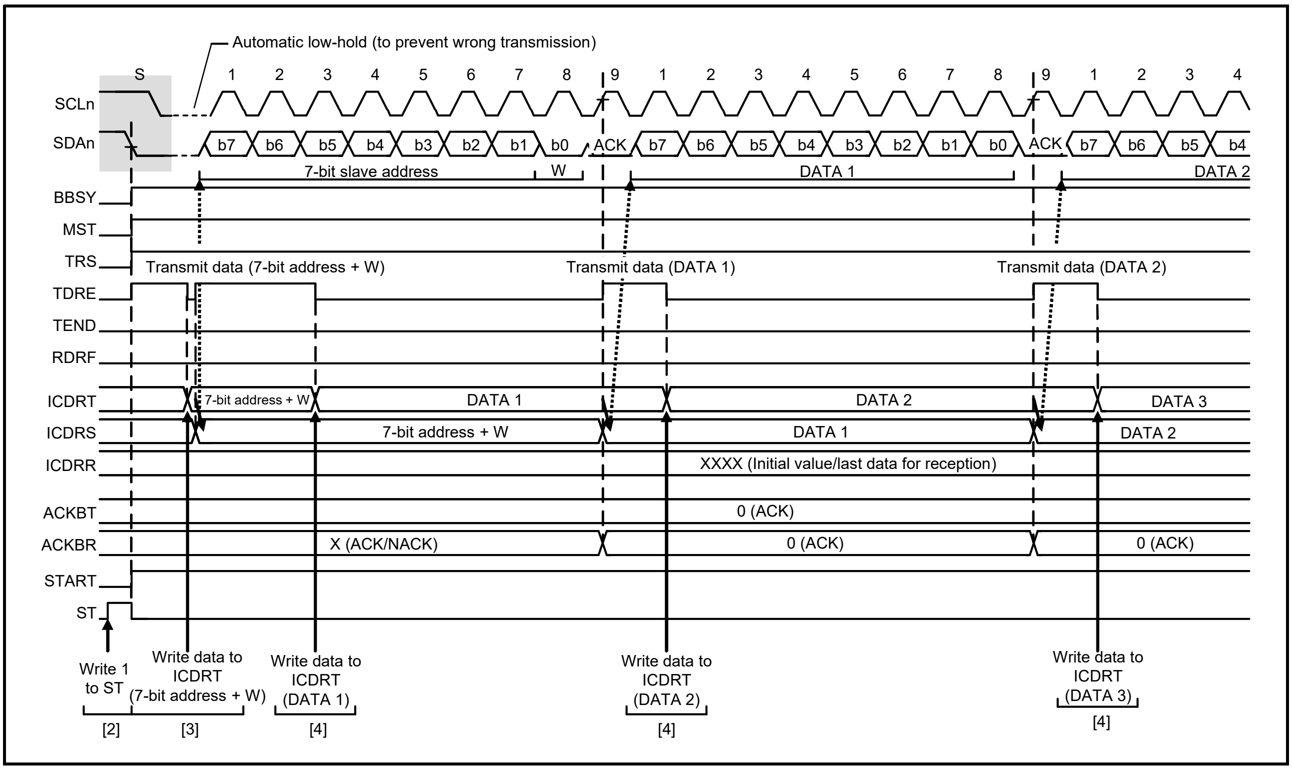

Automatic low-hold (to prevent wrong transmission)
S
1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4
SCLn
SDAn
b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4
7-bit slave address
W
DATA 1
DATA 2
BBSY
MST
TRS
Transmit data (7-bit address + W)
Transmit data (DATA 1)
Transmit data (DATA 2)
TDRE
TEND
RDRF
ICDRT
7-bit address + W
DATA 1
DATA 2
DATA 3
ICDRS
7-bit address + W
DATA 1
DATA 2
ICDRR
XXXX (Initial value/last data for reception)
ACKBT
0 (ACK)
ACKBR
X (ACK/NACK)
0 (ACK)
0 (ACK)
START
ST
Write 1 to ST
Write data to ICDRT (7-bit address + W)
Write data to ICDRT (DATA 1)
Write data to ICDRT (DATA 2)
Write data to ICDRT (DATA 3)
[2]
[3]
[4]
[4]
[4]

Figure 30.7 Master transmit operation timing (1) (7-bit address format)

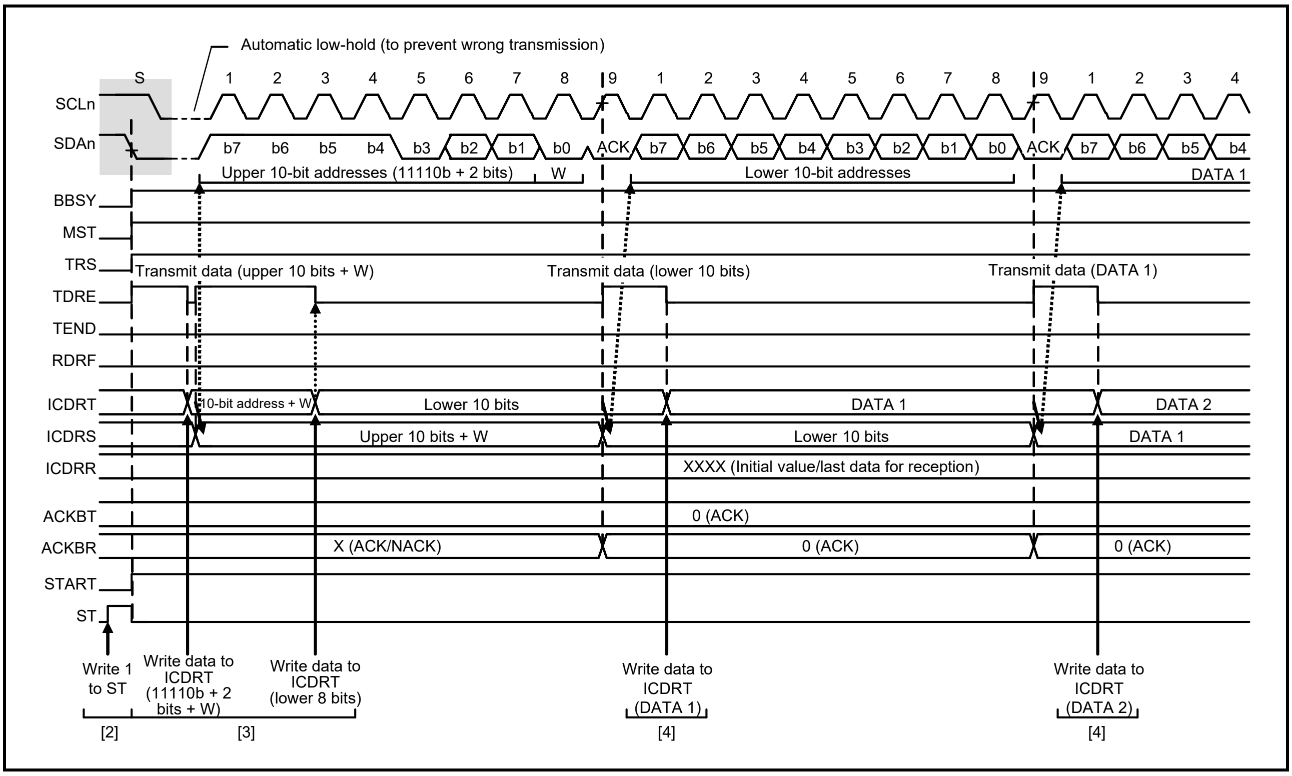

Automatic low-hold (to prevent wrong transmission)
S
1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4
SCLn
SDAn
b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4
Upper 10-bit addresses (11110b + 2 bits)
W
Lower 10-bit addresses
DATA 1
BBSY
MST
TRS
Transmit data (upper 10 bits + W)
Transmit data (lower 10 bits)
Transmit data (DATA 1)
TDRE
TEND
RDRF
ICDRT
10-bit address + W
Lower 10 bits
DATA 1
DATA 2
ICDRS
Upper 10 bits + W
Lower 10 bits
DATA 1
ICDRR
XXXX (Initial value/last data for reception)
ACKBT
0 (ACK)
ACKBR
X (ACK/NACK)
0 (ACK)
0 (ACK)
START
ST
Write 1 to ST
Write data to ICDRT (11110b + 2 bits + W)
Write data to ICDRT (lower 8 bits)
Write data to ICDRT (DATA 1)
Write data to ICDRT (DATA 2)
[2]
[3]
[4]
[4]

Figure 30.8 Master transmit operation timing (2) (10-bit address format)

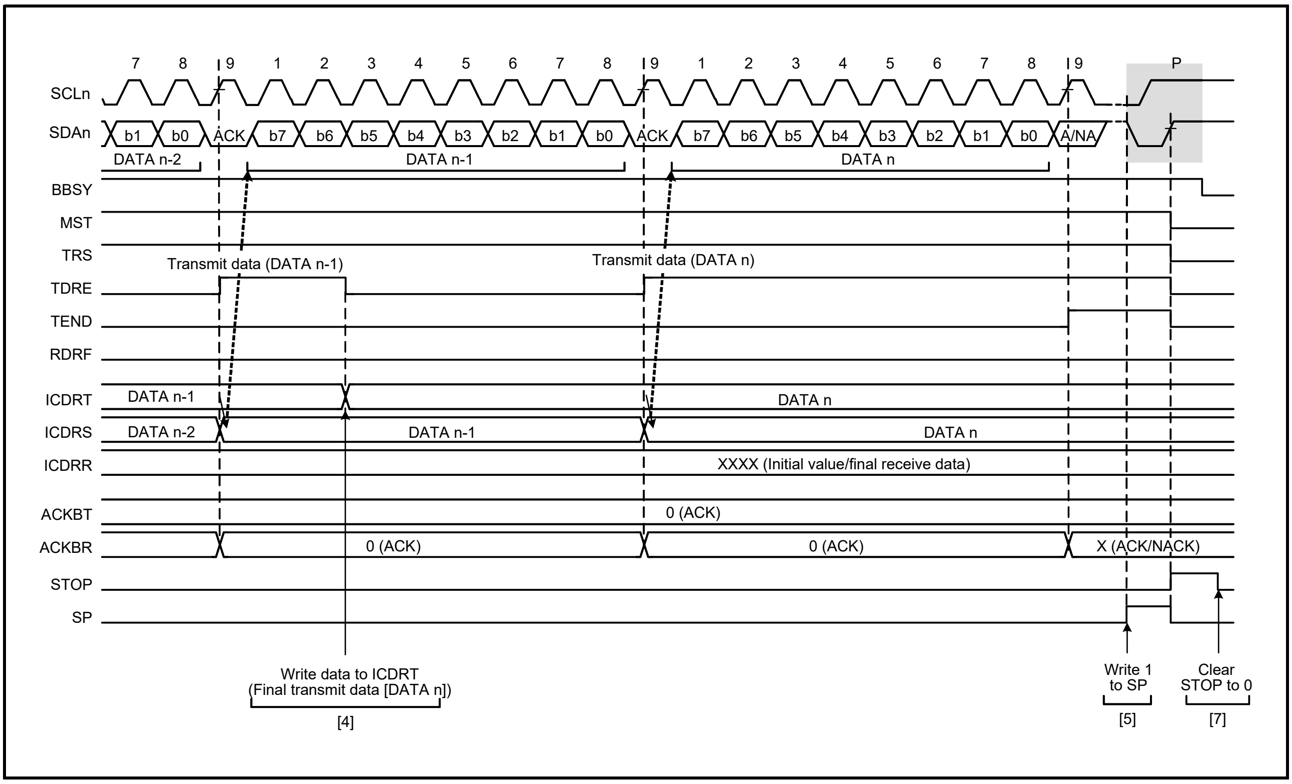

SCLn
SDAn
7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9
P
b1 b0 ACK b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4 b3 b2 b1 b0 A/NA
DATA n-2
DATA n-1
DATA n
BBSY
MST
TRS
TDRE
TEND
RDRF
ICDRT
ICDRS
ICDRR
ACKBT
ACKBR
STOP
SP
Transmit data (DATA n-1)
Transmit data (DATA n)
DATA n-1
DATA n
DATA n-2
DATA n-1
DATA n
XXXX (Initial value/final receive data)
0 (ACK)
0 (ACK)
0 (ACK)
X (ACK/NACK)
Write data to ICDRT
(Final transmit data [DATA n])
[4]
Write 1 to SP
[5]
Clear STOP to 0
[7]

Figure 30.9 Master transmit operation timing (3)

30.3.4 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.

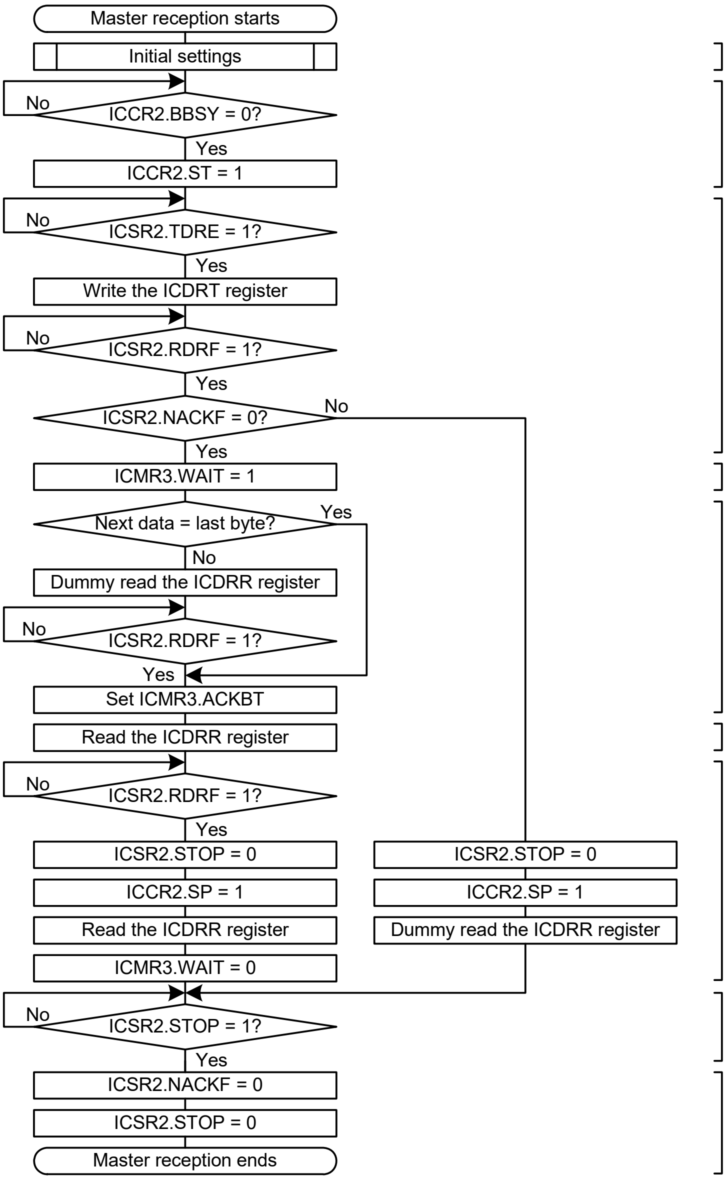
Figure 30.10 and Figure 30.11 show examples of master reception (7-bit address format). Figure 30.12 to Figure 30.14 show the timing of operations in master reception.

To set up and perform master reception:

1. Initialize the IIC using the procedure in section 30.3.2, 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. 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 start condition issue is successful as requested by the ST bit, 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 the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag is automatically set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag is set to 1. When the byte containing the slave address and R/W# bit is transmitted, the value of the ICCR2.TRS bit automatically updates to select transmit or receive mode 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. 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 is to end transfer for data reception after transfer of the next and final byte, set the ICMR3.ACKBT bit to 1 (NACK).
7. After reading the second-to-last byte from the ICDRR register, if the value of the ICSR2.RDRF flag is 1, write 1 to the SP bit in ICCR2 (to request stop condition), and then read the last byte from ICDRR. When ICDRR is read, the IIC is released from the wait state and issues the stop condition after low-level output in the 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.


Master reception starts
Initial settings
No
ICCR2.BBSY = 0?
Yes
ICCR2.ST = 1
No
ICSR2.TDRE = 1?
Yes
Write the ICDRT register
No
ICSR2.RDRF = 1?
Yes
ICSR2.NACKF = 0?
No
Yes
ICMR3.WAIT = 1
Next data = last byte?
Yes
No
Dummy read the ICDRR register
No
ICSR2.RDRF = 1?
Yes
Set ICMR3.ACKBT
Read the ICDRR register
No
ICSR2.RDRF = 1?
Yes
ICSR2.STOP = 0
ICCR2.SP = 1
Read the ICDRR register
ICMR3.WAIT = 0
ICSR2.STOP = 0
ICCR2.SP = 1
Dummy read the ICDRR register
No
ICSR2.STOP = 1?
Yes
ICSR2.NACKF = 0
ICSR2.STOP = 0
Master reception ends

(1) Initial settings.

(2) Check I^{2}C bus occupation and issue a start condition.

(3) Transmit the slave address followed by
R and check ACK.

(4) Set to WAIT.

(5) Set to NACK.
(When receiving 2 bytes, perform dummy
read.)

(6) Read received data.
(When receiving 1 byte, perform
dummy read.)

(7) Read the last data,
release SCL by the ACKBT setting,
and generate a stop condition.

(8) Confirm that the stop condition
has been generated.

(9) Processing for the next transfer operation.

Figure 30.10 Example of master reception flow with 7-bit address format, and 1 or 2 bytes

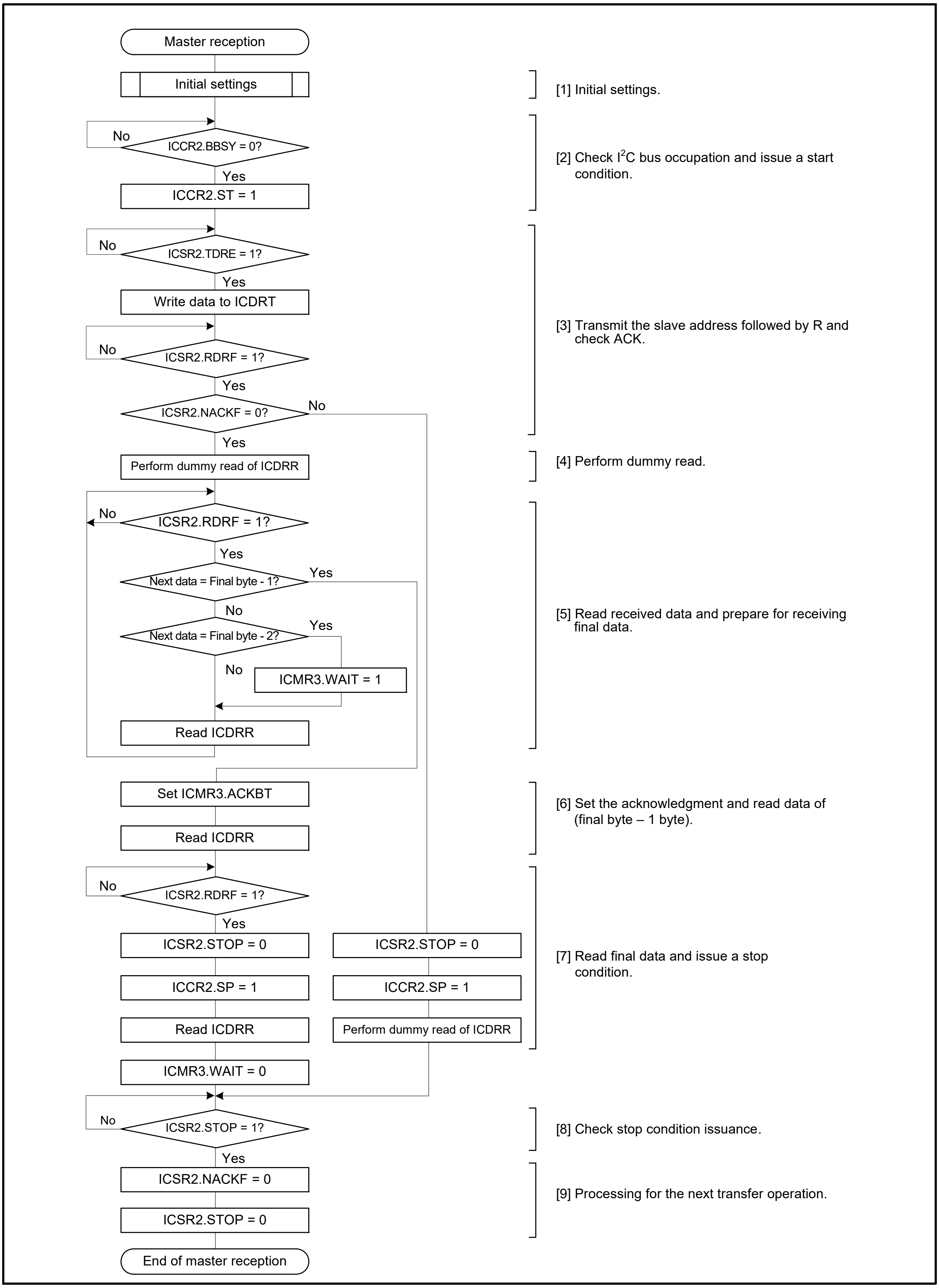

Master reception
Initial settings
No
ICCR2.BBSY = 0?
Yes
ICCR2.ST = 1
No
ICSR2.TDRE = 1?
Yes
Write data to ICDRT
No
ICSR2.RDRF = 1?
Yes
ICSR2.NACKF = 0?
No
Yes
Perform dummy read of ICDRR
No
ICSR2.RDRF = 1?
Yes
Next data = Final byte - 1?
Yes
No
Next data = Final byte - 2?
Yes
No
ICMR3.WAIT = 1
Read ICDRR
Set ICMR3.ACKBT
Read ICDRR
No
ICSR2.RDRF = 1?
Yes
ICSR2.STOP = 0
ICCR2.SP = 1
Read ICDRR
ICMR3.WAIT = 0
ICSR2.STOP = 0
ICCR2.SP = 1
Perform dummy read of ICDRR
No
ICSR2.STOP = 1?
Yes
ICSR2.NACKF = 0
ICSR2.STOP = 0
End of master reception
[1] Initial settings.
[2] Check I2C bus occupation and issue a start condition.
[3] Transmit the slave address followed by R and check ACK.
[4] Perform dummy read.
[5] Read received data and prepare for receiving final data.
[6] Set the acknowledgment and read data of (final byte – 1 byte).
[7] Read final data and issue a stop condition.
[8] Check stop condition issuance.
[9] Processing for the next transfer operation.

Figure 30.11 Example of master reception flow with 7-bit address format, and 3 bytes or more

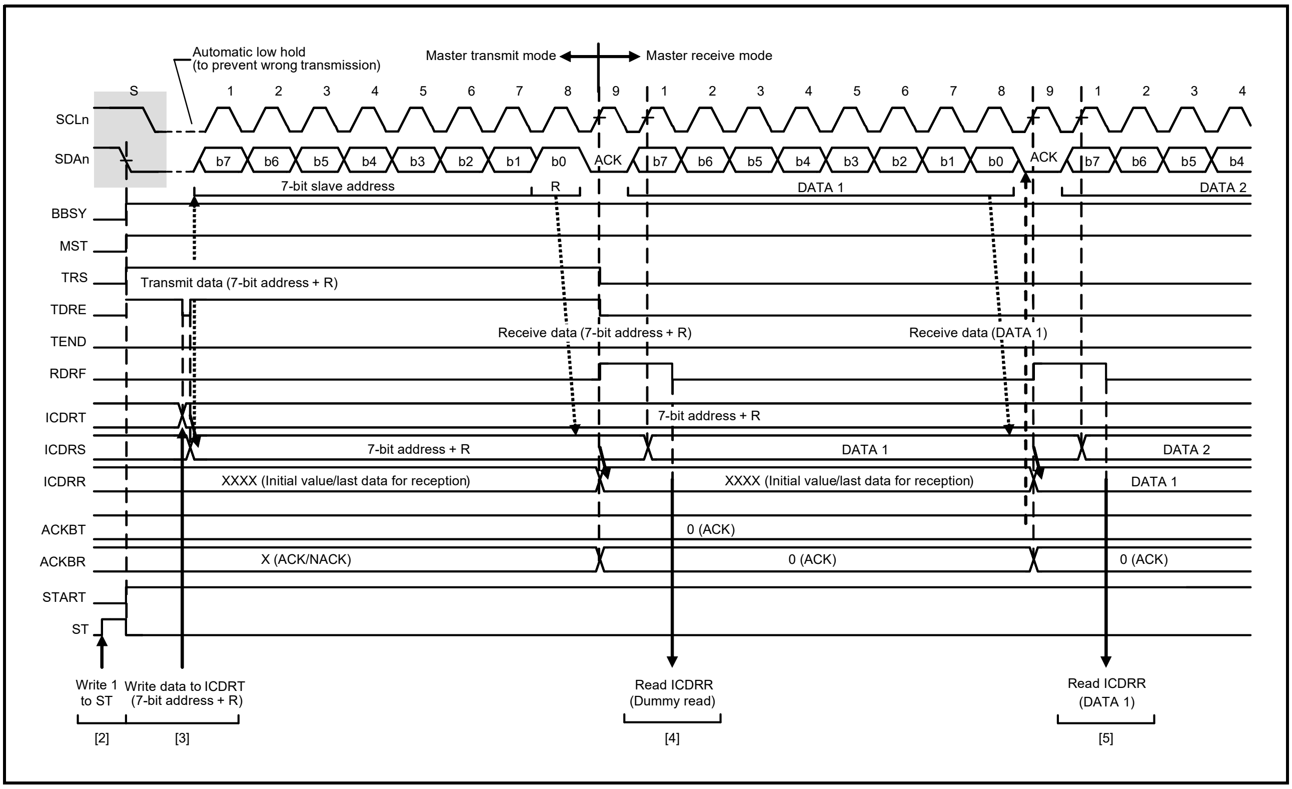

Automatic low hold
(to prevent wrong transmission)
Master transmit mode
Master receive mode
S
1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4
SCLn
SDAn
b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4
7-bit slave address
R
DATA 1
DATA 2
BBSY
MST
TRS
Transmit data (7-bit address + R)
TDRE
TEND
Receive data (7-bit address + R)
Receive data (DATA 1)
RDRF
ICDRT
7-bit address + R
ICDRS
7-bit address + R
DATA 1
DATA 2
ICDRR
XXXX (Initial value/last data for reception)
XXXX (Initial value/last data for reception)
DATA 1
ACKBT
0 (ACK)
ACKBR
X (ACK/NACK)
0 (ACK)
0 (ACK)
START
ST
Write 1 to ST
Write data to ICDRT (7-bit address + R)
Read ICDRR (Dummy read)
Read ICDRR (DATA 1)
[2]
[3]
[4]
[5]

Figure 30.12 Master receive operation timing (1) with 7-bit address format when RDRFS = 0

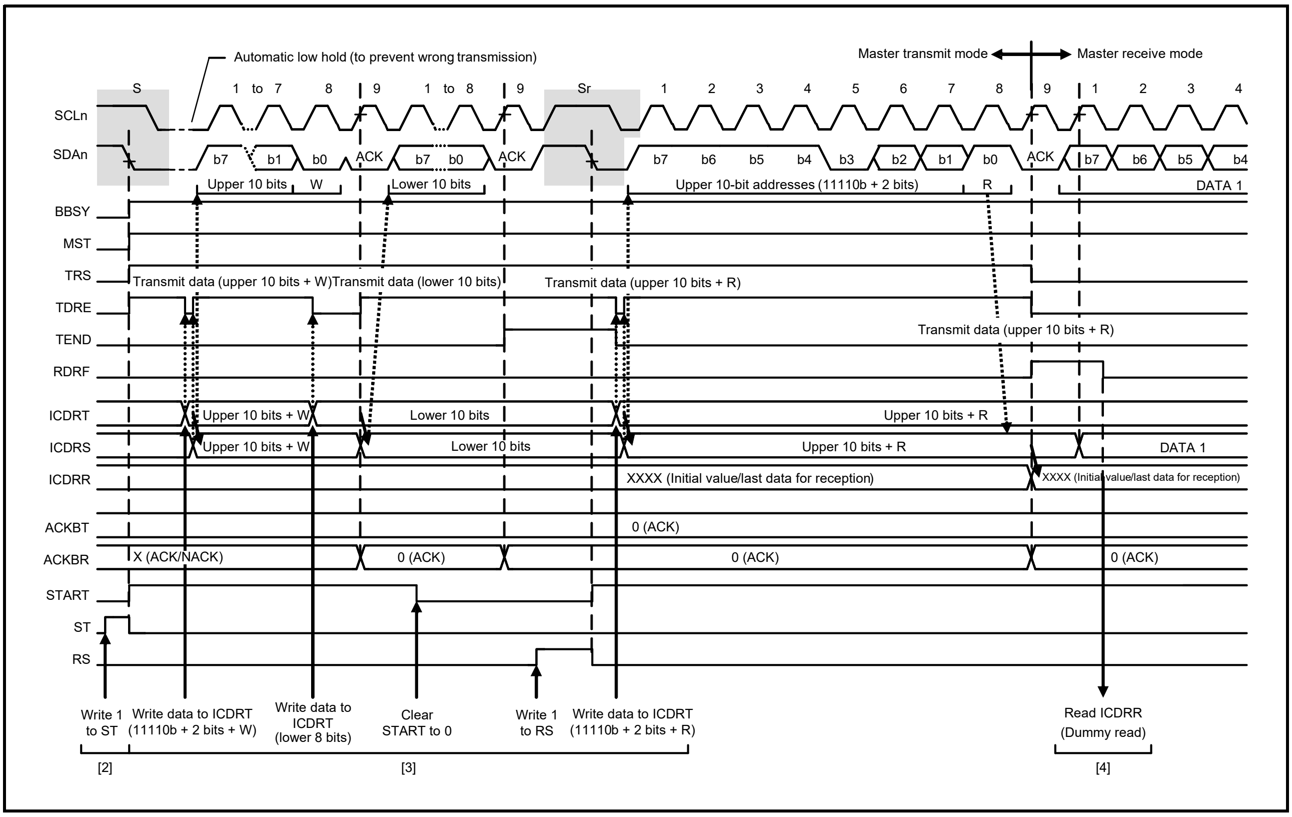

Automatic low hold (to prevent wrong transmission)
Master transmit mode
Master receive mode
S
1 to 7 8 9 1 to 8 9
Sr
1 2 3 4 5 6 7 8 9 1 2 3 4
SCLn
SDAn
b7 b1 b0 ACK b7 b0 ACK
b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4
Upper 10 bits
W
Lower 10 bits
Upper 10-bit addresses (11110b + 2 bits)
R
DATA 1
BBSY
MST
TRS
Transmit data (upper 10 bits + W)
Transmit data (lower 10 bits)
Transmit data (upper 10 bits + R)
TDRE
TEND
Transmit data (upper 10 bits + R)
RDRF
ICDRT
Upper 10 bits + W
Lower 10 bits
Upper 10 bits + R
ICDRS
Upper 10 bits + W
Lower 10 bits
Upper 10 bits + R
DATA 1
ICDRR
XXXX (Initial value/last data for reception)
XXXX (Initial value/last data for reception)
ACKBT
0 (ACK)
ACKBR
X (ACK/NACK)
0 (ACK)
0 (ACK)
0 (ACK)
START
ST
RS
Write 1 to ST
Write data to ICDRT (11110b + 2 bits + W)
Write data to ICDRT (lower 8 bits)
Clear START to 0
Write 1 to RS
Write data to ICDRT (11110b + 2 bits + R)
Read ICDRR (Dummy read)
[2]
[3]
[4]

Figure 30.13 Master receive operation timing (2) with 10-bit address format when RDRFS = 0

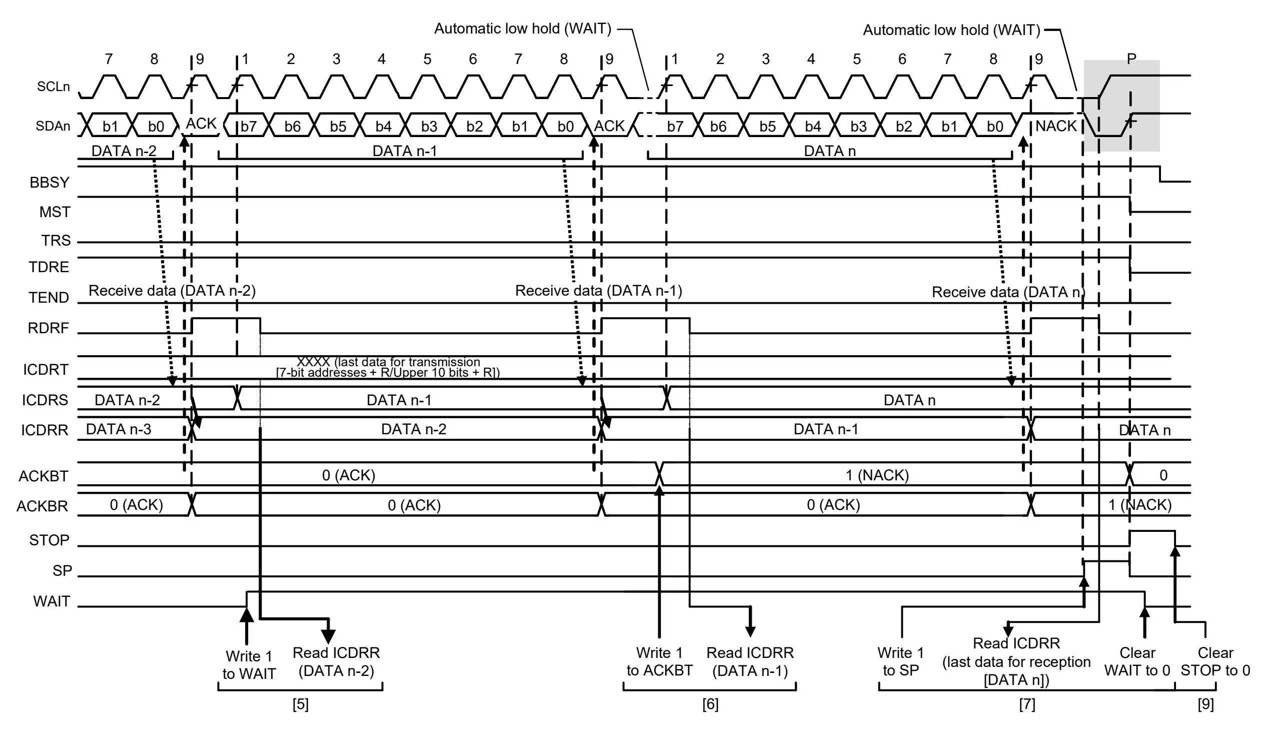

Automatic low hold (WAIT)
Automatic low hold (WAIT)
SCLn
SDAn
BBSY
MST
TRS
TDRE
TEND
RDRF
ICDRT
ICDRS
ICDRR
ACKBT
ACKBR
STOP
SP
WAIT
7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9
P
b1 b0 ACK b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4 b3 b2 b1 b0 NACK
DATA n-2
DATA n-1
DATA n
Receive data (DATA n-2)
Receive data (DATA n-1)
Receive data (DATA n)
XXXX (last data for transmission [7-bit addresses + R/Upper 10 bits + R])
DATA n-2
DATA n-1
DATA n
DATA n-3
DATA n-2
DATA n-1
DATA n
0 (ACK)
1 (NACK)
0
0 (ACK)
0 (ACK)
0 (ACK)
1 (NACK)
Write 1 to WAIT
Read ICDRR (DATA n-2)
[5]
Write 1 to ACKBT
Read ICDRR (DATA n-1)
[6]
Write 1 to SP
Read ICDRR (last data for reception [DATA n])
[7]
Clear WAIT to 0
Clear STOP to 0
[9]

Figure 30.14 Master receive operation timing (3) when RDRFS = 0

30.3.5 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 30.15 shows an example of slave transmission. Figure 30.16 and Figure 30.17 show the operation timing in slave transmission.

To set up and perform slave transmission:

1. To initialize the IIC, follow the procedure in section 30.3.2, 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. 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.

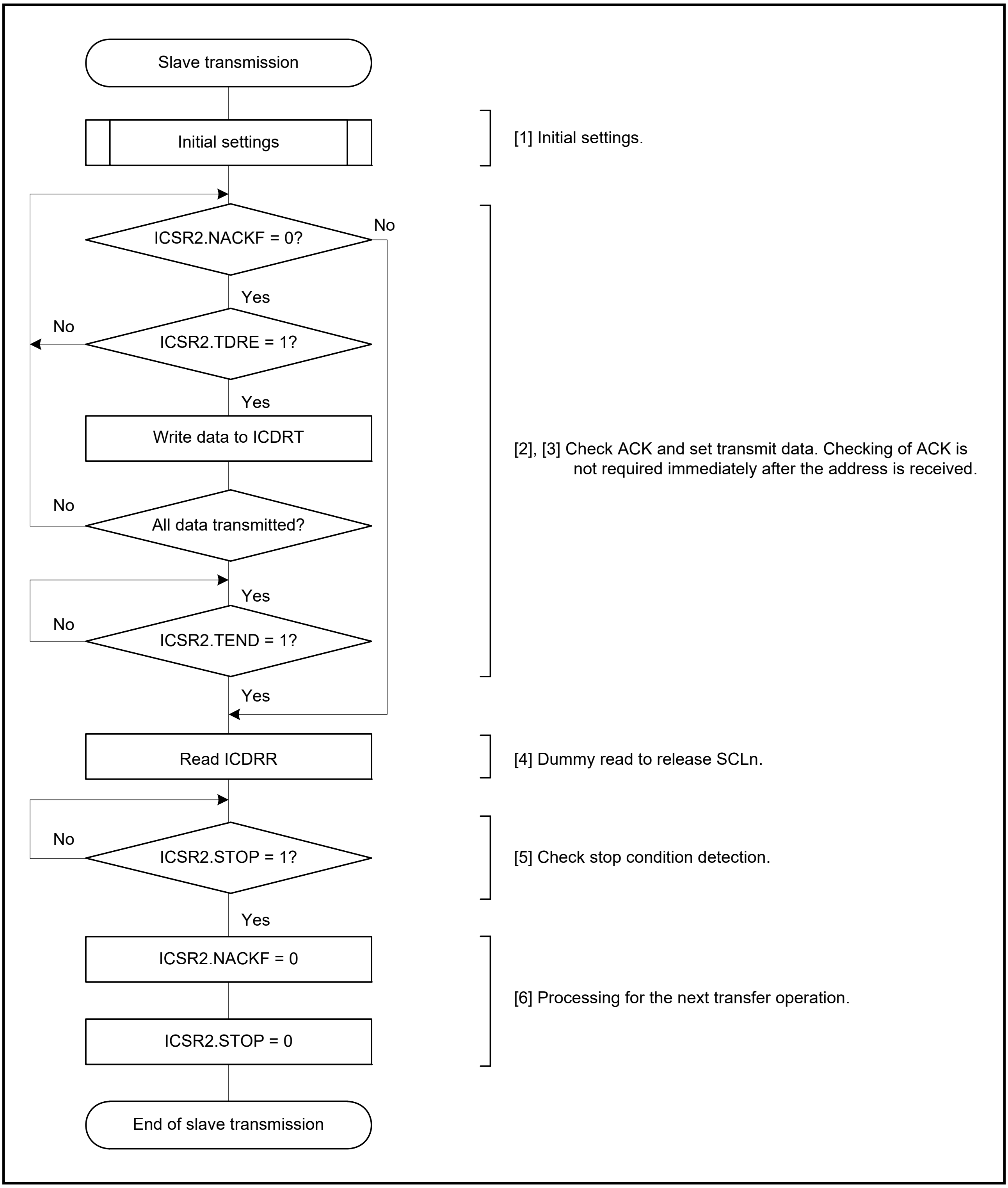

Slave transmission
Initial settings
[1] Initial settings.
ICSR2.NACKF = 0?
No
Yes
No
ICSR2.TDRE = 1?
Yes
Write data to ICDRT
No
All data transmitted?
[2], [3] Check ACK and set transmit data. Checking of ACK is not required immediately after the address is received.
Yes
No
ICSR2.TEND = 1?
Yes
Read ICDRR
[4] Dummy read to release SCLn.
No
ICSR2.STOP = 1?
[5] Check stop condition detection.
Yes
ICSR2.NACKF = 0
ICSR2.STOP = 0
[6] Processing for the next transfer operation.
End of slave transmission

Figure 30.15 Example of slave transmission flow

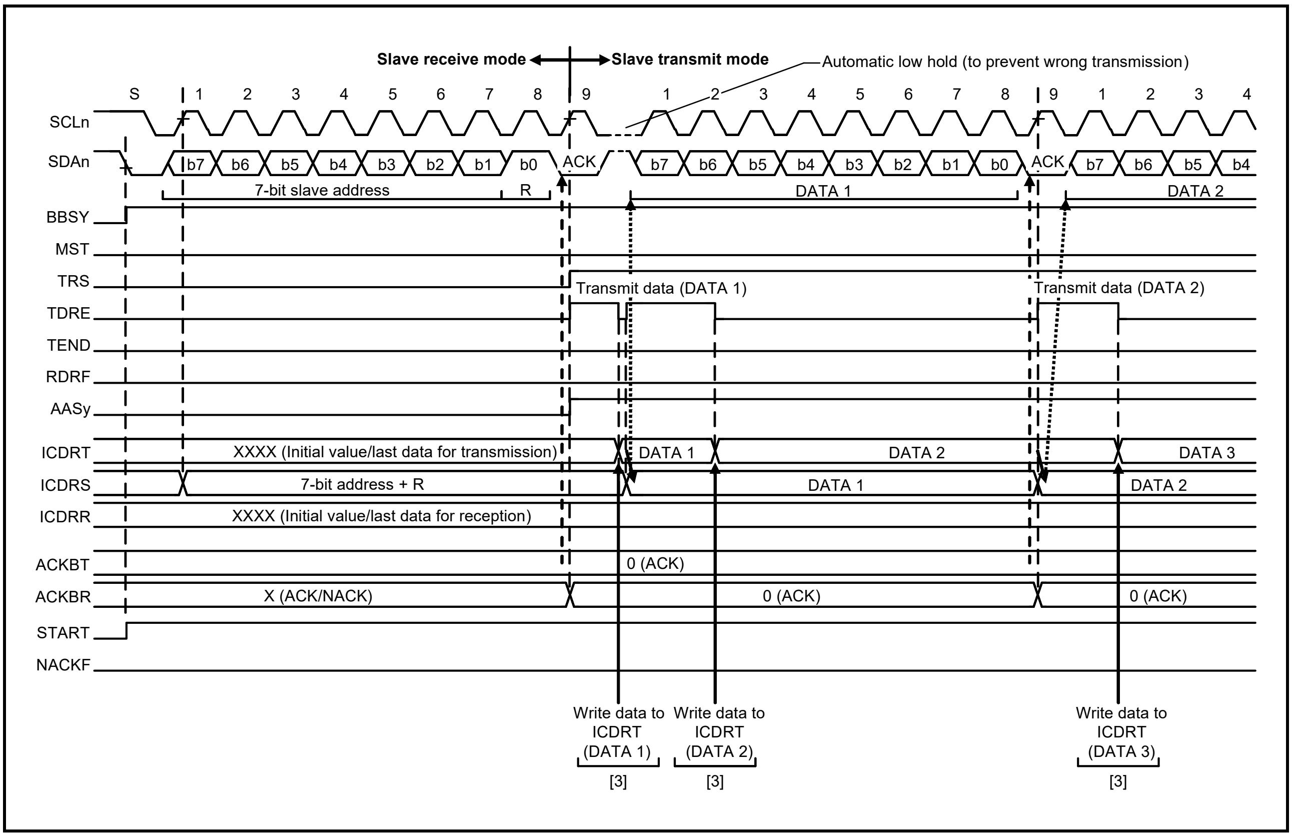

Slave receive mode
Slave transmit mode
Automatic low hold (to prevent wrong transmission)
S 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4
SCLn
SDAn
b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4
7-bit slave address
R
DATA 1
DATA 2
BBSY
MST
TRS
TDRE
TEND
RDRF
AASy
ICDRT
ICDRS
ICDRR
ACKBT
ACKBR
START
NACKF
Transmit data (DATA 1)
Transmit data (DATA 2)
XXXX (Initial value/last data for transmission)
DATA 1
DATA 2
DATA 3
7-bit address + R
DATA 1
DATA 2
XXXX (Initial value/last data for reception)
0 (ACK)
X (ACK/NACK)
0 (ACK)
0 (ACK)
Write data to ICDRT (DATA 1)
Write data to ICDRT (DATA 2)
Write data to ICDRT (DATA 3)
[3]
[3]
[3]

Figure 30.16 Slave transmit operation timing (1) with 7-bit address format

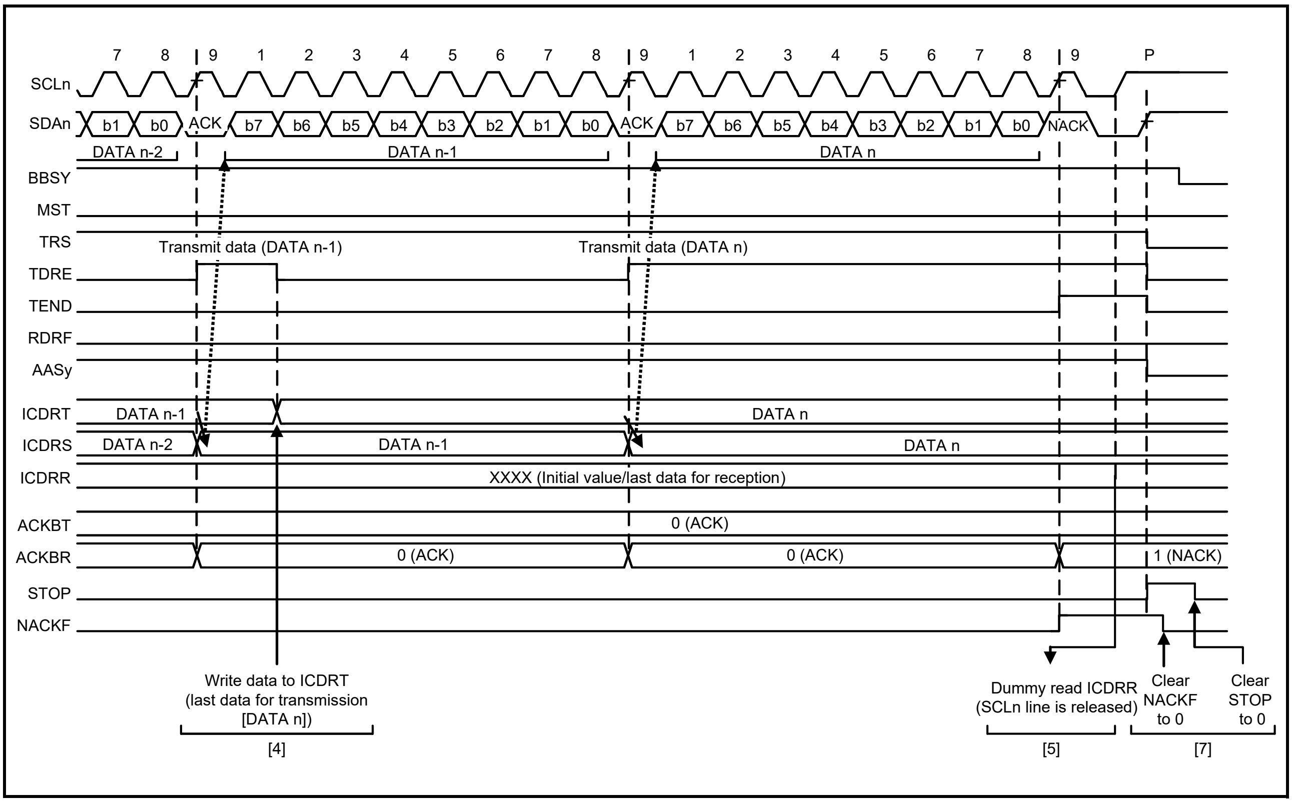

7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 P
SCLn
SDAn
b1 b0 ACK b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4 b3 b2 b1 b0 NACK
DATA n-2
DATA n-1
DATA n
BBSY
MST
TRS
TDRE
TEND
RDRF
AASy
ICDRT
ICDRS
ICDRR
ACKBT
ACKBR
STOP
NACKF
Transmit data (DATA n-1)
Transmit data (DATA n)
DATA n-1
DATA n
DATA n-2
DATA n-1
DATA n
XXXX (Initial value/last data for reception)
0 (ACK)
0 (ACK)
0 (ACK)
1 (NACK)
Write data to ICDRT (last data for transmission [DATA n])
[4]
Dummy read ICDRR (SCLn line is released)
[5]
Clear NACKF to 0
Clear STOP to 0
[7]

Figure 30.17 Slave transmit operation timing (2)

30.3.6 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 30.18 shows an example of slave reception. Figure 30.19 and Figure 30.20 show the timing of operations in slave reception.

To set up and perform slave reception:

1. To initialize the IIC, follow the procedure in section 30.3.2, 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 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 1 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, then set the ICSR2.STOP flag to 0 for the next transfer operation.

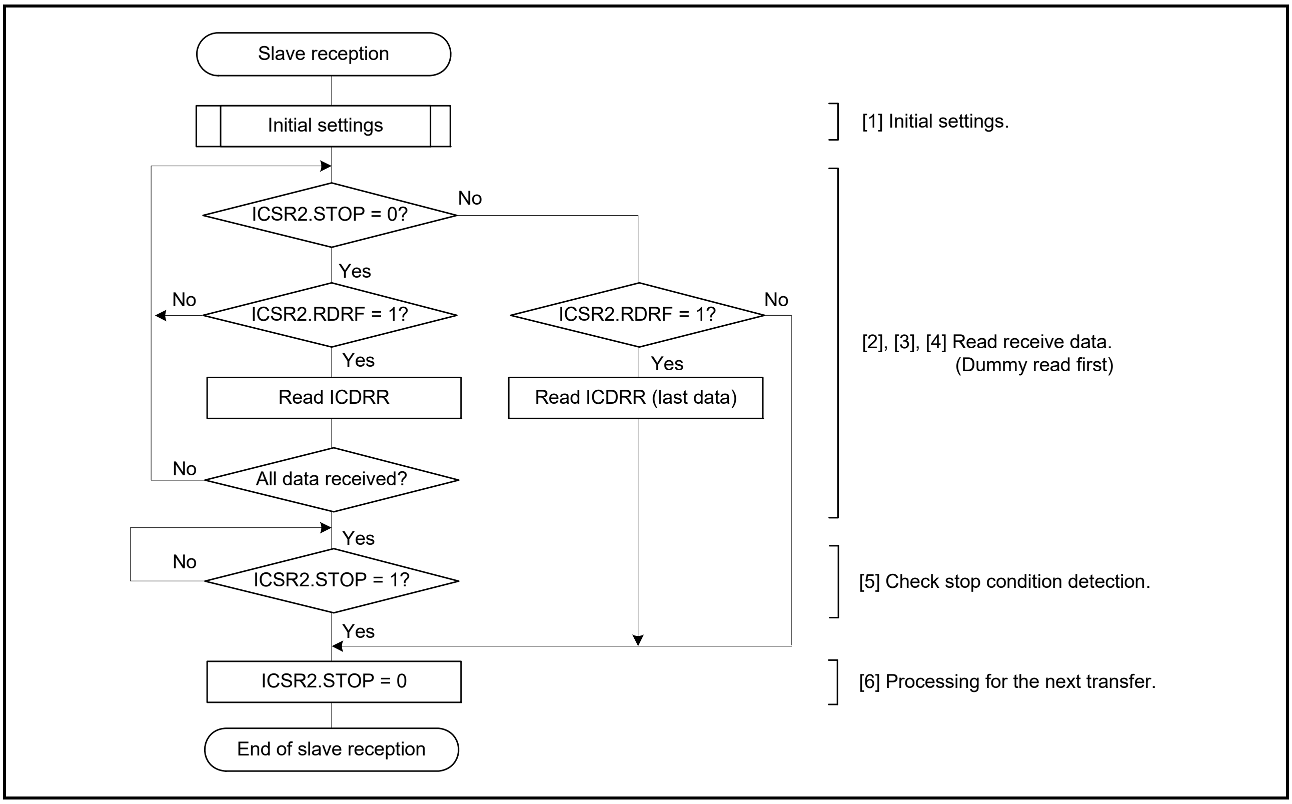

Slave reception
Initial settings
ICSR2.STOP = 0?
No
Yes
ICSR2.RDRF = 1?
No
Yes
Read ICDRR
All data received?
No
Yes
ICSR2.STOP = 1?
No
Yes
ICSR2.RDRF = 1?
No
Yes
Read ICDRR (last data)
ICSR2.STOP = 0
End of slave reception
[1] Initial settings.
[2], [3], [4] Read receive data.
(Dummy read first)
[5] Check stop condition detection.
[6] Processing for the next transfer.

Figure 30.18 Example slave reception flow

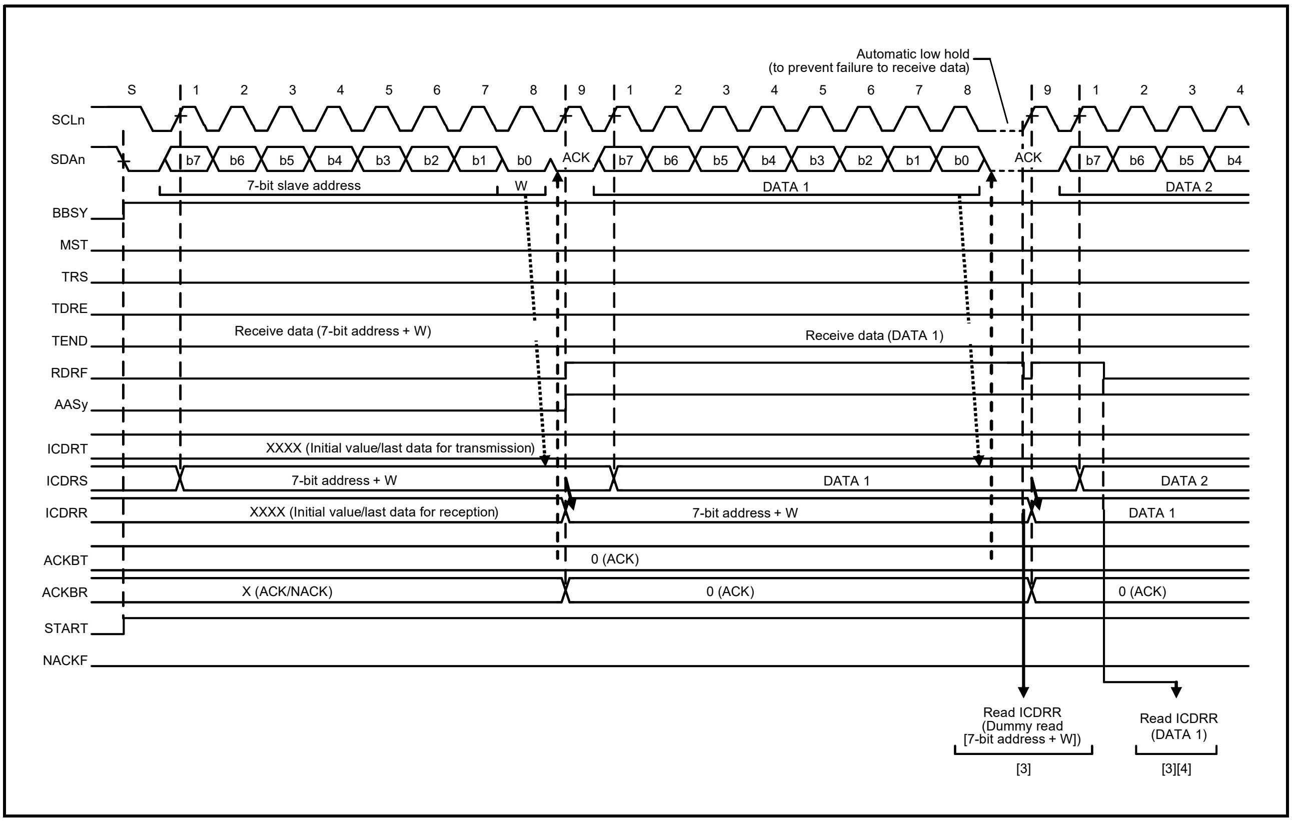

Automatic low hold
(to prevent failure to receive data)
S 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4
SCLn
SDAn
b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4
7-bit slave address
W
DATA 1
DATA 2
BBSY
MST
TRS
TDRE
TEND
Receive data (7-bit address + W)
Receive data (DATA 1)
RDRF
AASy
ICDRT
XXXX (Initial value/last data for transmission)
ICDRS
7-bit address + W
DATA 1
DATA 2
ICDRR
XXXX (Initial value/last data for reception)
7-bit address + W
DATA 1
ACKBT
0 (ACK)
ACKBR
X (ACK/NACK)
0 (ACK)
0 (ACK)
START
NACKF
Read ICDRR
(Dummy read
[7-bit address + W])
[3]
Read ICDRR
(DATA 1)
[3][4]

Figure 30.19 Slave receive operation timing (1) with 7-bit address format when RDRFS = 0

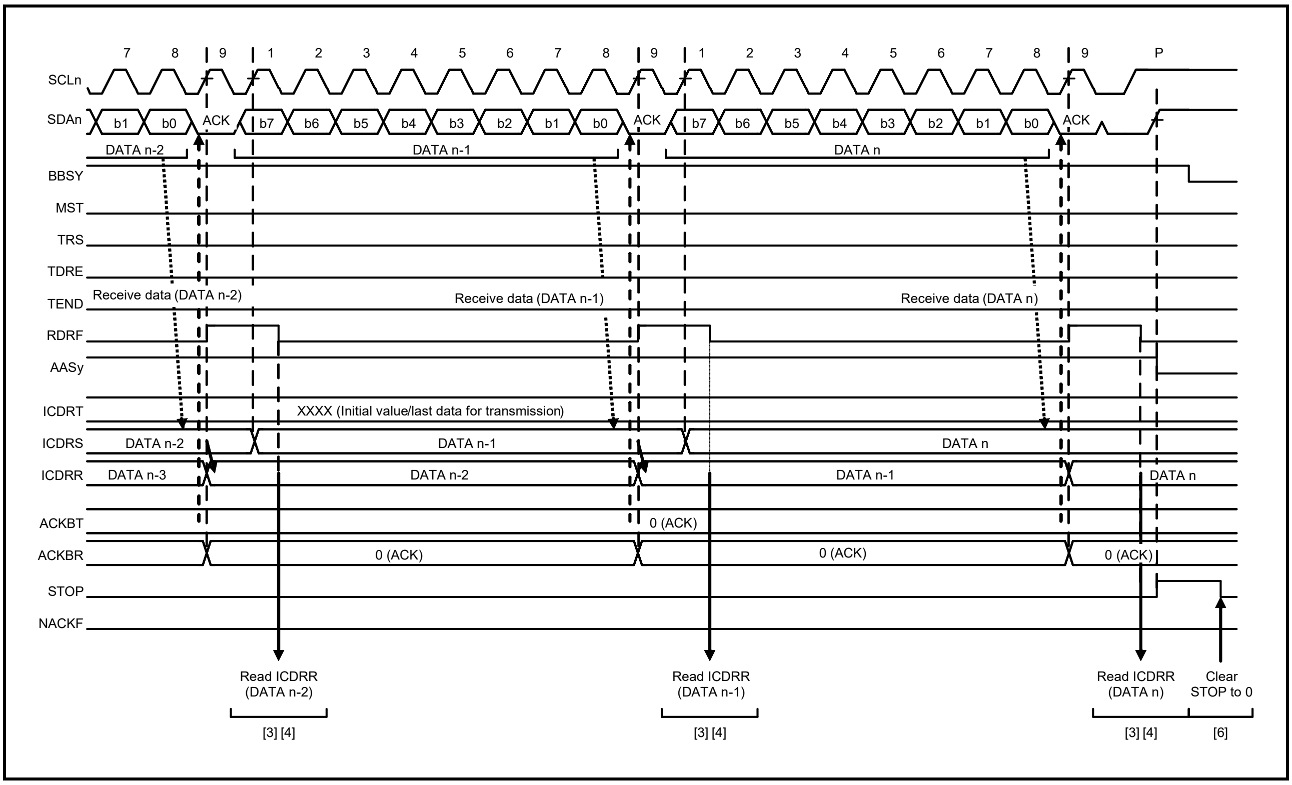

7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 P
SCLn
SDAn
b1 b0 ACK b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4 b3 b2 b1 b0 ACK
DATA n-2
DATA n-1
DATA n
BBSY
MST
TRS
TDRE
TEND
Receive data (DATA n-2)
Receive data (DATA n-1)
Receive data (DATA n)
RDRF
AASy
ICDRT
XXXX (Initial value/last data for transmission)
ICDRS
DATA n-2
DATA n-1
DATA n
ICDRR
DATA n-3
DATA n-2
DATA n-1
DATA n
ACKBT
0 (ACK)
ACKBR
0 (ACK)
0 (ACK)
0 (ACK)
STOP
NACKF
Read ICDRR
(DATA n-2)
[3] [4]
Read ICDRR
(DATA n-1)
[3] [4]
Read ICDRR
(DATA n)
[3] [4]
Clear
STOP to 0
[6]

Figure 30.20 Slave receive operation timing (2) when RDRFS = 0

30.4 SCL Synchronization Circuit

To generate 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 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 includes 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 performs the following:

1. Stops counting when it detects the falling edge.
2. Drives the level on the SCLn line low.
3. Starts counting the low-level period specified in ICBRL.

When the IIC finishes counting the low-level period, it stops driving the SCLn line low. 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. If the low-level period for the other master device ends, the SCL signal rises because the SCLn line is released.

When the IIC finishes outputting the low-level period of the SCL clock, the SCLn line is released and the SCL clock rises. That is, when SCL signals from more than one master are contending, the high-level period of the SCL signal is synchronized with that of the clock with the narrower period, and the low-level period of the SCL signal is synchronized with that of the clock with the broader period. However, such synchronization of the SCL signal is only enabled when the SCLE bit in ICFER is set to 1.

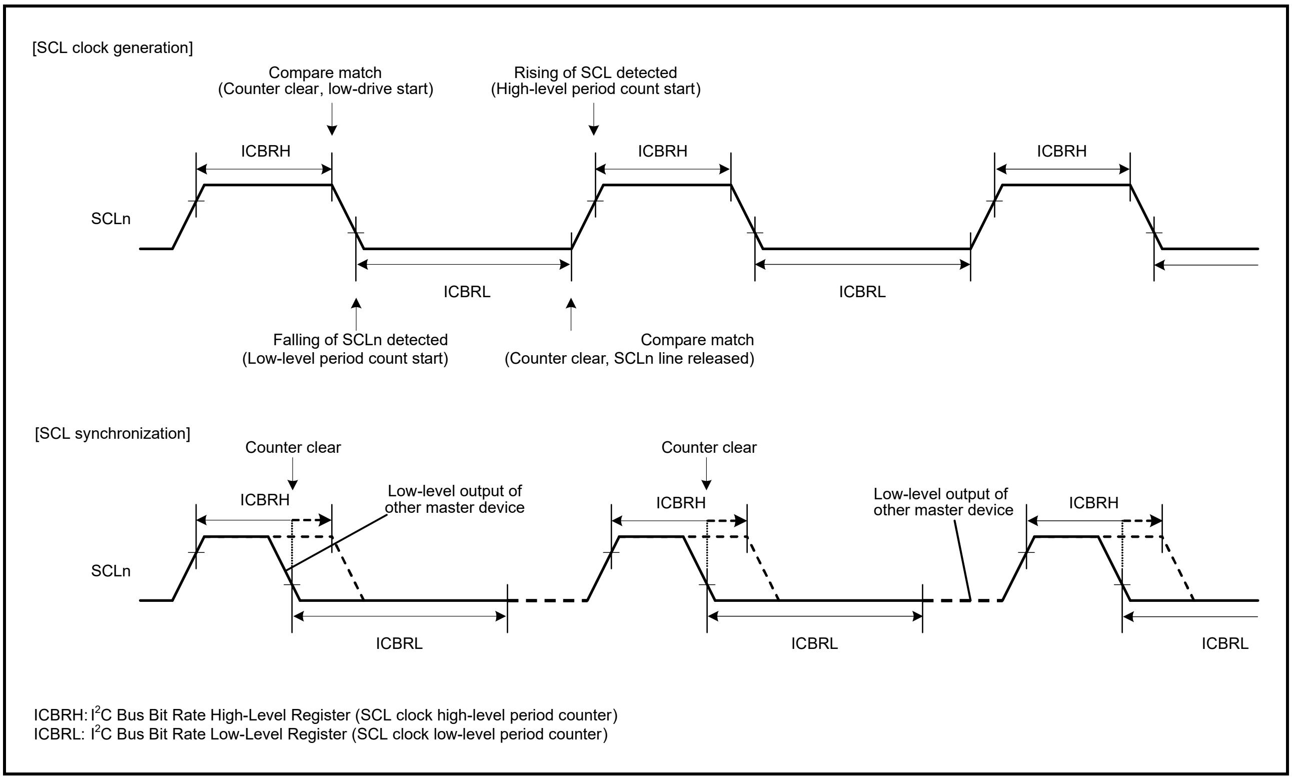

[SCL clock generation]
Compare match
(Counter clear, low-drive start)
Rising of SCL detected
(High-level period count start)
ICBRH
ICBRH
ICBRH
SCLn
ICBRL
ICBRL
Falling of SCLn detected
(Low-level period count start)
Compare match
(Counter clear, SCLn line released)
[SCL synchronization]
Counter clear
Counter clear
ICBRH
Low-level output of
other master device
ICBRH
Low-level output of
other master device
ICBRH
SCLn
ICBRL
ICBRL
ICBRL
ICBRH: I2C Bus Bit Rate High-Level Register (SCL clock high-level period counter)
ICBRL: I2C Bus Bit Rate Low-Level Register (SCL clock low-level period counter)

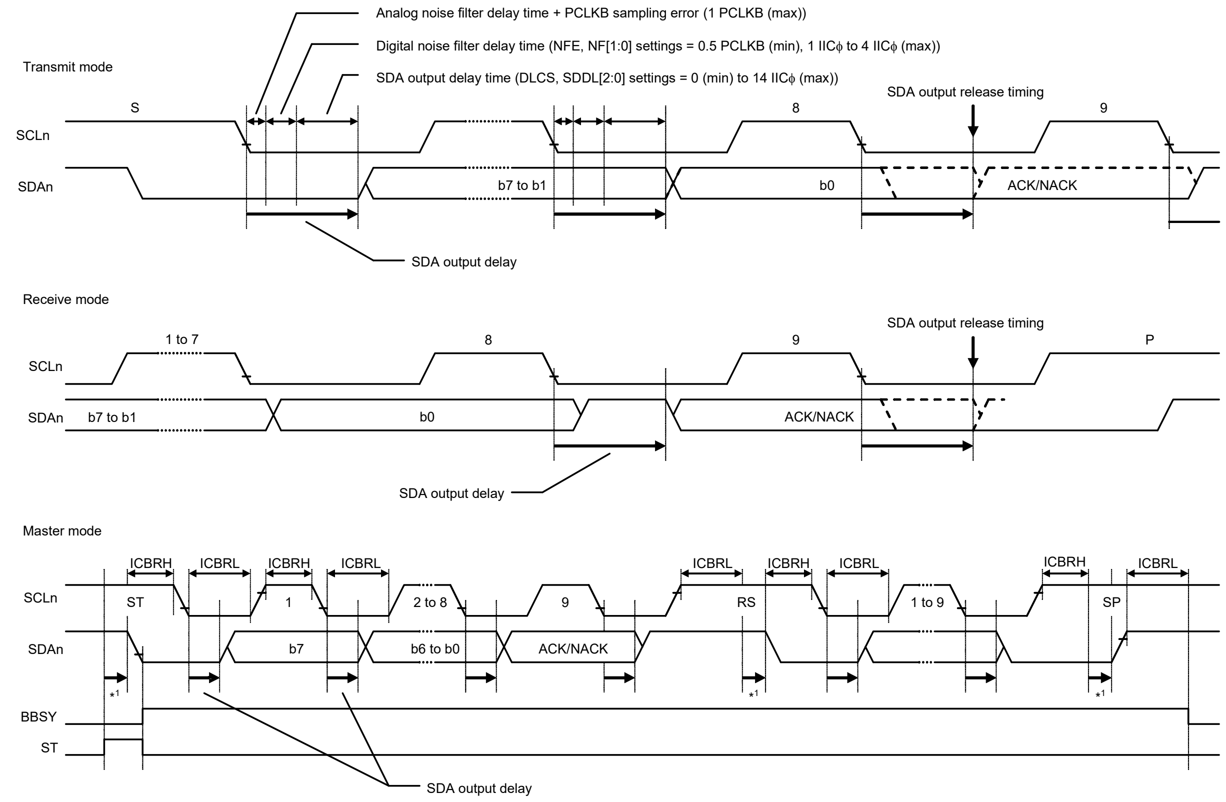
Figure 30.21 Generation and synchronization of SCL signal from IIC

30.5 SDA Output Delay Function

The IIC module provides 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 communication 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, the DLCS bit in ICMR2 selects the clock source for the SDA output delay counter, either as the internal base clock (IICϕ) for the IIC module or as the internal base clock divided by 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.


Analog noise filter delay time + PCLKB sampling error (1 PCLKB (max))
Digital noise filter delay time (NFE, NF[1:0] settings = 0.5 PCLKB (min), 1 IICϕ to 4 IICϕ (max))
SDA output delay time (DLCS, SDDL[2:0] settings = 0 (min) to 14 IICϕ (max))
Transmit mode
S
8
9
SDA output release timing
SCLn
SDAn
b7 to b1
b0
ACK/NACK
SDA output delay
Receive mode
SDA output release timing
1 to 7
8
9
P
SCLn
SDAn
b7 to b1
b0
ACK/NACK
SDA output delay
Master mode
ICBRH
ICBRL
ICBRH
ICBRL
ICBRL
ICBRH
ICBRL
ICBRH
ICBRL
SCLn
ST
1
2 to 8
9
RS
1 to 9
SP
SDAn
b7
b6 to b0
ACK/NACK
*1
*1
*1
BBSY
ST
SDA output delay

Note 1. The output delay function is set by the DLCS and SDDL[2:0] bits when a start (ST), restart (RS), or stop (SP) condition is issued.

Figure 30.22 SDA output delay function

30.6 Digital Noise Filter Circuits

The internal circuitry sees the states of the SCLn and SDAn pins through analog and digital noise-filter circuits. Figure 30.23 shows a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the IIC consists of four flip-flop circuit stages connected in series, and a match-detection circuit.

The number of valid stages in the digital noise filter is selected in the NF[1:0] bits in ICMR3. The selected number of valid stages determines the noise-filtering capability as a period from 1 to 4 IICϕ cycles.

The input signal to the SCLn pin (or SDAn pin) is sampled on falling edges of the IICϕ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected in the NF[1:0] bits in ICMR3, the signal level is seen in the subsequent stage. If the signal levels do not match, the previous value is saved.

If the ratio between the frequency of the internal operating clock (PCLKB) and the transfer rate is small, for example, for data transfer at 400 kbps with PCLKB at 4 MHz, the digital noise filter might lead to the elimination of the required signals as noise. In such cases, it is possible to disable the digital noise-filter circuit by setting the ICFER.NFE bit to 0, and use only the analog noise filter circuit.

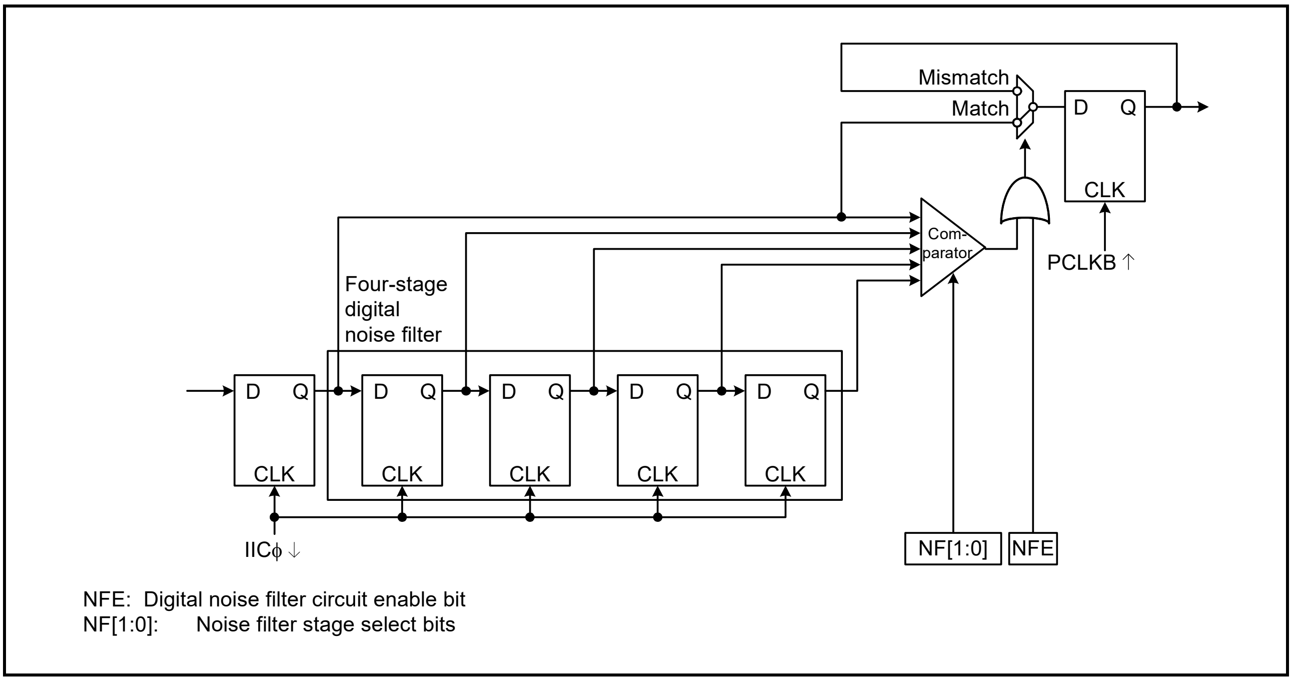

Mismatch
Match
D Q
CLK
PCLKB ↑
Com-
parator
Four-stage
digital
noise filter
D Q
CLK
D Q
CLK
D Q
CLK
D Q
CLK
D Q
CLK
IICϕ ↓
NF[1:0]
NFE
NFE: Digital noise filter circuit enable bit
NF[1:0]: Noise filter stage select bits

Figure 30.23 Digital noise filter circuit block diagram

30.7 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.

30.7.1 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 30.24 to Figure 30.26 show the AASy flag set timing in three cases.

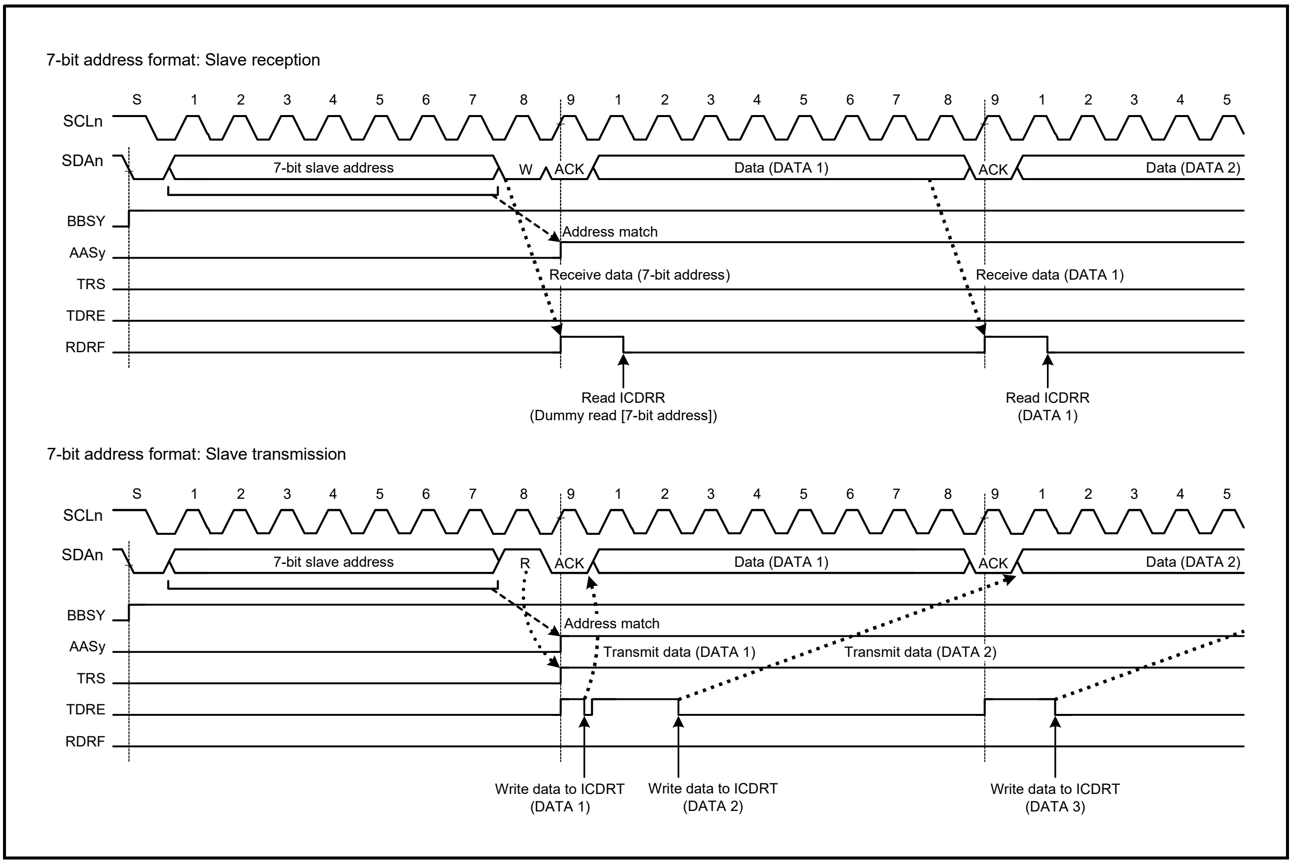

7-bit address format: Slave reception
S 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5
SCLn
SDAn
7-bit slave address
W
ACK
Data (DATA 1)
ACK
Data (DATA 2)
BBSY
AASy
TRS
TDRE
RDRF
Address match
Receive data (7-bit address)
Receive data (DATA 1)
Read ICDRR
(Dummy read [7-bit address])
Read ICDRR
(DATA 1)
7-bit address format: Slave transmission
S 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5
SCLn
SDAn
7-bit slave address
R
ACK
Data (DATA 1)
ACK
Data (DATA 2)
BBSY
AASy
TRS
TDRE
RDRF
Address match
Transmit data (DATA 1)
Transmit data (DATA 2)
Write data to ICDRT
(DATA 1)
Write data to ICDRT
(DATA 2)
Write data to ICDRT
(DATA 3)

Figure 30.24 AASy flag set timing with 7-bit address format selected

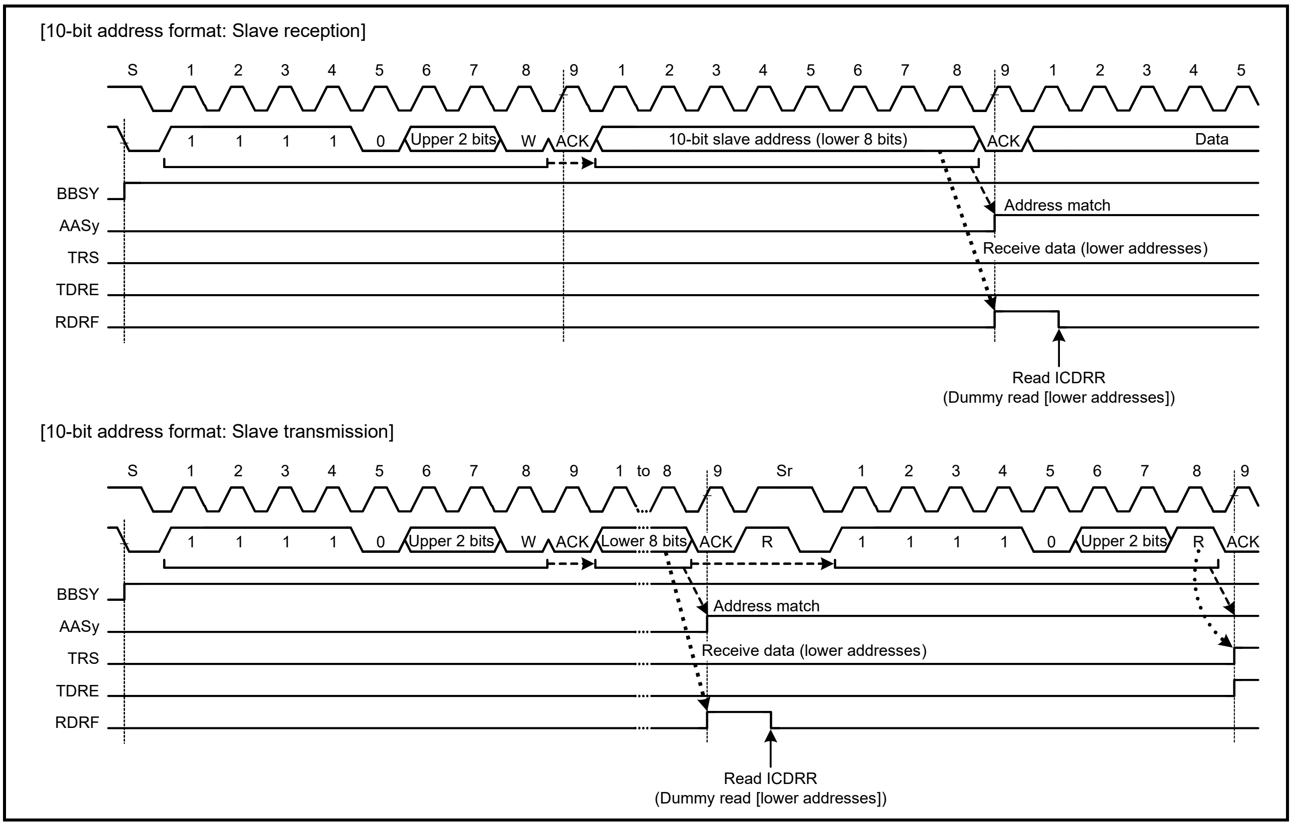

[10-bit address format: Slave reception]
S 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5
1 1 1 1 0
Upper 2 bits
W
ACK
10-bit slave address (lower 8 bits)
ACK
Data
BBSY
AASy
TRS
TDRE
RDRF
Address match
Receive data (lower addresses)
Read ICDRR
(Dummy read [lower addresses])
[10-bit address format: Slave transmission]
S 1 2 3 4 5 6 7 8 9 1 to 8 9 Sr 1 2 3 4 5 6 7 8 9
1 1 1 1 0
Upper 2 bits
W
ACK
Lower 8 bits
ACK
R
1 1 1 1 0
Upper 2 bits
R
ACK
BBSY
AASy
TRS
TDRE
RDRF
Address match
Receive data (lower addresses)
Read ICDRR
(Dummy read [lower addresses])

Figure 30.25 AASy flag set timing with 10-bit address format selected

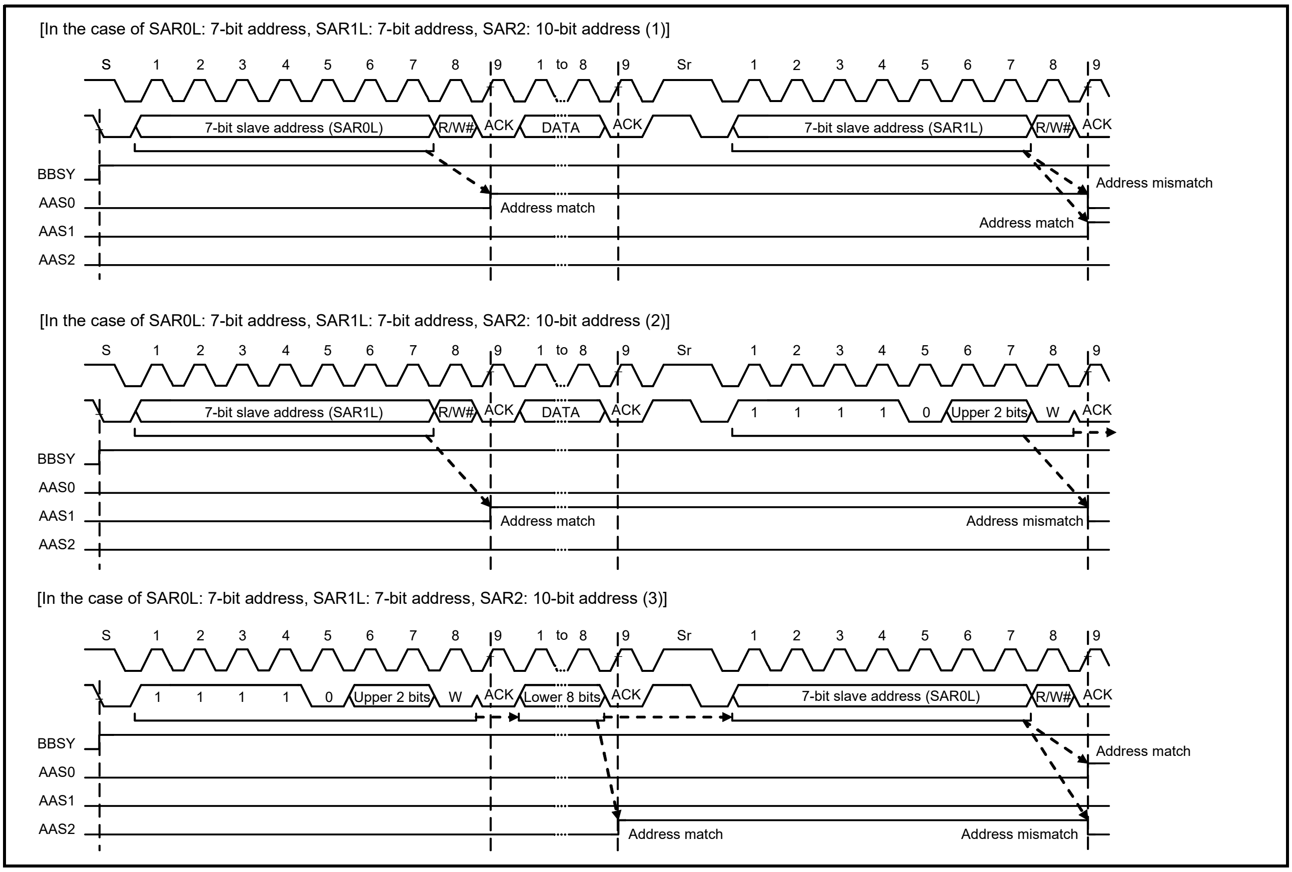

[In the case of SAR0L: 7-bit address, SAR1L: 7-bit address, SAR2: 10-bit address (1)]
S 1 2 3 4 5 6 7 8 9 1 to 8 9 Sr 1 2 3 4 5 6 7 8 9
7-bit slave address (SAR0L)
R/W#
ACK
DATA
ACK
7-bit slave address (SAR1L)
R/W#
ACK
BBSY
AAS0
AAS1
AAS2
Address match
Address mismatch
Address match
[In the case of SAR0L: 7-bit address, SAR1L: 7-bit address, SAR2: 10-bit address (2)]
S 1 2 3 4 5 6 7 8 9 1 to 8 9 Sr 1 2 3 4 5 6 7 8 9
7-bit slave address (SAR1L)
R/W#
ACK
DATA
ACK
1 1 1 1 0
Upper 2 bits
W
ACK
BBSY
AAS0
AAS1
AAS2
Address match
Address mismatch
[In the case of SAR0L: 7-bit address, SAR1L: 7-bit address, SAR2: 10-bit address (3)]
S 1 2 3 4 5 6 7 8 9 1 to 8 9 Sr 1 2 3 4 5 6 7 8 9
1 1 1 1 0
Upper 2 bits
W
ACK
Lower 8 bits
ACK
7-bit slave address (SAR0L)
R/W#
ACK
BBSY
AAS0
AAS1
AAS2
Address match
Address match
Address mismatch

Figure 30.26 AASy flag set and clear timing with 7-bit and 10-bit address formats mixed

30.7.2 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.

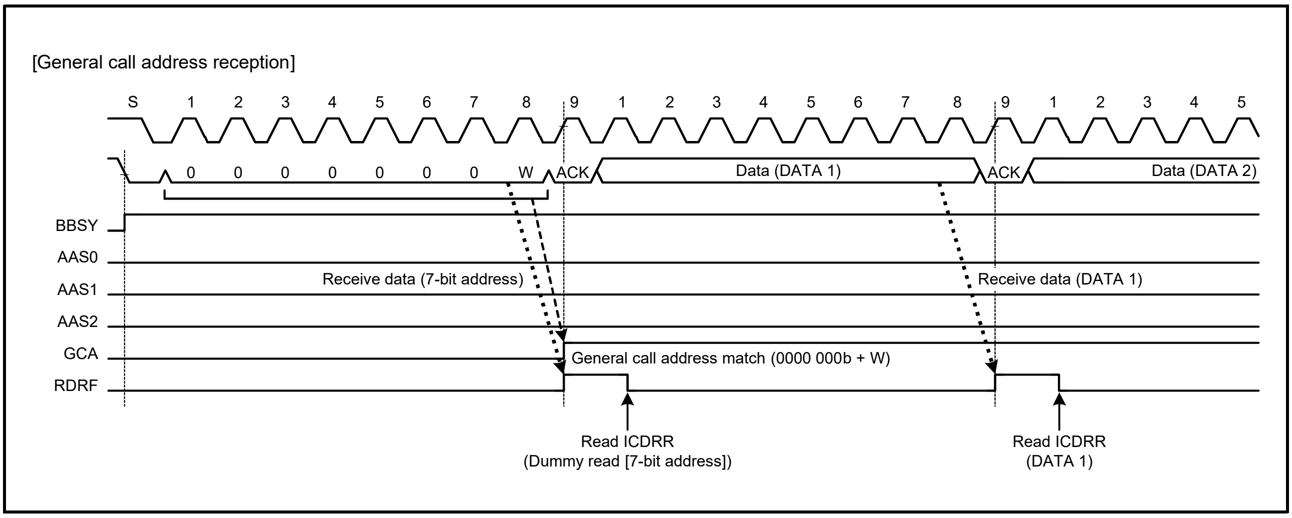

[General call address reception]
S 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5
0 0 0 0 0 0 0 W ACK
Data (DATA 1)
ACK
Data (DATA 2)
BBSY
AAS0
AAS1
AAS2
GCA
RDRF
Receive data (7-bit address)
Receive data (DATA 1)
General call address match (0000 000b + W)
Read ICDRR
(Dummy read [7-bit address])
Read ICDRR
(DATA 1)

Figure 30.27 Timing of GCA flag setting during reception of general call address

30.7.3 Device ID Address Detection

The IIC module provides detection of device ID address in compliance with the I^{2}C bus specification (Rev. 03). When the IIC receives 1111 100b as the first byte after a start condition 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, and 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.

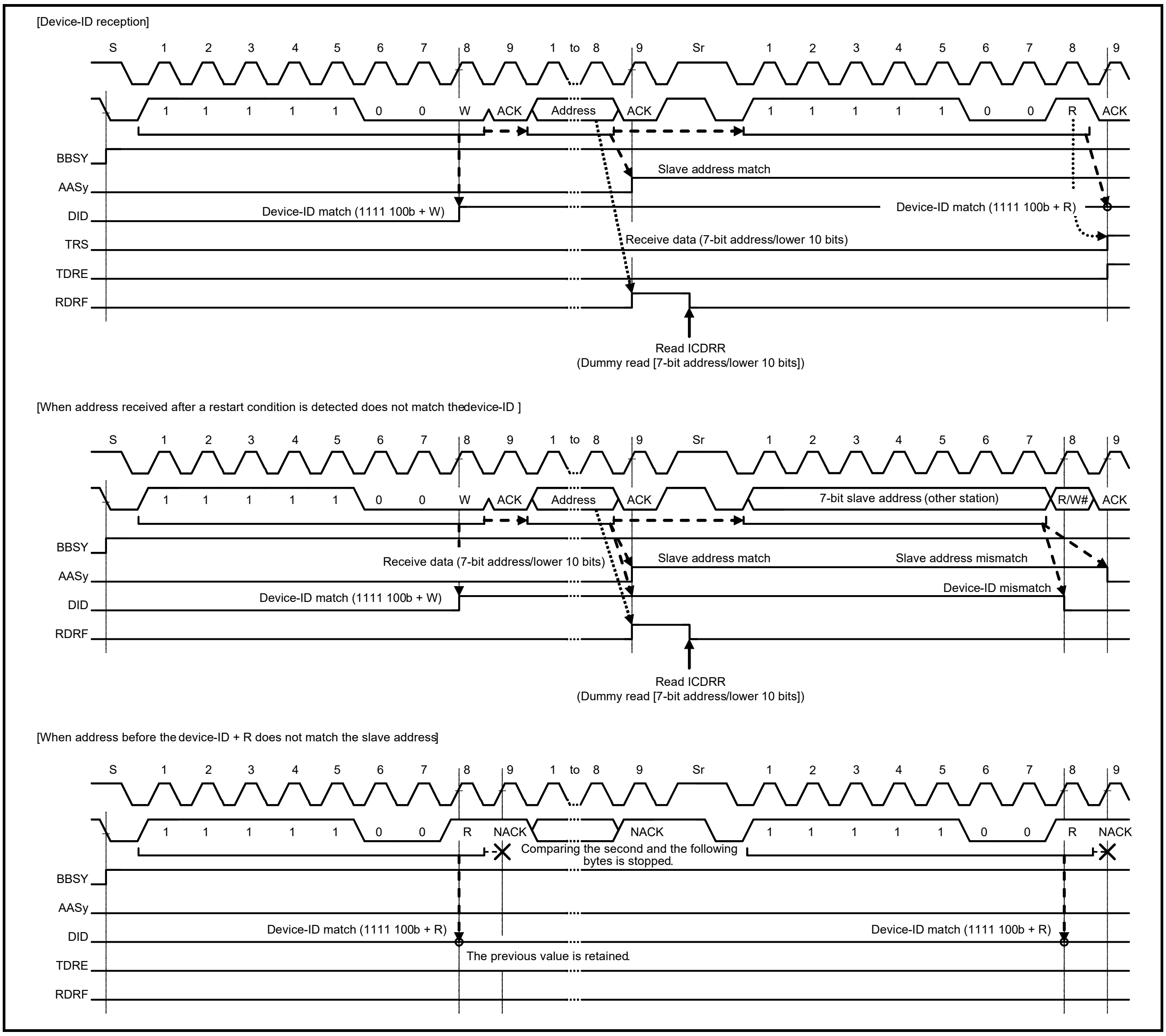

[Device-ID reception]
S 1 2 3 4 5 6 7 8 9 1 to 8 9 Sr 1 2 3 4 5 6 7 8 9
1 1 1 1 1 0 0 W ACK Address ACK 1 1 1 1 1 0 0 R ACK
BBSY
AASy
DID
TRS
TDRE
RDRF
Slave address match
Device-ID match (1111 100b + W)
Device-ID match (1111 100b + R)
Receive data (7-bit address/lower 10 bits)
Read ICDRR
(Dummy read [7-bit address/lower 10 bits])
[When address received after a restart condition is detected does not match the device-ID]
S 1 2 3 4 5 6 7 8 9 1 to 8 9 Sr 1 2 3 4 5 6 7 8 9
1 1 1 1 1 0 0 W ACK Address ACK 7-bit slave address (other station) R/W# ACK
BBSY
AASy
DID
RDRF
Receive data (7-bit address/lower 10 bits)
Slave address match
Slave address mismatch
Device-ID match (1111 100b + W)
Device-ID mismatch
Read ICDRR
(Dummy read [7-bit address/lower 10 bits])
[When address before the device-ID + R does not match the slave address]
S 1 2 3 4 5 6 7 8 9 1 to 8 9 Sr 1 2 3 4 5 6 7 8 9
1 1 1 1 1 0 0 R NACK NACK 1 1 1 1 1 0 0 R NACK
Comparing the second and the following bytes is stopped.
BBSY
AASy
DID
TDRE
RDRF
Device-ID match (1111 100b + R)
Device-ID match (1111 100b + R)
The previous value is retained.

Figure 30.28 AASy/DID flag set/clear timing during reception of device ID

30.7.4 Host Address Detection

The IIC provides host address detection when operating in SMBus mode. When the HOAE bit in ICSER is set to 1 while the SMBS bit in ICMR3 is 1, the IIC can detect the host address (0001 000b) in slave receive mode (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.

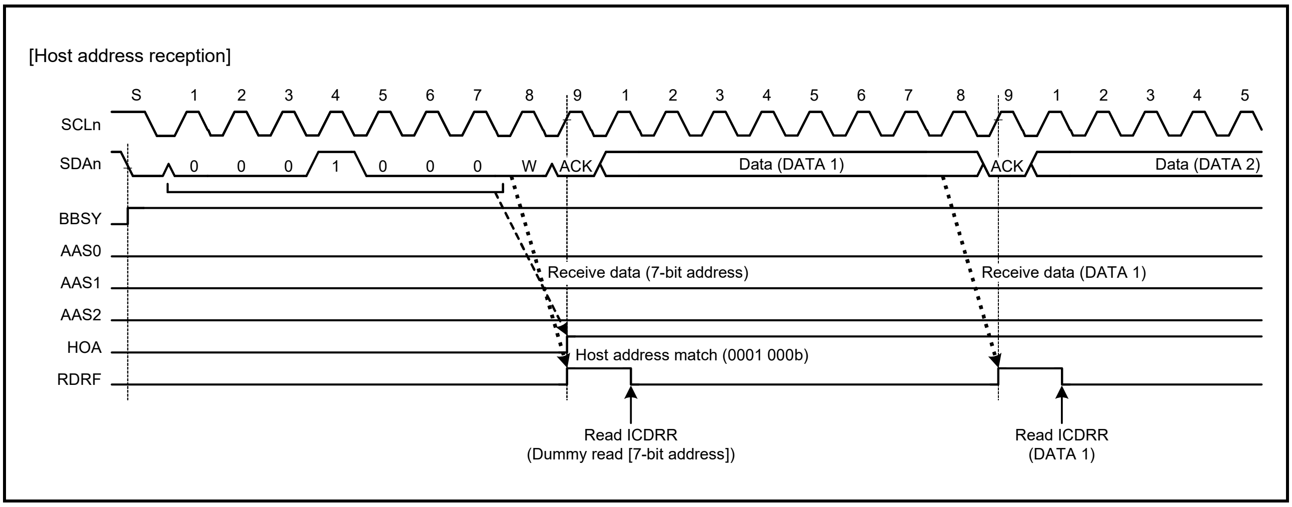

[Host address reception]
S 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5
SCLn
SDAn
0 0 0 1 0 0 0 W ACK
Data (DATA 1)
ACK
Data (DATA 2)
BBSY
AAS0
AAS1
AAS2
HOA
RDRF
Receive data (7-bit address)
Receive data (DATA 1)
Host address match (0001 000b)
Read ICDRR
(Dummy read [7-bit address])
Read ICDRR
(DATA 1)

Figure 30.29 HOA flag set timing during reception of host address

30.8 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 30.9 describes the behavior in these modes.

Table 30.9 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 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 Software Standby mode
- Do not transition to 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.

30.8.1 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:

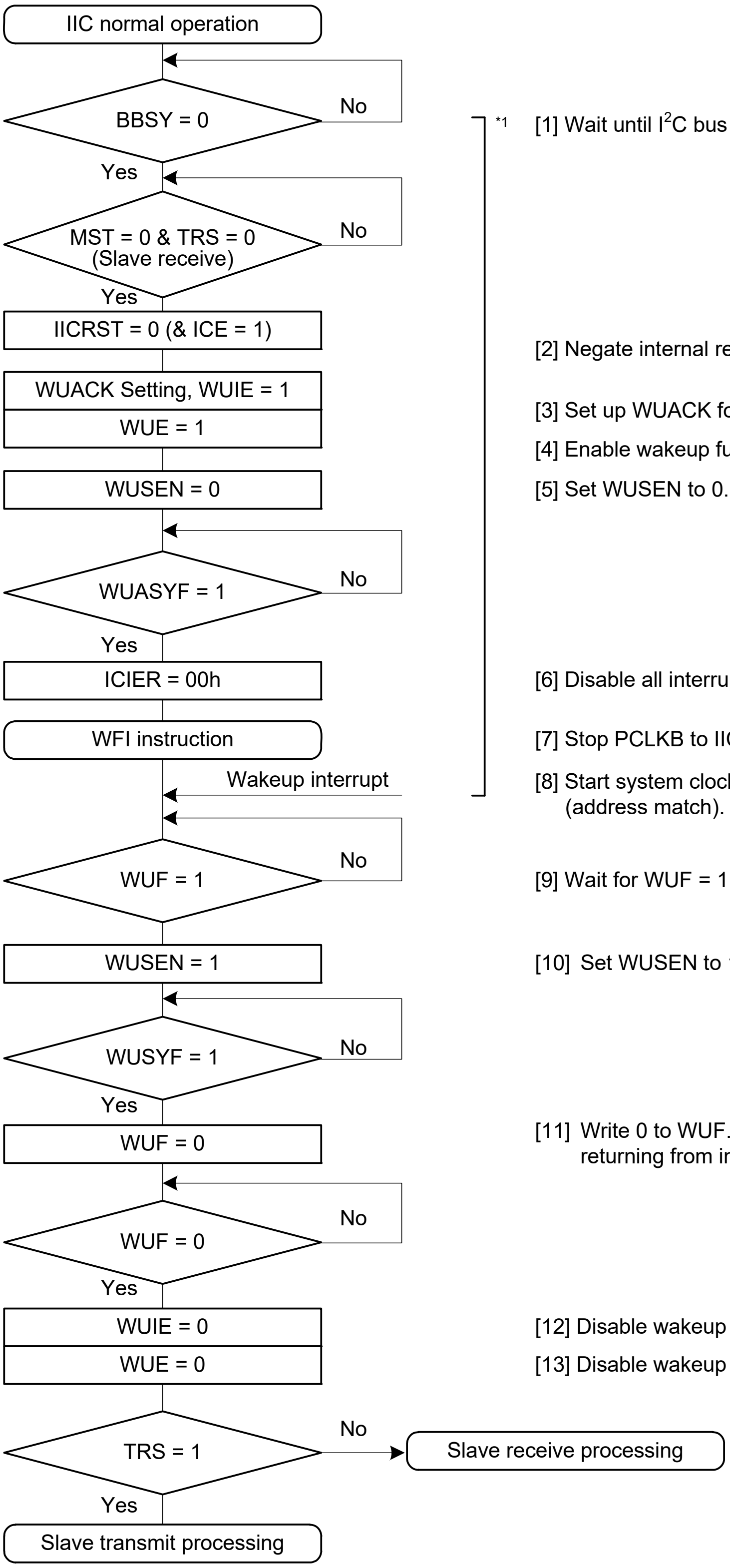
Before wakeup:	ACK is sent in response to the data received with its own slave address of the IIC.
During wakeup:	ACK response is made on the 9th clock cycle of SCL, after which SCL is held low.*1
After wakeup:	Normal operation continues.

If the slave address does not match, the SCL line is not held low after the 9th clock cycle of SCL, and the slave operation continues.

Figure 30.30 shows an operation example and Figure 30.32 shows the detailed timing.

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 the IRQn, the WUF flag is not set to 1. Figure 30.31 shows an operation example.


IIC normal operation
BBSY = 0
No
Yes
MST = 0 & TRS = 0
(Slave receive)
No
Yes
IICRST = 0 (& ICE = 1)
WUACK Setting, WUIE = 1
WUE = 1
WUSEN = 0
WUASYF = 1
No
Yes
ICIER = 00h
WFI instruction
Wakeup interrupt
WUF = 1
No
WUSEN = 1
WUSYF = 1
No
Yes
WUF = 0
WUF = 0
No
Yes
WUIE = 0
WUE = 0
TRS = 1
No
Slave receive processing
Yes
Slave transmit processing
*1

[1] Wait until I^{2}C bus is free and stay in standby state.

[2] Negate internal reset (if asserted).

[3] Set up WUACK for desired wakeup mode. Enable wakeup interrupt.

[4] Enable wakeup function.

[5] Set WUSEN to 0.

[6] Disable all interrupt requests except WUI.

[7] Stop PCLKB to IIC. IIC continues to receive.

[8] Start system clock and PCLKB to IIC on wakeup interrupt (address match).

[9] Wait for WUF = 1.

[10] Set WUSEN to 1.

[11] Write 0 to WUF. Read and check that WUF = 0 before returning from interrupt handling.

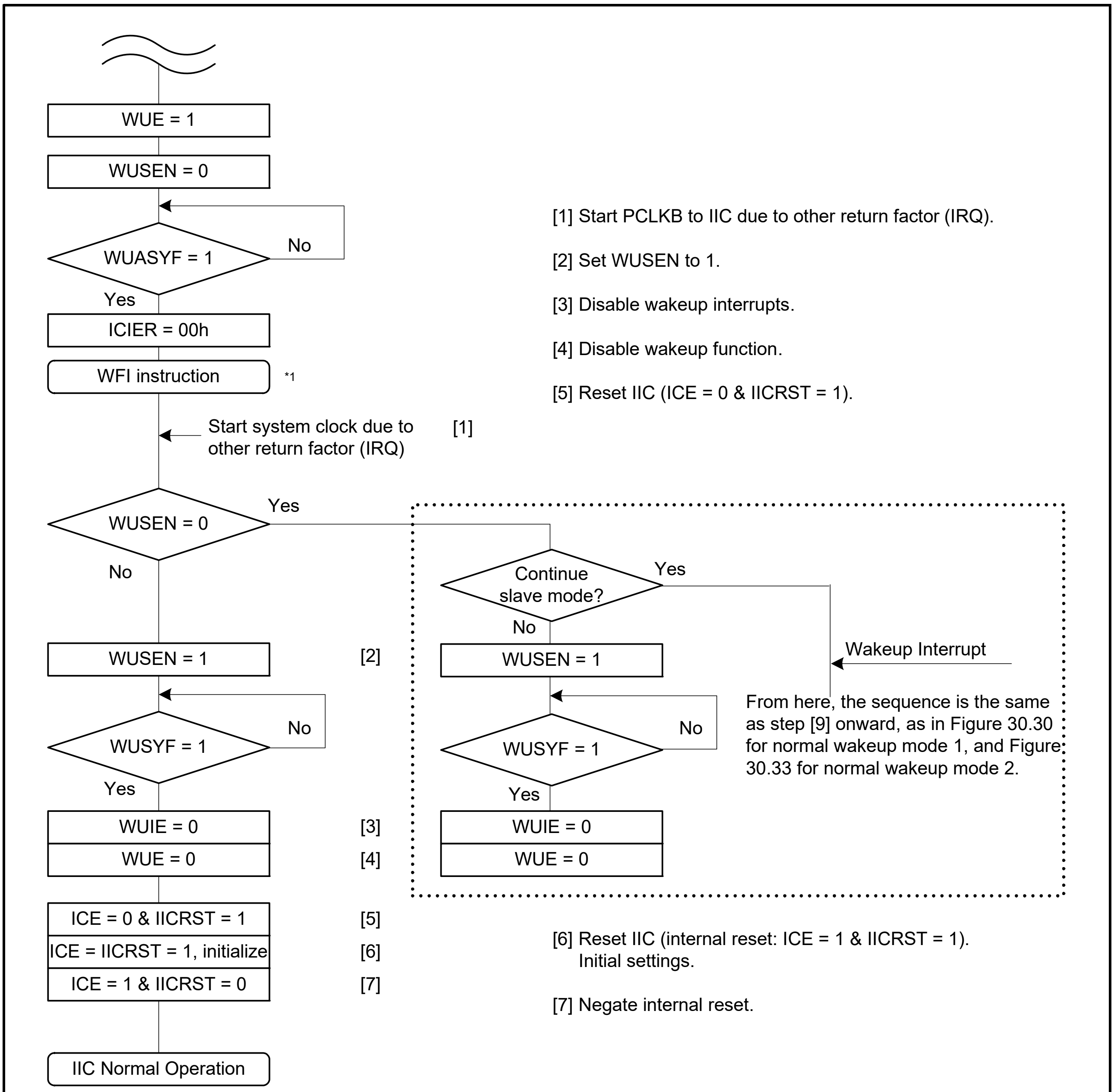
[12] Disable wakeup interrupt.

[13] Disable wakeup function.

Note 1. Do not issue start condition between BBSY = 0 and executing WFI instruction.

Figure 30.30 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.


WUE = 1
WUSEN = 0
WUASYF = 1
No
Yes
ICIER = 00h
WFI instruction
*1
Start system clock due to other return factor (IRQ)
[1]
WUSEN = 0
Yes
No
Continue slave mode?
Yes
No
WUSEN = 1
[2]
WUSEN = 1
Wakeup Interrupt
WUSYF = 1
No
Yes
WUSYF = 1
No
Yes
WUIE = 0
WUE = 0
[3]
[4]
WUIE = 0
WUE = 0
ICE = 0 & IICRST = 1
ICE = IICRST = 1, initialize
ICE = 1 & IICRST = 0
[5]
[6]
[7]
IIC Normal Operation

[1] Start PCLKB to IIC due to other return factor (IRQ).

[2] Set WUSEN to 1.

[3] Disable wakeup interrupts.

[4] Disable wakeup function.

[5] Reset IIC (ICE = 0 & IICRST = 1).

From here, the sequence is the same as step [9] onward, as in Figure 30.30 for normal wakeup mode 1, and Figure 30.33 for normal wakeup mode 2.

[6] Reset IIC (internal reset: ICE = 1 & IICRST = 1).
Initial settings.

[7] Negate internal reset.

Note 1. Before this step, the flow is the same as Figure 30.30.

Figure 30.31 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 30.3.2, Initial Settings.

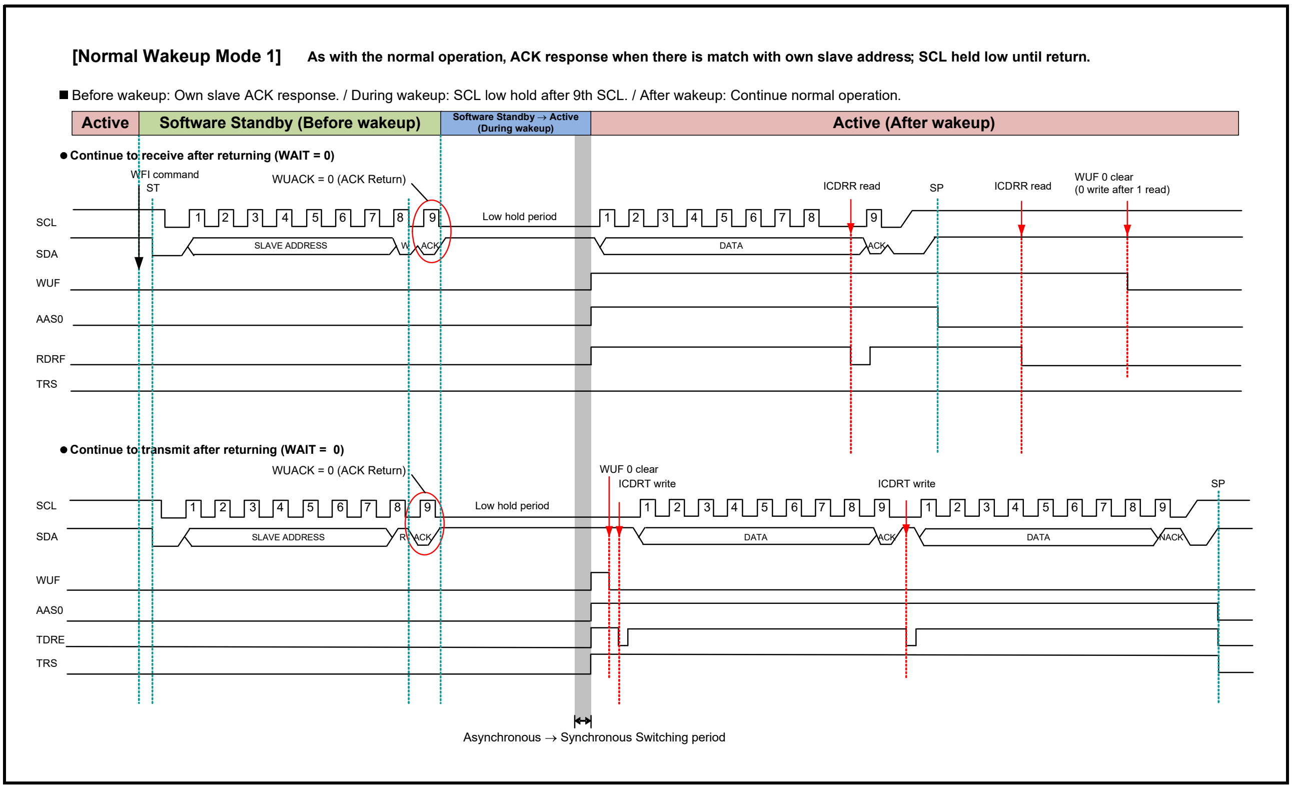

[Normal Wakeup Mode 1] As with the normal operation, ACK response when there is match with own slave address; SCL held low until return.
■ Before wakeup: Own slave ACK response. / During wakeup: SCL low hold after 9th SCL. / After wakeup: Continue normal operation.
Active
Software Standby (Before wakeup)
Software Standby → Active (During wakeup)
Active (After wakeup)
● Continue to receive after returning (WAIT = 0)
WFI command
ST
WUACK = 0 (ACK Return)
Low hold period
ICDRR read
SP
ICDRR read
WUF 0 clear (0 write after 1 read)
SCL
SDA
WUF
AAS0
RDRF
TRS
SLAVE ADDRESS
W
ACK
DATA
ACK
● Continue to transmit after returning (WAIT = 0)
WUACK = 0 (ACK Return)
WUF 0 clear
ICDRT write
ICDRT write
SP
SCL
SDA
WUF
AAS0
TDRE
TRS
SLAVE ADDRESS
R
ACK
DATA
ACK
DATA
NACK
Low hold period
Asynchronous → Synchronous Switching period

Figure 30.32 Timing of normal wakeup mode 1

30.8.2 Normal Wakeup Mode 2

This section describes the behavior, timing, and an example operation in normal wakeup mode 2.

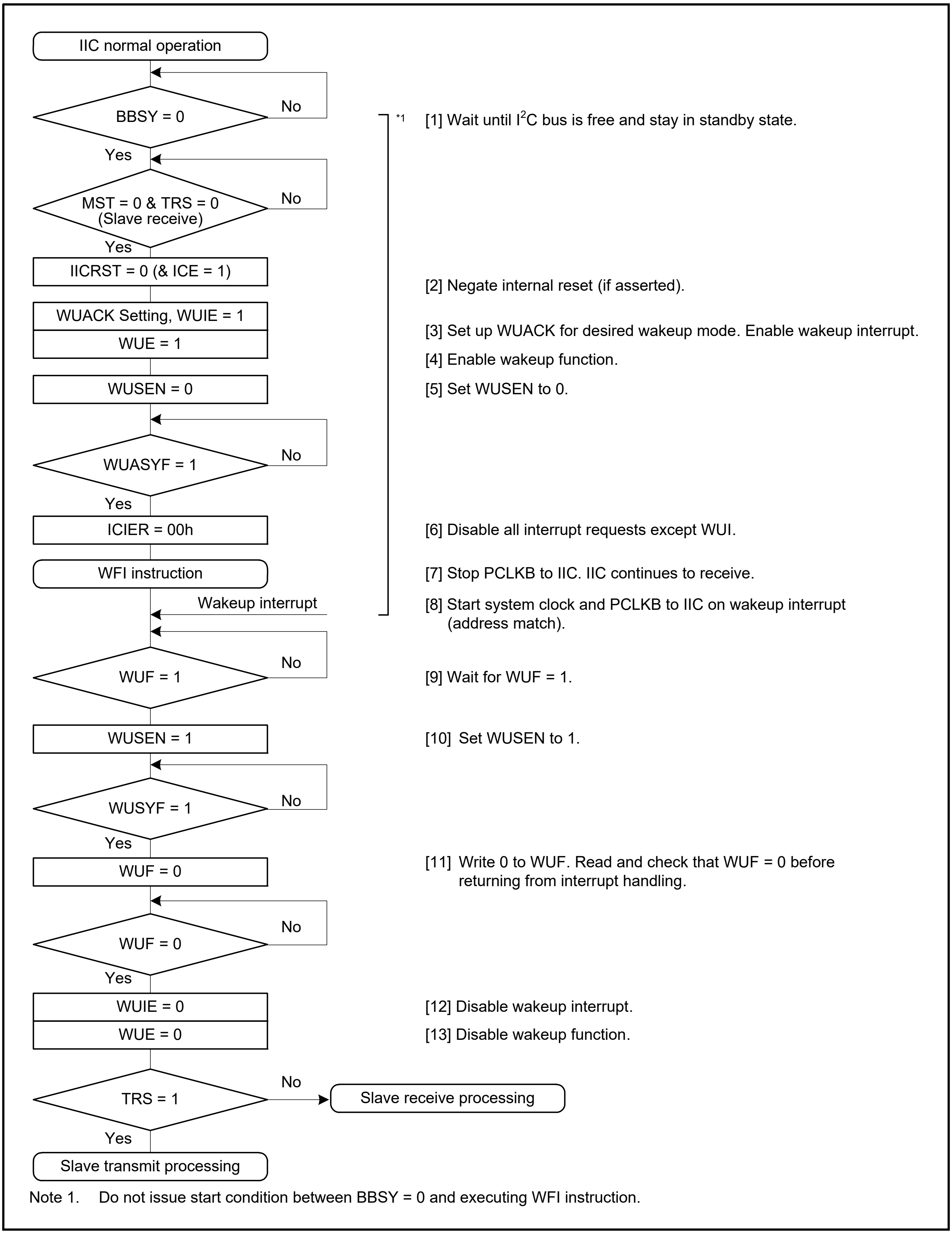
In normal wakeup mode 2, a wakeup interrupt triggered by the match of the slave address initiates the transition to normal operation as follows:

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.

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.

For an example operation in normal wakeup mode 2, see Figure 30.33. Figure 30.34 shows the detailed timing.

If the transition from Software Standby mode is triggered by an interrupt other than a wakeup interrupt, for example IRQ, the WUF flag is not set to 1. Figure 30.31shows an operation example.


IIC normal operation
BBSY = 0
No
Yes
MST = 0 & TRS = 0 (Slave receive)
No
Yes
IICRST = 0 (& ICE = 1)
WUACK Setting, WUIE = 1
WUE = 1
WUSEN = 0
WUASYF = 1
No
Yes
ICIER = 00h
WFI instruction
Wakeup interrupt
WUF = 1
No
WUSEN = 1
WUSYF = 1
No
Yes
WUF = 0
WUF = 0
No
Yes
WUIE = 0
WUE = 0
TRS = 1
No
Slave receive processing
Yes
Slave transmit processing
*1

[1] Wait until I²C bus is free and stay in standby state.

[2] Negate internal reset (if asserted).

[3] Set up WUACK for desired wakeup mode. Enable wakeup interrupt.

[4] Enable wakeup function.

[5] Set WUSEN to 0.

[6] Disable all interrupt requests except WUI.

[7] Stop PCLKB to IIC. IIC continues to receive.

[8] Start system clock and PCLKB to IIC on wakeup interrupt (address match).

[9] Wait for WUF = 1.

[10] Set WUSEN to 1.

[11] Write 0 to WUF. Read and check that WUF = 0 before returning from interrupt handling.

[12] Disable wakeup interrupt.

[13] Disable wakeup function.

Note 1. Do not issue start condition between BBSY = 0 and executing WFI instruction.

Figure 30.33 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.

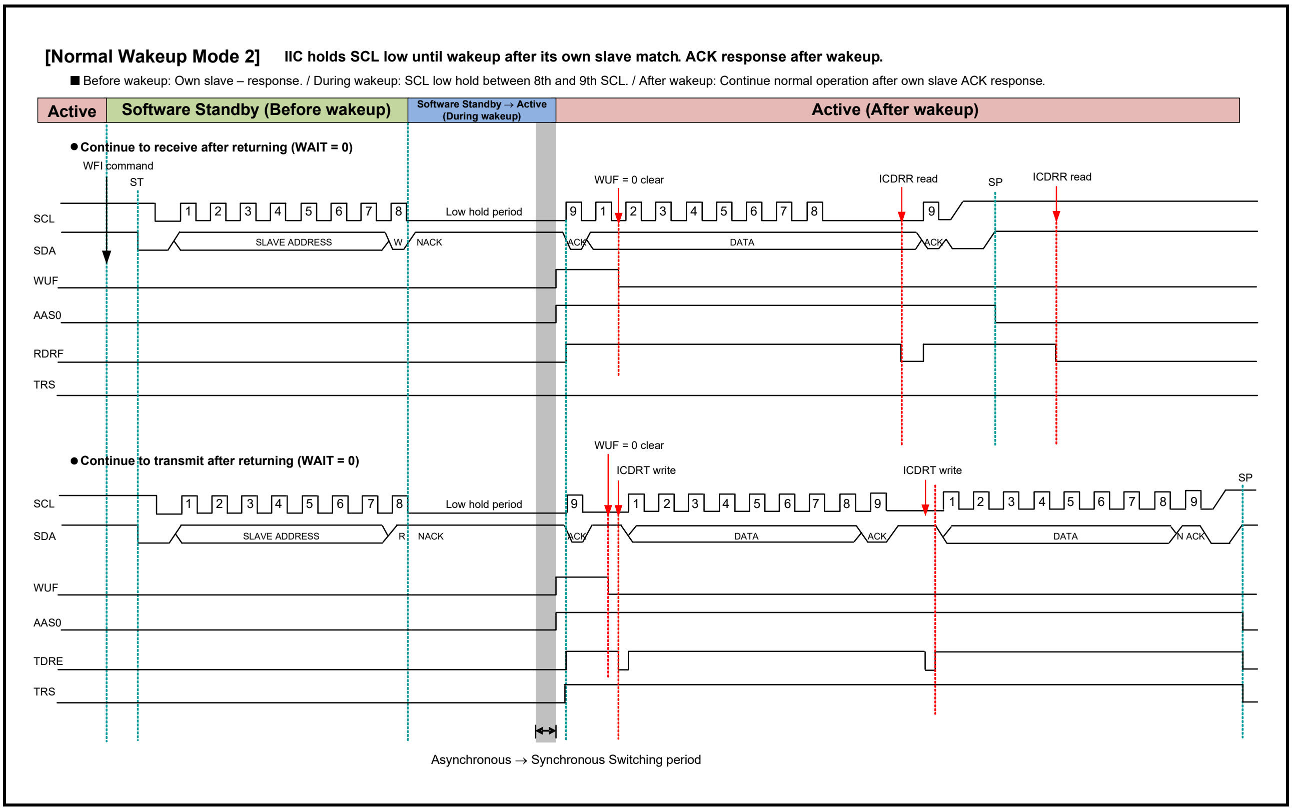

[Normal Wakeup Mode 2] IIC holds SCL low until wakeup after its own slave match. ACK response after wakeup.
■ Before wakeup: Own slave – response. / During wakeup: SCL low hold between 8th and 9th SCL. / After wakeup: Continue normal operation after own slave ACK response.
Active
Software Standby (Before wakeup)
Software Standby → Active (During wakeup)
Active (After wakeup)
● Continue to receive after returning (WAIT = 0)
WFI command
ST
SCL
SDA
WUF
AAS0
RDRF
TRS
SLAVE ADDRESS
W
NACK
Low hold period
WUF = 0 clear
ACK
DATA
ICDRR read
SP
ICDRR read
● Continue to transmit after returning (WAIT = 0)
ICDRT write
ICDRT write
R
N ACK
TDRE
Asynchronous → Synchronous Switching period

Figure 30.34 Timing of normal wakeup mode 2

30.8.3 Command Recovery 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 I^{2}C bus during this period.

This section describes the behavior, timing, and an example operation in command recovery and EEP response modes.

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.

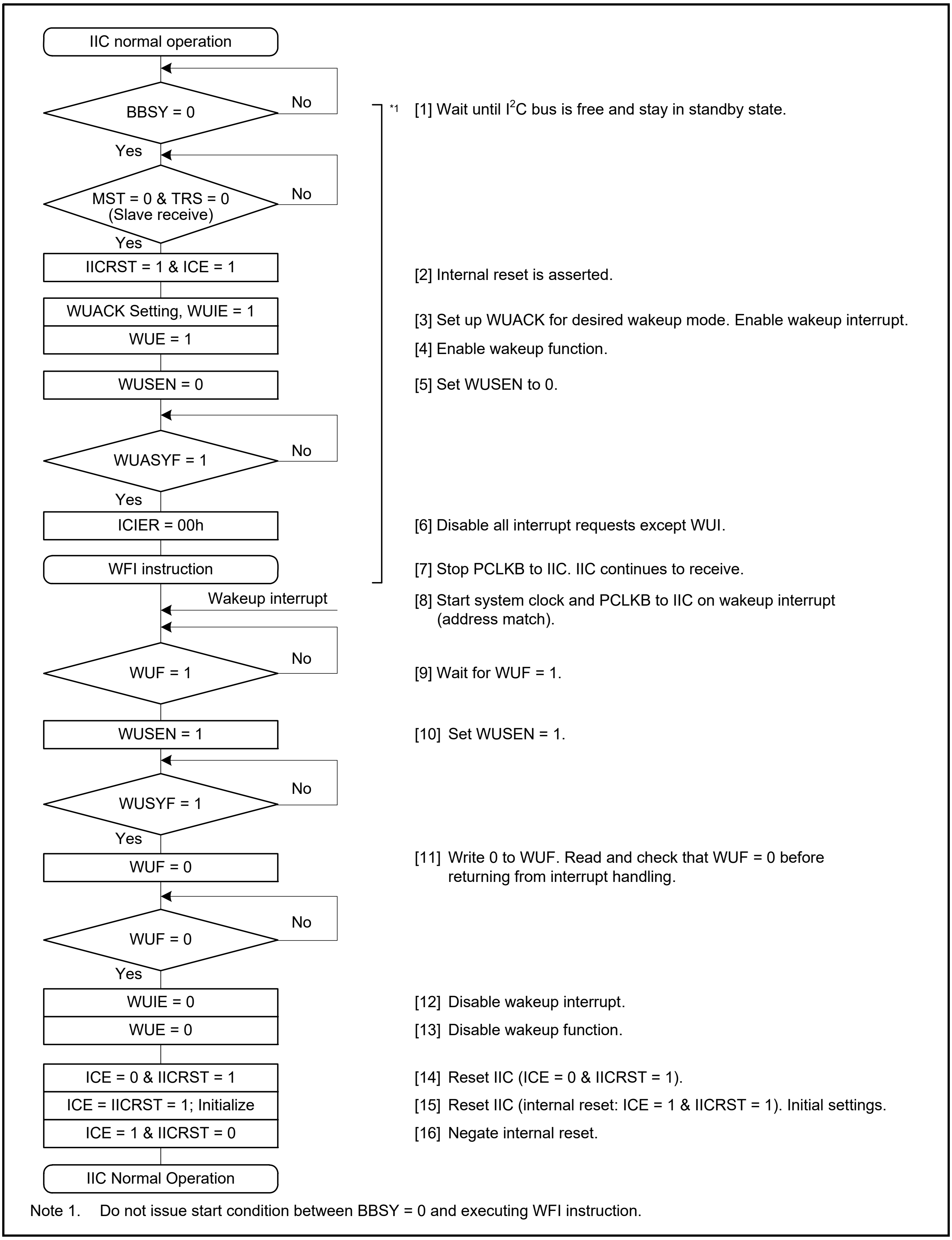
If the slave address does not match, the slave operation continues.

For an example operation in command recovery and EEP response modes, see Figure 30.35. Figure 30.37 shows the detailed timing.

Note: Because the SCL line is not held low during wakeup, transmission or reception of the data that follows the slave address is not possible.

Note: The command recovery and EEP response modes are internal reset states (ICE = IICRST = 1). Therefore, the match of the slave address does not set the flags HOA, GCA, 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. Figure 30.36 shows an operation example.


IIC normal operation
BBSY = 0
No
Yes
MST = 0 & TRS = 0
(Slave receive)
No
Yes
IICRST = 1 & ICE = 1
WUACK Setting, WUIE = 1
WUE = 1
WUSEN = 0
WUASYF = 1
No
Yes
ICIER = 00h
WFI instruction
Wakeup interrupt
WUF = 1
No
WUSEN = 1
WUSYF = 1
No
Yes
WUF = 0
WUF = 0
No
Yes
WUIE = 0
WUE = 0
ICE = 0 & IICRST = 1
ICE = IICRST = 1; Initialize
ICE = 1 & IICRST = 0
IIC Normal Operation
*1

[1] Wait until I^2C bus is free and stay in standby state.

[2] Internal reset is asserted.

[3] Set up WUACK for desired wakeup mode. Enable wakeup interrupt.

[4] Enable wakeup function.

[5] Set WUSEN to 0.

[6] Disable all interrupt requests except WUI.

[7] Stop PCLKB to IIC. IIC continues to receive.

[8] Start system clock and PCLKB to IIC on wakeup interrupt (address match).

[9] Wait for WUF = 1.

[10] Set WUSEN = 1.

[11] Write 0 to WUF. Read and check that WUF = 0 before returning from interrupt handling.

[12] Disable wakeup interrupt.

[13] Disable wakeup function.

[14] Reset IIC (ICE = 0 & IICRST = 1).

[15] Reset IIC (internal reset: ICE = 1 & IICRST = 1). Initial settings.

[16] Negate internal reset.

Note 1. Do not issue start condition between BBSY = 0 and executing WFI instruction.

Figure 30.35 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.

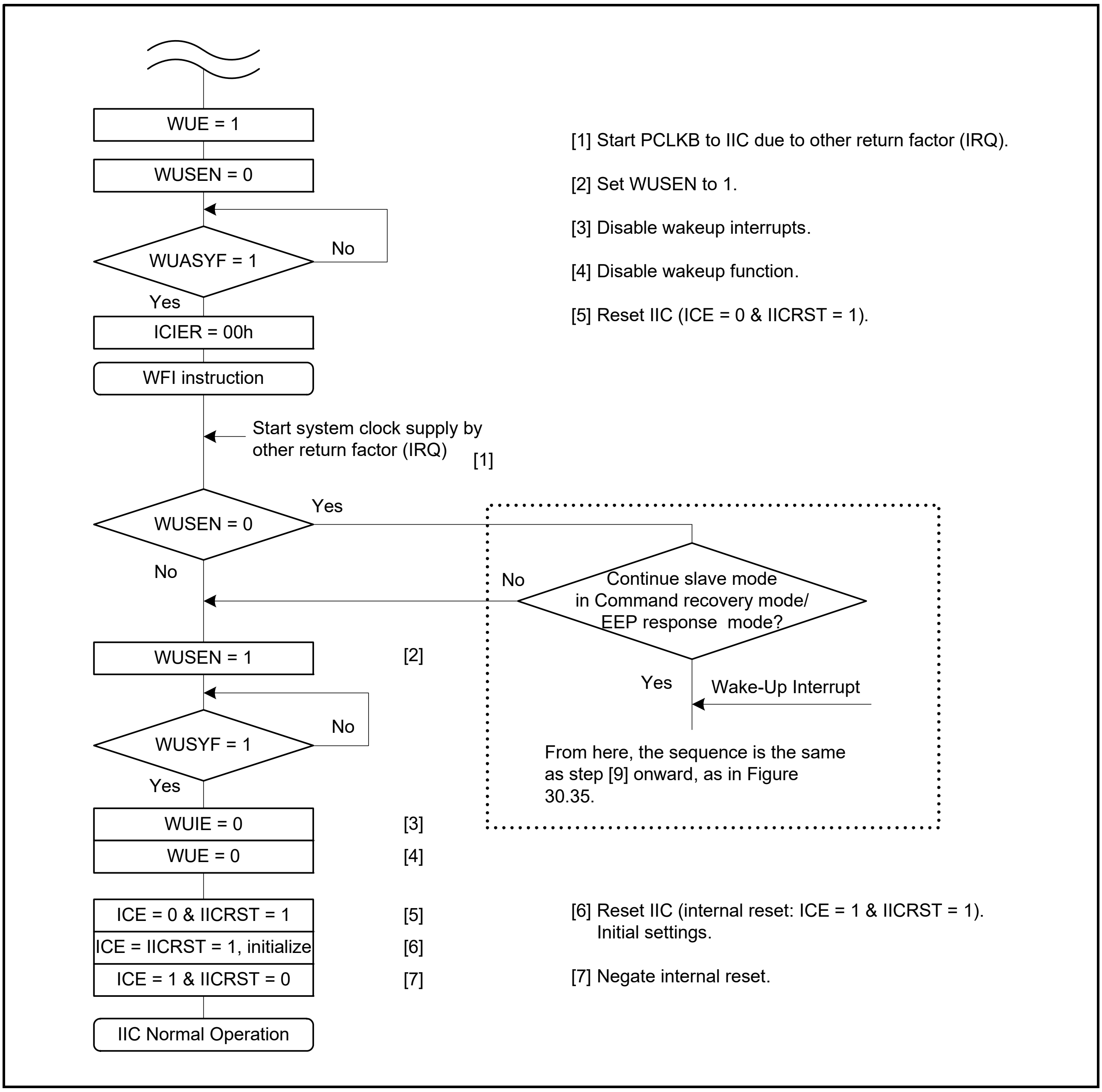

WUE = 1
WUSEN = 0
WUASYF = 1
No
Yes
ICIER = 00h
WFI instruction
Start system clock supply by other return factor (IRQ)
[1]
WUSEN = 0
Yes
No
Continue slave mode in Command recovery mode/ EEP response mode?
No
Yes
Wake-Up Interrupt
From here, the sequence is the same as step [9] onward, as in Figure 30.35.
WUSEN = 1
[2]
WUSYF = 1
No
Yes
WUIE = 0
[3]
WUE = 0
[4]
ICE = 0 & IICRST = 1
[5]
ICE = IICRST = 1, initialize
[6]
ICE = 1 & IICRST = 0
[7]
IIC Normal Operation
[1] Start PCLKB to IIC due to other return factor (IRQ).
[2] Set WUSEN to 1.
[3] Disable wakeup interrupts.
[4] Disable wakeup function.
[5] Reset IIC (ICE = 0 & IICRST = 1).
[6] Reset IIC (internal reset: ICE = 1 & IICRST = 1). Initial settings.
[7] Negate internal reset.

Figure 30.36 **Example operation of command recovery and EEP response modes when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example, the IRQn**

Note: For details of the IIC initial settings, see section 30.3.2, Initial Settings.

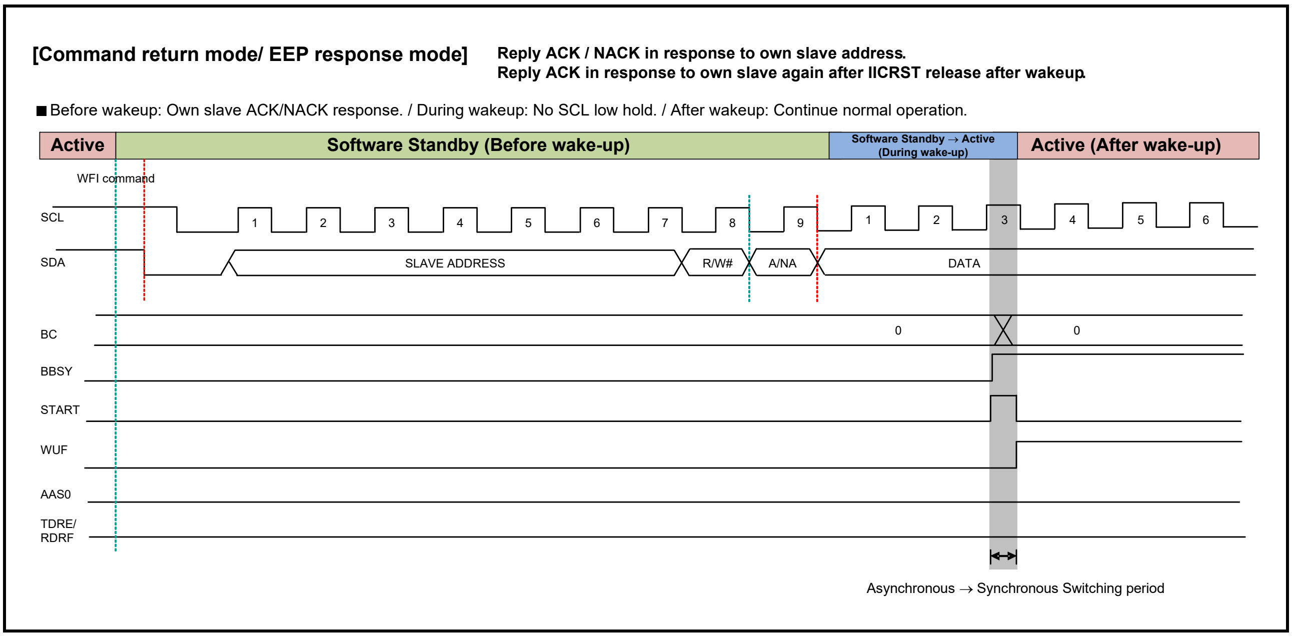

[Command return mode/ EEP response mode]
Reply ACK / NACK in response to own slave address.
Reply ACK in response to own slave again after IICRST release after wakeup.
■ Before wakeup: Own slave ACK/NACK response. / During wakeup: No SCL low hold. / After wakeup: Continue normal operation.
Active
Software Standby (Before wake-up)
Software Standby → Active (During wake-up)
Active (After wake-up)
WFI command
SCL
1
2
3
4
5
6
7
8
9
1
2
3
4
5
6
SDA
SLAVE ADDRESS
R/W#
A/NA
DATA
BC
0
0
BBSY
START
WUF
AAS0
TDRE/
RDRF
Asynchronous → Synchronous Switching period

Figure 30.37 Timing of command recovery and EEP response modes

30.8.4 Precautions for WFI Instruction Execution

In the wakeup function examples shown in Figure 30.30, Figure 30.33, and Figure 30.35, 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.

30.9 Automatic Low-Hold Function for SCL

30.9.1 Function to Prevent Wrong Transmission of Transmit Data

If the I^{2}C Bus Shift Register (ICDRS) is empty when data has not been written to the I^{2}C Bus Transmit Data Register (ICDRT) with the IIC in transmission mode (ICCR2.TRS = 1), the SCLn line is automatically held low over the subsequent intervals. This low-hold period is extended until the transmit data is written, which prevents the unintended transmission of erroneous data.

Master transmit mode:

- Low-level interval after a start or restart condition is issued
- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

Slave transmit mode:

- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

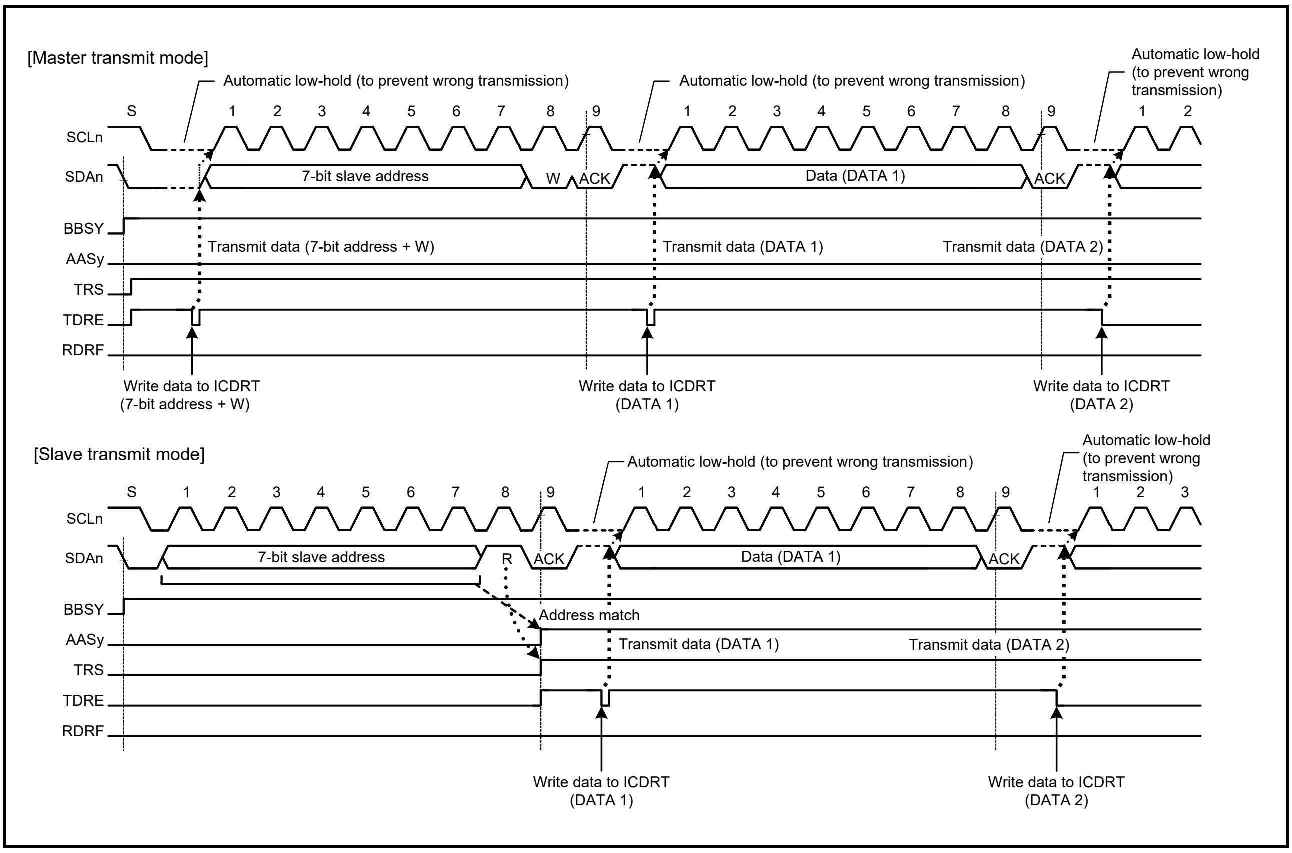

[Master transmit mode]
Automatic low-hold (to prevent wrong transmission)
Automatic low-hold (to prevent wrong transmission)
Automatic low-hold (to prevent wrong transmission)
S
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2
SCLn
SDAn
7-bit slave address
W
ACK
Data (DATA 1)
ACK
BBSY
AASy
TRS
TDRE
RDRF
Transmit data (7-bit address + W)
Transmit data (DATA 1)
Transmit data (DATA 2)
Write data to ICDRT (7-bit address + W)
Write data to ICDRT (DATA 1)
Write data to ICDRT (DATA 2)
[Slave transmit mode]
Automatic low-hold (to prevent wrong transmission)
Automatic low-hold (to prevent wrong transmission)
S
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3
SCLn
SDAn
7-bit slave address
R
ACK
Data (DATA 1)
ACK
BBSY
Address match
AASy
Transmit data (DATA 1)
Transmit data (DATA 2)
TRS
TDRE
RDRF
Write data to ICDRT (DATA 1)
Write data to ICDRT (DATA 2)

Figure 30.38 Automatic low-hold operation in transmit mode

30.9.2 NACK Reception Transfer Suspension Function

This function suspends transfer operation when NACK is received in transmit mode (the ICCR2.TRS bit is 1). This function is enabled when the NACKE bit in ICFER is set to 1 (transfer suspension enabled). If the next transmit data is written (ICSR2.TDRE flag is 0) when the 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 flag is 1), transmit and receive operations are discontinued. To restore transmit and receive operations, set the NACKF flag to 0. In master transmit mode, after issuing a restart or stop condition, set the NACKF flag to 0, and then issue a start condition again.

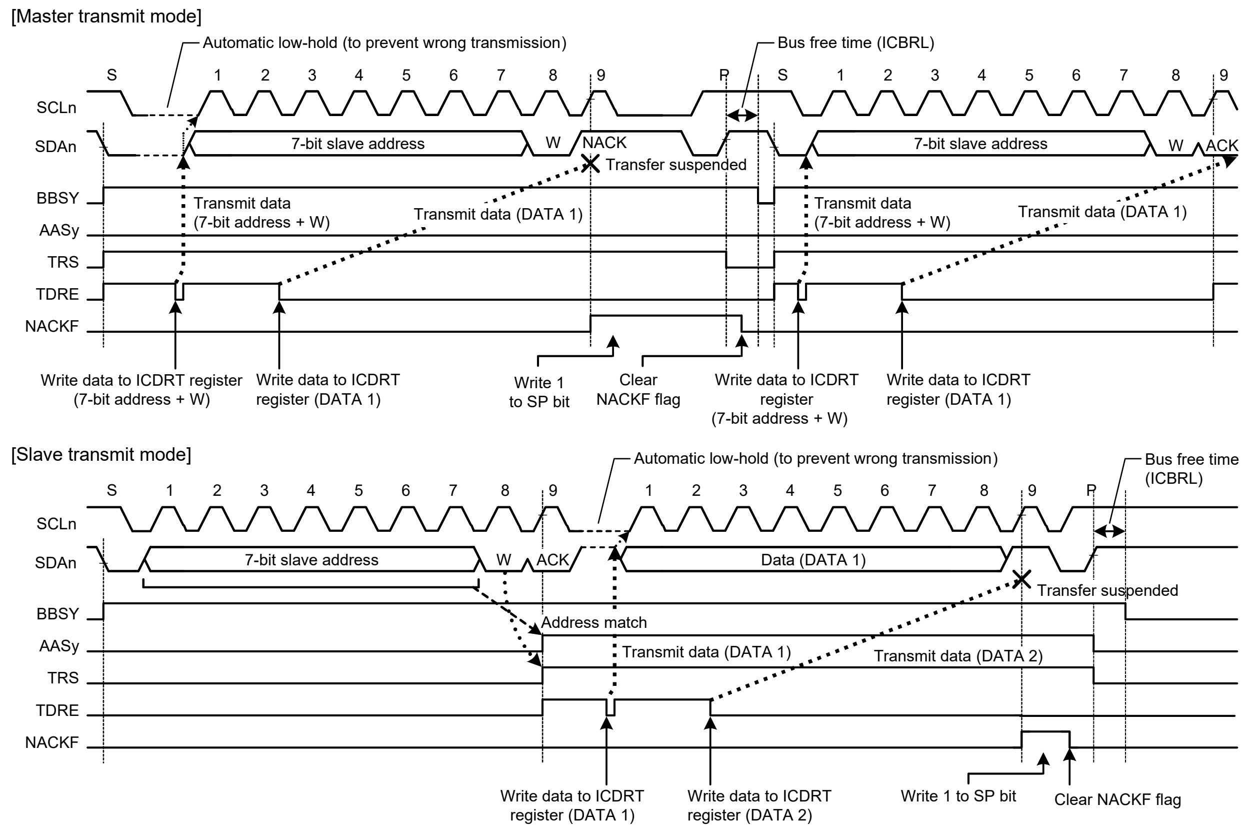

[Master transmit mode]
Automatic low-hold (to prevent wrong transmission)
Bus free time (ICBRL)
S
1 2 3 4 5 6 7 8 9
P
S
1 2 3 4 5 6 7 8 9
SCLn
SDAn
7-bit slave address
W
NACK
7-bit slave address
W
ACK
Transfer suspended
BBSY
AASy
TRS
TDRE
NACKF
Transmit data (7-bit address + W)
Transmit data (DATA 1)
Transmit data (7-bit address + W)
Transmit data (DATA 1)
Write data to ICDRT register (7-bit address + W)
Write data to ICDRT register (DATA 1)
Write 1 to SP bit
Clear NACKF flag
Write data to ICDRT register (7-bit address + W)
Write data to ICDRT register (DATA 1)
[Slave transmit mode]
Automatic low-hold (to prevent wrong transmission)
Bus free time (ICBRL)
S
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
P
SCLn
SDAn
7-bit slave address
W
ACK
Data (DATA 1)
Transfer suspended
BBSY
AASy
TRS
TDRE
NACKF
Address match
Transmit data (DATA 1)
Transmit data (DATA 2)
Write data to ICDRT register (DATA 1)
Write data to ICDRT register (DATA 2)
Write 1 to SP bit
Clear NACKF flag

Figure 30.39 Suspension of data transfer when NACK is received (NACKE = 1)

30.9.3 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 interfere 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 ICMR3, but cannot be released by reading data from ICDRR, which enables receive operation through the ACK or NACK transmission control based on the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master receive mode or slave receive mode.

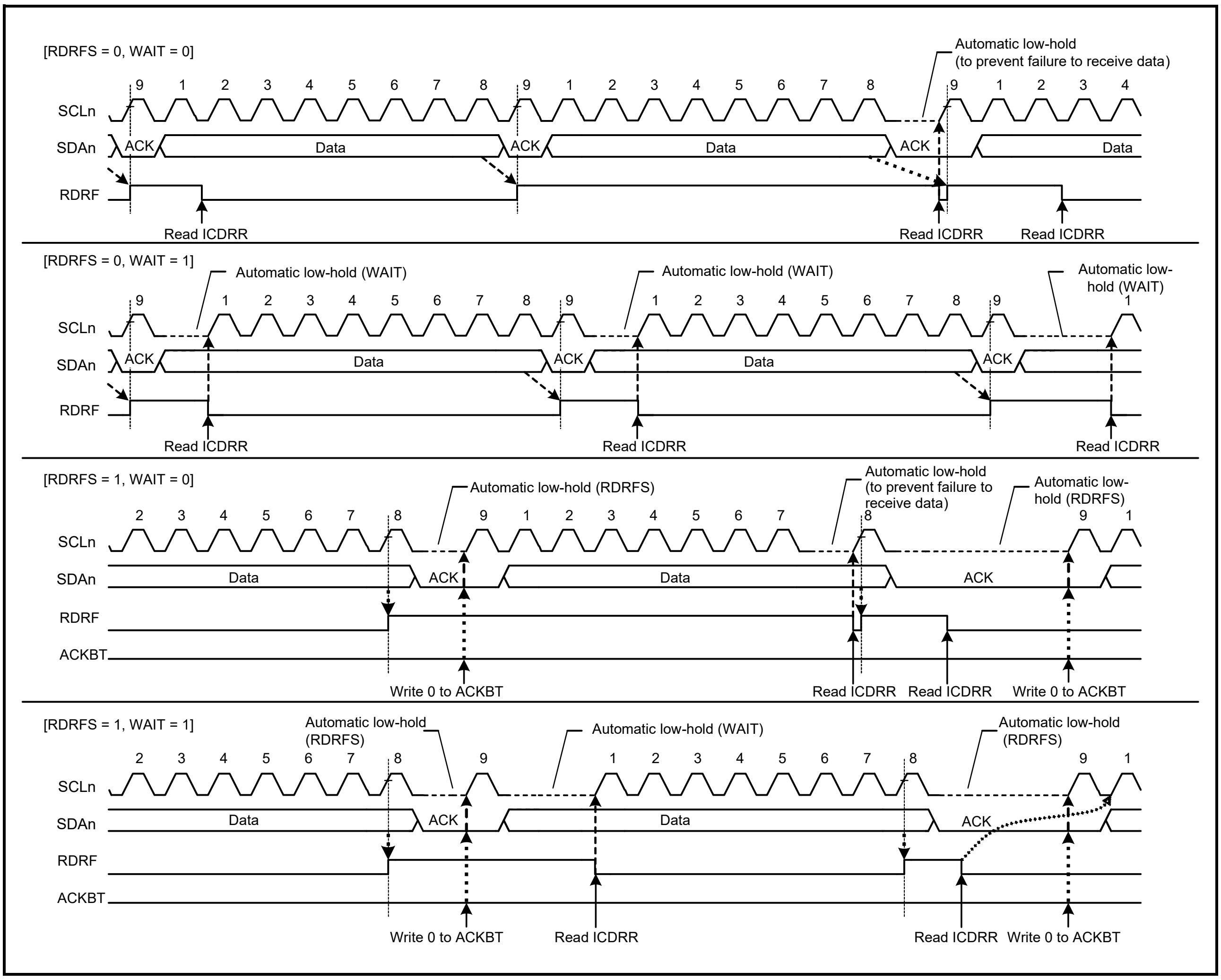

[RDRFS = 0, WAIT = 0]
Automatic low-hold (to prevent failure to receive data)
9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4
SCLn
SDAn
ACK
Data
RDRF
Read ICDRR
Read ICDRR
Read ICDRR
[RDRFS = 0, WAIT = 1]
Automatic low-hold (WAIT)
Automatic low-hold (WAIT)
Automatic low-hold (WAIT)
Read ICDRR
Read ICDRR
Read ICDRR
[RDRFS = 1, WAIT = 0]
Automatic low-hold (RDRFS)
Automatic low-hold (to prevent failure to receive data)
Automatic low-hold (RDRFS)
ACKBT
Write 0 to ACKBT
Read ICDRR
Read ICDRR
Write 0 to ACKBT
[RDRFS = 1, WAIT = 1]
Automatic low-hold (RDRFS)
Automatic low-hold (WAIT)
Automatic low-hold (RDRFS)
Write 0 to ACKBT
Read ICDRR
Read ICDRR
Write 0 to ACKBT

Figure 30.40 Automatic low-hold operation in receive mode using RDRFS and WAIT bits

30.10 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.

30.10.1 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]

- Mismatching of the internal level for output on SDA and the level on the SDAn line after a start condition was issued by setting the ST bit in ICCR2 to 1 while the 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).

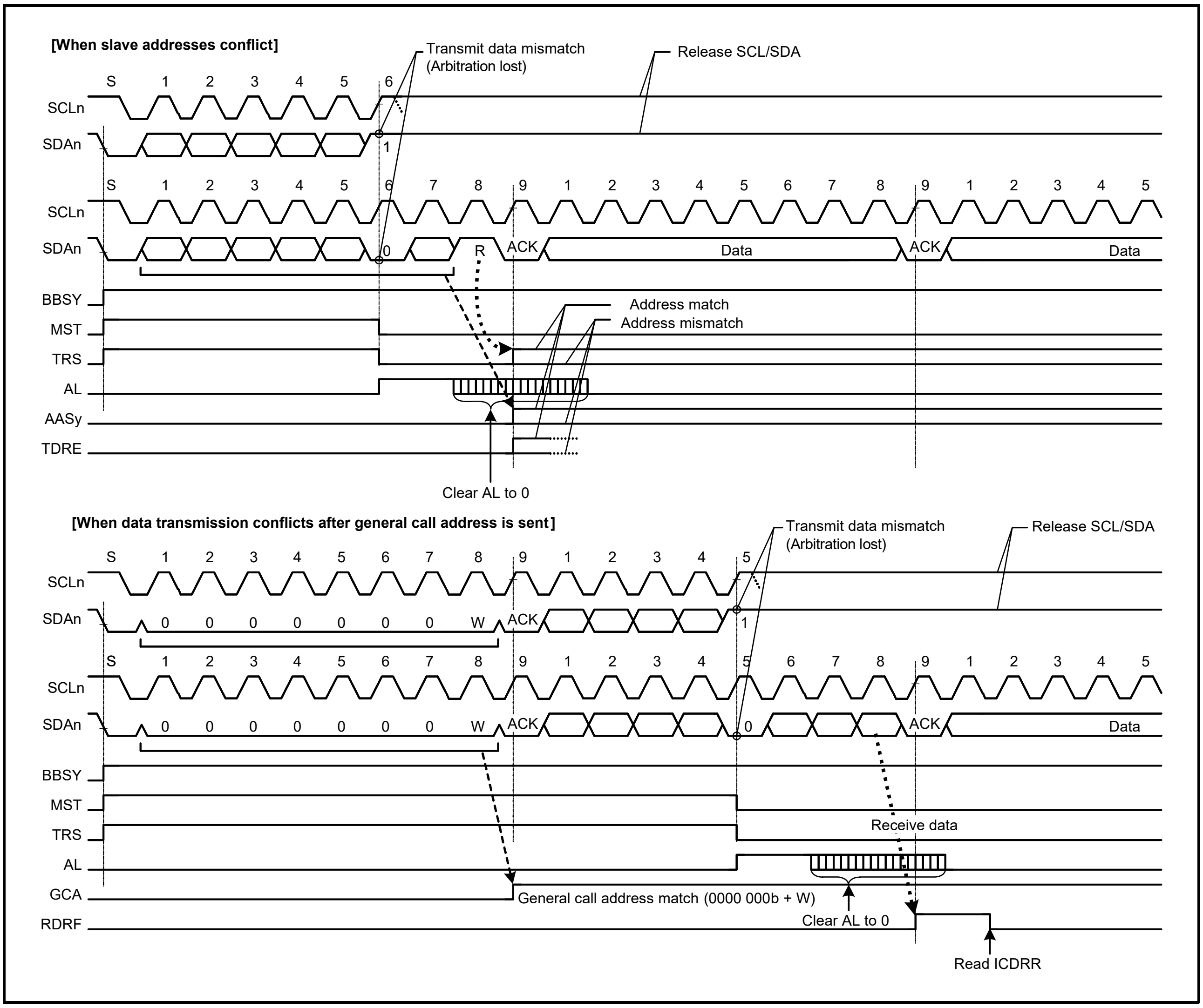

[When slave addresses conflict]
Transmit data mismatch (Arbitration lost)
Release SCL/SDA
S 1 2 3 4 5 6
SCLn
SDAn
1
S 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5
SCLn
SDAn
0 R ACK Data ACK Data
BBSY
MST
TRS
AL
AASy
TDRE
Address match
Address mismatch
Clear AL to 0
[When data transmission conflicts after general call address is sent]
Transmit data mismatch (Arbitration lost)
Release SCL/SDA
S 1 2 3 4 5 6 7 8 9 1 2 3 4 5
SCLn
SDAn
0 0 0 0 0 0 0 W ACK 1
S 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5
SCLn
SDAn
0 0 0 0 0 0 0 W ACK 0 ACK Data
BBSY
MST
TRS
AL
GCA
RDRF
Receive data
General call address match (0000 000b + W)
Clear AL to 0
Read ICDRR

Figure 30.41 Examples of master arbitration-lost detection (MALE = 1)

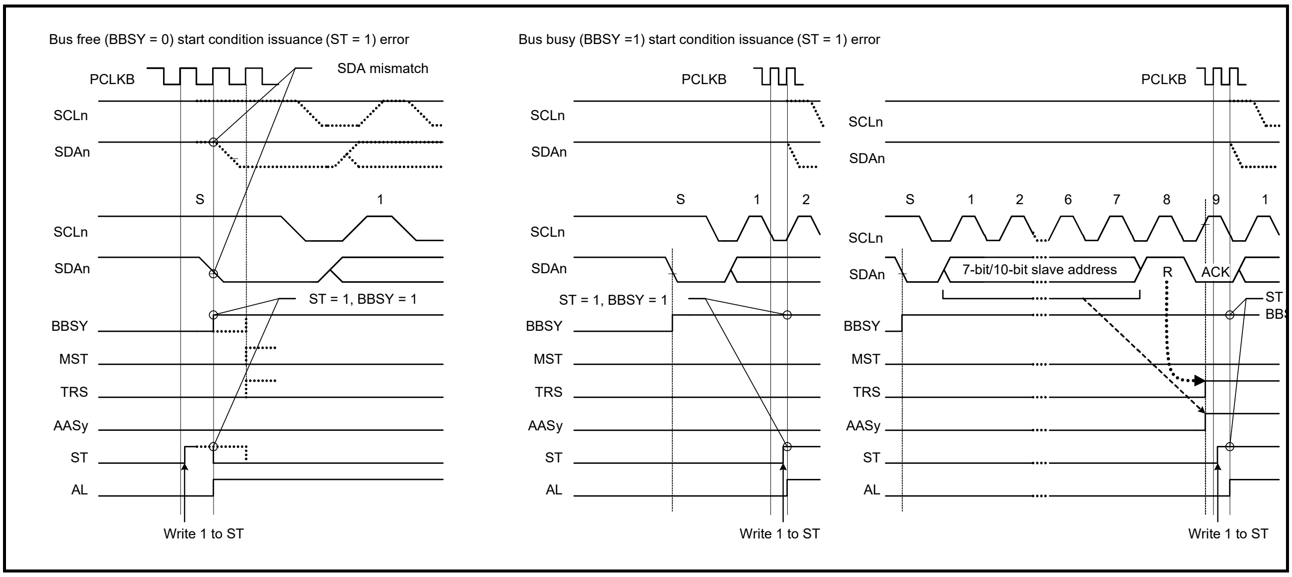

Bus free (BBSY = 0) start condition issuance (ST = 1) error
PCLKB
SDA mismatch
SCLn
SDAn
S
1
SCLn
SDAn
ST = 1, BBSY = 1
BBSY
MST
TRS
AASy
ST
AL
Write 1 to ST
Bus busy (BBSY =1) start condition issuance (ST = 1) error
PCLKB
SCLn
SDAn
S
1
2
ST = 1, BBSY = 1
BBSY
MST
TRS
AASy
ST
AL
Write 1 to ST
PCLKB
SCLn
SDAn
S
1
2
6
7
8
9
1
7-bit/10-bit slave address
R
ACK
ST
BBS
BBSY
MST
TRS
AASy
ST
AL
Write 1 to ST

Figure 30.42 Arbitration-lost when start condition is issued (MALE = 1)

30.10.2 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 and ACK transmissions when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such a conflict occurs when multiple master devices send or receive the same information through a single slave device. Figure 30.43 shows an example of arbitration-lost detection during transmission of NACK.

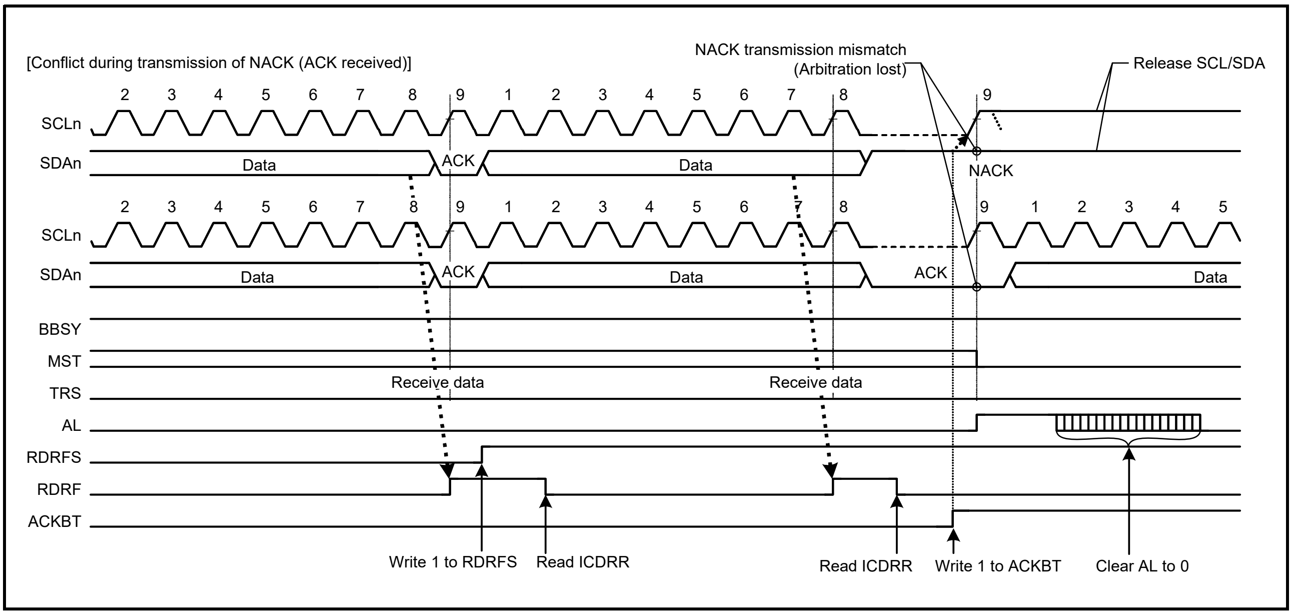

[Conflict during transmission of NACK (ACK received)]
NACK transmission mismatch
(Arbitration lost)
Release SCL/SDA
2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9
SCLn
SDAn
Data
ACK
Data
NACK
2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5
SCLn
SDAn
Data
ACK
Data
ACK
Data
BBSY
MST
TRS
Receive data
Receive data
AL
RDRFS
RDRF
ACKBT
Write 1 to RDRFS
Read ICDRR
Read ICDRR
Write 1 to ACKBT
Clear AL to 0

Figure 30.43 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 either 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 required 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 required if the UDID (Unique Device Identifier) of the assigned address does not match in the Get UDID general processing after the Assign Address command.

The IIC detects arbitration-lost during transmission of NACK when the following condition is met with the NALE bit in ICFER set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the SDAn line (ACK is received) during transmission of NACK (ICMR3.ACKBT = 1).

30.10.3 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 ICFER set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When transmit data excluding acknowledge (internal SDA output level) does not match the SDAn line in slave transmit mode (MST and TRS bits = 01b in ICCR2).

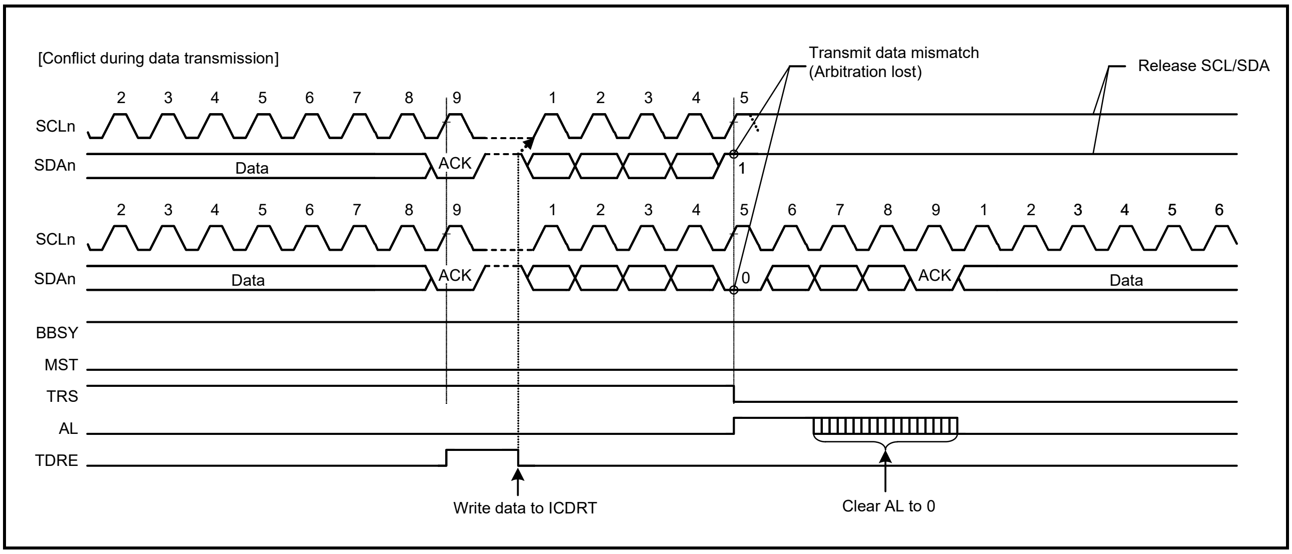


Figure 30.44 Example of slave arbitration-lost detection (SALE = 1)

30.11 Start, Restart, and Stop Condition Issuing Function

30.11.1 Issuing a Start Condition

The IIC issues a start condition when the ST bit in ICCR2 is set to 1.

When the ST bit is set to 1, a start condition request is made. The IIC issues a start condition when the BBSY flag in ICCR2 is 0 (bus free state). When a start condition is issued normally, the IIC automatically shifts to the master transmit mode.

To issue a start condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure that the time set in ICBRH and the start condition hold time elapse.
3. Drive the SCLn line low (high level to low level).
4. Detect low level of the SCLn line and ensure that the low-level period of the SCLn line set in ICBRL elapses.

30.11.2 Issuing a Restart Condition

The IIC issues a restart condition when the RS bit in ICCR2 is set to 1.

When the RS bit is set to 1, a restart condition request is made. The IIC issues a restart condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a restart condition:

1. Release the SDAn line.
2. Ensure the low-level period of SCLn line set in ICBRL 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 ICBRL and the restart condition setup time elapse.
5. Drive the SDAn line low (high level to low level).
6. Ensure the time set in ICBRH and the restart condition hold time elapse.
7. Drive the SCLn line low (high level to low level).
8. Detect a low level of the SCLn line and ensure the low-level period of SCLn line set in ICBRL elapses.

Note: When issuing restart condition requests, write the slave address to ICDRT after confirming that ICCR2.RS is 0. Data written while ICCR2.RS = 1 is not forwarded because of the retransmission condition before the occurrence.

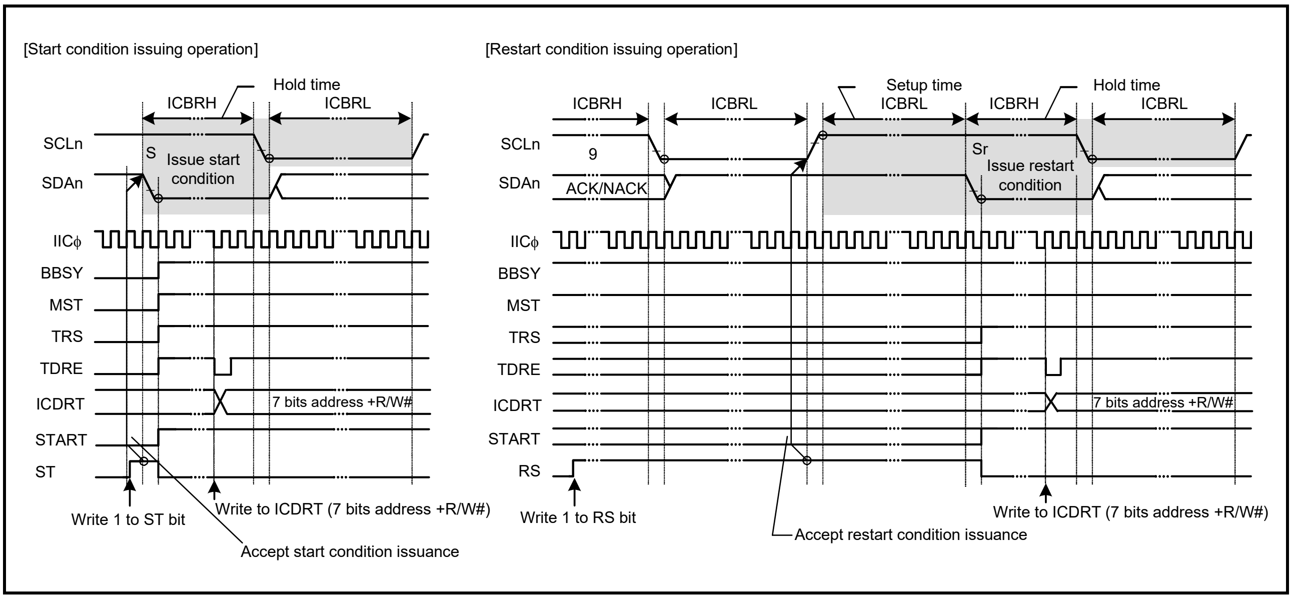

[Start condition issuing operation]
[Restart condition issuing operation]
Hold time
ICBRH
ICBRL
Setup time
SCLn
SDAn
S
Issue start condition
Sr
Issue restart condition
9
ACK/NACK
IICϕ
BBSY
MST
TRS
TDRE
ICDRT
START
ST
RS
7 bits address +R/W#
Write 1 to ST bit
Write to ICDRT (7 bits address +R/W#)
Accept start condition issuance
Write 1 to RS bit
Accept restart condition issuance

Figure 30.45 Start or restart condition issue timing using the ST and RS bits

Figure 30.46 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 30.3.2, 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 is also automatically set 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 ICDRT. After 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 set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode according to the value of the transmitted R/W# bit. If the value of the R/W# bit is 0, the IIC continues in master transmit mode. If the ICSR2.NACKF flag is 1 at this time, indicating that no slave device recognized the address or there was an error in communications, write 1 to ICCR2.SP bit to issue a stop condition.

   To transmit data with an address in the 10-bit format, start by writing 1111 0b, the 2 upper bits of the slave address, and W to ICDRT as the first address transmission. Then, as the second address transmission, write the 8 lower bits of the slave address to 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 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 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.

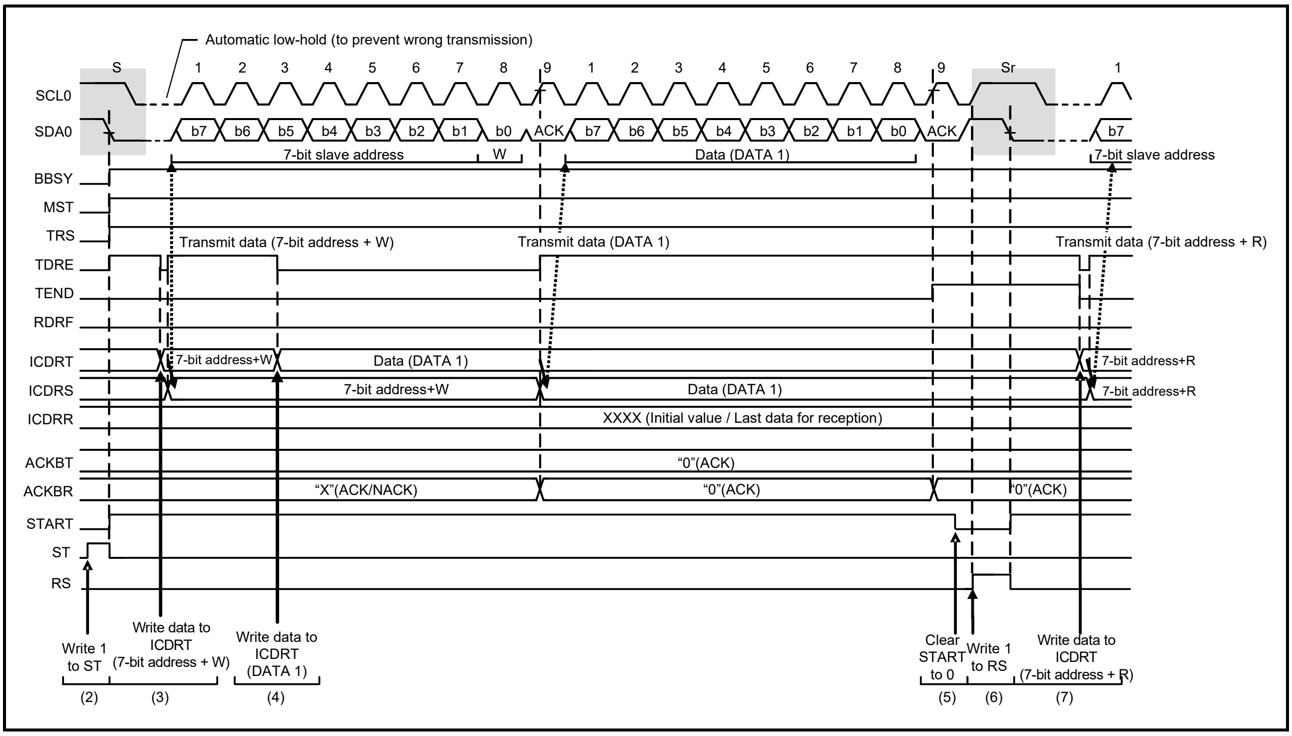

Automatic low-hold (to prevent wrong transmission)
S
1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9
Sr
1
SCL0
SDA0
b7 b6 b5 b4 b3 b2 b1 b0 ACK b7 b6 b5 b4 b3 b2 b1 b0 ACK b7
7-bit slave address
W
Data (DATA 1)
7-bit slave address
BBSY
MST
TRS
TDRE
TEND
RDRF
ICDRT
ICDRS
ICDRR
ACKBT
ACKBR
START
ST
RS
Transmit data (7-bit address + W)
Transmit data (DATA 1)
Transmit data (7-bit address + R)
7-bit address+W
Data (DATA 1)
7-bit address+R
7-bit address+W
Data (DATA 1)
7-bit address+R
XXXX (Initial value / Last data for reception)
"0"(ACK)
"X"(ACK/NACK)
"0"(ACK)
"0"(ACK)
Write 1 to ST
Write data to ICDRT (7-bit address + W)
Write data to ICDRT (DATA 1)
Clear START to 0
Write 1 to RS
Write data to ICDRT (7-bit address + R)
(2)
(3)
(4)
(5)
(6)
(7)

Figure 30.46 Restart condition issue timing after master transmission

30.11.3 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. 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 the low-level period of SCLn line set in ICBRL 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 ICBRH and the stop condition setup time elapse.
5. Release the SDAn line (low level to high level).
6. Ensure the time set in ICBRL and the bus free time elapse.
7. Clear the BBSY flag to 0 to release the bus mastership.

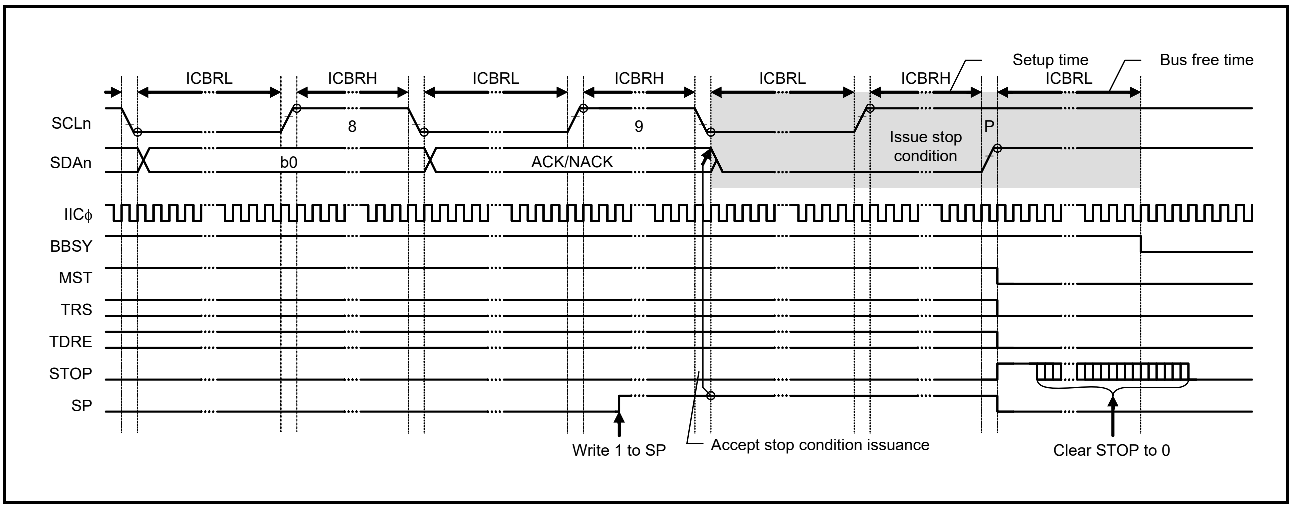


Figure 30.47 Stop condition issue timing using the SP bit

30.12 Bus Hanging

If the clock signals from the master and slave devices are out of synchronization because of noise or other factors, the I^{2}C 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 ICCR1, it is possible to determine whether the IIC or its communicating partner is placing the low level on the SCLn or SDAn lines.

30.12.1 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 ICMR1 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 ICMR2. If both TMOL and TMOH bits are set to 0, the internal counter is disabled.

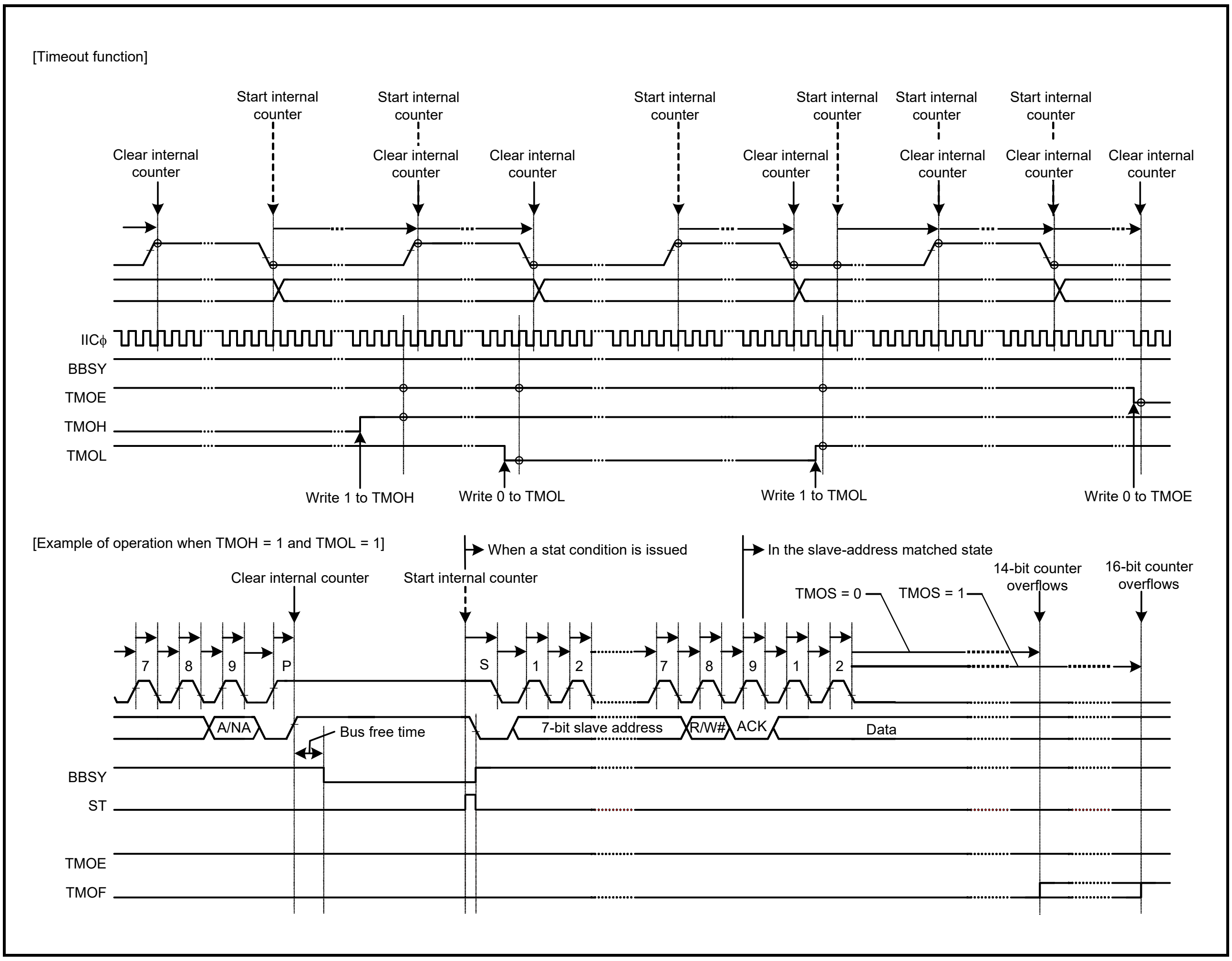

[Timeout function]
Start internal counter
Clear internal counter
IICφ
BBSY
TMOE
TMOH
TMOL
Write 1 to TMOH
Write 0 to TMOL
Write 1 to TMOL
Write 0 to TMOE
[Example of operation when TMOH = 1 and TMOL = 1]
When a stat condition is issued
In the slave-address matched state
Clear internal counter
Start internal counter
TMOS = 0
TMOS = 1
14-bit counter overflows
16-bit counter overflows
7
8
9
P
S
1
2
A/NA
Bus free time
7-bit slave address
R/W#
ACK
Data
ST
TMOF

Figure 30.48 Timeout function using the TMOE, TMOS, TMOH, and TMOL bits

30.12.2 Extra SCL Clock Cycle Output Function

In master mode, this function outputs extra SCL clock cycles to release the SDAn line of the slave device from being held low because the master is out of synchronization with the slave device.

This function 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 SDAI bit in ICCR1. 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 30.49 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

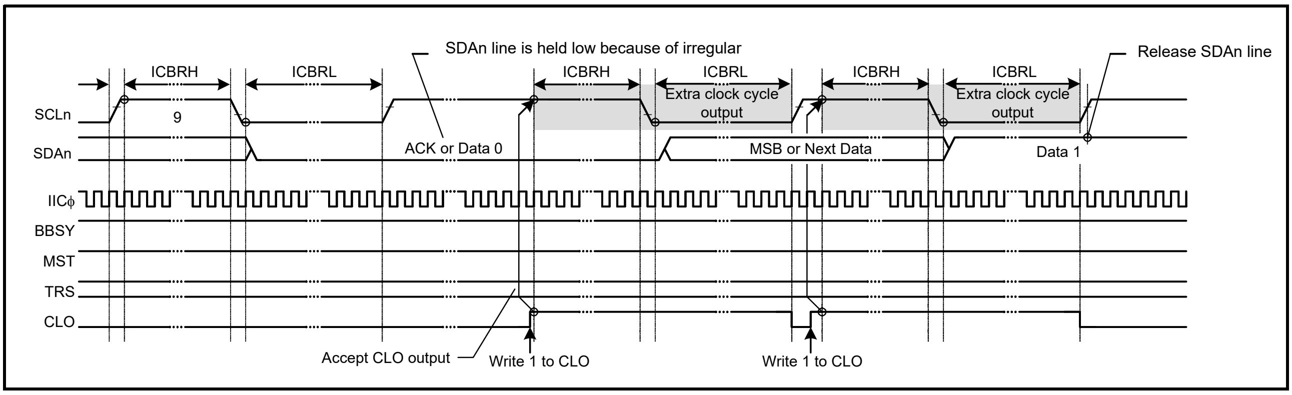


Figure 30.49 Extra SCL clock cycle output function using the CLO bit

30.12.3 IIC Reset and Internal Reset

The IIC has two types of resets:

- IIC reset, which initializes all registers, including the BBSY flag in ICCR2
- 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 resets are valid for release from bus-hung states, because both restore the output state of the SCLn and SDAn pins to the high-impedance state.

Issuing a reset during slave operation might lead to a loss of synchronization between the master device clock and the slave device clock, so avoid this when possible. In addition, monitoring 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 30.15, State of Registers when Issuing each Condition.

30.13 SMBus Operation

The IIC supports data communication conforming to the SMBus Specification (version 2.0). To perform SMBus communication, set the SMBS bit in ICMR3 to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus standard, set the ICMR1.CKS[2:0] bits, 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 ICBRL must be set to a value longer than the data setup time of 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 or 10-bit address format select) in SARUy (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the ICFER.SALE bit to 1 to enable the slave arbitration-lost detection function.

30.13.1 SMBus Timeout Measurement

(1) Measuring slave device timeout

The following period (timeout interval: $T_{LOW: SEXT}$) must be measured for slave devices in SMBus communication:

- From start condition to stop condition.

To measure timeout for slave devices, measure the period from start condition detection to stop condition detection with the GPT using the IIC start condition detection interrupt (STIn) and stop condition detection interrupt (SPIn). The measured timeout period must be within the total clock low-level period [slave device] $T_{LOW: SEXT}$: 25 ms (maximum) of the SMBus standard.

If the time measured with the GPT exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (minimum) of the SMBus standard, the slave device must release the bus by writing 1 to the ICCR1.IICRST to issue an internal reset of the IIC. When an internal reset is issued, the IIC stops driving the bus for the SCLn and SDAn pins 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).

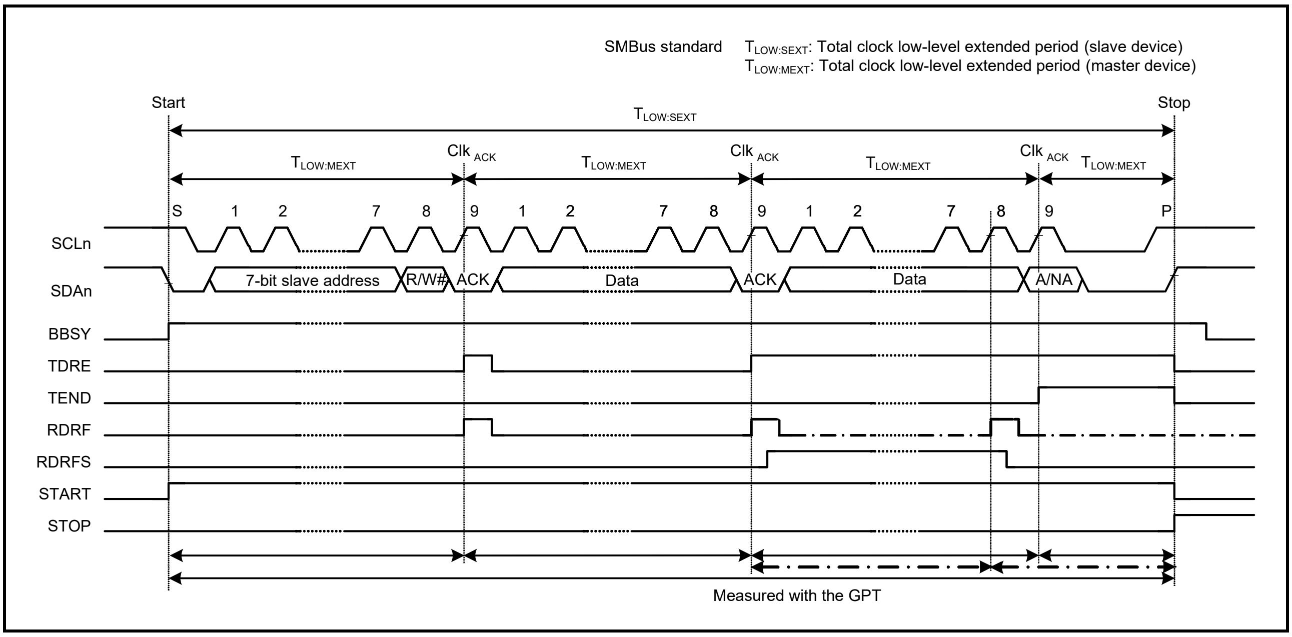

SMBus standard
TLOW:SEXT: Total clock low-level extended period (slave device)
TLOW:MEXT: Total clock low-level extended period (master device)
Start
TLOW:SEXT
Stop
TLOW:MEXT
Clk ACK
TLOW:MEXT
Clk ACK
TLOW:MEXT
Clk ACK
TLOW:MEXT
S 1 2 7 8 9 1 2 7 8 9 1 2 7 8 9 P
SCLn
SDAn
7-bit slave address
R/W#
ACK
Data
ACK
Data
A/NA
BBSY
TDRE
TEND
RDRF
RDRFS
START
STOP
Measured with the GPT

Figure 30.50 SMBus timeout measurement

30.13.2 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 33, 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 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.

30.13.3 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.

For a product using the MCU to operate as an SMBus host (or ARP master), the host address (0001 000b) sent from the slave device must be detected as a slave address, and so the IIC 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.

30.14 Interrupt Sources

The IIC issues five types of interrupt requests:

- Transfer error or event generation (arbitration-lost, NACK detection, timeout detection, start condition detection, and stop condition detection)
- Receive data full
- Transmit data empty
- Transmit end
- Address match during wakeup function.

Table 30.10 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 30.10 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 the actual writing to the module. When an interrupt flag is cleared or masked, read the relevant flag again to check whether clearing or masking 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 ICDRR.

Note 3. When using the IICn_TEI interrupt, clear the ICSR2.TEND flag in the IICn_TEI interrupt handling.
The ICSR2.TEND is automatically set to 0 when transmit data is written to ICDRT 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 1).

Clear or mask each flag during interrupt handling.

30.14.1 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.

30.15 State of Registers when Issuing each Condition

The IIC has two dedicated resets, IIC reset and internal reset. Table 30.11 lists the register states when issuing each condition.

Table 30.11 State of registers when issuing each condition (1 of 2)

Registers		Reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
ICCR1	ICE, IICRST	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
ICWUR2	WUSEN	Reset	Reset	Saved	Saved	Saved
	Others					Saved or Set or Reset
SARL0, SARL1, SARL2 SARU0, SARU1, SARU2		Reset	Reset	Saved	Saved	Saved
ICBRH, ICBRL		Reset	Reset	Saved	Saved	Saved
ICDRT		Reset	Reset	Saved	Saved	Saved

Table 30.11 State of registers when issuing each condition (2 of 2)

Registers	Reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
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

30.16 Event Link Output

IIC0 to IIC1 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.

30.16.1 Interrupt Handling and Event Linking

Each of the IIC interrupt types (see Table 30.10) has an enable bit to control enabling and disabling of the associated interrupt signal. An interrupt request signal is output to the CPU when an interrupt source condition is satisfied while the associated enable bit is set.

The associated event link output signals are sent to other modules as event signals by the ELC when the interrupt source conditions are satisfied, regardless of the interrupt enable bit settings. For details on interrupt sources, see Table 30.10.

30.17 Usage Notes

30.17.1 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.

30.17.2 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 unexpected 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.

31. Controller Area Network (CAN) Module

31.1 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.

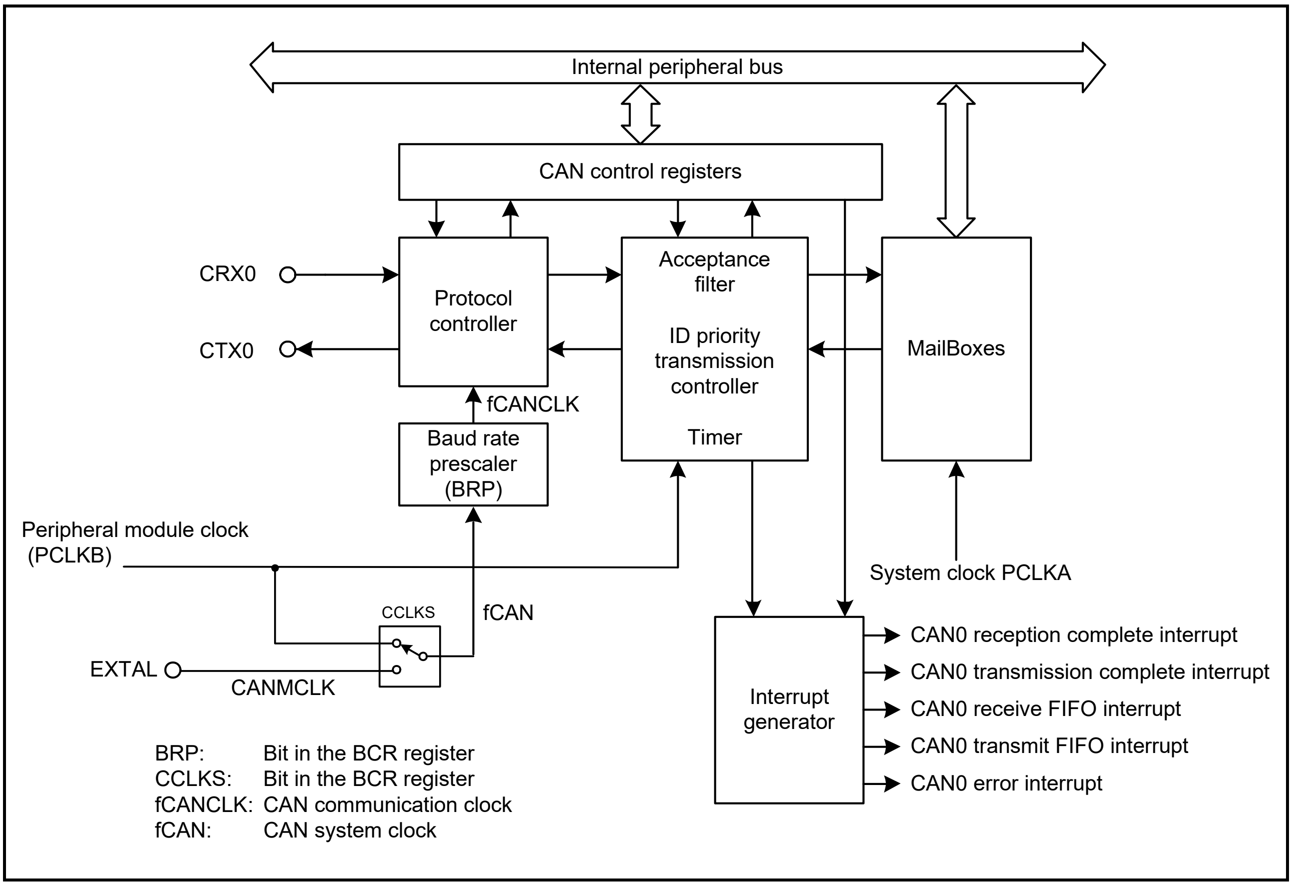
Table 31.1 lists the features of the CAN module and Figure 31.1 shows a block diagram.

Table 31.1 CAN module specifications (1 of 2)

Parameter	Description
Data transfer rate	• 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 transmission or reception • FIFO mode: 24 mailboxes independently configurable for 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 mask 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 for reading of error counters.
Time stamping	• Time stamp function using a 16-bit counter • Reference clock selectable from 1-bit, 2-bit, 4-bit and 8-bit time periods.
Interrupt function	Support for five interrupt sources: • Reception complete • Transmission complete • Receive FIFO • Transmit FIFO • Error interrupts.
CAN sleep mode	• CAN clock stopped to reduce power consumption

Table 31.1 CAN module specifications (2 of 2)

Parameter	Description
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.
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


Internal peripheral bus
CAN control registers
CRX0
CTX0
Protocol controller
Acceptance filter
ID priority transmission controller
Timer
MailBoxes
fCANCLK
Baud rate prescaler (BRP)
Peripheral module clock (PCLKB)
System clock PCLKA
CCLKS
fCAN
EXTAL
CANMCLK
Interrupt generator
CAN0 reception complete interrupt
CAN0 transmission complete interrupt
CAN0 receive FIFO interrupt
CAN0 transmit FIFO interrupt
CAN0 error interrupt

BRP: Bit in the BCR register
CCLKS: Bit in the BCR register
fCANCLK: CAN communication clock
fCAN: CAN system clock

Figure 31.1 CAN module block diagram

The CAN module constitutes the following blocks:

- CAN input and output pins
  CRX0 and CTX0
- Protocol controller
  Handles CAN protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, and error handling
- Mailboxes
  Consists of 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 value.

- Interrupt generator for five types of interrupts:
  - CAN0 reception complete interrupt
  - CAN0 transmission complete interrupt
  - CAN0 receive FIFO interrupt
  - CAN0 transmit FIFO interrupt
  - CAN0 error interrupt.

Table 31.2 lists the CAN module pins. These pins are multiplexed with other signals on the MCU. For details, see section 20, I/O Ports.

Table 31.2 Pin configuration

Pin name	I/O	Function
CRX0	Input	Data receive pin
CTX0	Output	Data transmit pin

31.2 Register Descriptions

31.2.1 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 IDs and extended IDs. In normal mailbox mode, use the associated IDE bit to differentiate standard or extended IDs. In FIFO mailbox mode, the associated IDE bit are used for mailboxes 0 to 23, the IDE bits in FIDCR0 and FIDCR1 are used for the receive FIFO, and the IDE bit in 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

Bit	Symbol	Bit name	Description	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
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 CAN sleep mode 1: Enter CAN 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 at the end of bus-off state 1 1: Entry to CAN halt mode during bus-off recovery period through software request.	R/W
b13	RBOC	Forcible Return from Bus-Off*2	0: No return occurred 1: Forced return from bus-off.*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 is automatically set 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 only to 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 31.3 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. All mailboxes, including the receive FIFO are set to either overwrite mode or overrun mode.

When this bit is 0, all mailboxes are set to overwrite mode. Any new message received overwrites the pre-existing message.

When this 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 and is discarded.

TPM bit (Transmission Priority Mode Select)

The TPM bit specifies the priority when transmitting messages.

The 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 complies with the CAN bus arbitration rule, as defined in the ISO11898-1 CAN specification. In ID priority transmit mode, mailboxes 0 to 31 (in normal mailbox mode), 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 this bit is set to 1, the TSR register is set to 0000h. This 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-bit, 2-bit, 4-bit, 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 by the SLPM bit. For details, see section 31.3, Modes of Operation. When the CAN module enters CAN halt mode based on the BOM[1:0] bits 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 31.3, 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 CAN 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 the CTLR register are set to 10b to enter the 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 the 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 the 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 TECR and RECR are set to 00h. However, 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, a bus-off recovery interrupt request is generated.

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 (Forcible 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. This 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 the STR register is set to 0, indicating that the CAN module is not in bus-off state. The other registers remain unchanged even 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 31.3 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 are 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 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.

31.2.2 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	These bits are 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 31.4, 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 1,023), 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.

31.2.3 Mask Register k (MKRk) (k = 0 to 7)

Address(es): CAN0.MKR0 4005 0400h to CAN0.MKR7 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 31.6, Acceptance Filtering and Masking Functions.

Write to MKR0 to MKR7 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. These bits 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. These bits are used to receive both standard ID and extended ID messages.

When an SID[10:0] bits 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 bit.

31.2.4 FIFO Received ID Compare Registers 0 and 1 (FIDCR0 and FIDCR1)

Address(es): CAN0.FIDCR0 4005 0420h, CAN0.FIDCR1 4005 0424h

	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	Extended ID of the data and remote frames	R/W

Bit	Symbol	Bit name	Description	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 bits in mailbox 28 to mailbox 31 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 31.6, 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. These bits are used to receive extended ID messages.

SID[10:0] bits (Standard ID)

The SID[10:0] bits set the standard ID of data frames and remote frames. These bits are used to receive both standard ID and extended ID messages.

RTR bit (Remote Transmission Request)

The RTR bit sets the specified frame format of data frames or remote frames:

- When the RTR bits in both FIDCR0 and FIDCR1 registers are set to 0, only data frames can be received
- When the RTR bits in both FIDCR0 and FIDCR1 registers are set to 1, only remote frames can be received
- When the RTR bits in both FIDCR0 and FIDCR1 registers are set to different values, both data frames and remote frames can be 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 CTLR are 10b (mixed ID mode):

- When the IDE bits in both the FIDCR0 and FIDCR1 registers are set to 0, only standard ID frames can be received
- When the IDE bits in both the FIDCR0 and FIDCR1 registers are set to 1, only extended ID frames can be received
- When the IDE bits in both the FIDCR0 and FIDCR1 registers are set to different values, both standard ID and extended ID frames can be received.

31.2.5 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 the MKIVLR register corresponds to mailbox 0 (MB0) and bit [31] corresponds to mailbox 31 (MB31).*1

Note 1. Set bits [31:24] to 0 in FIFO mailbox mode.

When an MBn bit is set to 1, the corresponding 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 exactly matches the mailbox ID. Write to the MKIVLR register in CAN reset mode or CAN halt mode.

31.2.6 Mailbox Register j (MBj_ID, MBj_DL, MBj_Dm, MBj_TS) (j = 0 to 31, m = 0 to 7)

Table 31.4 lists the CAN0 mailbox memory mapping, and Table 31.5 lists the CAN data frame configuration. The value of the CAN0 mailbox is undefined after reset.

Write to the MBj_ID, MBj_DL, MBj_Dm and MBj_TS registers only when the related MCTL_TXj or MCTL_RXj (j = 0 to 31) register is 00h and the associated mailbox does not process an abort request. See Table 31.4 for details on register addresses.

Table 31.4 CAN0 mailbox memory mapping (1 of 2)

Address for CAN0	Mapped message content
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 31.4 CAN0 mailbox memory mapping (2 of 2)

Address for CAN0	Mapped message content
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 31.5 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 IDFM[1:0] bits in the CTLR register are 10b (mixed ID mode). When the 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. 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

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA6							
Value after reset:	x	x	x	x	x	x	x	x

Address(es): CAN0.MB0_D7 4005 020Dh to CAN0.MB31_D7 4005 03FDh

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA7							
Value after reset:	x	x	x	x	x	x	x	x

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 bytes, 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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSH[7:0]								TSL[7:0]							
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
b7 to b0	TSL[7:0]	Time Stamp Lower Byte	Bits TSH[7:0] and TSL[7:0] 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		R/W

EID[17:0] bits (Extended ID)

The EID[17:0] bits set the extended ID of data and remote frames. These bits 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. These bits 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 only transmits frames with the 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 specified in the RTR bit in the relevant 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 are 10b (mixed ID mode):

- The receive mailbox receives only 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 bit in the FIDCR0 and FIDCR1 registers
- The transmit FIFO mailbox transmits messages with the standard ID or extended ID settings 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, this field specifies the requested data length.

When a data frame is received, the received data length is stored in this field. When a remote frame is received, this field stores the requested data length.

31.2.7 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: Interrupt disabled 1: Interrupt enabled. 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 a mailbox with the same number. These bits enable or disable transmission and reception complete interrupts for the associated mailboxes as follows:

- 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) register is 00h and the associated mailbox does not process a transmission or reception abort request.

31.2.8 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: Interrupt disabled 1: Interrupt enabled. Bit [23] corresponds to mailbox 23 (MB23) and bit [0] corresponds to mailbox 0 (MB0).	R/W
b24	MB24	Transmit FIFO Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled.	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: Interrupt disabled 1: Interrupt enabled.	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 on 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 of the same number. These bits enable or disable transmission and reception complete interrupts for the associated mailboxes:

- Bit [0] in MIER_FIFO is associated with mailbox 0 (MB0)
- Bit [23] in MIER_FIFO is associated with mailbox 23 (MB23).

The MB24, MB25, MB28 and MB29 bits specify whether transmit and receive FIFO interrupts are enabled or disabled and the timing when interrupt requests are generated.

Write to the MIER_FIFO register only when the related MCTL_TXj or MCTL_RXj (j = 0 to 31) register is 00h and the associated mailbox does not process a transmission or reception abort request. In addition, change the bits in MIER_FIFO for the related FIFO only when all the following conditions are true:

- The TFE bit in TFCR is 0 and the TFEST bit is 1
- The RFE bit in RFCR is 0 and the RFEST flag in RFCR is 1.

31.2.9 Message Control Registers 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_TX0 4005 0820h to CAN0.MCTL_TX31 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 transmission is 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 flags 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 the SENDDATA, TRMACTIVE, and TRMABT flags to 0 simultaneously.

The MCTL_TXj register sets the mailbox j to transmit mode or receive mode. In transmit mode, MCTL_TXj also controls and indicates the status of transmission. Do not access the MCTL_TXj register if the mailbox j is in receive mode. Only write to the MCTL_TXj register in CAN operation mode or CAN halt mode. Do not use the MCTL_TX24 through 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. This flag is set to 0 through a software write.

To set this flag to 0, first set the TRMREQ bit to 0. The SENTDATA flag and the TRMREQ bit 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 loses the 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 SENTDATA flag is set to 1 and 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 due to 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 flag is set to 1.

RECREQ bit (Receive Mailbox Request)

The RECREQ bit selects the receive modes listed in Table 31.10.

When the RECREQ bit is set to 1, the associated mailbox is configured for reception of a data frame or 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 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 the 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 flags to 0 before changing to reception.

Note: MCTL_TXj.RECREQ is the mirror bit of MCTL_RXj.RECREQ.

TRMREQ bit (Transmit Mailbox Request)

The TRMREQ bit selects the transmit modes listed in Table 31.10.

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 flags to 0 before changing to transmission.

Note: MCTL_TXj.TRMREQ is the mirror bit of MCTL_RXj.TRMREQ.

31.2.10 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_RX0 4005 0820h to CAN0.MCTL_RX31 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 NEWDATA flag 1: New message is being stored or was stored in 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 in this register, write 1 to the NEWDATA and MSGLOST flags 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 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 status of reception.

Do not access the MCTL_RXj register if mailbox j is in transmit mode. Only write to the MCTL_RXj register 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. Always set this bit to 1 simultaneously 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 while 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 through a software write during the 5 PCLKB cycles following the 6th 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)

The RECREQ bit selects the receive modes listed in Table 31.10.

When the RECREQ bit is set to 1, the associated mailbox is configured for reception of a data frame 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 the 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 the acceptance filter processing
  - If no mailbox is specified to receive the message, after the 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, and then set the SENTDATA and TRMABT flags to 0 before changing to reception.

Note: MCTL_RXj.RECREQ is the mirror bit of MCTL_TXj.REQREQ.

TRMREQ bit (Transmit Mailbox Request)

The TRMREQ bit selects the transmit modes listed in Table 31.10.

When the TRMREQ bit is set to 1, the associated mailbox is configured for transmission of a data frame or a remote frame.

If the TRMREQ bit changes from 1 to 0 to cancel the associated transmission request, either the TRMABT flag or the 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, and then set the NEWDATA and MSGLOST flags to 0 before changing to transmission.

Note: MCTL_RXj.TRMREQ is the mirror bit of MCTL_TXj.TRMREQ.

31.2.11 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

Bit	Symbol	Bit name	Description	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 has no buffer warning 1: Receive FIFO has 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 flag setting.

Do not set this bit to 1 in normal mailbox mode (MBM = 0). Due to hardware protection, the RFE bit is not set to 0 through a software write during the following period:

- Hardware protection is started from the 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. This flag 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 receive a message, the RFMLF flag cannot be set to 0 (no receive FIFO message 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 buffer warning) when the number of unread messages in the receive FIFO is 3. The RFWST flag is 0 (no receive FIFO 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 31.2 shows the receive FIFO mailbox operation.

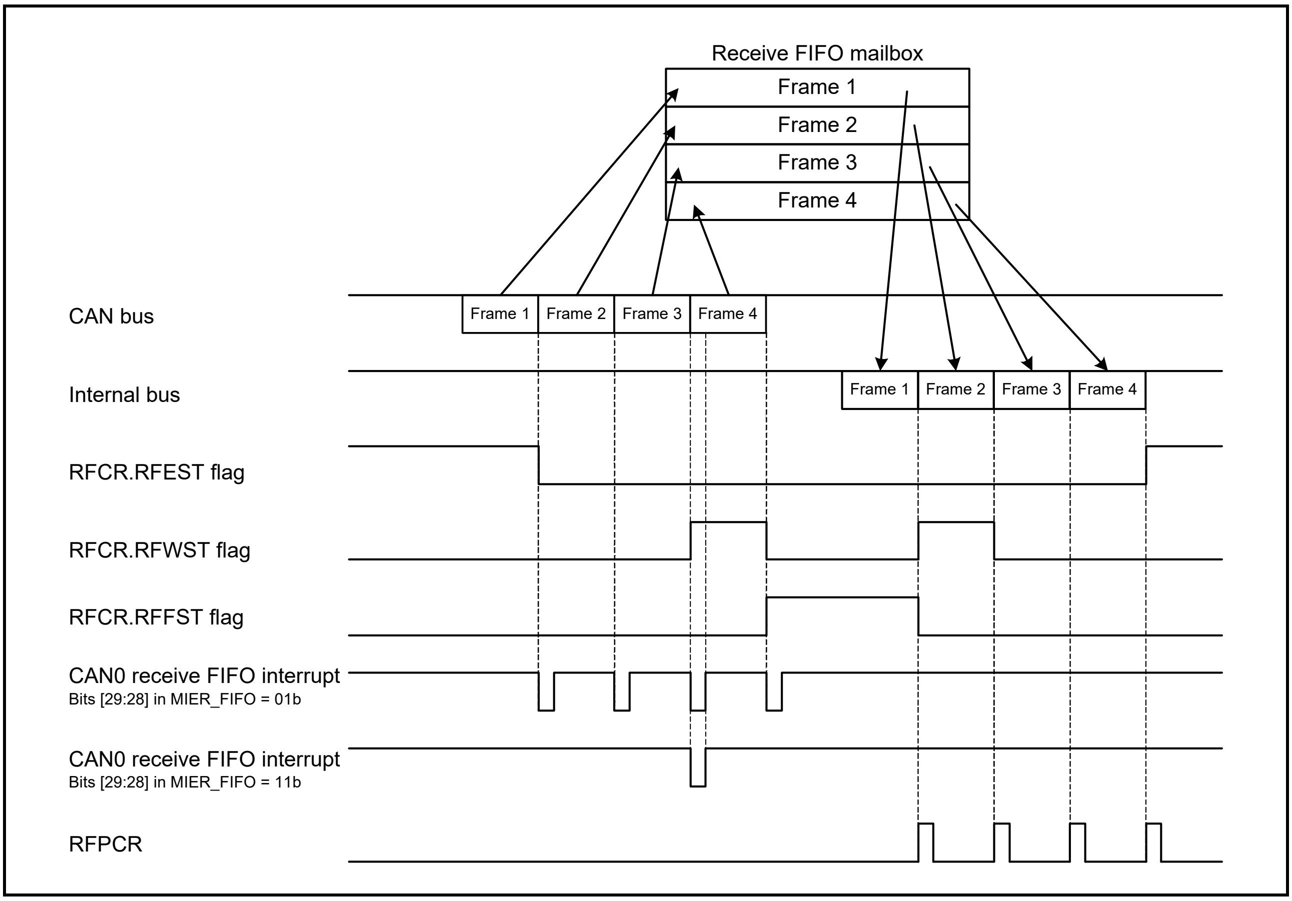

Receive FIFO mailbox
Frame 1
Frame 2
Frame 3
Frame 4
CAN bus
Frame 1
Frame 2
Frame 3
Frame 4
Internal bus
Frame 1
Frame 2
Frame 3
Frame 4
RFCR.RFEST flag
RFCR.RFWST flag
RFCR.RFFST flag
CAN0 receive FIFO interrupt
Bits [29:28] in MIER_FIFO = 01b
CAN0 receive FIFO interrupt
Bits [29:28] in MIER_FIFO = 11b
RFPCR

Figure 31.2 Receive FIFO mailbox operation with bits [29:28] in MIER_FIFO = 01b or 11b

31.2.12 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 the RFPCR register when the RFE bit in RFCR is 0 (receive FIFO disabled).

Both the CAN and CPU pointers are incremented when a new message is received and the RFFST flag is 1 (receive FIFO is full) in overwrite mode. When the RFMLF flag is 1 in this condition, the CPU pointer cannot be incremented on a software write to RFPCR.

31.2.13 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 is not full 1: Transmit FIFO is 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 TFCR in CAN operation mode or CAN halt mode.

TFE bit (Transmit FIFO Enable)

When the TFE bit is set to 1, the transmit FIFO is enabled.

When the TFE bit is set to 0, the transmit FIFO becomes empty (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
- Following the 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 (MBM bit in CTLR = 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 the TFE bit is set to 0 and transmission is aborted or completed.

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 aborts. 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 31.3 shows the transmit FIFO mailbox operation.

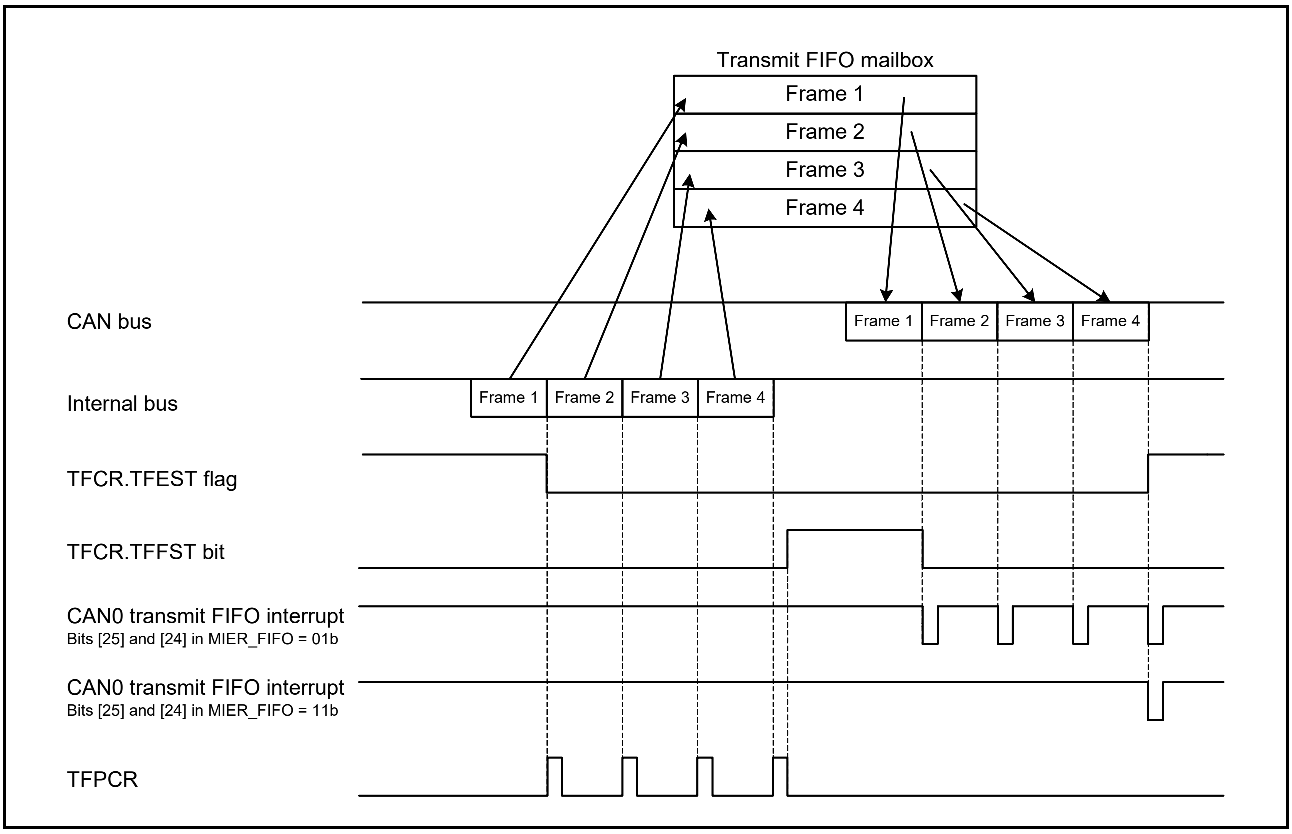

Transmit FIFO mailbox
Frame 1
Frame 2
Frame 3
Frame 4
CAN bus
Frame 1
Frame 2
Frame 3
Frame 4
Internal bus
Frame 1
Frame 2
Frame 3
Frame 4
TFCR.TFEST flag
TFCR.TFFST bit
CAN0 transmit FIFO interrupt
Bits [25] and [24] in MIER_FIFO = 01b
CAN0 transmit FIFO interrupt
Bits [25] and [24] in MIER_FIFO = 11b
TFPCR

Figure 31.3 Transmit FIFO mailbox operation when MIER_FIFO[25:24] = 01b or 11b

31.2.14 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 TFE bit in TFCR is 0 (transmit FIFO disabled).

31.2.15 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 = 1 1: 1 or more mailboxes with NEWDATA = 1.	R
b1	SDST	SENTDATA Status Flag	0: No mailbox with SENTDATA = 1 1: 1 or more mailboxes with SENTDATA = 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 = 1 1: One or more mailboxes with MSGLOST = 1.	R
b5	FMLST	FIFO Mailbox Message Lost Status Flag	0: RFMLF = 0 1: RFMLF = 1.	R
b6	TABST	Transmission Abort Status Flag	0: No mailbox with TRMABT = 1 1: 1 or more mailboxes with TRMABT = 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
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 flag in the MCTL_RXj (j = 0 to 31) registers is 1, regardless of the value of the MIER or MIER_FIFO registers. The NDST flag is set to 0 when all the NEWDATA flags are 0.

SDST flag (SENTDATA Status Flag)

The SDST flag is set to 1 when at least one SENTDATA flag in the MCTL_TXj (j = 0 to 31) registers is 1, regardless of the value of the MIER or MIER_FIFO registers. The SDST flag is set to 0 when all the SENTDATA flags 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 flag 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 flags are 0.

FMLST flag (FIFO Mailbox Message Lost Status Flag)

The FMLST flag is set to 1 when the RFMLF flag in RFCR is 1, regardless of the value of MIER_FIFO. The FMLST flag is set to 0 when the RFMLF flag is 0.

TABST flag (Transmission Abort Status Flag)

The TABST flag is set to 1 when at least one TRMABT flag in the MCTL_TXj (j = 0 to 31) registers is 1, regardless of the value of the MIER or MIER_FIFO registers. The TABST flag is set to 0 when all TRMABT flags 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 the EIER register. The EST flag is set to 0 when no error is detected by EIFR.

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. Even when the state changes 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. Even when the state is changed 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 in the TECR or RECR registers exceeds 127 and the CAN module is in the error-passive state ($128 \leq TEC < 256$ or $128 \leq REC < 256$). 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 in the TECR register exceeds 255 and the CAN module is in the bus-off state ($TEC \geq 256$). 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.

31.2.16 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 flag in the MCTL_RXj (j = 0 to 31) registers for the normal mailbox, and the RFEST flag in the RFCR register.

When the MBSM[1:0] bits are 01b, transmit mailbox search mode is selected. In this mode, the search target is the SENTDATA flag in the MCTL_TXj register.

When the MBSM[1:0] bits are 10b, message lost search mode is selected. In this mode, the search targets are the MSGLOST flag in the MCTL_RXj register for the normal mailbox, and the RFMLF flag in the RFCR register.

When the MBSM[1:0] bits are 11b, channel search mode is selected. In this mode, the search target is the CSSR register. See section 31.2.18, Channel Search Support Register (CSSR).

31.2.17 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	R
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 mailbox number found. 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 flag is set to 0 for a mailbox output by

MBNST[4:0]

- When the respective NEWDATA, SENTDATA or MSGLOST flag 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 31.6 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 the mailboxes. For example, in transmit mailbox search mode, the SEST bit is set to 1 when no SENTDATA flag is 1 for any mailbox. The SEST bit is set to 0 when at least one SENTDATA flag is 1. When the SEST bit is 1, the value of the MBNST[4:0] bits is undefined.

Table 31.6 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 flag for the normal mailboxes is set to 1 (no 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 flag for the normal mailboxes is set to 1 (no message is overwritten or overrun) and the RFCR.RFMLF flag is set to 1 (receive FIFO message was lost) in the receive FIFO
11b		Mailbox 28 is not output

31.2.18 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 it 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 31.4 shows the write and read operations of the CSSR and MSSR registers.

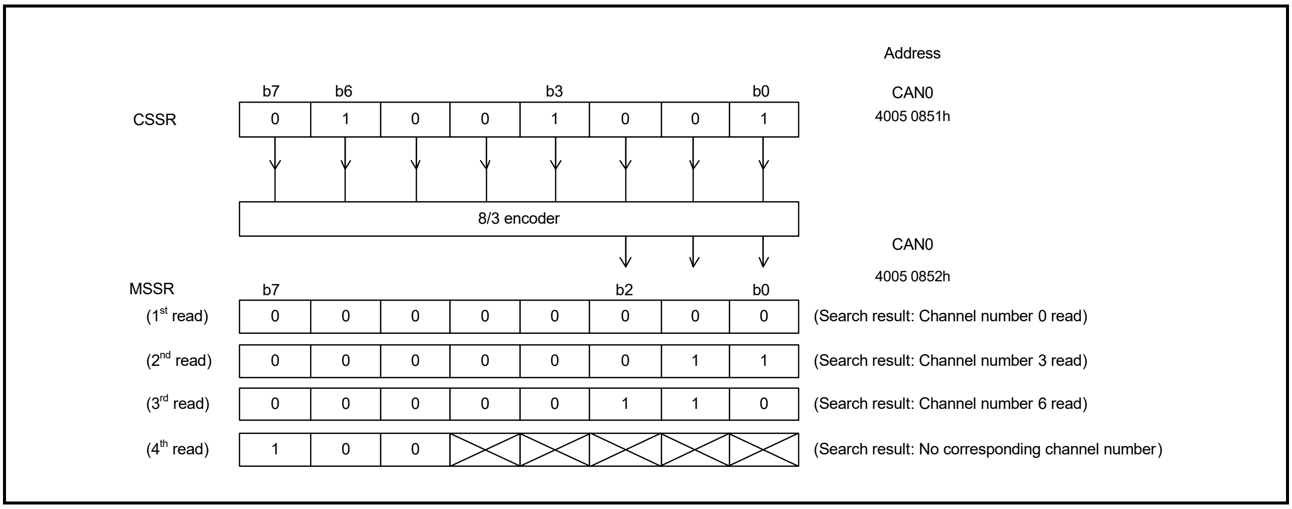

Address
CSSR
b7 b6 b3 b0
0 1 0 0 1 0 0 1
CAN0
4005 0851h
8/3 encoder
CAN0
4005 0852h
MSSR
b7 b2 b0
(1st read) 0 0 0 0 0 0 0 0 (Search result: Channel number 0 read)
(2nd read) 0 0 0 0 0 0 1 1 (Search result: Channel number 3 read)
(3rd read) 0 0 0 0 0 1 1 0 (Search result: Channel number 6 read)
(4th read) 1 0 0 (Search result: No corresponding channel number)

Figure 31.4 Write and read operations of 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.

31.2.19 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.

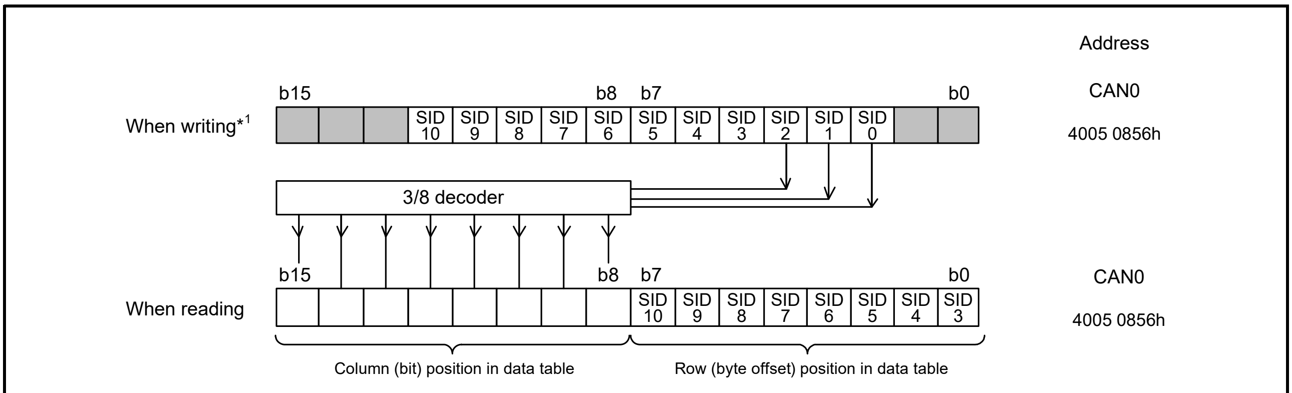
The acceptance filter support unit (ASU) can be used for data table (8 bits × 256) search. In the data table, all standard IDs that you create are set to be valid or invalid in bit units. When the AFSR register is written with data in 16-bit units including the SID[10:0] bit in the MBj_ID (j = 0 to 31) register, 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 IDs 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 31.5 shows the write and read operation in the AFSR register.


Address
When writing*1
b15
b8
b7
b0
CAN0
4005 0856h
SID 10
SID 9
SID 8
SID 7
SID 6
SID 5
SID 4
SID 3
SID 2
SID 1
SID 0
3/8 decoder
b15
b8
b7
b0
CAN0
4005 0856h
When reading
SID 10
SID 9
SID 8
SID 7
SID 6
SID 5
SID 4
SID 3
Column (bit) position in data table
Row (byte offset) position in data table

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 31.5 Write and read operations in the AFSR register

31.2.20 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 error interrupt for each error interrupt source. 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 BEIF flag in the EIFR register is 1. When the BEIE bit is 1, an error interrupt request is generated if the BEIF flag 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 EWIF flag in the EIFR register is 1. When the

EWIE bit is 1, an error interrupt request is generated if the EWIF flag 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 EPIF flag in the EIFR register is 1. When the EPIE bit is 1, an error interrupt request is generated if the EPIF flag 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 BOEIF flag in the EIFR register is 1. When the BOEIE bit is 1, an error interrupt request is generated if the BOEIF flag is set to 1.

BORIE bit (Bus-Off Recovery Interrupt Enable)

When the BORIE bit is 0, no error interrupt request is generated even if the BORIF flag in the EIFR register is 1. When the BORIE bit is set to 1, an error interrupt request is generated if the BORIF flag is set to 1.

ORIE bit (Overrun Interrupt Enable)

When the ORIE bit is 0, no error interrupt request is generated even if the ORIF flag in the EIFR register is 1. When the ORIE bit is 1, an error interrupt request is generated if the ORIF flag 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 OLIF flag in the EIFR register is 1. When the OLIE bit is 1, an error interrupt request is generated if the OLIF flag 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 BLIF flag in the EIFR register is 1. When the BLIE bit is 1, an error interrupt request is generated if the BLIF flag is set to 1.

31.2.21 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 an EIFR flag occurs, the associated bit in EIFR is set to 1, regardless of the setting of EIER.

Clear the bits to 0 through a software write. If a bit is set to 1 at the same time that 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 on 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. This flag is set to 1 only when the REC or TEC value initially exceeds 95. If software writes 0 to this flag while the REC or TEC value remains greater than 95, the EWIF flag is not set to 1 until the REC or TEC value goes below 95, and 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 REC or TEC value exceeds 127. This flag is set to 1 only when the REC or TEC value initially exceeds 127. If software writes 0 to this flag while the REC or TEC value remains greater than 127, the EPIF flag is not set to 1 until the REC or TEC value goes below 127, and 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, while the TEC value exceeds 255. The BOEIF flag is also set to 1 when the BOM[1:0] bits in CTLR 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 BOM[1:0] bits in CTLR are 00b
- When the BOM[1:0] bits in CTLR are 10b
- When the BOM[1:0] bits in CTLR are 11b.

The BORIF flag is not set to 1 if the CAN module recovers from the bus-off state in the following conditions:

- When the CANM[1:0] bits in CTLR are set to 01b or 11b (CAN reset mode)
- When the RBOC bit in CTLR is set to 1 (forced return from bus-off)
- When the BOM[1:0] bits in CTLR are set to 01b
- When the BOM[1:0] bits in CTLR are set to 11b and the CANM[1:0] bits in CTLR are set to 10b (CAN halt mode) before normal recovery occurs.

Table 31.7 shows the behavior of the BOEIF and BORIF flags for each CTLR.BOM[1:0] bit setting.

Table 31.7 Behavior of BOEIF and BORIF flags according to CTLR.BOM[1:0] bit setting

BOM[1:0] bits	BOEIF flag	BORIF flag
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 flag is not set to 1 in overwrite mode. In this mode, a reception complete interrupt request is generated if an overwrite condition occurs and the ORIF flag is not set to 1.

In overrun mode with normal mailbox mode, if an overrun occurs in any of the mailboxes 0 through 31, the ORIF flag is set to 1. In overrun mode with FIFO mailbox mode, if an overrun occurs in any of the mailboxes 0 through 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 performs transmission or reception.

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 flag is set to 1, the 32 consecutive dominant bits are detected again in either of the following conditions:

- Recessive bits are detected after the BLIF flag changes from 0 to 1
- The CAN module enters CAN reset mode or CAN halt mode and then enters CAN operation mode again, after the BLIF flag changes from 0 to 1.

31.2.22 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 or decrement conditions of the receive error counter. The value of the RECR register in the bus-off state is undefined.

31.2.23 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.

31.2.24 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
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 the related bits are set to 1.

The ECSR register can be used to monitor whether an error occurred on the CAN bus. See the CAN specification (ISO11898-1) for the conditions when each error occurs.

Clear all the bits except for the EDPM bit to 0 through a software write. If an ECSR bit is set to 1 by the CAN module at the same time that software writes 0 to it, the bit is set to 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 ECSR. 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.

31.2.25 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: 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 CTLR. 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 in the TSL[7:0] and TSH[7:0] bits in the MBj_TS register when a received message is stored in a receive mailbox.

31.2.26 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 the 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 the 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 31.6 shows the connection when listen-only mode is selected.

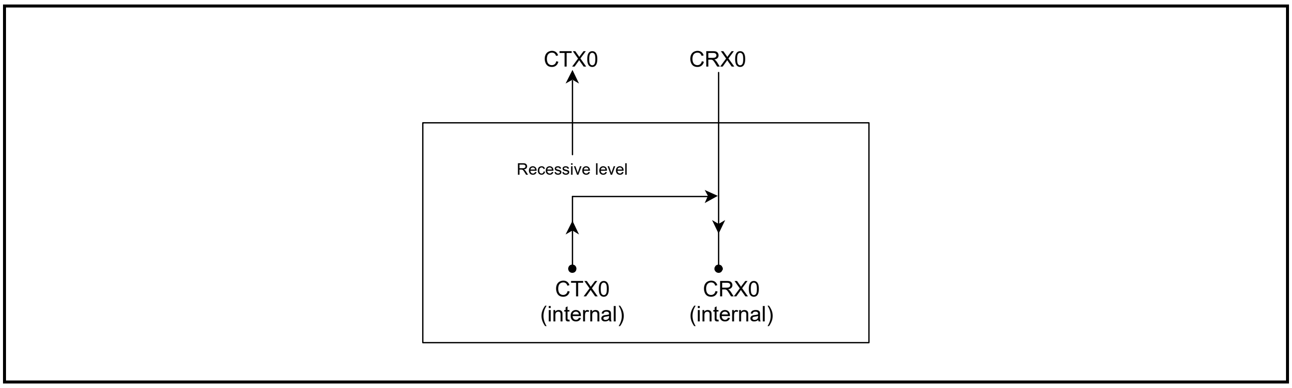


Figure 31.6 Connection when listen-only mode is selected

(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 31.7 shows the connection when self-test mode 0 is selected.

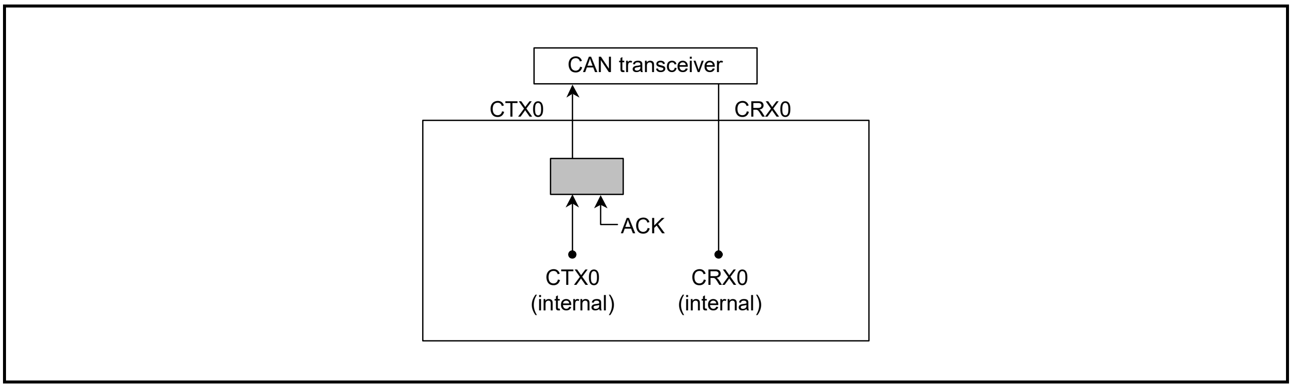


Figure 31.7 Connection when self-test mode 0 is selected

(3) Self-test mode 1 (internal loopback)

Self-test mode 1 is provided for self-test functions. In this mode, 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 an internal feedback 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 31.8 shows the connection when self-test mode 1 is selected.

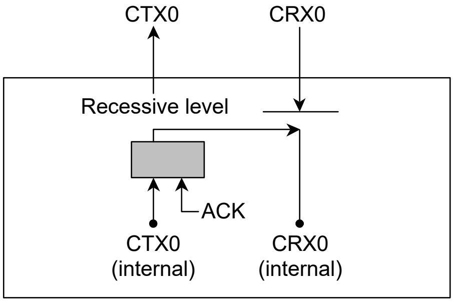

CTX0
CRX0
Recessive level
ACK
CTX0
(internal)
CRX0
(internal)

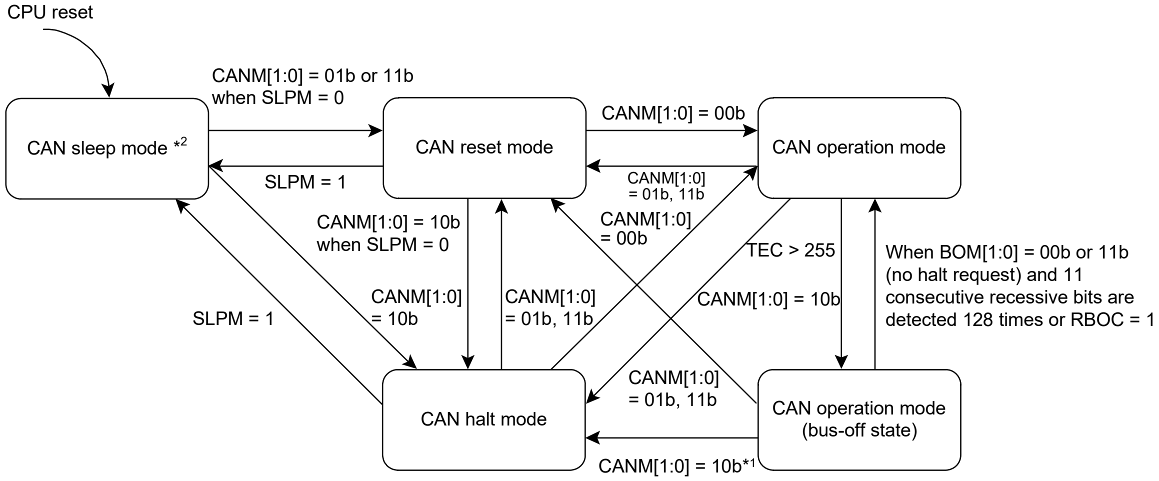
Figure 31.8 Connection when self-test mode 1 is selected

31.3 Modes of Operation

The CAN module operation modes include:

- CAN reset mode
- CAN halt mode
- CAN operation mode
- CAN sleep mode.

Figure 31.9 shows the transition between different modes of operation.


CPU reset
CAN sleep mode *2
CAN reset mode
CAN operation mode
CAN halt mode
CAN operation mode
(bus-off state)
CANM[1:0] = 01b or 11b
when SLPM = 0
SLPM = 1
CANM[1:0] = 00b
CANM[1:0]
= 01b, 11b
CANM[1:0] = 10b
when SLPM = 0
CANM[1:0]
= 00b
TEC > 255
When BOM[1:0] = 00b or 11b
(no halt request) and 11
consecutive recessive bits are
detected 128 times or RBOC = 1
SLPM = 1
CANM[1:0]
= 10b
CANM[1:0]
= 01b, 11b
CANM[1:0] = 10b
CANM[1:0]
= 01b, 11b
CANM[1:0] = 10b*1

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 31.9 Transition between different modes of operation

31.3.1 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 flag is then set to 1. Do not change the

CTLR.CANM[1:0] bits until the RSTST flag is set to 1. Set the BCR register before exiting CAN reset mode to enter any other modes.

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.

31.3.2 CAN Halt Mode

CAN halt mode is used for mailbox configuration and test mode setting. When the CANM[1:0] bits in the CTLR register are set to 10b, CAN halt mode is selected and the HLTST bit in the STR register is set to 1. Do not change the CANM[1:0] bits in the CTLR register until the HLTST bit is 1. See Table 31.8 for the state transition conditions when transmitting or receiving.

All registers except for the RSTST, HLTST, and SLPST bits in the STR register remain unchanged when the CAN enters CAN halt mode.

Do not change the CTLR register (except for the CANM[1:0] and SLPM bits) and the 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 31.8 Operation in CAN reset mode and CAN halt mode

Operation mode	Receiver	Transmitter	Bus-off
CAN reset mode (forced 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 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 flag 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.

31.3.3 CAN Sleep Mode

CAN sleep mode reduces power 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 the CTLR register is set to 1, the CAN module enters CAN sleep mode and the SLPST bit in STR is 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.

31.3.4 CAN Operation Mode (Excluding Bus-Off State)

CAN operation mode is used for CAN communication. When the CANM[1:0] bits in the CTLR register are set to 00b, the CAN module enters CAN operation mode. The RSTST and HLTST bits in STR are set to 0. Do not change the value of the CANM[1:0] bits until the RSTST and HLTST bits are set to 0.

If 11 consecutive recessive bits are detected after entering CAN operation mode, the CAN module is in the following states:

- 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 may be 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 31.10 shows the sub-modes of CAN operation mode.

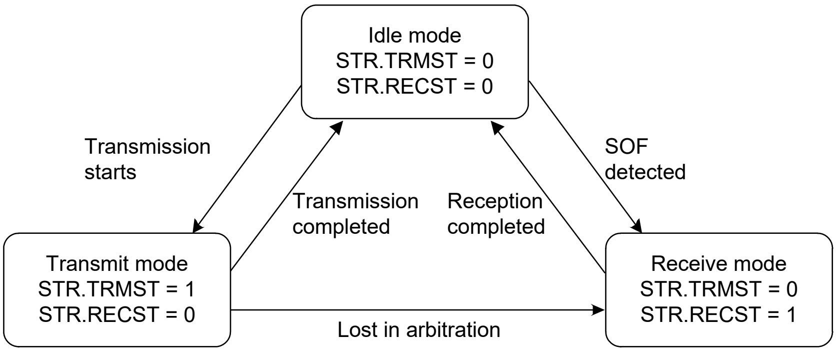


Figure 31.10 Sub-modes of CAN operation mode

31.3.5 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-related registers remain unchanged, except for those in STR, EIFR, RECR, TECR, and TSR.

(1) When BOM[1:0] bits in CTLR are 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 BORIF flag in the EIFR register is set to 1 (bus-off recovery detected).

(2) When RBOC bit in CTLR is set to 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 flag is not set to 1.

(3) When BOM[1:0] bits are 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 flag is not set to 1.

(4) When BOM[1:0] bits are 10b (automatic transition to CAN halt mode at bus-off end)

The CAN module enters CAN halt mode when it completes the recovery from bus-off. The BORIF flag is set to 1.

(5) When BOM[1:0] bits are 11b (automatic transition to CAN halt mode through software) and CANM[1:0] bits in CTLR are set to 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 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.

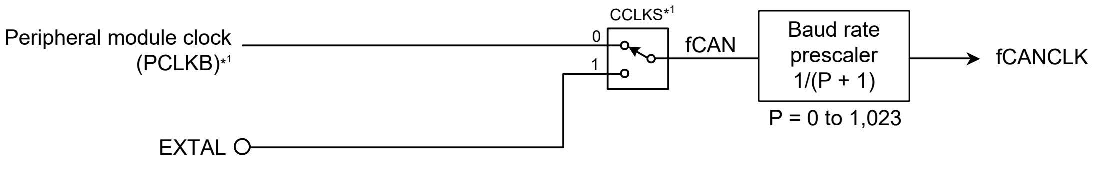
31.4 Data Transfer Rate Configuration

This section describes how to configure the data transfer rate.

31.4.1 Clock Setting

The CAN module has a CAN clock generator. The CAN clock can be set by the CCLKS bit and the BRP[9:0] bits in the BCR register.

Figure 31.11 shows a block diagram of the CAN clock generator.



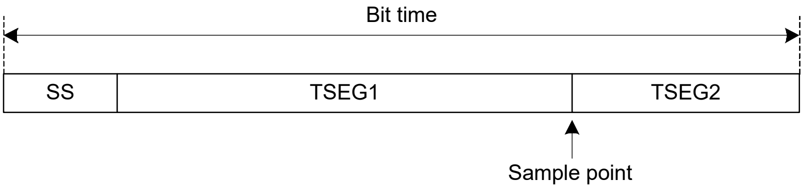
CCLKS: Bit in the BCR register
fCAN: CAN system clock
P: Value selected by BRP[9:0] bits in BCR (P = 0 to 1,023)
fCANCLK: CAN communication clock (fCANCLK = fCAN/(P + 1))

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 31.11 Block diagram of CAN clock generator

31.4.2 Bit Time Setting

The bit time setting consists of three segments as Figure 31.12 shows.



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 31.12 Bit timing

31.4.3 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 Tq count for 1 bit time.

$$\textbf{Data transfer rate [bps]} = \frac{\textbf{fCAN}}{\textbf{Baud rate prescaler division value}^{*1} \times \textbf{Tq count for 1 bit time}} = \frac{\textbf{fCANCLK}}{\textbf{Tq count for 1 bit time}}$$

Note 1. Division value of baud rate prescaler = P + 1 (P: 0 to 1,023), where P is the BRP[9:0] setting in the BCR register.

Table 31.9 lists data transfer rate examples.

Table 31.9 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

31.5 Mailbox and Mask Register Structure

Figure 31.13 shows the structure of the 32 mailbox registers MBj_ID, MBj_DL, MBj_Dm, and MBj_TS.

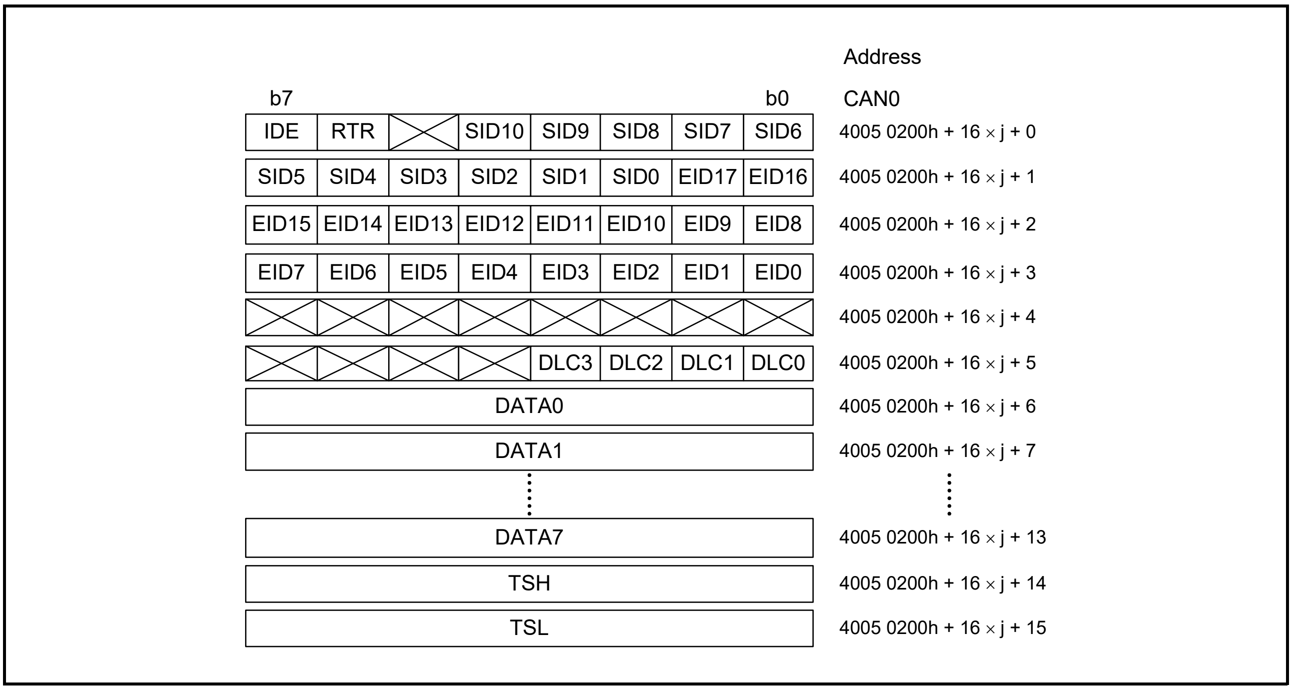

Address
b7
b0
CAN0
IDE RTR SID10 SID9 SID8 SID7 SID6
4005 0200h + 16 × j + 0
SID5 SID4 SID3 SID2 SID1 SID0 EID17 EID16
4005 0200h + 16 × j + 1
EID15 EID14 EID13 EID12 EID11 EID10 EID9 EID8
4005 0200h + 16 × j + 2
EID7 EID6 EID5 EID4 EID3 EID2 EID1 EID0
4005 0200h + 16 × j + 3
4005 0200h + 16 × j + 4
DLC3 DLC2 DLC1 DLC0
4005 0200h + 16 × j + 5
DATA0
4005 0200h + 16 × j + 6
DATA1
4005 0200h + 16 × j + 7
DATA7
4005 0200h + 16 × j + 13
TSH
4005 0200h + 16 × j + 14
TSL
4005 0200h + 16 × j + 15

Figure 31.13 Structure of a mailbox register (j = 0 to 31)

Figure 31.14 shows the structure of the eight mask registers MKRk.

b7							b0	Address CAN0
			SID10	SID9	SID8	SID7	SID6	4005 0400h + 4 × k + 0
SID5	SID4	SID3	SID2	SID1	SID0	EID17	EID16	4005 0400h + 4 × k + 1
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	4005 0400h + 4 × k + 2
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	4005 0400h + 4 × k + 3

MKRk register

Figure 31.14 Structure of MKRk (k = 0 to 7)

Figure 31.15 shows the structure of the two FIFO received ID compare registers, FIDCR0 and FIDCR1.

b7							b0	Address CAN0
IDE	RTR		SID10	SID9	SID8	SID7	SID6	4005 0420h + 4 × n + 0
SID5	SID4	SID3	SID2	SID1	SID0	EID17	EID16	4005 0420h + 4 × n + 1
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	4005 0420h + 4 × n + 2
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	4005 0420h + 4 × n + 3

FIDCRn register

Figure 31.15 Structure of the FIDCRn registers (n = 0, 1)

31.6 Acceptance Filtering and Masking Functions

The acceptance filtering and masking functions allow 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 IDFM[1:0] bits in the CTLR register 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 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 required, set the same mask value and the same ID in both the FIFO ID and mask registers.

Figure 31.16 shows the associations between mask registers and mailboxes. Figure 31.17 shows the acceptance filtering.

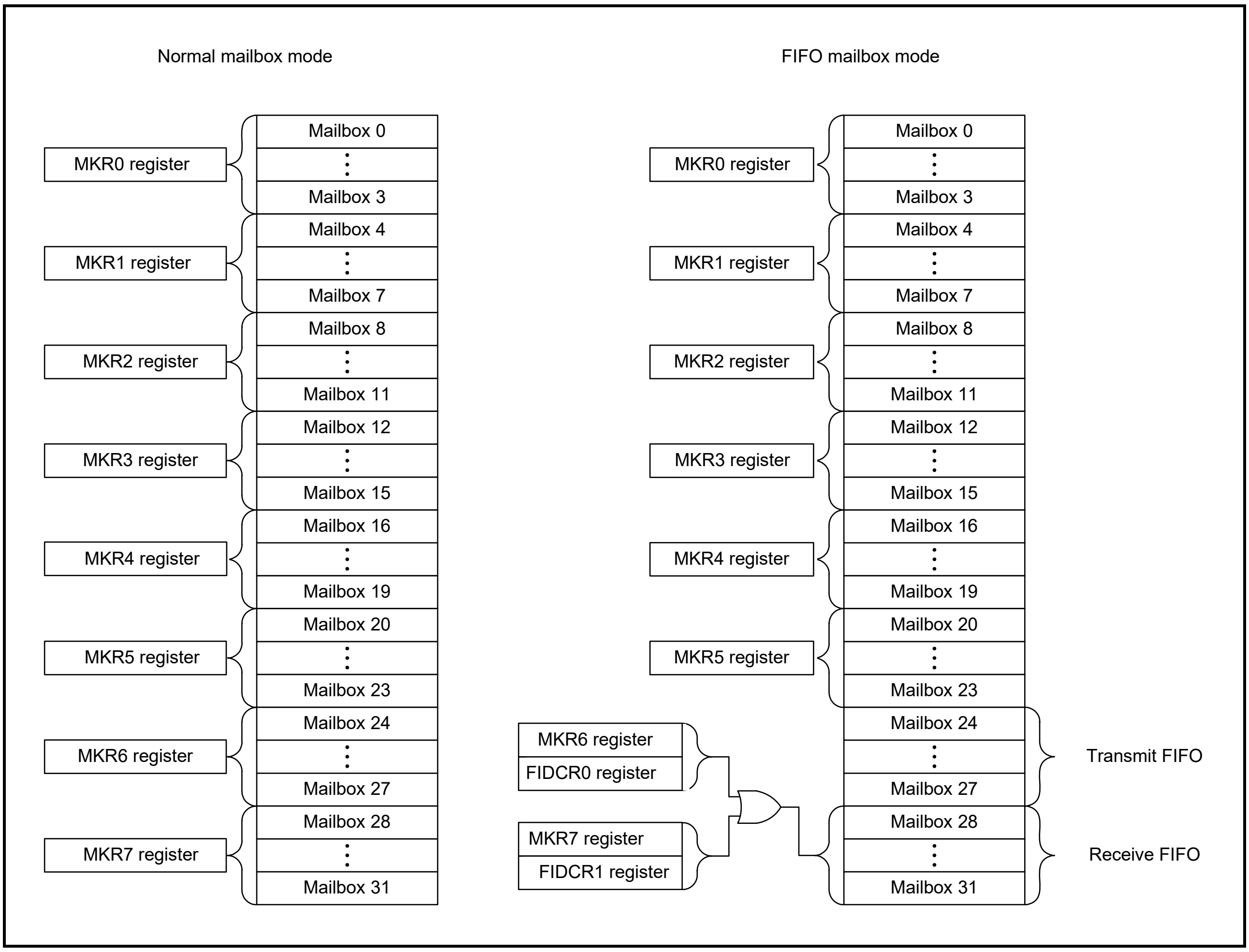
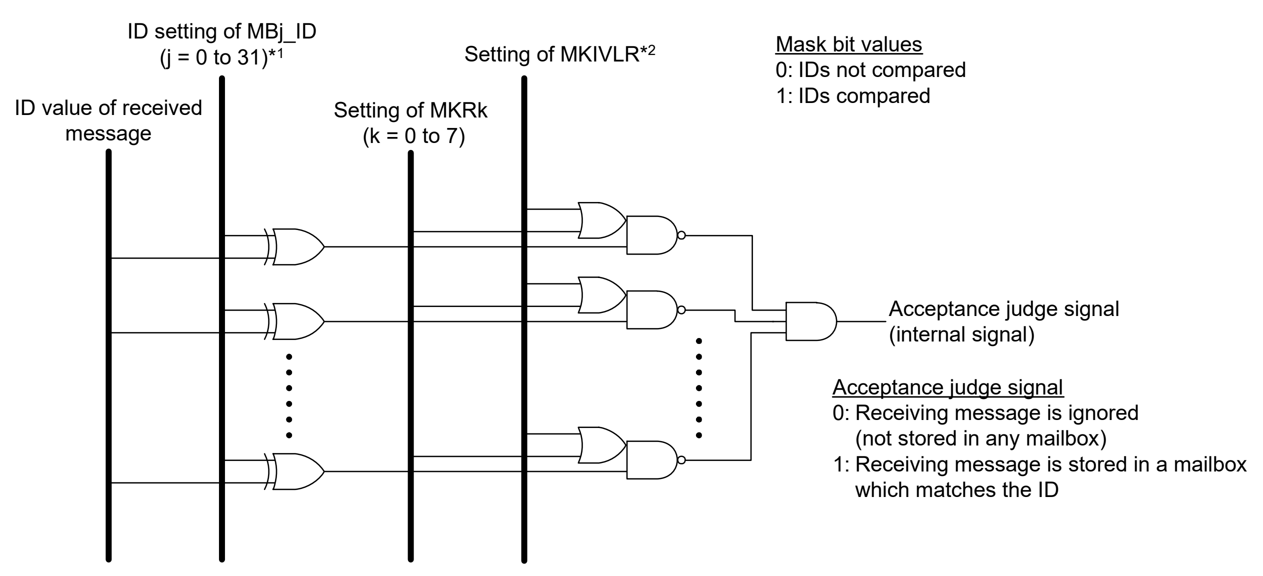

Normal mailbox mode
MKR0 register
MKR1 register
MKR2 register
MKR3 register
MKR4 register
MKR5 register
MKR6 register
MKR7 register
Mailbox 0
Mailbox 3
Mailbox 4
Mailbox 7
Mailbox 8
Mailbox 11
Mailbox 12
Mailbox 15
Mailbox 16
Mailbox 19
Mailbox 20
Mailbox 23
Mailbox 24
Mailbox 27
Mailbox 28
Mailbox 31
FIFO mailbox mode
MKR0 register
MKR1 register
MKR2 register
MKR3 register
MKR4 register
MKR5 register
MKR6 register
FIDCR0 register
MKR7 register
FIDCR1 register
Mailbox 0
Mailbox 3
Mailbox 4
Mailbox 7
Mailbox 8
Mailbox 11
Mailbox 12
Mailbox 15
Mailbox 16
Mailbox 19
Mailbox 20
Mailbox 23
Mailbox 24
Mailbox 27
Mailbox 28
Mailbox 31
Transmit FIFO
Receive FIFO

Figure 31.16 Associations between mask registers and mailboxes


ID setting of MBj_ID
(j = 0 to 31)*1
Setting of MKIVLR*2
Mask bit values
0: IDs not compared
1: IDs compared
ID value of received message
Setting of MKRk
(k = 0 to 7)
Acceptance judge signal
(internal signal)
Acceptance judge signal
0: Receiving message is ignored
(not stored in any mailbox)
1: Receiving message is stored in a mailbox which matches the ID

Note 1. The settings of FIDCR0 and FIDCR1 are used in FIFO mailbox mode.
Note 2. Invalid in FIFO mailboxes.

Figure 31.17 Acceptance filtering

31.7 Reception and Transmission

Table 31.10 lists the CAN communication mode settings.

Table 31.10 Settings for CAN receive and transmit modes

MCTL_TXj.TRMREQ and MCTL_RXj.TRMREQ	MCTL_TXj.RECREQ and MCTL_RXj.RECREQ	MCTL_TXj.ONESHOT and MCTL_RX.ONESHOT	Mailbox communication mode
0	0	0	Mailbox disabled or transmission 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 the 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:

- Before configuring the mailbox, ensure that the MCTL_TXj register is 00h and that there is no pending abort process.

31.7.1 Reception

Figure 31.18 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 the MCTL_RXj (j = 0 to 31) register.

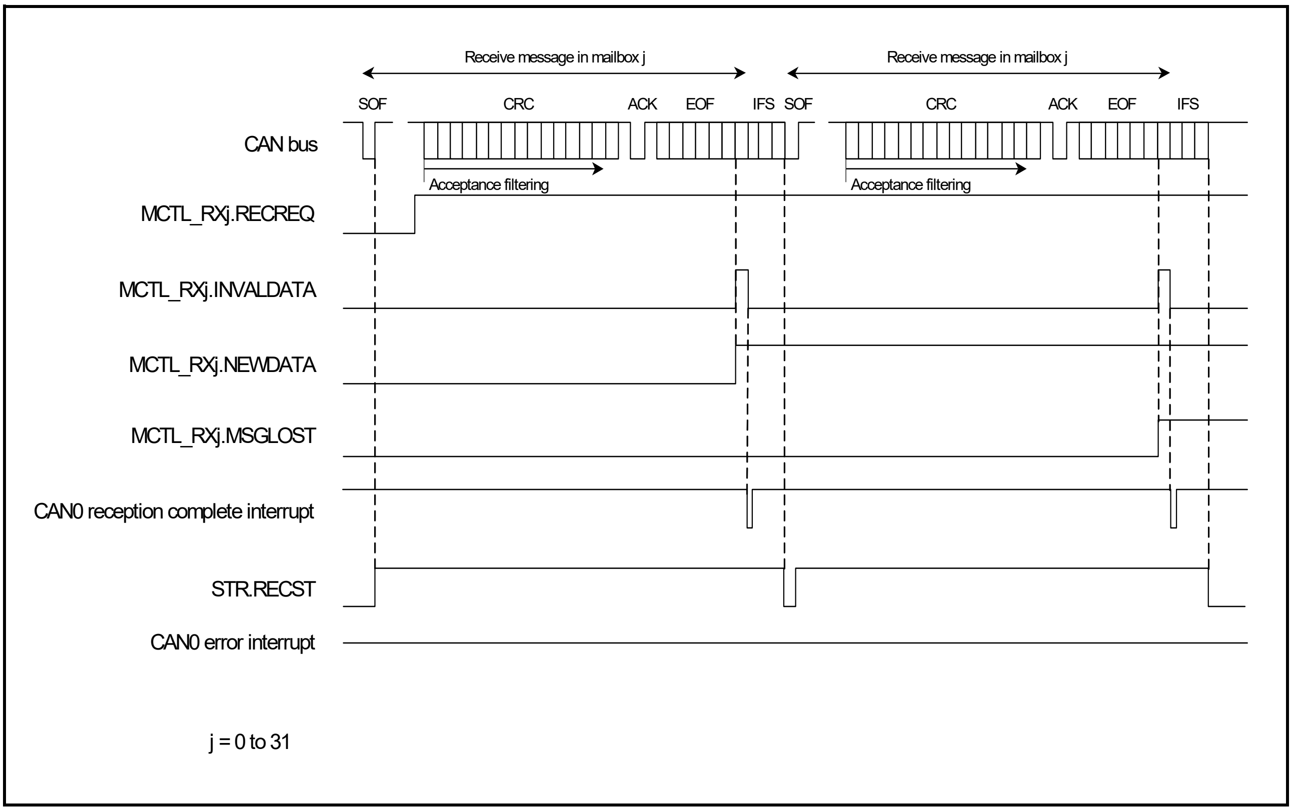

Receive message in mailbox j
Receive message in mailbox j
SOF
CRC
ACK
EOF
IFS
SOF
CRC
ACK
EOF
IFS
CAN bus
Acceptance filtering
Acceptance filtering
MCTL_RXj.RECREQ
MCTL_RXj.INVALDATA
MCTL_RXj.NEWDATA
MCTL_RXj.MSGLOST
CAN0 reception complete interrupt
STR.RECST
CAN0 error interrupt
j = 0 to 31

Figure 31.18 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, MCTL_RXj.NEWDATA for the receive mailbox is set to 1 (new message is being stored or was stored to the mailbox). The INVALDATA flag in the MCTL_RXj register 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. When the interrupt enable bit in the MIER register for the receive mailbox is 1 (interrupt enabled), the INVALDATA flag is set to 0, which triggers a CAN0 reception complete interrupt request.
5. After reading the message from the mailbox, the NEWDATA flag 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 flag 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 31.19 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).

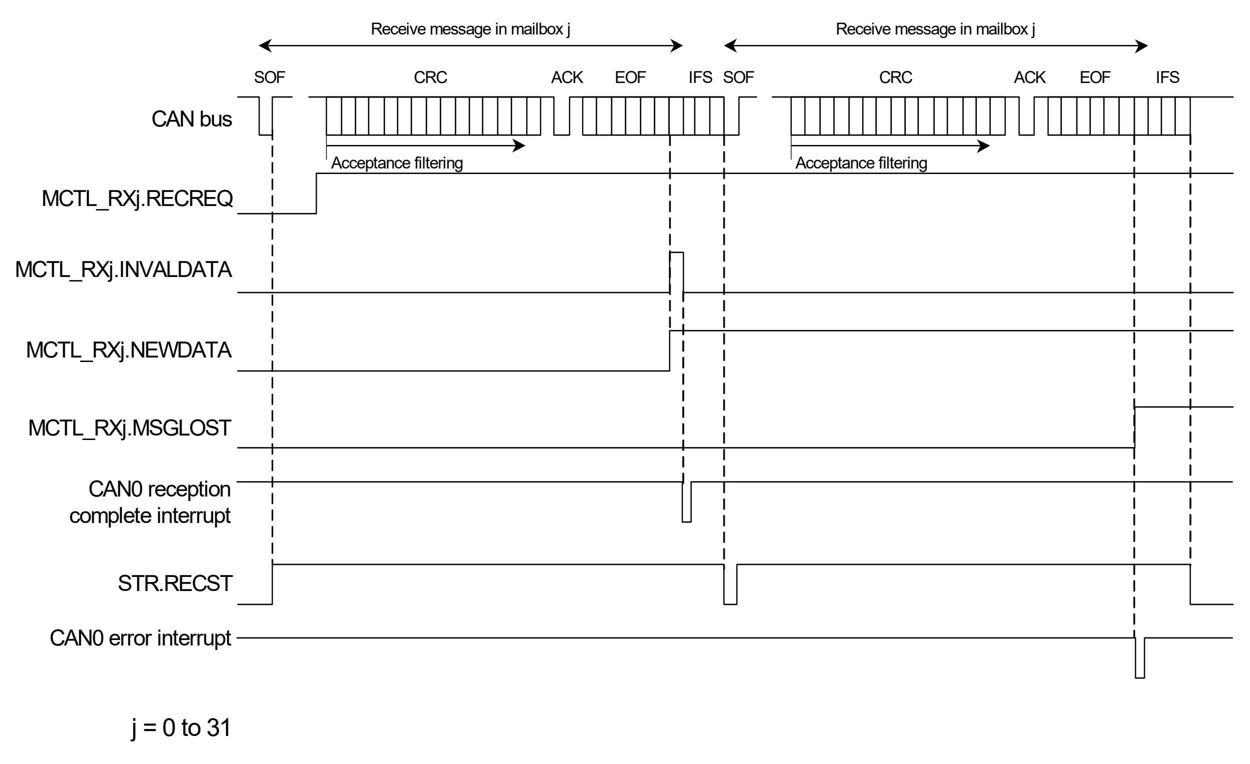

Receive message in mailbox j
Receive message in mailbox j
SOF
CRC
ACK
EOF
IFS
SOF
CRC
ACK
EOF
IFS
CAN bus
Acceptance filtering
Acceptance filtering
MCTL_RXj.RECREQ
MCTL_RXj.INVALDATA
MCTL_RXj.NEWDATA
MCTL_RXj.MSGLOST
CAN0 reception complete interrupt
STR.RECST
CAN0 error interrupt
j = 0 to 31

Figure 31.19 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 flag in MCTL_RXj is set to 0, the MSGLOST flag 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 set to 1 (interrupt enabled).

31.7.2 Transmission

Figure 31.20 shows an example operation of data frame transmission.

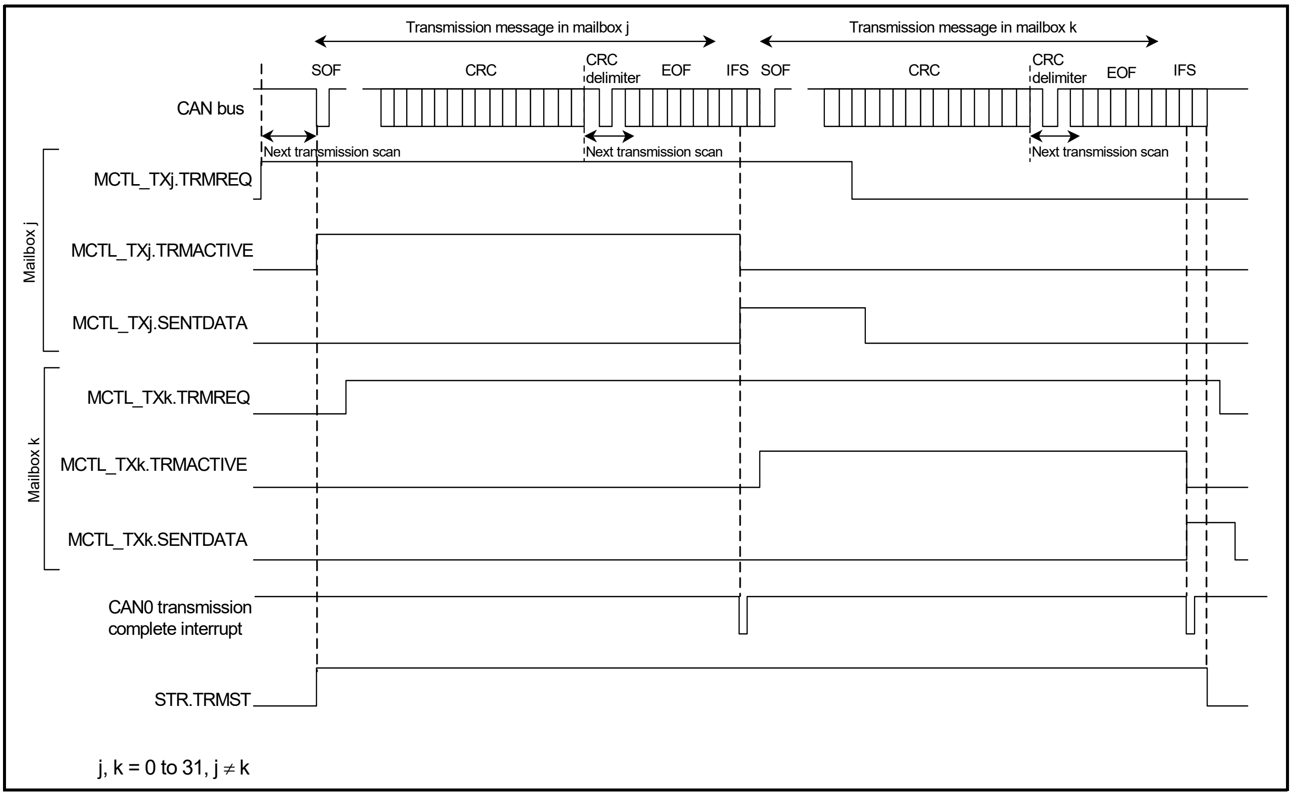

Transmission message in mailbox j
Transmission message in mailbox k
SOF
CRC
CRC delimiter
EOF
IFS
CAN bus
Next transmission scan
Mailbox j
MCTL_TXj.TRMREQ
MCTL_TXj.TRMACTIVE
MCTL_TXj.SENTDATA
Mailbox k
MCTL_TXk.TRMREQ
MCTL_TXk.TRMACTIVE
MCTL_TXk.SENTDATA
CAN0 transmission complete interrupt
STR.TRMST
j, k = 0 to 31, j ≠ k

Figure 31.20 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, the mailbox scanning determines 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 error/arbitration-lost), the TRMST bit in STR 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 completed without losing arbitration, the SENTDATA flag 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, and 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.

31.8 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 the CAN0 error interrupts. Check the EIFR register to determine the interrupt sources:

- Bus error
- Error-warning
- Error-passive
- Bus-off entry
- Bus-off recovery
- Receive overrun
- Overload frame transmission
- Bus lock.

Table 31.11 lists the CAN interrupts.

Table 31.11 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

31.9 Usage Notes

31.9.1 Settings for the Module-Stop State

The Module Stop Control Register B (MSTPCRB) can enable or disable CAN 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.

31.9.2 Settings for the Operating Clock

The settings for the operating clock can be made as follows:

- 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.

32. Serial Peripheral Interface (SPI)

32.1 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 32.1 lists the SPI specifications, and Figure 32.1 shows a 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.

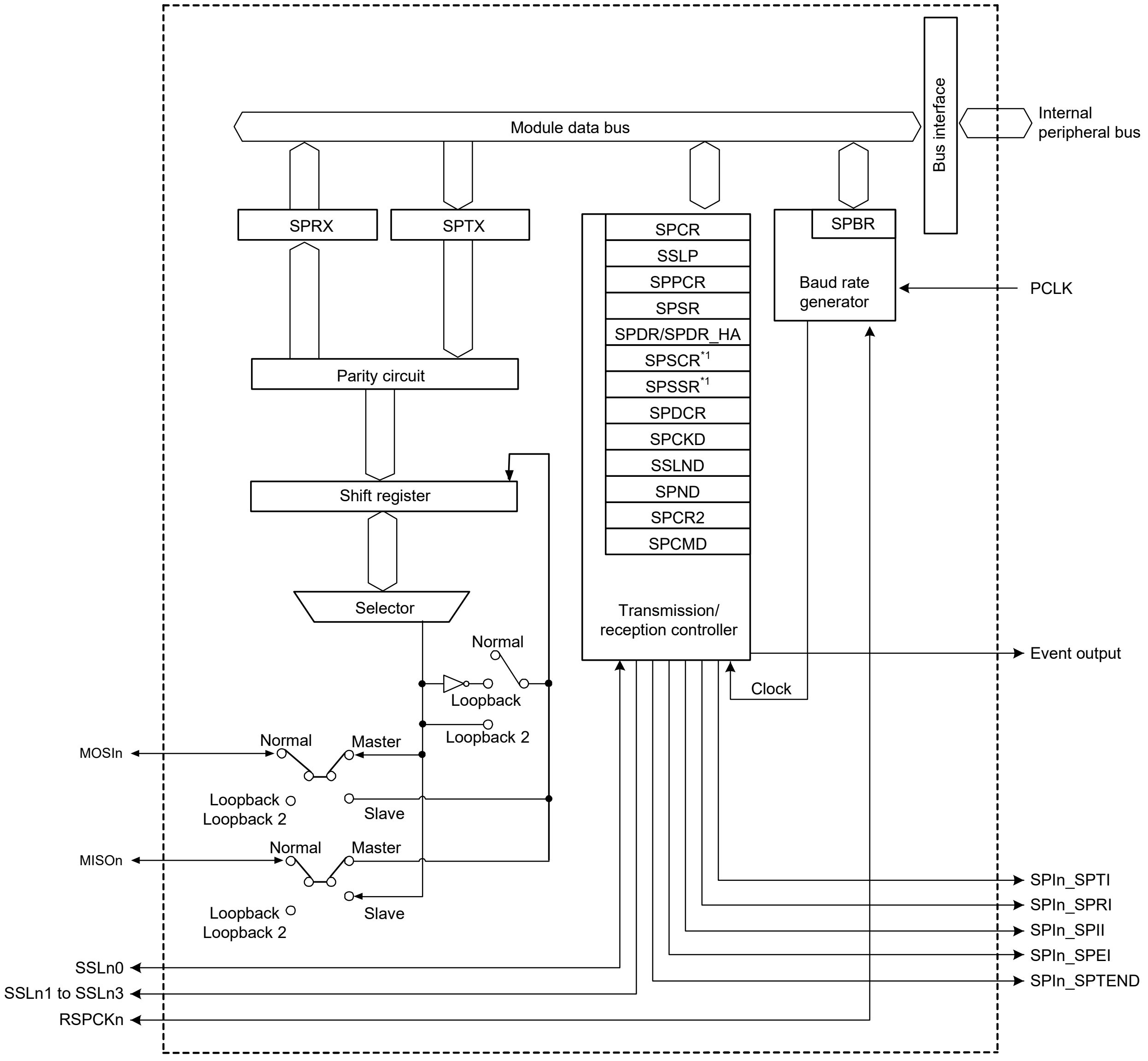
Table 32.1 SPI specifications (1 of 2)

Parameter	Description
Number of channels	Two channels
SPI transfer functions	• MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (SPI clock) signals enable 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 selectable • Transfer bit length selectable as 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits • SPI0: 128 bit transmit and receive buffers, with capability to transfer up to four frames in one round of transmission or reception (each frame consisting of up to 32 bits) • SPI1: 32 bits transmit and receive buffers, with capability to transfer one frame in one round of transmission or reception.
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 RSPCK frequency is that of PCLK divided by 6). Width at high level: 3 PCLK cycles Width at low level: 3 PCLK cycles.
Buffer configuration	• Double buffer configuration for the transmit and receive buffers • SPI0: 128 bits for the transmit and receive buffers • SPI1: 32 bits for the transmit and receive buffers.
Error detection	• Mode fault error detection • Underrun error detection • Overrun error detection*1 • Parity error detection.
SSL control function	• Four SSL pins (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 for 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.

Table 32.1 SPI specifications (2 of 2)

Parameter	Description
Control in master transfer	• A transfer of up to eight commands (for SPI0) can be executed sequentially in looped execution • Support for the following commands: - SPI0: SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, MSB- or LSB-first, burst, RSPCK delay, SSL negation delay, and next-access delay - SPI1: SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, MSB- or LSB-first, 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.
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 • Transmit buffer empty • Mode fault, underrun, overrun, or parity error • SPI idle • Transmission-completed.
Others	• Function for initializing the SPI • Loopback mode.
Module-stop function	Module-stop state can be set to reduce power consumption

Note 1. In master reception, when the RSPCK auto-stop function is enabled, an overrun error does not occur because the transfer clock is stopped on overrun error detection.


Module data bus
Bus interface
Internal peripheral bus
SPRX
SPTX
SPCR
SSLP
SPPCR
SPSR
SPDR/SPDR_HA
SPSCR*1
SPSSR*1
SPDCR
SPCKD
SSLND
SPND
SPCR2
SPCMD
SPBR
Baud rate generator
PCLK
Parity circuit
Shift register
Selector
Transmission/ reception controller
Event output
Normal
Loopback
Loopback 2
Clock
MOSIn
Normal
Master
Loopback
Loopback 2
Slave
MISOn
Normal
Master
Loopback
Loopback 2
Slave
SSLn0
SSLn1 to SSLn3
RSPCKn
SPIn_SPTI
SPIn_SPRI
SPIn_SPII
SPIn_SPEI
SPIn_SPTEND

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[*1]:	SPI Sequence Control Register
SPSSR[*1]:	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 SPI0: 0 to 7 (eight registers) SPI1: 0 (one register)
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

Note 1. SPI0 only

Figure 32.1 SPI block diagram

Table 32.2 lists the I/O pins used in the SPI. The SPI automatically switches the I/O direction of the SSLn0 pin. SSLn0 is set as an output when the SPI is a single master and 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 32.3.2, Controlling the SPI Pins.

Table 32.2 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
	SSLB3	Output	Slave selection output

32.2 Register Descriptions

32.2.1 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: Select SPI operation (4-wire method) 1: Select clock synchronous operation (3-wire method).	R/W
b1	TXMD	Communications Operating Mode Select	0: Select full-duplex synchronous serial communications 1: Select serial communications consisting of only transmit operations.	R/W
b2	MODFEN	Mode Fault Error Detection Enable	0: Disable detection of mode fault errors 1: Enable detection of mode fault errors.	R/W
b3	MSTR	SPI Master/Slave Mode Select	0: Select slave mode 1: Select master mode.	R/W
b4	SPEIE	SPI Error Interrupt Enable	0: Disable SPI error interrupt requests 1: Enable SPI error interrupt requests.	R/W
b5	SPTIE	Transmit Buffer Empty Interrupt Enable	0: Disable transmit buffer empty interrupt requests 1: Enable transmit buffer empty interrupt requests.	R/W
b6	SPE	SPI Function Enable	0: Disable SPI function 1: Enable SPI function.	R/W
b7	SPRIE	SPI Receive Buffer Full Interrupt Enable	0: Disable SPI receive buffer full interrupt requests 1: Enable SPI receive buffer full interrupt requests.	R/W

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 CPHA bit in the SPCMDm register 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 32.3.6, 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 detection of mode fault errors (see section 32.3.8, 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 32.3.2, Controlling the SPI Pins).

MSTR bit (SPI Master/Slave Mode Select)

The MSTR bit selects master or slave mode for the SPI. Based on the MSTR bit settings, the SPI determines the direction of the RSPCKn, MOSIn, MISOn, and 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 32.3.8, 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 32.3.8, Error Detection.

Setting the SPE bit to 0 disables the SPI function and initializes a part of the module function. For details, see section 32.3.9, Initializing the SPI. Additionally, 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 when the SPI detects a receive buffer full write after completing a serial transfer.

32.2.2 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*1	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

Note 1. This bit is not available in SPI1.

If the contents of SSLP are changed when the SPCR.SPE bit is 1, do not perform subsequent operations.

32.2.3 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 low 1: The level output on the MOSIn pin during MOSI idling is 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 the SPPCR register 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, establishing 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 for both SPI0 and SPI1, including the SSL retention period during a burst transfer for the SPI0.

MOIFE bit (MOSI Idle Value Fixing Enable)

The MOIFE bit fixes the MOSIn output value when the SPI in master mode is in an SSL negation period for both SPI0 and SPI1, including the SSL retention period during a burst transfer for the SPI0. 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.

32.2.4 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 idle state 1: SPI is in 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 as the MODF flag.
Note 3. The write value should be 1.

OVRF flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error. In master mode (SPCR.MSTR = 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 32.3.8.1, 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.

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

- When the SPCR.SPE bit is 1, enabling the SPI function.

[Clearing condition]

Master mode

- When either condition 1. is satisfied or conditions 2., 3., and 4. are satisfied for SPI0, and when either condition 1. is satisfied or conditions 2. and 4. are satisfied for SPI1.

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]

Master mode

- When the input level of the SSLni pin changes to an active level while the SPCR.MSTR bit is 1 (master mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), triggering a mode fault error.

Slave mode

- When either condition 1. or 2. is satisfied.
    1. The SSLni pin is negated before the RSPCK cycle required for data transfer ends, when 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), 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.

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.

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), 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.

SPTEF flag (SPI Transmit Buffer Empty Flag)

The SPTEF flag indicates the status of the transmit buffer for the SPI Data Register (SPDR/SPDR_HA).

[Setting conditions]

- When either condition 1. or 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]

- SPI0: Data written to the SPDR/SPDR_HA register equals the number of frames set by the number of frames specification bits, SPFC[1:0], in the SPI Data Control Register (SPDCR).
- SPI1: Data written to SPDR/SPDR_HA.

Data can only be written to SPDR/SPDR_HA when the SPTEF bit is 1. If data is written to the transmit buffer of SPDR/SPDR_HA when the SPTEF bit is 0, data in the transmit buffer is not updated.

SPRF flag (SPI Receive Buffer Full Flag)

The SPRF flag indicates the status of the receive buffer for the SPI Data Register (SPDR/SPDR_HA).

[Setting conditions]

- SPI0: When receive data with the number of frames specified in the SPDCR.SPFC[1:0] bits is transferred from the shift register to SPDR/SPDR_HA, while the SPCR.TXMD bit is 0, and the SPRF flag is 0. When the OVRF flag is 1, however, this flag does not change from 0 to 1.
- SPI1: When receive data is transferred from the shift register to SPDR/SPDR_HA, while the SPCR.TXMD bit is 0, and the SPRF flag is 0. When the OVRF flag is 1, however, this flag does not change from 0 to 1.

[Clearing condition]

- When received data is read from SPDR/SPDR_HA.

32.2.5 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

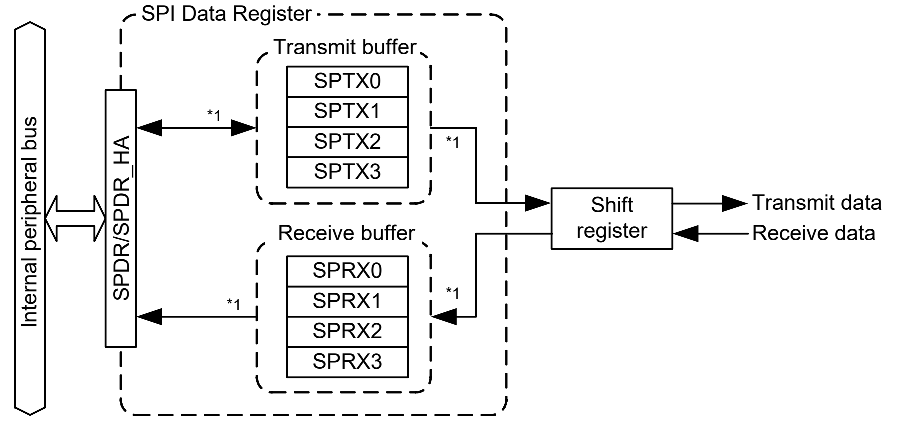
Address(es): SPI0.SPDR_HA 4007 2004h, SPI1.SPDR_HA 4007 2104h

	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

The SPDR/SPDR_HA register is the interface with the buffers that hold data for transmission and reception by the SPI.

When accessing this register in words (SPLW = 1), access the SPDR register. When accessing it in halfwords (SPLW = 0), access the SPDR_HA register.

The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to SPDR/SPDR_HA. Figure 32.2 and Figure 32.3 show the configuration of the SPDR/SPDR_HA register for SPI0 and SPI1 channels, respectively.


Internal peripheral bus
SPDR/SPDR_HA
SPI Data Register
Transmit buffer
SPTX0
SPTX1
SPTX2
SPTX3
Receive buffer
SPRX0
SPRX1
SPRX2
SPRX3
*1
*1
*1
*1
Shift register
Transmit data
Receive data

Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 32.2 Configuration of SPDR/SPDR_HA (SPI0)

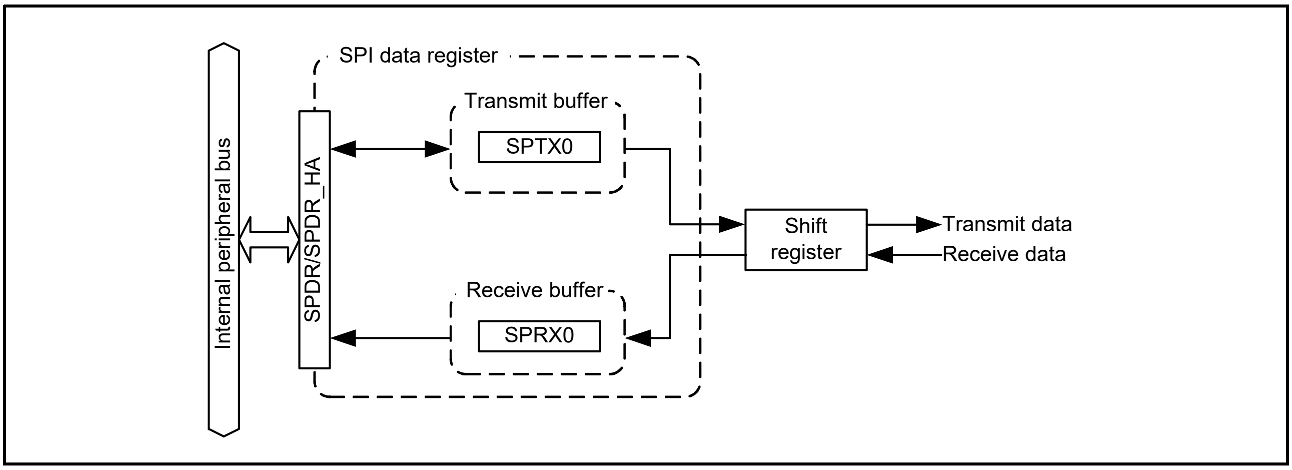


Figure 32.3 Configuration of SPDR/SPDR_HA (SPI1)

The transmit and receive buffers each have four stages for SPI0 and one stage for SPI1. The number of stages used for SPI0 is selectable by the number of frames specification bits, SPDCR, in the SPDCR register. These stages of the buffer are all mapped to the single address of the SPDR/SPDR_HA register.

Data written to the SPDR/SPDR_HA register is written to a transmit-buffer stage (SPTXn) (n = 0 to 3 for SPI0, n = 0 for SPI1), 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, the bits not referred to in SPTXn (n = 0 to 3 for SPI0, n = 0 for SPI1) are stored in the associated bits in SPRXn (n = 0 to 3 for SPI0, n = 0 for SPI1). For example, if the data length is 9 bits, received data is stored in the SPRXn[8:0] bits and the SPTXn[31:9] bits are stored in the SPRXn[31:9] bits.

(1) Bus interface

SPDR/SPDR_HA is an interface with 32-bit wide transmit and receive buffers, each of which has four stages for SPI0 and one stage for SPI1, 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, SPLW, in the SPDCR register.

Flush the transmission data at the LSB end of the register and store the received data at the LSB end.

The following sections describe the operations involved in writing to and reading from the SPDR/SPDR_HA register.

(a) Writing

Data written to SPDR/SPDR_HA is written to a transmit buffer SPTXn (n = 0 to 3 for SPI0, n = 0 for SPI1). This is not affected by the value of the SPDCR.SPRDTD bit, unlike when reading from SPDR/SPDR_HA.

The transmit buffer includes a write pointer that is automatically updated to reference the next stage each time data is written to SPDR/SPDR_HA.

Figure 32.4 and Figure 32.5 show the configuration of the bus interface with the transmit buffer, for writes to SPDR/SPDR_HA, for SPI0 and SPI1 respectively.

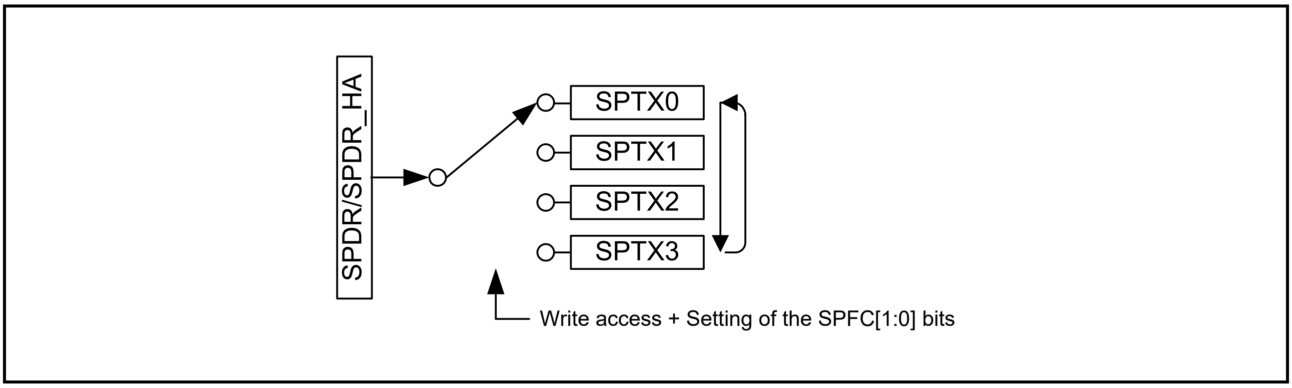


Figure 32.4 Configuration of SPDR/SPDR_HA for write access (SPI0)

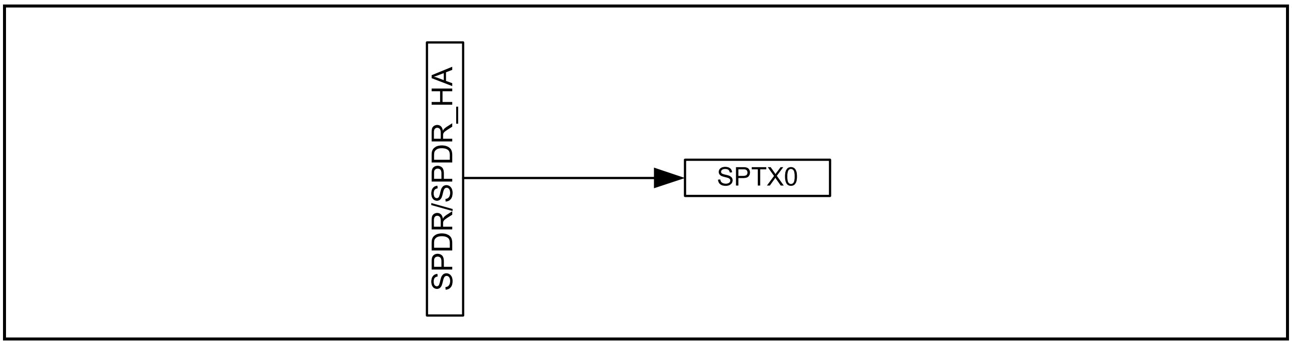


Figure 32.5 Configuration of SPDR/SPDR_HA for write access (SPI1)

For SPI0, the sequence for switching the transmit buffer write pointer changes with the setting of the number of frames specification bits, SPFC[1:0], in the SPDCR register.

- 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, SPE, in the SPI Control Register (SPCR) while the bit is 0, SPTX0 is the destination for the next write.

When writing to the transmit buffer SPTXn (n = 0 to 3 for SPI0, n = 0 for SPI1) after generating the transmit buffer empty interrupt (when SPSR.SPTEF is 1), write the number of frames set in the number of frames specification bits, SPFC[1:0], in the SPDCR register. The value of the buffer is not updated after completion of the writing and before the next transmit buffer empty interrupt is generated (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 a receive buffer SPRXn (n = 0 to 3 for SPI0, n = 0 for SPI1) or a transmit buffer SPTXn (n = 0 to 3 for SPI0, n = 0 for SPI1). The setting of the SPI receive or transmit data select bit, SPRDTD, in the SPDCR register selects whether reading is from the receive or transmit buffer.

The sequence of reading the SPDR/SPDR_HA register is controlled by the independent pointers, receive buffer read pointer and transmit buffer read pointer.

Figure 32.6 and Figure 32.7 show the configuration of a bus interface with the receive and transmit buffers for reading from SPDR/SPDR_HA for SPI0 and SPI1, respectively.

- SPI0

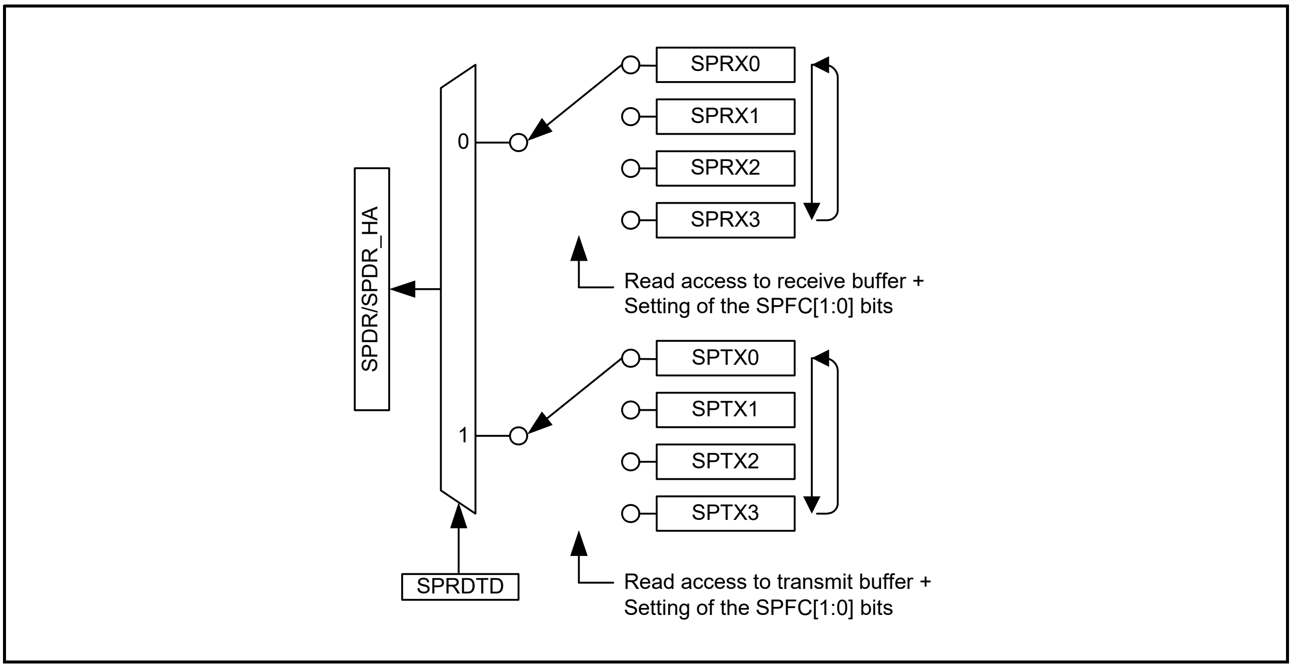

SPDR/SPDR_HA
0
1
SPRX0
SPRX1
SPRX2
SPRX3
Read access to receive buffer +
Setting of the SPFC[1:0] bits
SPTX0
SPTX1
SPTX2
SPTX3
SPRDTD
Read access to transmit buffer +
Setting of the SPFC[1:0] bits

Figure 32.6 Configuration of SPDR/SPDR_HA for read access (SPI0)

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, SPE, in the SPI Control Register (SPCR) when 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 is not updated 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 a transmit buffer empty interrupt is generated, and when the transmit buffer becomes full again (the number of frames of data specified in the number of frames specification bits, SPDCR.SPFC[1:0], are written to the transmit buffer), reading from the transmit buffer returns all 0s until the next transmit buffer empty interrupt is generated.

- SPI1

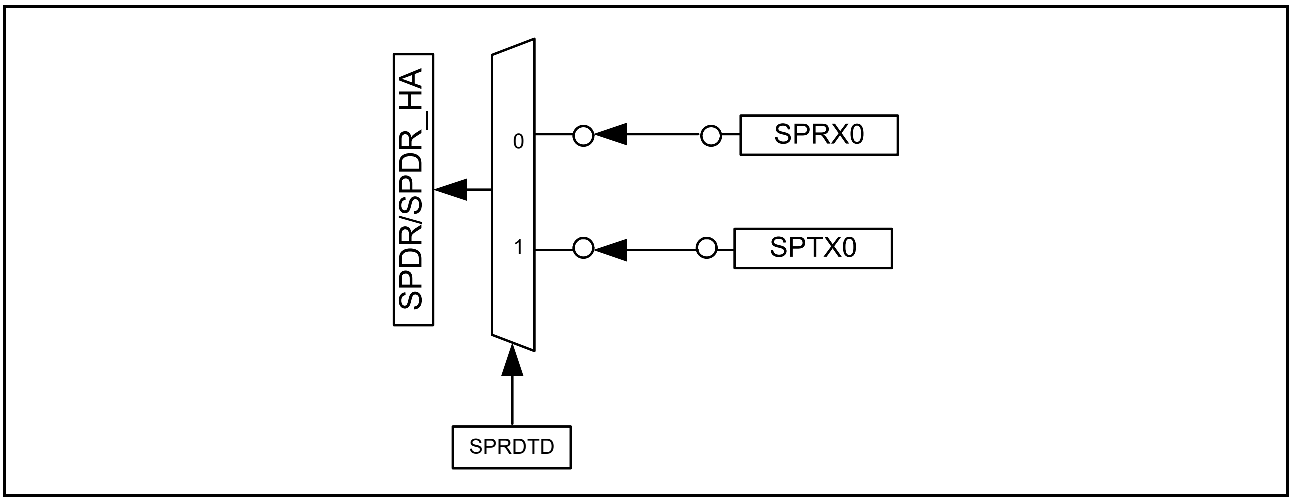

SPDR/SPDR_HA
0
1
SPRX0
SPTX0
SPRDTD

Figure 32.7 Configuration of SPDR/SPDR_HA for read access (SPI1)

The transmit buffer read pointer is updated when writing to SPDR/SPDR_HA, and not updated 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 a transmit buffer empty interrupt is generated, and when the transmit buffer becomes full again, reading from the transmit buffer returns all 0s until the next transmit buffer empty interrupt is generated (when SPTEF is 0).

32.2.6 SPI Sequence Control Register (SPSCR)

Address(es): SPI0.SPSCR 4007 2008h

	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 (Number) 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 the 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 the 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.

32.2.7 SPI Sequence Status Register (SPSSR)

Address(es): SPI0.SPSSR 4007 2009h

	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

Bit	Symbol	Bit name	Description	R/W
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 32.3.10.1, 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 SPECM[2:0] bit values have no meaning.

For the SPI error detection function, see section 32.3.8, Error Detection. For the SPI sequence control, see section 32.3.10.1, Master mode operation.

32.2.8 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 in slave mode, the bit rate depends on the bit rate of the input clock, regardless of the settings in the SPBR and the SPCMDm.BRDV[1:0] bits (bit rate division setting bits). Use bit rates that satisfy the electrical characteristics of the device.

The bit rate is determined by combination of the SPBR and BRDV[1:0] settings in the SPI Command Register, SPCMDm (SPCMD0 to SPCMD7 for SPI0, SPCMD0 for SPI1). 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 32.3 lists examples of the relationship between the SPBR settings, the BRDV[1:0] settings, and bit rates.

Table 32.3 Relationship between SPBR settings, BRDV[1:0] settings, and bit rates (1 of 2)

SPBR (n)	BRDV[1:0] bits (N)	Division ratio	Bit rate	
			PCLK = 32 MHz	PCLK = 48 MHz
0	0	2	16.0 Mbps	-

Table 32.3 Relationship between SPBR settings, BRDV[1:0] settings, and bit rates (2 of 2)

SPBR (n)	BRDV[1:0] bits (N)	Division ratio	Bit rate	
			PCLK = 32 MHz	PCLK = 48 MHz
1	0	4	8.00 Mbps	12.0 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

32.2.9 SPI Data Control Register (SPDCR)

Address(es): SPI0.SPDCR 4007 200Bh

	b7	b6	b5	b4	b3	b2	b1	b0
(SPI0)	—	—	SPLW	SPRDTD	—	—	SPFC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Address(es): SPI1.SPDCR 4007 210Bh

	b7	b6	b5	b4	b3	b2	b1	b0
(SPI1)	—	—	SPLW	SPRDTD	—	—	—	—
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	• SPI0: b1 b0 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames.	R/W
	—	Reserved	• SPI1: These bits are read as 0. The write value should be 0.	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: Read SPDR/SPDR_HA values from the receive buffer 1: Read SPDR/SPDR_HA values from the transmit buffer (but only if the transmit buffer is empty).	R/W
b5	SPLW	SPI Word Access/Halfword Access Specification	0: Set SPDR_HA to valid for halfword access 1: Set SPDR to valid for word access.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Up to four frames for SPI0 and one frame for SPI1 can be transmitted or received in one round of transmission or reception. The amount of data in each transfer for SPI0 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. The amount of data in each transfer for SPI1 is controlled by the combination of the SPCMD0.SPB[3:0] 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 with 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 with the number of frames specified by the SPFC[1:0] bits are transmitted to the shift register, the SPI generates a transmission buffer empty interrupt (SPSR.SPTEF = 1).

When the data with the number of frames specified by the SPFC[1:0] bits is received, the SPI generates a receive buffer full interrupt (SPSR.SPRF = 1).

The SPFC[1:0] bits are reserved for SPI1.

Table 32.4 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 register reads values from the receive buffer or from the transmit buffer. If reading is from the transmit buffer, the last value written to the SPDR/SPDR_HA register is read. Reading the transmit buffer for SPI0 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 = 1). Reading the transmit buffer for SPI1 must be done after generation of the transmit buffer empty interrupt (SPSR.SPTEF = 1).

For details, see section 32.2.5, 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 the SPDR_HA register 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 SPI data length setting bits, SPCMDm.SPB[3:0], from 8 to 16 bits. When a data length of 20, 24, or 32 bits is specified, do not perform any operations.

32.2.10 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.

32.2.11 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.

32.2.12 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 the 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 the 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.

32.2.13 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 of receive data not checked 1: Parity bit added to transmit data and parity bit of receive data checked (when SPCR.TXMD = 0). Parity bit added to transmit data but parity bit of receive data not checked (when SPCR.TXMD = 1).	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: Idle interrupt requests disabled 1: Idle interrupt requests enabled.	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 SPCR2 is changed while the SPE bit in the SPCR register is 1, do not perform subsequent operations.

SPPE bit (Parity Enable)

The SPPE bit enables or disables the parity function.

When the SPCR.TXMD bit is 0 and this bit is 1, the parity bit is added to transmit data and parity checking is performed for receive data. When the SPCR.TXMD bit is 1 and this bit is 1, the parity bit is added to transmit data, but parity checking is not performed for receive data.

SPOE bit (Parity Mode)

The SPOE bit specifies odd or even parity.

When even parity is set, parity bit addition is performed so that the total number of bits whose value is 1 in the transmit or receive character plus the parity bit is even. Similarly, when odd parity is set, parity bit addition is performed so that the total number of bits whose value is 1 in the transmit or receive character plus the parity bit is odd.

The SPOE bit is 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 an SPI 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 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 32.3.8.1, Overrun errors.

32.2.14 SPI Command Registers (SPCMDm) (m =0 to 7 for SPI0; m = 0 for SPI1)

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(SPI0)	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

Address(es): SPI1.SPCMD0 4007 2110h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(SPI1)	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]				—	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1

Bit	Symbol	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: Set RSPCK low when idle 1: Set 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

Bit	Symbol	Bit name	Description	R/W
b6 to b4	SSLA[2:0] *1	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	• SPI0 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
	—	Reserved	• SPI1 This bit is read as 0. The write value should be 0.	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: A next-access delay of 1 RSPCK + 2 PCLK 1: A next-access delay equal to the setting of 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 equal to the setting of the SPI Clock Delay register (SPCKD).	R/W

Note 1. This bit is not available in SPI1.

- SPI0

The SPCMDm register sets the transfer format for the SPI in master mode. SPI0 has eight 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 based on 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 that 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 SPCMDm are changed while the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1, do not perform subsequent operations.

- SPI1

SPI1 has one SPI Command Register (SPCMD). The SPI Command Register (SPCMD) sets the transfer format in master mode. Some of the bits in the SPCMD0 register are used to set the transfer mode for the SPI in slave mode. If the contents of the SPCMD register are changed while the SPCR.SPE bit is 1, do not perform subsequent operations.

CPHA bit (RSPCK Phase Setting)

The CPHA bit sets the RSPCK phase of 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 of 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. A bit rate is determined by combination of the settings in the BRDV[1:0] bits and the SPBR register. See section 32.2.8, SPI Bit Rate Register (SPBR). The SPBR settings determine the base bit rate. The BRDV[1:0] settings select a bit rate obtained by dividing the base bit rate by 1, 2, 4, or 8. In the SPCMDm register for SPI0, different BRDV[1:0] bit settings can be specified, enabling the 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 value set in the associated SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state, as the SSLn0 pin acts as input.

When using the SPI in slave mode, set the SSLA[2:0] bits to 000b.

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 32.3.10.1, Master mode operation (4) Burst transfer. When using the SPI in slave mode, set the SSLKP bit to 0.

The SSLKP bit is reserved for SPI1.

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 SPI data length setting bits, SPCMDm.SPB[3:0], 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.

32.3 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.

32.3.1 Overview of SPI Operations

The SPI is capable of synchronous serial transfers in the following modes:

- Slave mode (SPI operation)
- Single-master mode (SPI operation)
- Multi-master mode (SPI operation)
- Slave mode (clock synchronous operation)
- Master mode (clock synchronous operation).

The SPI mode can be selected with the MSTR, MODFEN, and SPMS bits in SPCR. Table 32.5 lists the relationship between SPI modes and SPCR settings, and a description of each mode.

Table 32.5 Relationship between SPCR settings and SPI modes (1 of 2)

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
RSPCKn 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	Supported in SPI0	Supported in SPI0	Supported in SPI0	—	—
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 on generation of a transmit buffer empty interrupt request (SPTEF is 1)	Transmit buffer is written to on generation of a transmit buffer empty interrupt request (SPTEF is 1)	RSPCK oscillation	Transmit buffer is written to on generation of a transmit buffer empty interrupt request (SPTEF is 1)
Sequence control	Not supported	Supported in SPI0	Supported in SPI0	Not supported	Supported in SPI0
Transmit buffer empty detection	Supported				

Table 32.5 Relationship between SPCR settings and SPI modes (2 of 2)

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
Receive buffer full detection	Supported*2				
Overrun error detection	Supported*2	Supported*2, *4	Supported*2, *4	Supported*2	Supported*2
Parity error detection	Supported*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.

32.3.2 Controlling the SPI Pins

The SPI can switch pin states based on the MSTR, MODFEN, and SPMS bit settings in the SPCR register and the PmnPFS.NCODR bit for I/O ports. Table 32.6 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 32.6 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 or multi-master mode (SPI operation) determines MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer for SPI0) based on the MOIFE and MOIFV bit settings in SPPCR, as listed in Table 32.7.

Table 32.7 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

32.3.3 SPI System Configuration Examples

32.3.3.1 Single master and single slave with the MCU configured as a master

Figure 32.8 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 outputs of the MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave 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 the SPCMDm.CPHA is 0, the SSL signal for some slave devices cannot be fixed to an active level. In this case, always connect the SSLni output of the MCU to the SSL input of the slave device.

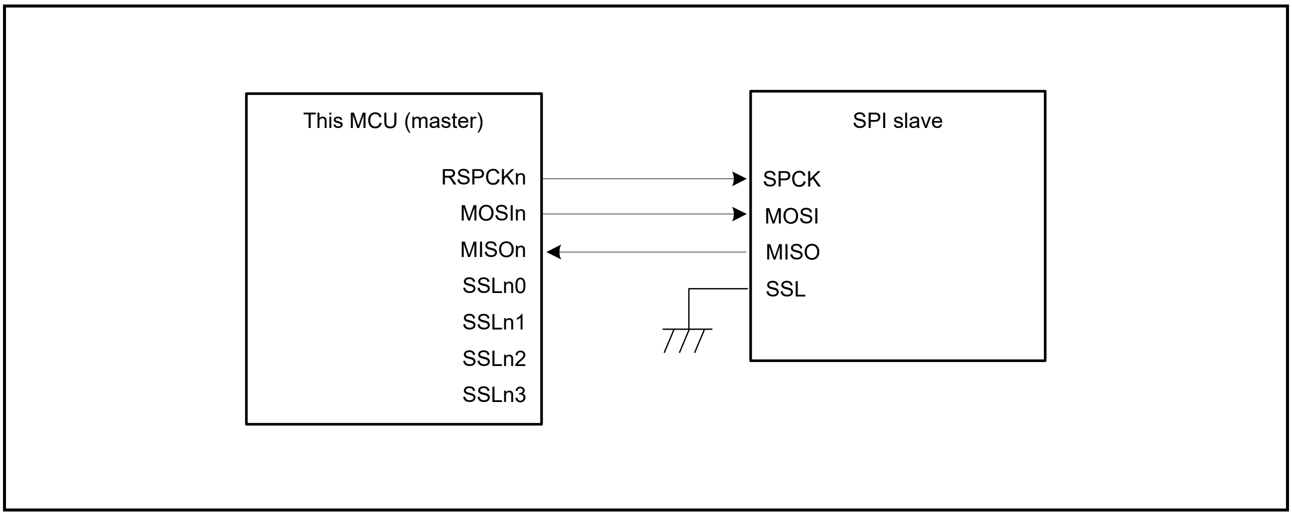


Figure 32.8 Single-master and single-slave configuration example with the MCU as the master

32.3.3.2 Single master and single slave with the MCU configured as a slave

Figure 32.9 shows a single-master and single-slave SPI system configuration example where the MCU is a slave. When the MCU operates as a slave, the SSLn0 pin is used as SSL input. The SPI master drives the SPCK and MOSI signals. The MCU (slave) drives the MISOn signal.*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, and the MCU (slave) stays selected. This enables serial transfer (Figure 32.10).

Note 1. When SSLn0 is at a non-active level, the pin state is Hi-Z.

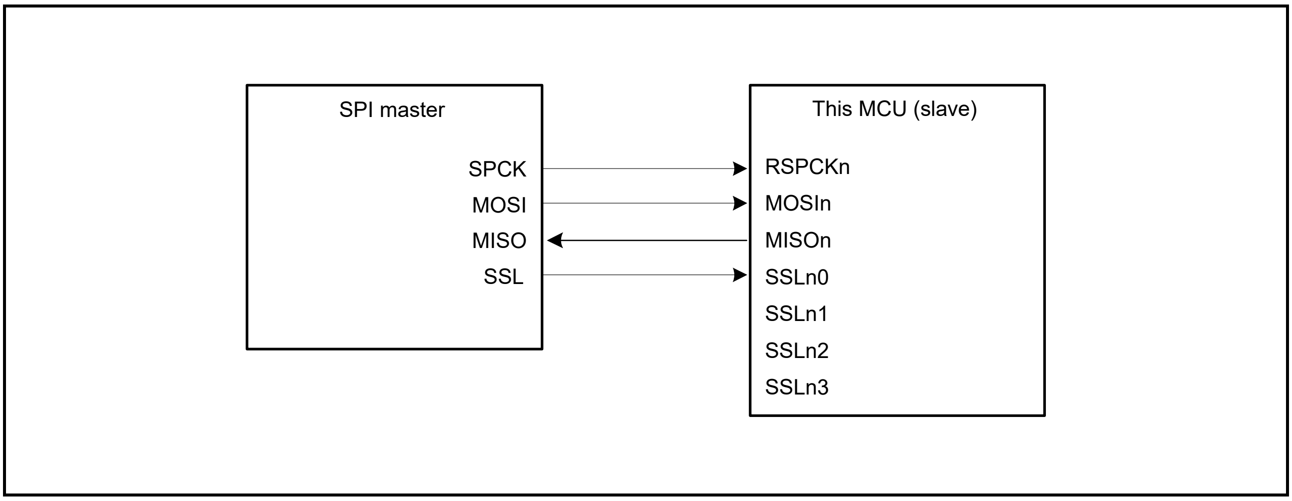

SPI master
SPCK
MOSI
MISO
SSL
This MCU (slave)
RSPCKn
MOSIn
MISOn
SSLn0
SSLn1
SSLn2
SSLn3

Figure 32.9 **Single-master and single-slave configuration example with the MCU as a slave and CPHA = 0**

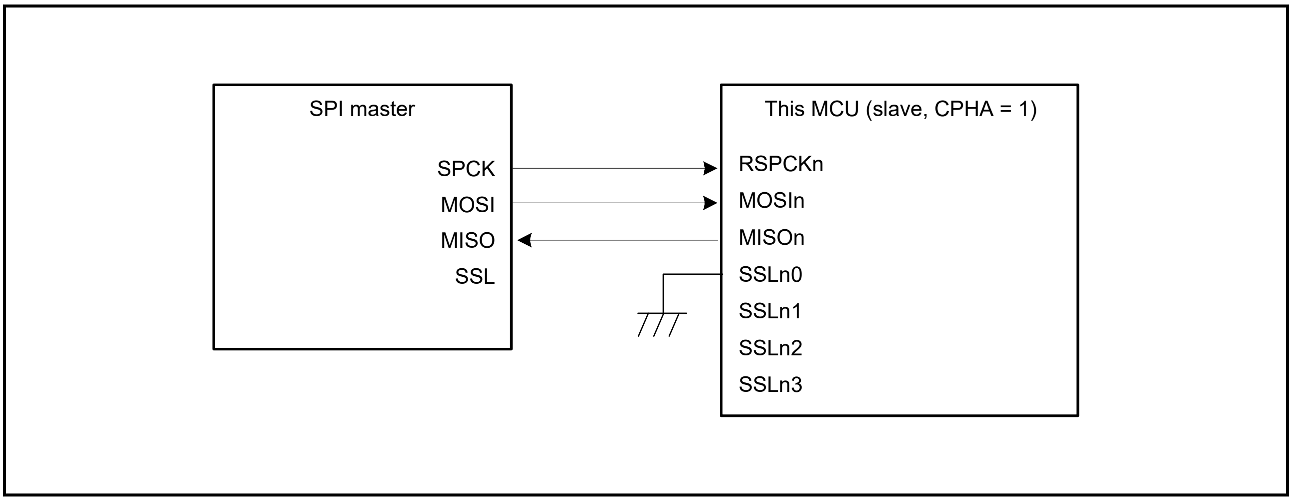

SPI master
SPCK
MOSI
MISO
SSL
This MCU (slave, CPHA = 1)
RSPCKn
MOSIn
MISOn
SSLn0
SSLn1
SSLn2
SSLn3

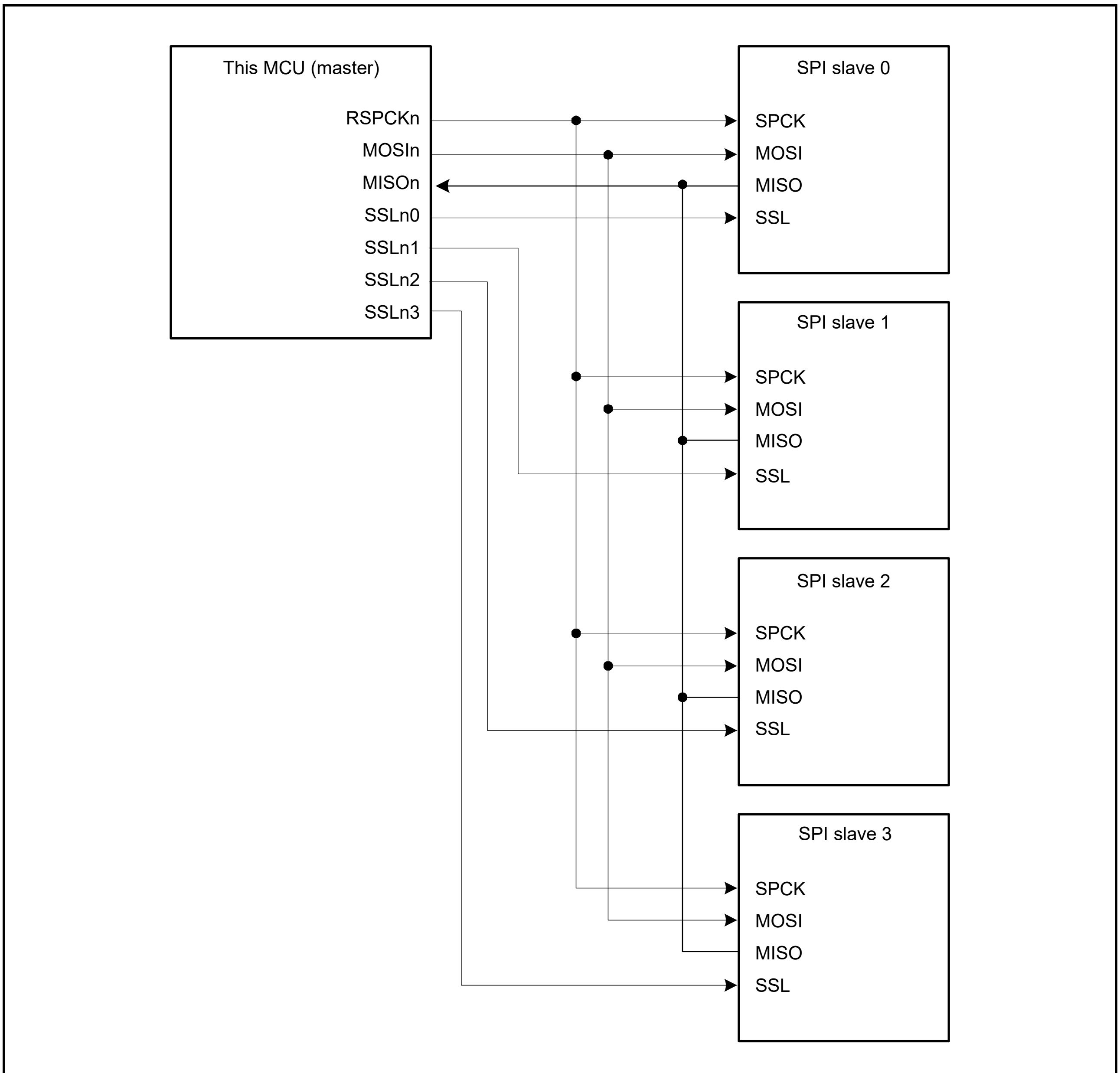
Figure 32.10 **Single-master and single-slave configuration example with the MCU as a slave and CPHA = 1**

32.3.3.3 Single-master and multi-slave with the MCU configured as a master

Figure 32.11 shows a single-master and multi-slave SPI system configuration example where the MCU is a master. In the example shown in Figure 32.11, the SPI system includes 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 SPCK and MOSI inputs of SPI slave 0 to SPI slave 3. The MISO outputs of SPI slave 0 to SPI slave 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 the RSPCKn, MOSIn signals, and SSLn0 to SSLn3 pins. Of the SPI slave 0 to SPI slave 3, the slave that receives low-level input into the SSL input drives the MISO signal.


This MCU (master)
RSPCKn
MOSIn
MISOn
SSLn0
SSLn1
SSLn2
SSLn3
SPI slave 0
SPCK
MOSI
MISO
SSL
SPI slave 1
SPCK
MOSI
MISO
SSL
SPI slave 2
SPCK
MOSI
MISO
SSL
SPI slave 3
SPCK
MOSI
MISO
SSL

Note: SSLB2 is not available.

Figure 32.11 Single-master and multi-slave configuration example with the MCU as a master

32.3.3.4 Single master and multi-slave with the MCU configured as a slave

Figure 32.12 shows a single-master and multi-slave SPI system configuration example where the MCU is a slave. In the example shown in Figure 32.12, the SPI system includes 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 the MISOn signal.

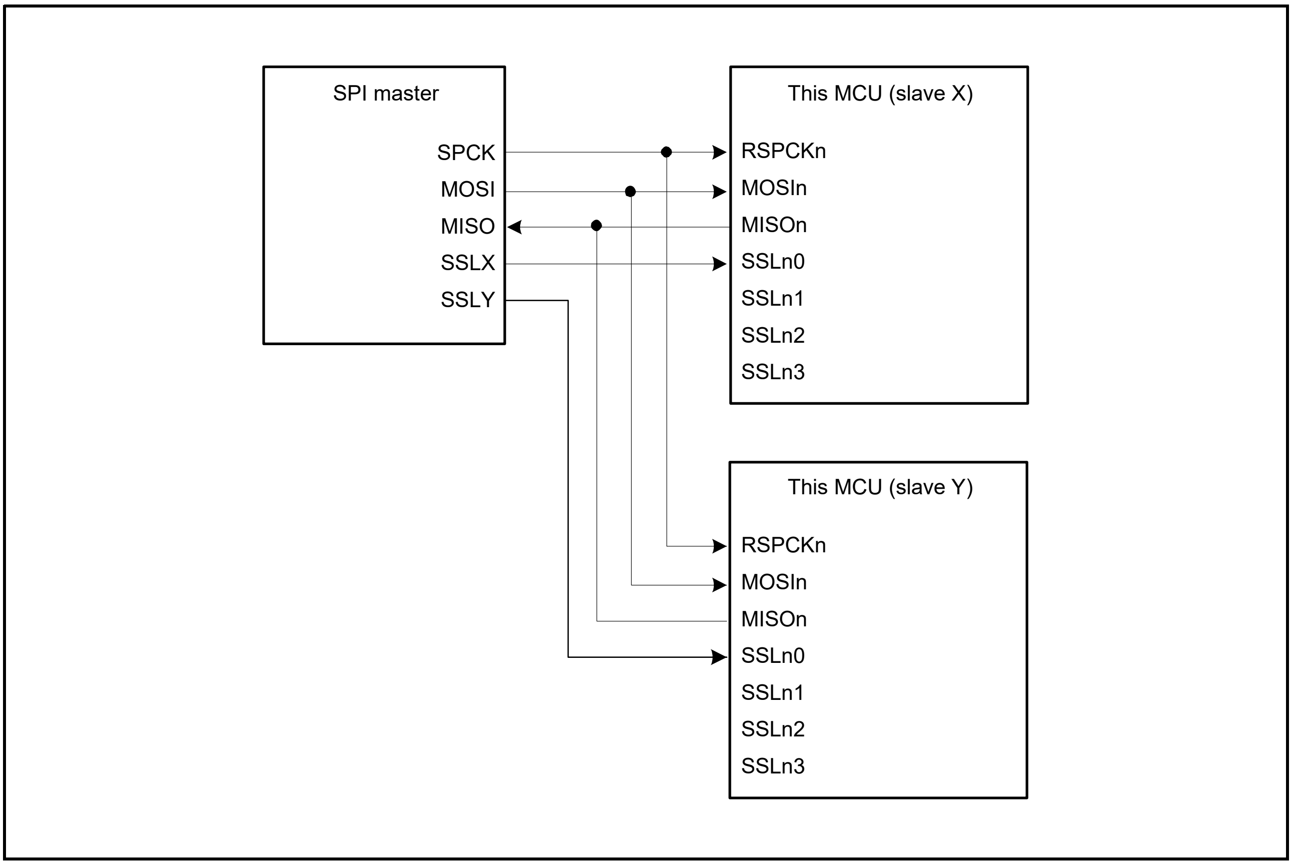


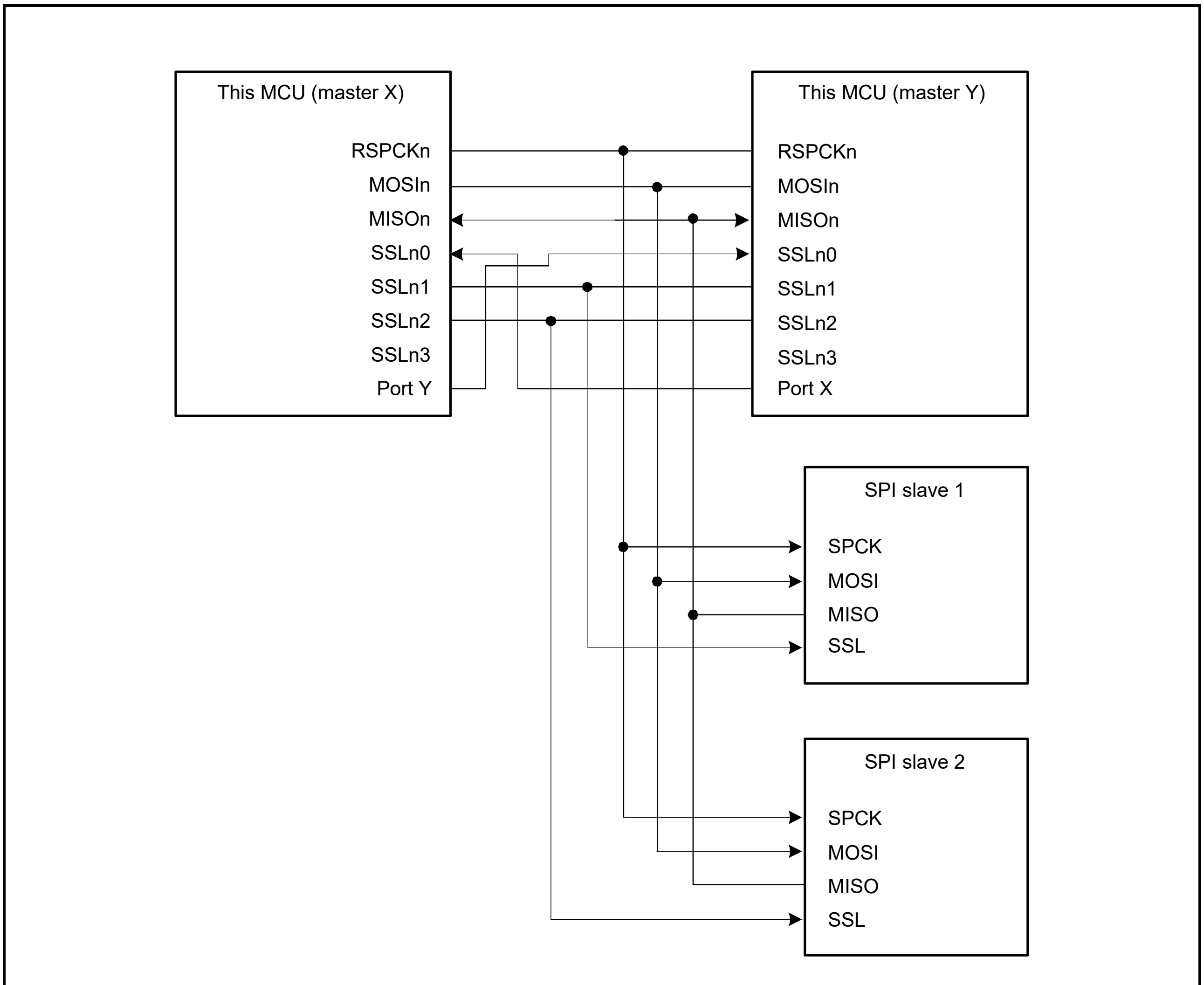
Figure 32.12 Single-master and multi-slave configuration example with the MCU as a slave

32.3.3.5 Multi-master and multi-slave with the MCU configured as a master

Figure 32.13 shows a multi-master and multi-slave SPI system configuration example where the MCU is a master. In the example shown in Figure 32.13, the SPI system includes 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. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives the MISO signal.


This MCU (master X)
RSPCKn
MOSIn
MISOn
SSLn0
SSLn1
SSLn2
SSLn3
Port Y
This MCU (master Y)
RSPCKn
MOSIn
MISOn
SSLn0
SSLn1
SSLn2
SSLn3
Port X
SPI slave 1
SPCK
MOSI
MISO
SSL
SPI slave 2
SPCK
MOSI
MISO
SSL

Note: SSLB2 is not available.

Figure 32.13 Multi-master and multi-slave configuration example with the MCU as a master

32.3.3.6 Master and slave in clock synchronous mode with the MCU configured as a master

Figure 32.14 shows a master and slave clock synchronous mode configuration where the MCU is a master. In the master and slave clock synchronous mode configuration, 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.

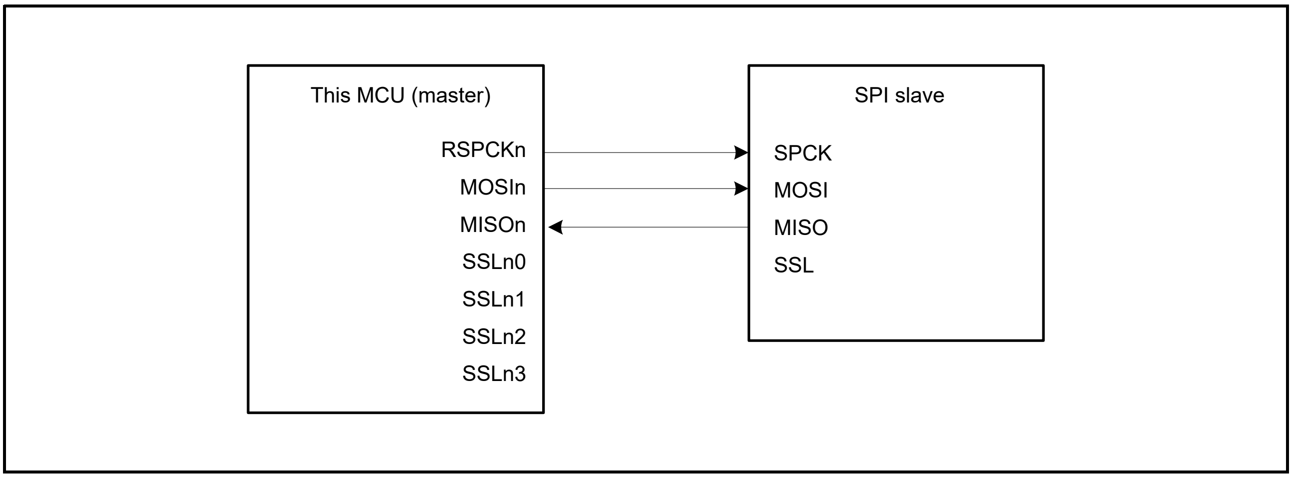

This MCU (master)
RSPCKn
MOSIn
MISOn
SSLn0
SSLn1
SSLn2
SSLn3
SPI slave
SPCK
MOSI
MISO
SSL

Figure 32.14 Configuration example of master and slave in clock synchronous mode with the MCU as a master

32.3.3.7 Master and slave in clock synchronous mode with the MCU configured as a slave

Figure 32.15 shows a master and slave in clock synchronous mode configuration example where the MCU is a slave. When the MCU is to operate as a slave in clock synchronous operation, the MCU (slave) drives the MISOn signal and the SPI master drives the SPCK and MOSI signals. The SSLn0 to SSLn3 of the MCU (slave) are not used.

The MCU (slave) can only execute serial transfer in the single-slave configuration when the SPCMDm.CPHA is set to 1.

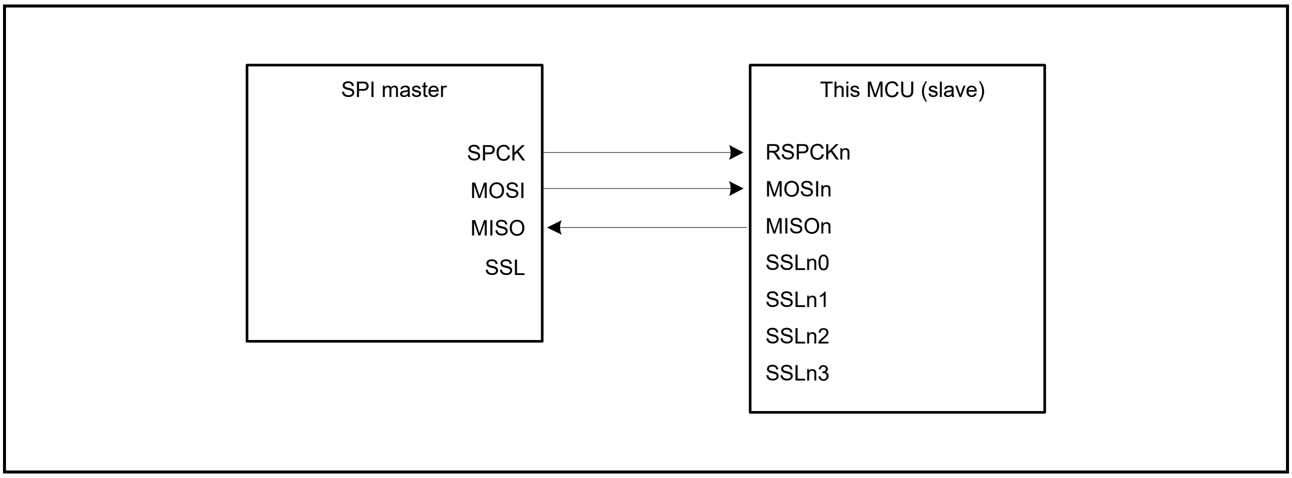

SPI master
SPCK
MOSI
MISO
SSL
This MCU (slave)
RSPCKn
MOSIn
MISOn
SSLn0
SSLn1
SSLn2
SSLn3

Figure 32.15 Configuration example of master and slave in clock synchronous mode with the MCU as a slave and CPHA = 1

32.3.4 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, SPPE, in the SPI Control Register 2 (SPCR2). Regardless of whether the ordering is MSB- 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) Data format with parity disabled

When parity is disabled, transmission or reception of data proceeds with the bit-length selected in the SPI data length setting bits in the SPI Command Register m (SPCMDm.SPB[3:0] for SPI0, SPCMD0.SPB[3:0] for SPI1).

(b) Data format with parity enabled

When parity is enabled, transmission or reception of data proceeds with the bit-length selected in the SPI data length setting bits in SPI Command Register m (SPCMDm.SPB[3:0] for SPI0, SPCMD0.SPB[3:0] for SPI1). In this case, however, the last bit is a parity bit.

- SPI0

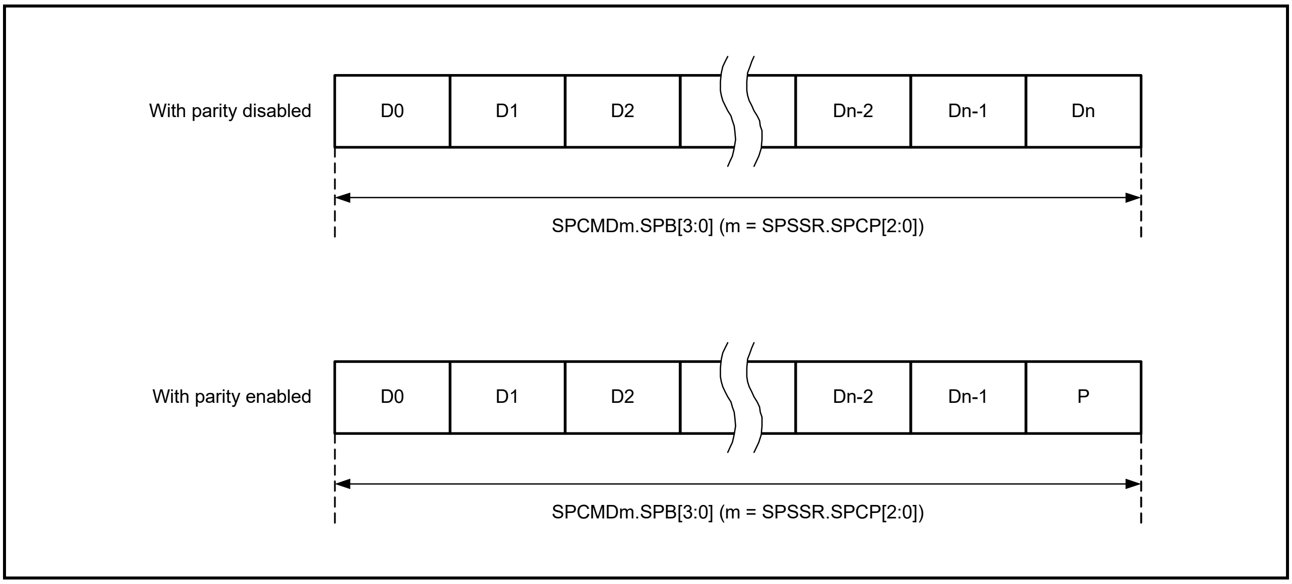

With parity disabled
D0
D1
D2
Dn-2
Dn-1
Dn
SPCMDm.SPB[3:0] (m = SPSSR.SPCP[2:0])
With parity enabled
D0
D1
D2
Dn-2
Dn-1
P
SPCMDm.SPB[3:0] (m = SPSSR.SPCP[2:0])

Figure 32.16 Data format with parity disabled and enabled (SPI0)

- SPI1

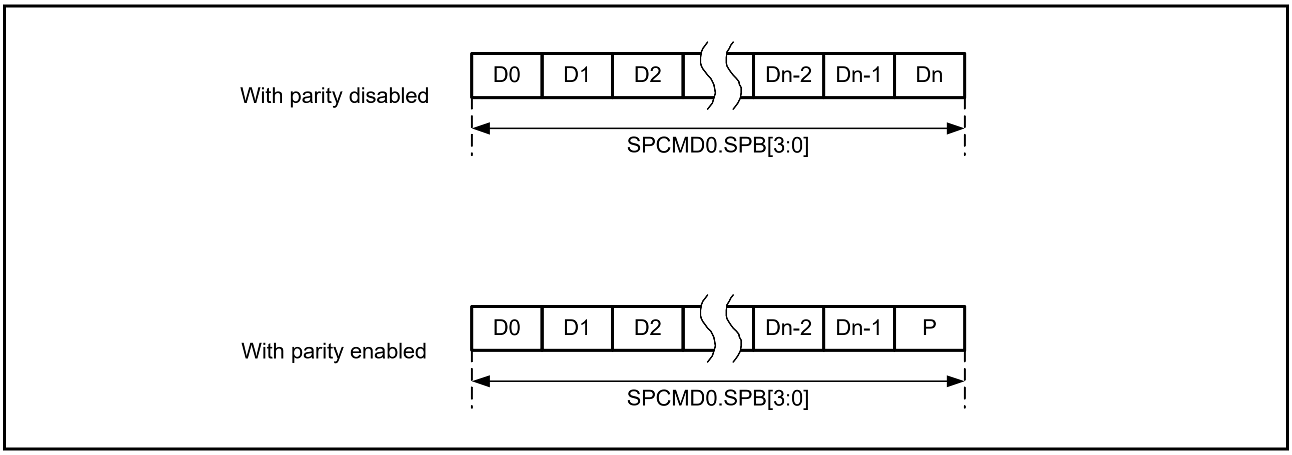

With parity disabled
D0
D1
D2
Dn-2
Dn-1
Dn
SPCMD0.SPB[3:0]
With parity enabled
D0
D1
D2
Dn-2
Dn-1
P
SPCMD0.SPB[3:0]

Figure 32.17 Data format with parity disabled and enabled (SPI1)

32.3.4.1 Operation when parity is disabled (SPCR2.SPPE = 0)

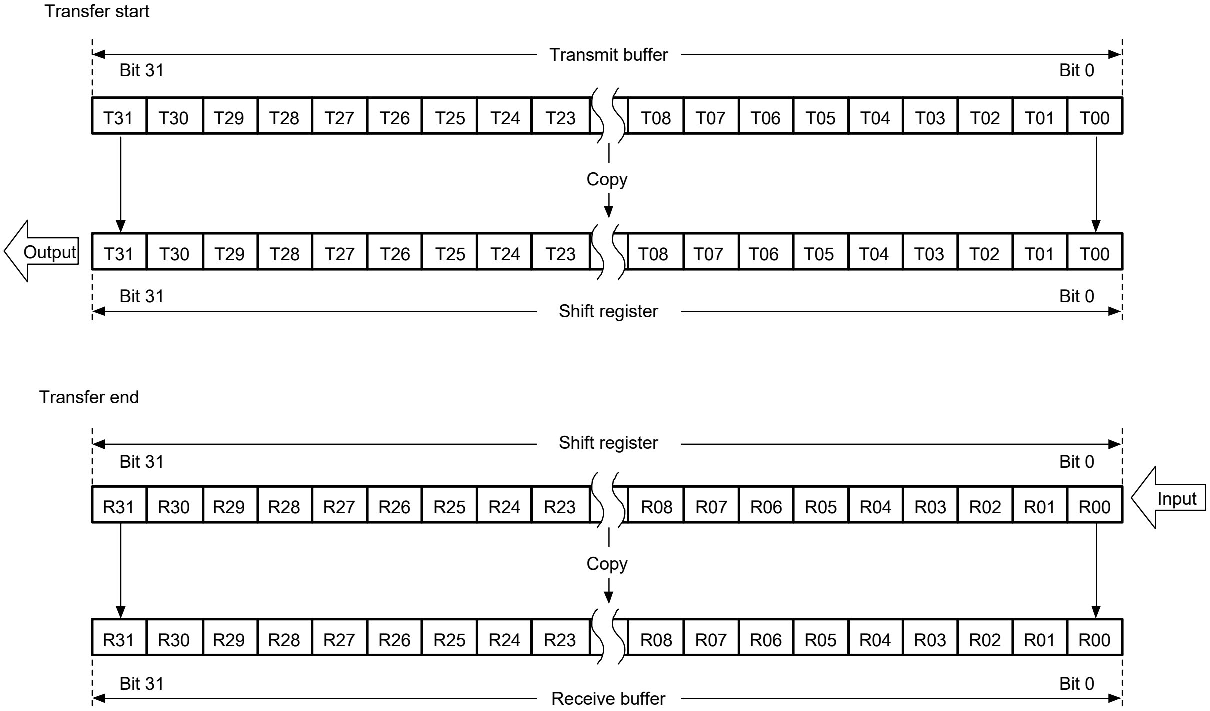
When parity is disabled, data for transmission is copied to the shift register with no prior processing. This section describes the connection between the SPI Data Register (SPDR/SPDR_HA) and the shift register in terms of the combination of MSB- or LSB-first order and data length.

(1) MSB-first transfer with 32-bit data

Figure 32.18 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 32 bits, and MSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission is shifted out from the shift register from T31 to T30, and continuing to T00.

In reception, received data is shifted in bit by bit through bit 0 of the shift register. When the R31 to R00 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.


Transfer start
Transmit buffer
Bit 31
Bit 0
T31 T30 T29 T28 T27 T26 T25 T24 T23 T08 T07 T06 T05 T04 T03 T02 T01 T00
Copy
Output
T31 T30 T29 T28 T27 T26 T25 T24 T23 T08 T07 T06 T05 T04 T03 T02 T01 T00
Bit 31
Bit 0
Shift register
Transfer end
Shift register
Bit 31
Bit 0
R31 R30 R29 R28 R27 R26 R25 R24 R23 R08 R07 R06 R05 R04 R03 R02 R01 R00
Input
Copy
R31 R30 R29 R28 R27 R26 R25 R24 R23 R08 R07 R06 R05 R04 R03 R02 R01 R00
Bit 31
Bit 0
Receive buffer

Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

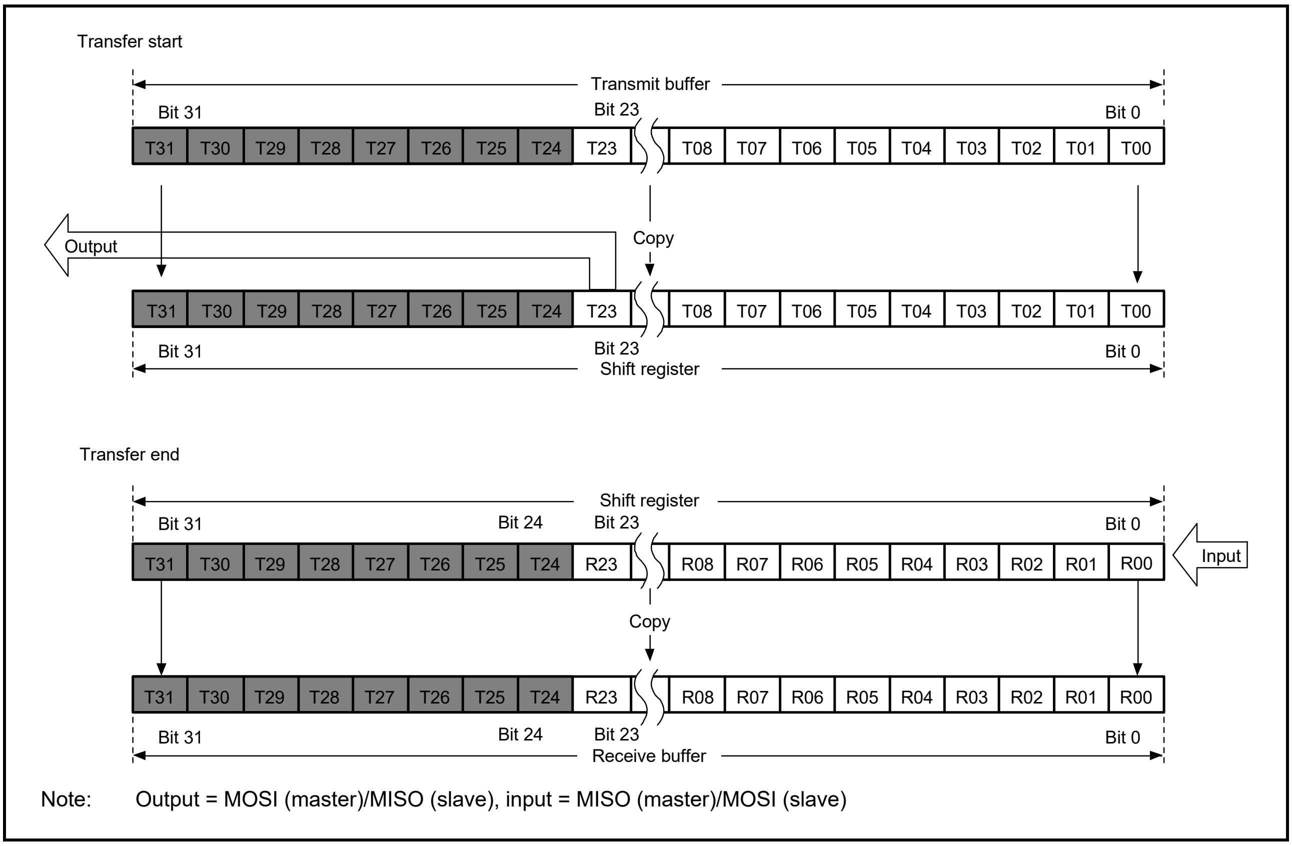
Figure 32.18 MSB-first transfer with 32-bit data and parity disabled

(2) MSB-first transfer with 24-bit data

Figure 32.19 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 24 bits for an example that is not 32 bits, and MSB-first selected.

In transmission, the lower 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission is shifted out from the shift register from T23 to T22, and continuing to T00.

In reception, received data is shifted in bit by bit through bit [0] of the shift register. When bits R23 to R00 are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. The upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the transmission results in 0 being inserted in the upper 8 bits of the receive buffer.


Transfer start
Transmit buffer
Bit 31
Bit 23
Bit 0
T31 T30 T29 T28 T27 T26 T25 T24 T23 T08 T07 T06 T05 T04 T03 T02 T01 T00
Output
Copy
T31 T30 T29 T28 T27 T26 T25 T24 T23 T08 T07 T06 T05 T04 T03 T02 T01 T00
Bit 31
Bit 23
Bit 0
Shift register
Transfer end
Shift register
Bit 31
Bit 24
Bit 23
Bit 0
T31 T30 T29 T28 T27 T26 T25 T24 R23 R08 R07 R06 R05 R04 R03 R02 R01 R00
Input
Copy
T31 T30 T29 T28 T27 T26 T25 T24 R23 R08 R07 R06 R05 R04 R03 R02 R01 R00
Bit 31
Bit 24
Bit 23
Bit 0
Receive buffer

Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

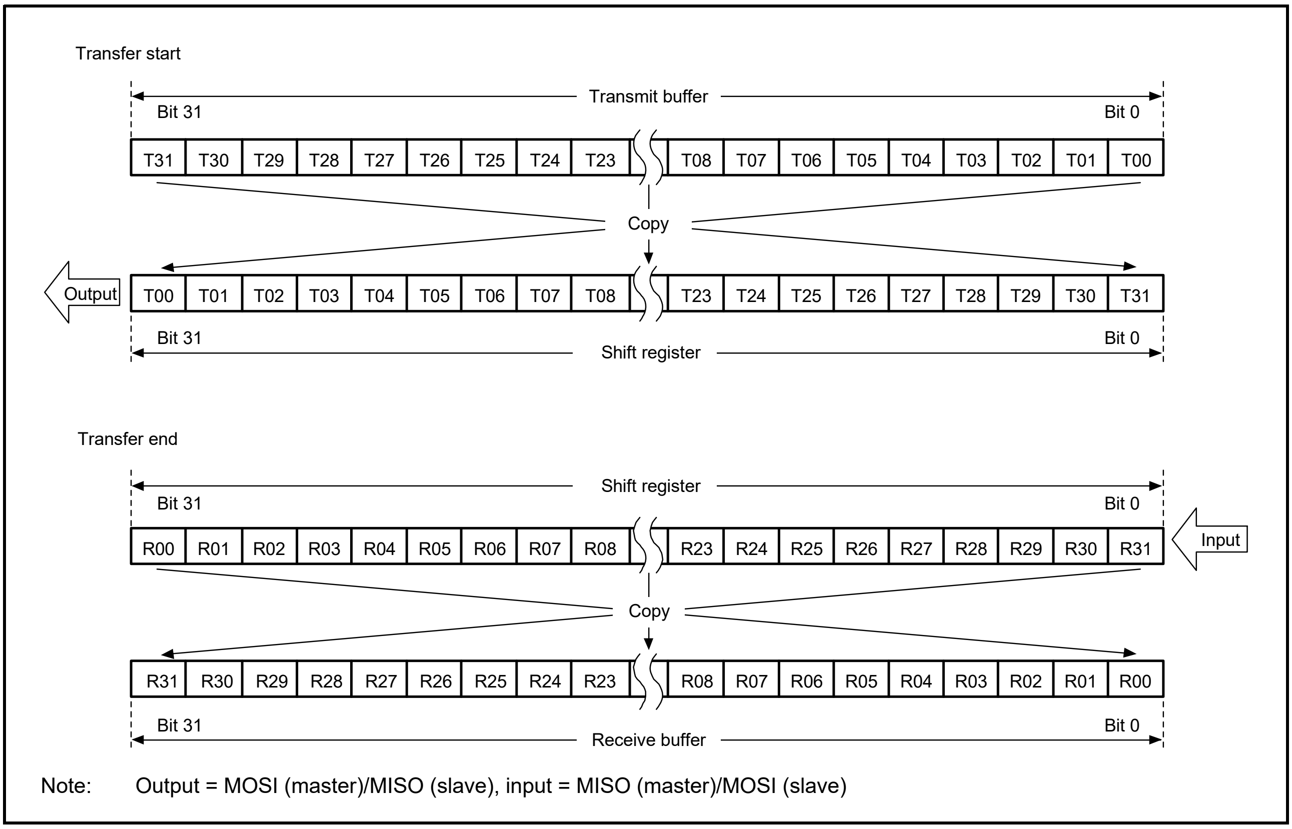
Figure 32.19 MSB-first transfer with 24-bit data and parity disabled

(3) LSB-first transfer with 32-bit data

Figure 32.20 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 32 bits, and LSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission is shifted out from the shift register 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 RSPCK cycles, the value in the shift register is copied to the receive buffer.


Transfer start
Transmit buffer
Bit 31
Bit 0
T31 T30 T29 T28 T27 T26 T25 T24 T23 T08 T07 T06 T05 T04 T03 T02 T01 T00
Copy
Output
T00 T01 T02 T03 T04 T05 T06 T07 T08 T23 T24 T25 T26 T27 T28 T29 T30 T31
Bit 31
Shift register
Bit 0
Transfer end
Shift register
Bit 31
Bit 0
R00 R01 R02 R03 R04 R05 R06 R07 R08 R23 R24 R25 R26 R27 R28 R29 R30 R31
Input
Copy
R31 R30 R29 R28 R27 R26 R25 R24 R23 R08 R07 R06 R05 R04 R03 R02 R01 R00
Bit 31
Receive buffer
Bit 0

Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 32.20 LSB-first transfer with 32-bit data and parity disabled

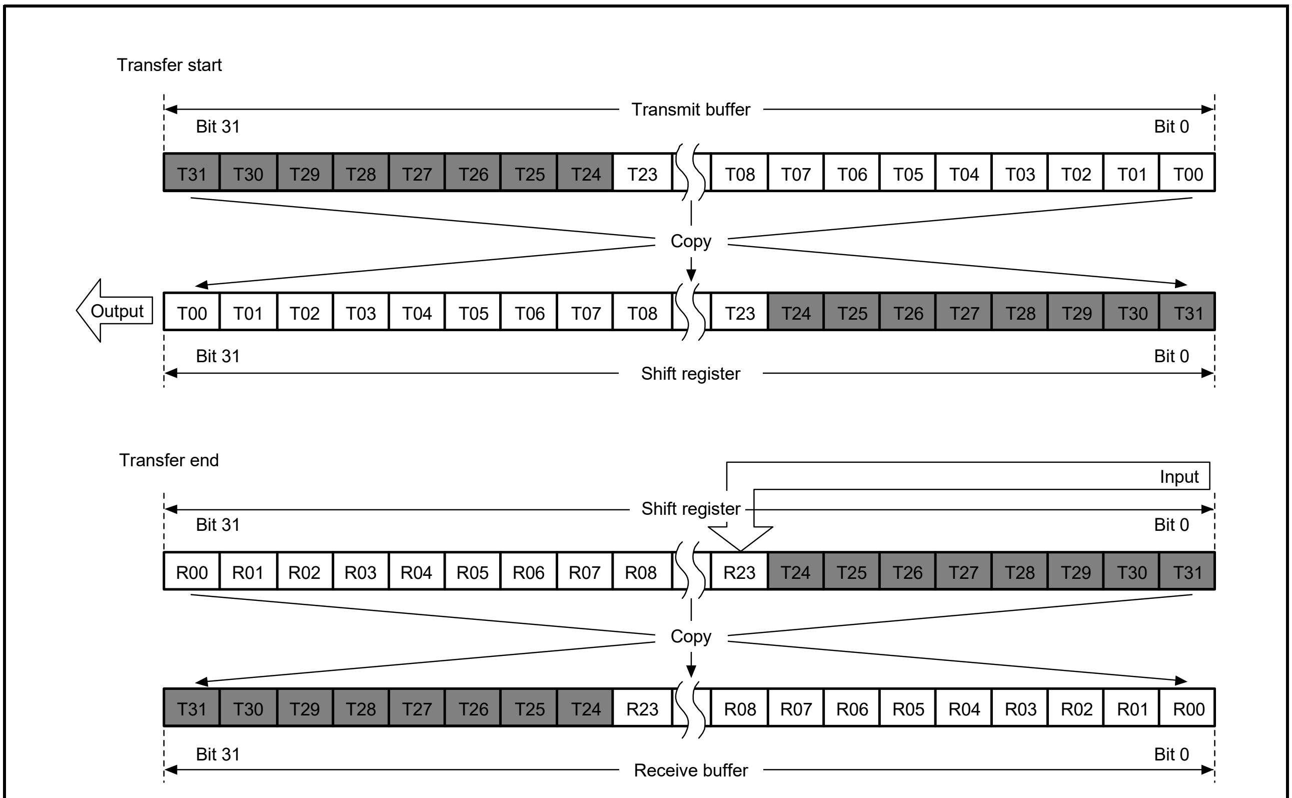
(4) LSB-first transfer with 24-bit data

Figure 32.21 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 24 bits for an example that is not 32 bits, and 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 RSPCK cycles, the value in the shift register is copied to the receive buffer.

The upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 during transmission results in 0 being inserted in the upper 8 bits of the receive buffer.


Transfer start
Transmit buffer
Bit 31
Bit 0
T31 T30 T29 T28 T27 T26 T25 T24 T23 T08 T07 T06 T05 T04 T03 T02 T01 T00
Copy
Output
T00 T01 T02 T03 T04 T05 T06 T07 T08 T23 T24 T25 T26 T27 T28 T29 T30 T31
Bit 31
Bit 0
Shift register
Transfer end
Input
Shift register
Bit 31
Bit 0
R00 R01 R02 R03 R04 R05 R06 R07 R08 R23 T24 T25 T26 T27 T28 T29 T30 T31
Copy
T31 T30 T29 T28 T27 T26 T25 T24 R23 R08 R07 R06 R05 R04 R03 R02 R01 R00
Bit 31
Bit 0
Receive buffer

Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 32.21 LSB-first transfer with 24-bit data and parity disabled

32.3.4.2 When parity is enabled (SPCR2.SPPE = 1)

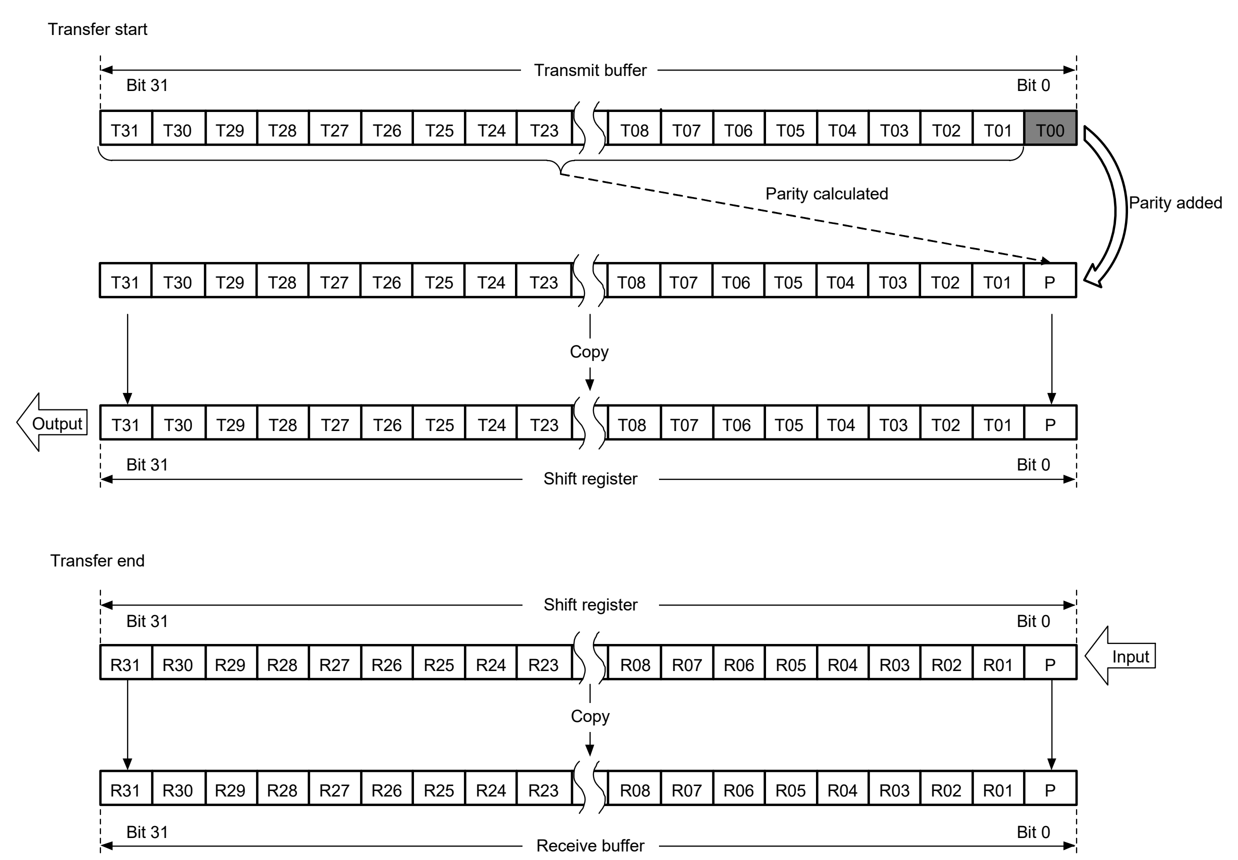
When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

(1) MSB-first transfer with 32-bit data

Figure 32.22 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an SPI data length of 32 bits, and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole value is copied to the shift register. Data is transmitted 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 RSPCK cycles, the value in the shift register is copied to the receive buffer. After copying data to the shift register, the data from R31 to P is checked for parity.


Transfer start
Transmit buffer
Bit 31
Bit 0
T31 T30 T29 T28 T27 T26 T25 T24 T23 T08 T07 T06 T05 T04 T03 T02 T01 T00
Parity calculated
Parity added
T31 T30 T29 T28 T27 T26 T25 T24 T23 T08 T07 T06 T05 T04 T03 T02 T01 P
Copy
Output
Shift register
Transfer end
Input
R31 R30 R29 R28 R27 R26 R25 R24 R23 R08 R07 R06 R05 R04 R03 R02 R01 P
Receive buffer

Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

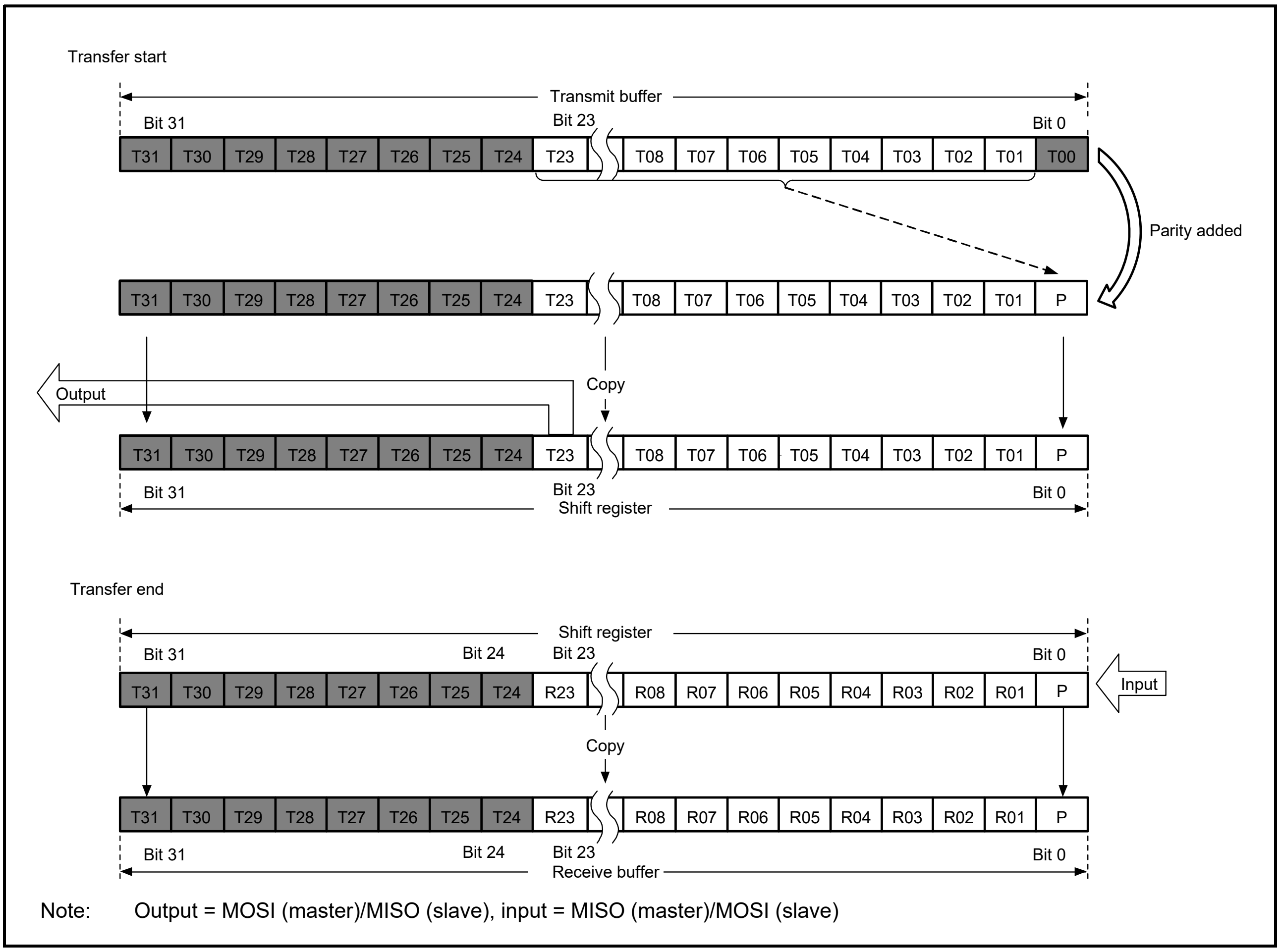
Figure 32.22 MSB-first transfer with 32-bit data and parity enabled

(2) MSB-first transfer with 24-bit data

Figure 32.23 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, a SPI data length of 24 bits 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 from 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 RSPCK cycles, the value in the shift register is copied to the receive buffer. After data is copied 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 during transmission results in 0 being inserted in the upper 8 bits of the receive buffer.


Transfer start
Transmit buffer
Bit 31
Bit 23
Bit 0
T31 T30 T29 T28 T27 T26 T25 T24 T23 T08 T07 T06 T05 T04 T03 T02 T01 T00
Parity added
T31 T30 T29 T28 T27 T26 T25 T24 T23 T08 T07 T06 T05 T04 T03 T02 T01 P
Output
Copy
T31 T30 T29 T28 T27 T26 T25 T24 T23 T08 T07 T06 T05 T04 T03 T02 T01 P
Bit 31
Bit 23
Shift register
Bit 0
Transfer end
Shift register
Bit 31
Bit 24
Bit 23
Bit 0
T31 T30 T29 T28 T27 T26 T25 T24 R23 R08 R07 R06 R05 R04 R03 R02 R01 P
Input
Copy
T31 T30 T29 T28 T27 T26 T25 T24 R23 R08 R07 R06 R05 R04 R03 R02 R01 P
Bit 31
Bit 24
Bit 23
Receive buffer
Bit 0

Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

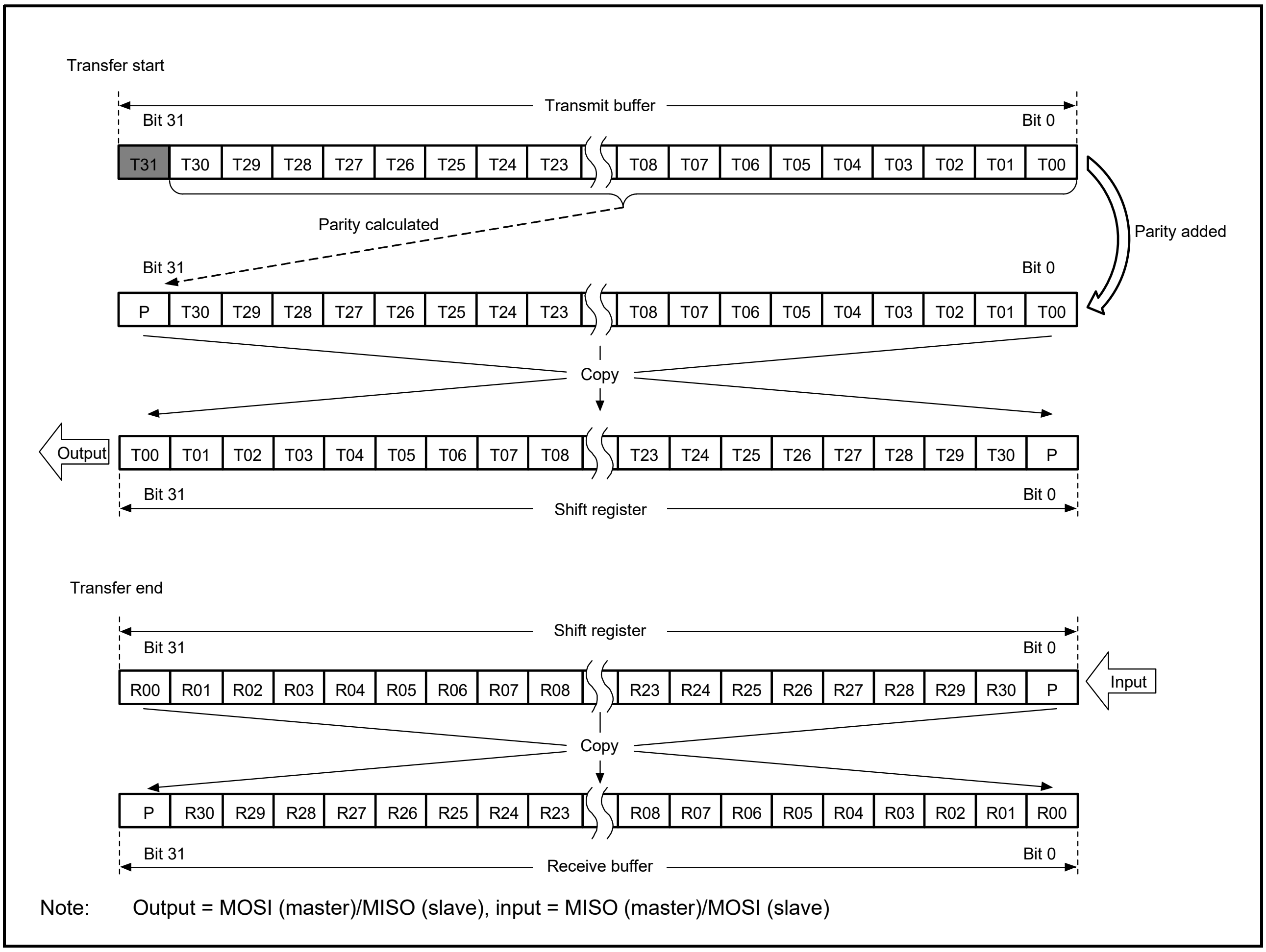
Figure 32.23 MSB-first transfer with 24-bit data and parity enabled

(3) LSB-first transfer with 32-bit data

Figure 32.24 shows the operation 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 RSPCK cycles, the value in the shift register is copied to the receive buffer. After data is copied to the shift register, the data from R00 to P is checked for parity.


Transfer start
Transmit buffer
Bit 31
Bit 0
T31 T30 T29 T28 T27 T26 T25 T24 T23 T08 T07 T06 T05 T04 T03 T02 T01 T00
Parity calculated
Parity added
P T30 T29 T28 T27 T26 T25 T24 T23 T08 T07 T06 T05 T04 T03 T02 T01 T00
Copy
Output
T00 T01 T02 T03 T04 T05 T06 T07 T08 T23 T24 T25 T26 T27 T28 T29 T30 P
Shift register
Transfer end
Input
R00 R01 R02 R03 R04 R05 R06 R07 R08 R23 R24 R25 R26 R27 R28 R29 R30 P
P R30 R29 R28 R27 R26 R25 R24 R23 R08 R07 R06 R05 R04 R03 R02 R01 R00
Receive buffer

Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

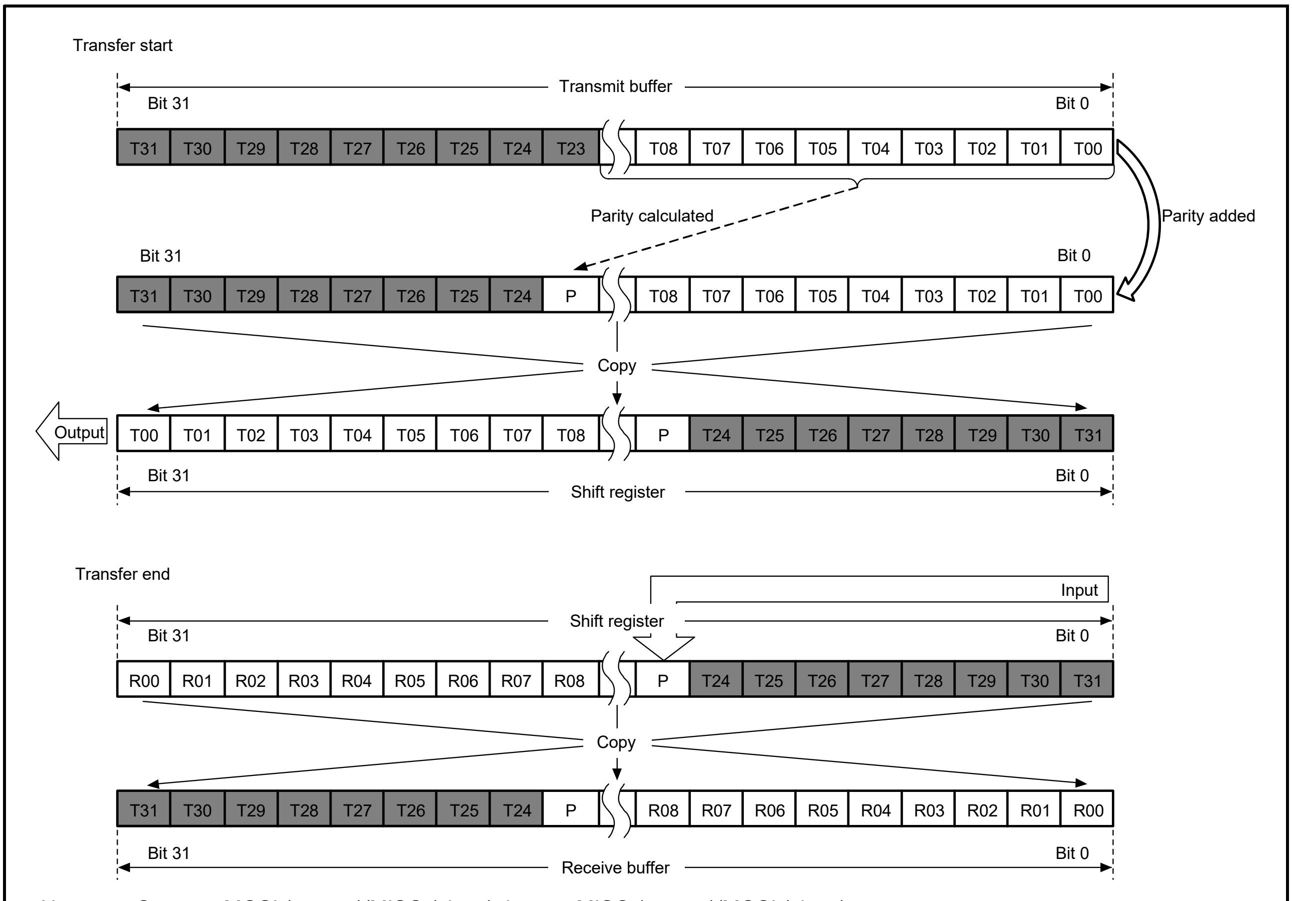
Figure 32.24 LSB-first transfer with 32-bit data and parity enabled

(4) LSB-first transfer with 24-bit data

Figure 32.25 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, a SPI data length of 24 bits 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 from 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 RSPCK cycles, the value in the shift register is copied to the receive buffer. After data is copied 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 during transmission results in 0 being inserted in the upper 8 bits of the receive buffer.


Transfer start
Transmit buffer
Bit 31
Bit 0
T31 T30 T29 T28 T27 T26 T25 T24 T23
T08 T07 T06 T05 T04 T03 T02 T01 T00
Parity calculated
Parity added
Bit 31
Bit 0
T31 T30 T29 T28 T27 T26 T25 T24 P
T08 T07 T06 T05 T04 T03 T02 T01 T00
Copy
Output
T00 T01 T02 T03 T04 T05 T06 T07 T08
P T24 T25 T26 T27 T28 T29 T30 T31
Bit 31
Bit 0
Shift register
Transfer end
Input
Shift register
Bit 31
Bit 0
R00 R01 R02 R03 R04 R05 R06 R07 R08
P T24 T25 T26 T27 T28 T29 T30 T31
Copy
T31 T30 T29 T28 T27 T26 T25 T24 P
R08 R07 R06 R05 R04 R03 R02 R01 R00
Bit 31
Bit 0
Receive buffer

Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 32.25 LSB-first transfer with 24-bit data and parity enabled

32.3.5 Transfer Format

32.3.5.1 CPHA = 0

Figure 32.26 shows an example transfer format for the serial transfer of 8-bit data when SPCMDm.CPHA 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 32.26, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when SPCMDm.CPOL is 0 and RSPCKn (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI settings. For details, see section 32.3.2, Controlling the SPI Pins.

When SPCMDm.CPHA 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 cycle 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 32.3.10.1, Master mode operation.

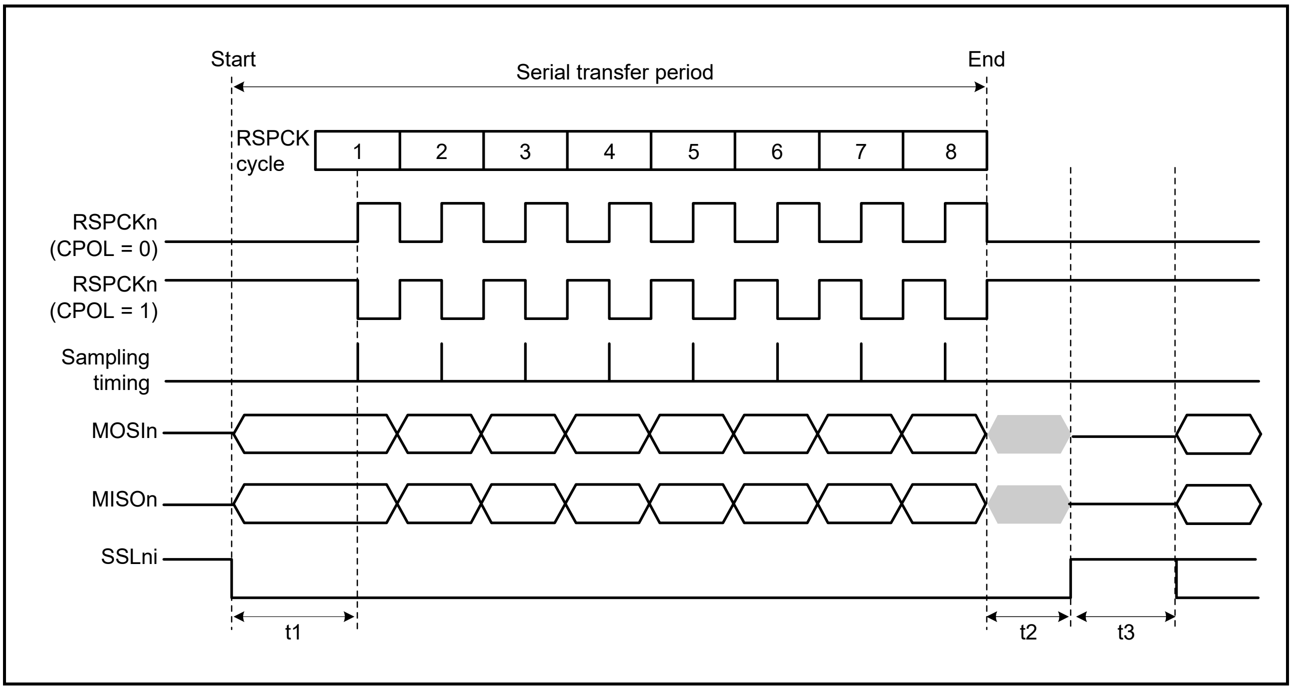


Figure 32.26 SPI transfer format with CPHA = 0

32.3.5.2 CPHA = 1

Figure 32.27 shows an example 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 RSPCKn, MOSIn, and MISOn signals handle communications. In Figure 32.27, RSPCK (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0 and RSPCK 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 32.3.2, Controlling the 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 cycle 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 32.3.10.1, Master mode operation.

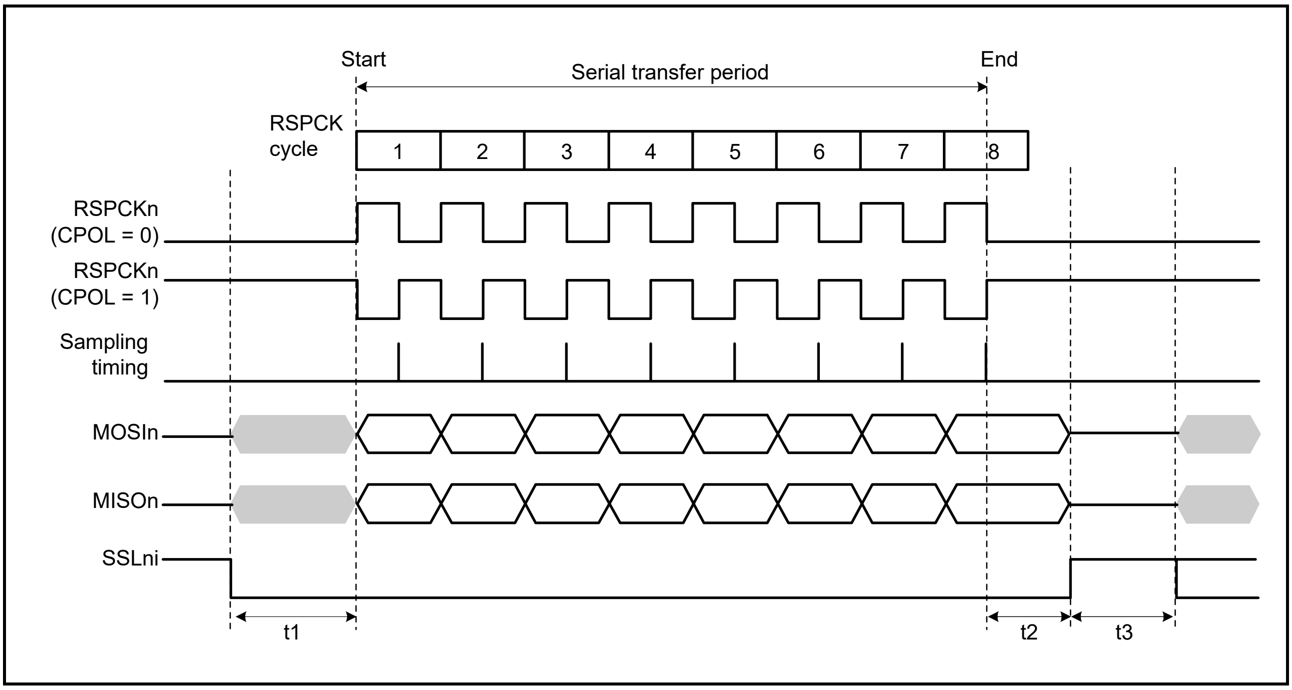

Start
Serial transfer period
End
RSPCK cycle
1
2
3
4
5
6
7
8
RSPCKn (CPOL = 0)
RSPCKn (CPOL = 1)
Sampling timing
MOSIn
MISOn
SSLni
t1
t2
t3

Figure 32.27 SPI transfer format with CPHA = 1

32.3.6 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 register accesses shown in Figure 32.28 and Figure 32.29 indicate the condition of access to the SPDR/SPDR_HA register, where W denotes a write cycle.

32.3.6.1 Full-duplex synchronous serial communications (SPCR.TXMD = 0)

Figure 32.28 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 for SPI0, and in which the SPCMD0.CPHA bit is 1 and the SPCMD0.CPOL bit is 0 for SPI1. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, meaning the number of transferred bits.

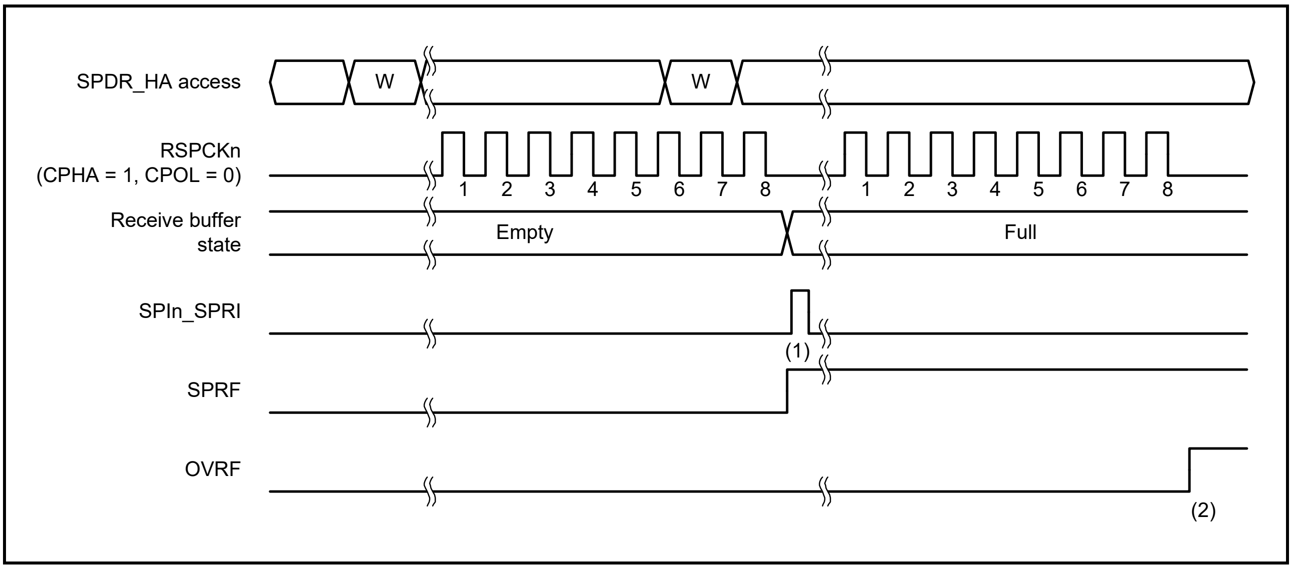

SPDR_HA access
W
W
RSPCKn (CPHA = 1, CPOL = 0)
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
Receive buffer state
Empty
Full
SPIn_SPRI
(1)
SPRF
OVRF
(2)

Figure 32.28 Operation example when SPCR.TXMD = 0

The operation of the flags at timings shown in (1) and (2) in Figure 32.28 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 SPI sets the SPSR.SPRF flag to 1, and the received data is copied from the shift register to the receive buffer.

(2) When a serial transfer ends with the SPDR_HA receive buffer 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.

32.3.6.2 Transmit-only operations (SPCR.TXMD = 1)

Figure 32.29 shows an operation example where the communications operating mode select bit (SPCR.TXMD) is set to 1. In the example in Figure 32.29, 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 for SPI0, and in which the SPCMD0.CPHA bit is 1 and the SPCMD0.CPOL bit is 0 for SPI1. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, meaning the number of transferred bits.

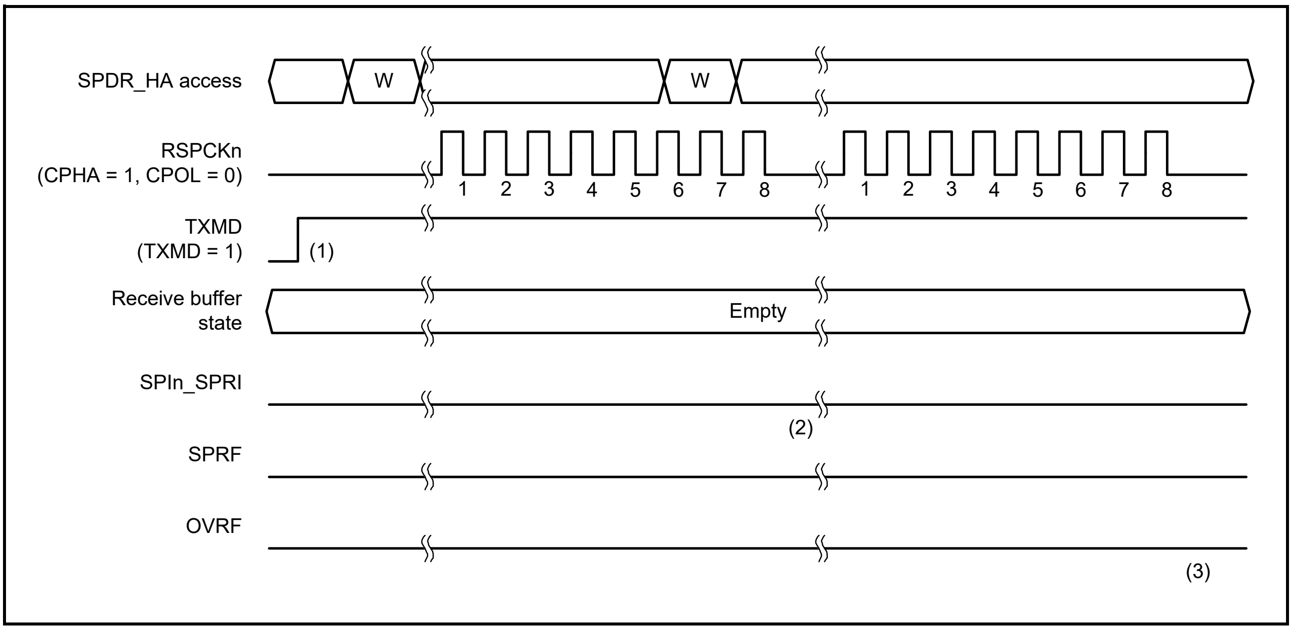


Figure 32.29 Operation example when SPCR.TXMD = 1

The operation of the flags at timings shown in (1) to (3) in Figure 32.29 is as follows:

(1) Make sure there is no data left in the receive buffer (SPSR.SPRF flag = 0) and the SPSR.OVRF flag is 0 before entering the transmit-only operation mode (SPCR.TXMD = 1).

(2) When a serial transfer ends with the receive buffer of SPDR_HA empty, if the transmit-only mode is selected (SPCR.TXMD = 1), the SPSR.SPRF flag remains 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 remains 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 but does not receive data. Therefore, the SPSR.SPRF and SPSR.OVRF flags remain 0 at timings (1) to (3).

32.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts

Figure 32.30 and Figure 32.31 show operation examples of the transmit buffer empty interrupt (SPIn_SPTI) and the receive buffer full interrupt (SPIn_SPRI). The register accesses shown in these figures indicate the conditions of access to the SPDR_HA register, where W denotes a write cycle, and R a read cycle. In Figure 32.26, 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 for SPI0, and in which the SPCR.TXMD bit is 0, the SPCMD0.CPHA bit is 0, and the

SPCMD0.CPOL bit is 0 for SPI1.

In Figure 32.27, 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 for SPI0, and in which the SPCR.TXMD bit is 0, the SPCMD0.CPHA bit is 1, and the SPCMD0.CPOL bit is 0 for SPI1. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, meaning the number of transferred bits.

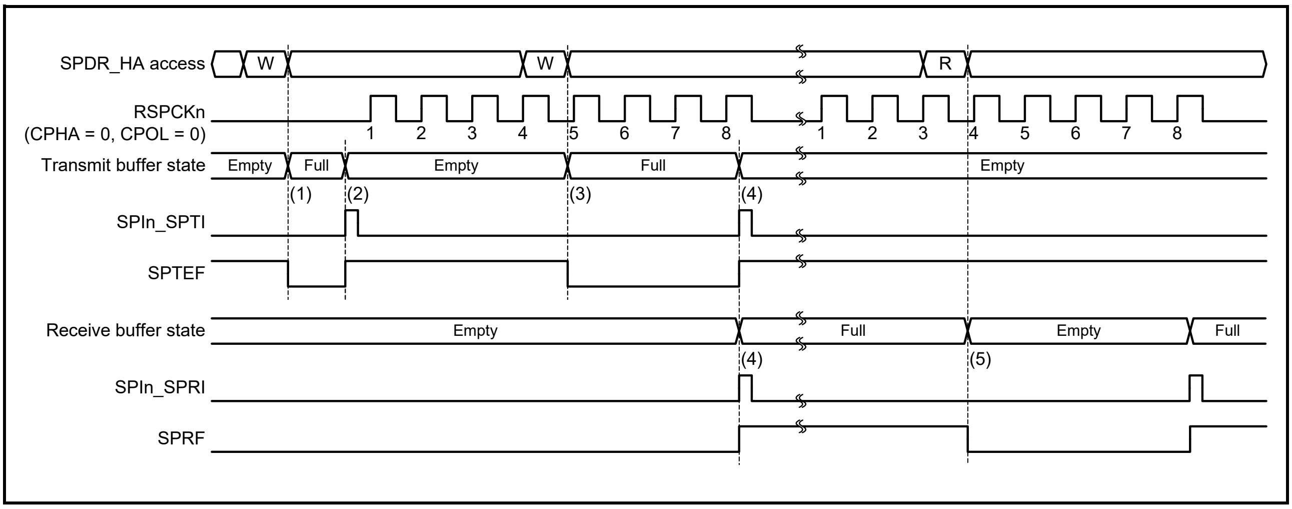

SPDR_HA access
W
W
R
RSPCKn
(CPHA = 0, CPOL = 0)
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
Transmit buffer state
Empty
Full
Empty
Full
Empty
(1)
(2)
(3)
(4)
SPIn_SPTI
SPTEF
Receive buffer state
Empty
Full
Empty
Full
(4)
(5)
SPIn_SPRI
SPRF

Figure 32.30 Operation example of SPIn_SPTI and SPIn_SPRI interrupts when CPHA = 0 and CPOL = 0

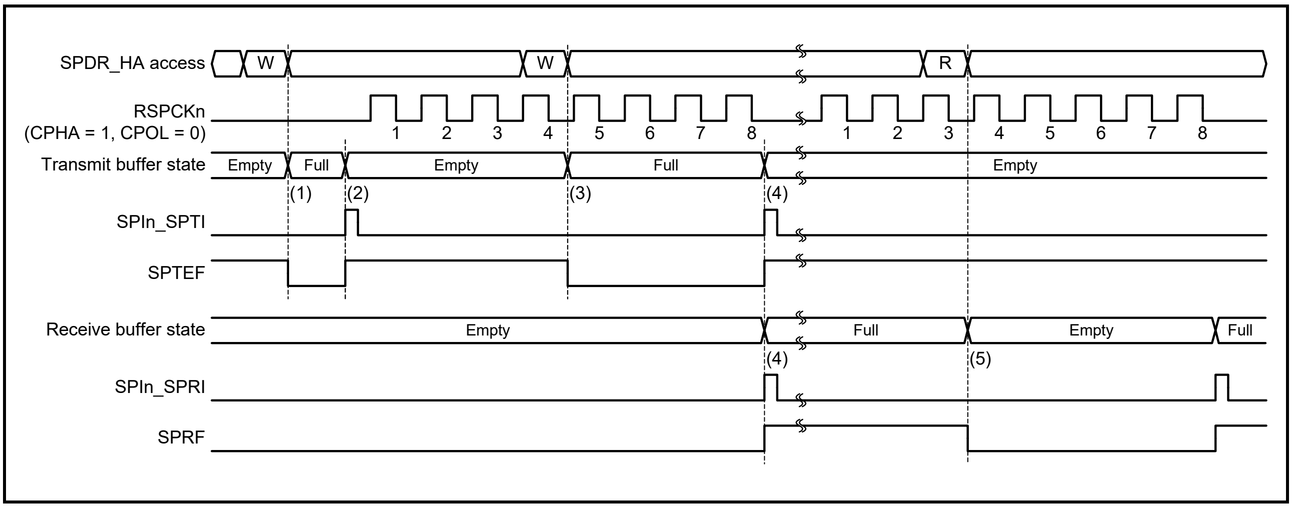

SPDR_HA access
W
W
R
RSPCKn
(CPHA = 1, CPOL = 0)
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
Transmit buffer state
Empty
Full
Empty
Full
Empty
(1)
(2)
(3)
(4)
SPIn_SPTI
SPTEF
Receive buffer state
Empty
Full
Empty
Full
(4)
(5)
SPIn_SPRI
SPRF

Figure 32.31 Operation example of SPIn_SPTI and SPIn_SPRI interrupts when CPHA = 1 and CPOL = 0

The operation of the SPI at timings shown in (1) to (5) in Figure 32.31 is as follows:

(1) When transmit data is written to the SPDR_HA register with the transmit buffer of SPDR_HA empty and data for the next transfer 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 data in the transmit buffer to the shift register, 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 32.3.10, SPI Operation, and section 32.3.11, Clock Synchronous Operation.

(3) When transmit data is written to the SPDR_HA register either by the transmit buffer empty interrupt routine, or by the processing of transmit buffer empty using SPTEF flag, the SPI writes data to the transmit buffer and clears the SPTEF flag to 0. Because the serially transferred data 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 the 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 by the processing of the receive buffer full interrupt using the SPRF flag, the receive data can be read.

If SPDR_HA is written to when the transmit buffer holds untransmitted data (SPTEF = 0), the SPI does not update the data in the transmit buffer. When writing to SPDR_HA, make sure 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 (SPCR.SPE = 0), set the SPTIE bit to 0.

When serial transfer ends with the receive buffer full (SPRF = 1), the SPI does not copy data from the shift register to the receive buffer, and it detects an overrun error (see section 32.3.8, 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 IELSRj.IR flags in the ICU, where j is the interrupt vector number, can be used to confirm the states of the transmission and reception buffers. Similarly, the SPTEF and the 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.

32.3.8 Error Detection

In normal SPI serial transfer, data written to the transmit buffer of SPDR/SPDR_HA is transmitted, and 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 a serial transfer.

If an abnormal transfer occurs, the SPI detects the event as an underrun error, overrun error, parity error, or mode fault error. Table 32.8 shows the relationship between non-normal transfer operations and the SPI error detection function.

Table 32.8 Relationship between non-normal transfer operations and SPI error detection function (1 of 2)

Operation	Occurrence condition	SPI operation	Error detection
1	SPDR/SPDR_HA is written when the transmit buffer is full	• The contents of the transmit buffer are kept • Write data is missing.	None
2	SPDR/SPDR_HA is read when the receive buffer is empty	The contents of the receive buffer and previously received data are output	None
3	Serial transfer is started in slave mode when the SPI is not able to transmit data	• Serial transfer is suspended • Transmit or receive data is missing • Driving of the 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 • Receive data is missing.	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, and 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, and SSLn1 to SSLn3 output signals is stopped • SPI function is disabled.	Mode fault error

Table 32.8 Relationship between non-normal transfer operations and SPI error detection function (2 of 2)

Operation	Occurrence condition	SPI operation	Error detection
8	The SSLn0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended • Missing transmit/receive data • Driving of the MISOn output signal is stopped • SPI function is disabled.	Mode fault error

In operation 1 described in Table 32.8, the SPI does not detect an error. To prevent data omission during writes to SPDR/SPDR_HA, the writes to SPDR/SPDR_HA must be executed using a transmit buffer empty interrupt request (when 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 32.3.8.4, Underrun errors.
- Overrun errors, indicated in operation 4, see section 32.3.8.1, Overrun errors.
- Parity errors, indicated in operation 5, see section 32.3.8.2, Parity errors.
- Mode fault errors, indicated in operations 6 to 8, see section 32.3.8.3, Mode fault errors.
- Transmit and receive interrupts, see section 32.3.7, Transmit Buffer Empty and Receive Buffer Full Interrupts.

32.3.8.1 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 before the error occurrence is saved in the receive buffer. To set the OVRF flag to 0, write 0 to it after the CPU reads SPSR with the OVRF flag set to 1.

Figure 32.32 shows an example operation of OVRF and SPRF flags. The SPSR and SPDR_HA accesses shown in Figure 32.32 indicate the condition of accesses to the 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.

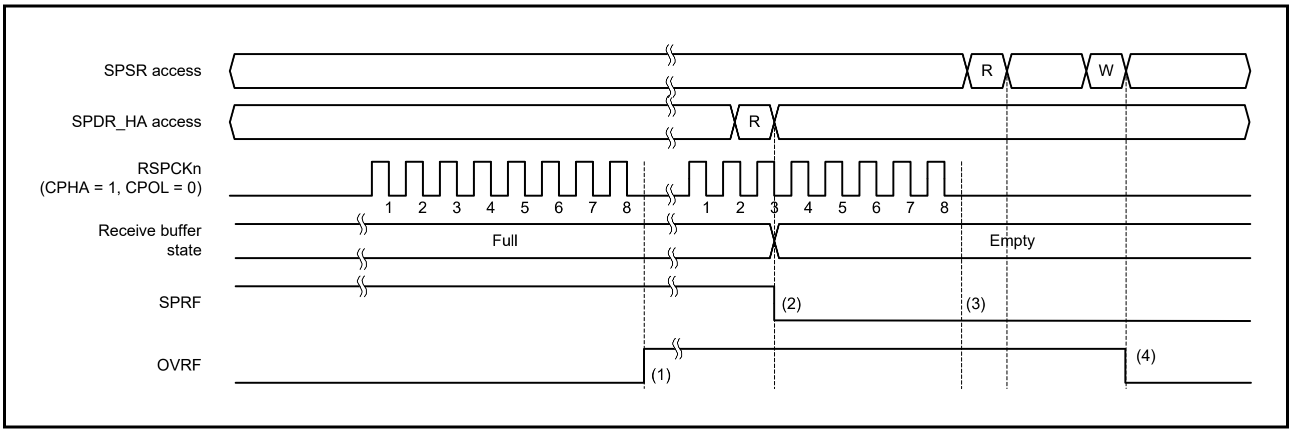


Figure 32.32 Operation example of OVRF and SPRF flags

The operation of the flags at the timing shown in (1) to (4) in Figure 32.32 is as follows:

(1) If a serial transfer terminates with the SPRF flag set to 1 (the receive buffer full), the SPI detects an overrun error and sets the OVRF flag to 1. The SPI does not copy data in the shift register to the receive buffer. Even if the SPPE bit is 1, parity errors are not detected. In master mode for SPI0, the SPI copies the value of the SPCMDm pointer to the SPSSR.SPECM[2:0] bits.

(2) When SPDR/SPDR_HA is read, the SPI outputs the data in the receive buffer. The SPRF flag is then set to 0. The emptying of the receive buffer does not set the OVRF flag to 0.

(3) If the serial transfer ends with the OVRF flag set to 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 when the SPPE bit is 1, parity errors are not detected. When in master mode for SPI0, the SPI does not update the SPSSR.SPECM[2:0] bits. When an overrun error occurs 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 occurrence of an overrun error can be checked 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/SPDR_HA is read. When the SPI is in master mode for SPI0, the value of the SPCMDm pointer at the error occurrence 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 32.33 and Figure 32.34 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

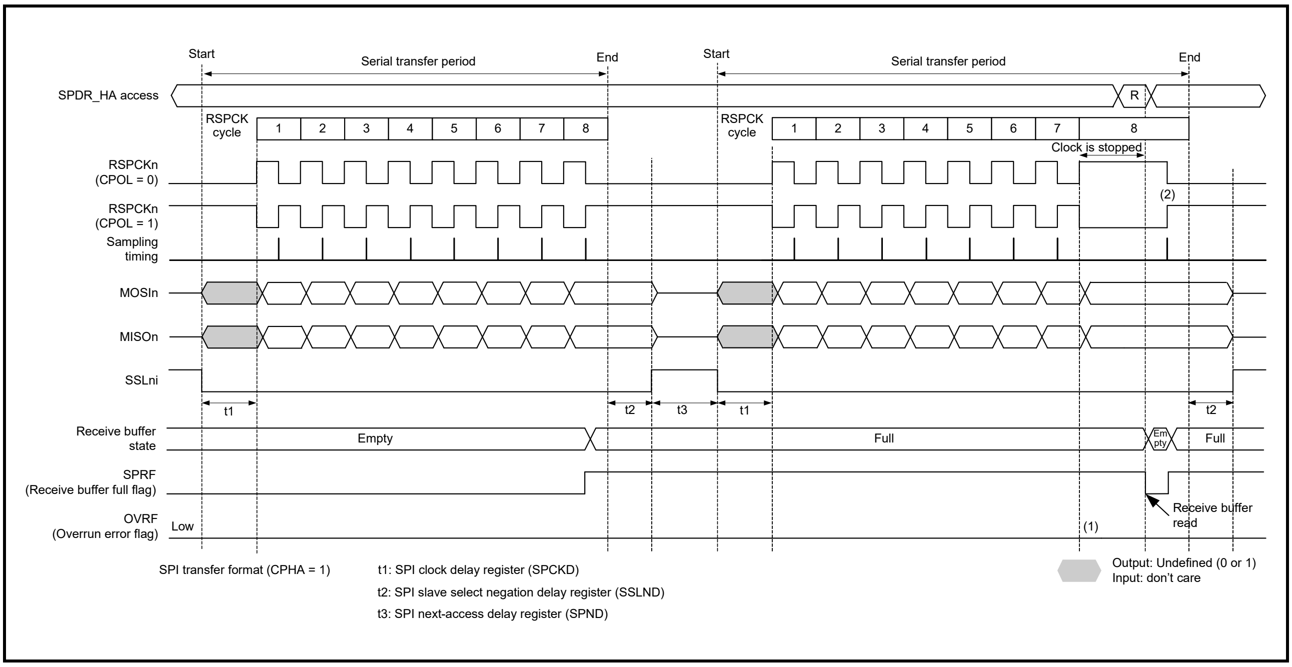

Start
Serial transfer period
End
Start
Serial transfer period
End
SPDR_HA access
R
RSPCK cycle
1
2
3
4
5
6
7
8
RSPCK cycle
1
2
3
4
5
6
7
8
Clock is stopped
RSPCKn (CPOL = 0)
RSPCKn (CPOL = 1)
(2)
Sampling timing
MOSIn
MISOn
SSLni
t1
t2
t3
t1
t2
Receive buffer state
Empty
Full
Em pty
Full
SPRF (Receive buffer full flag)
OVRF (Overrun error flag)
Low
(1)
Receive buffer read
SPI transfer format (CPHA = 1)
t1: SPI clock delay register (SPCKD)
t2: SPI slave select negation delay register (SSLND)
t3: SPI next-access delay register (SPND)
Output: Undefined (0 or 1)
Input: don’t care

Figure 32.33 Clock stop waveform when serial transfer continues while receive buffer is full in master mode with CPHA = 1

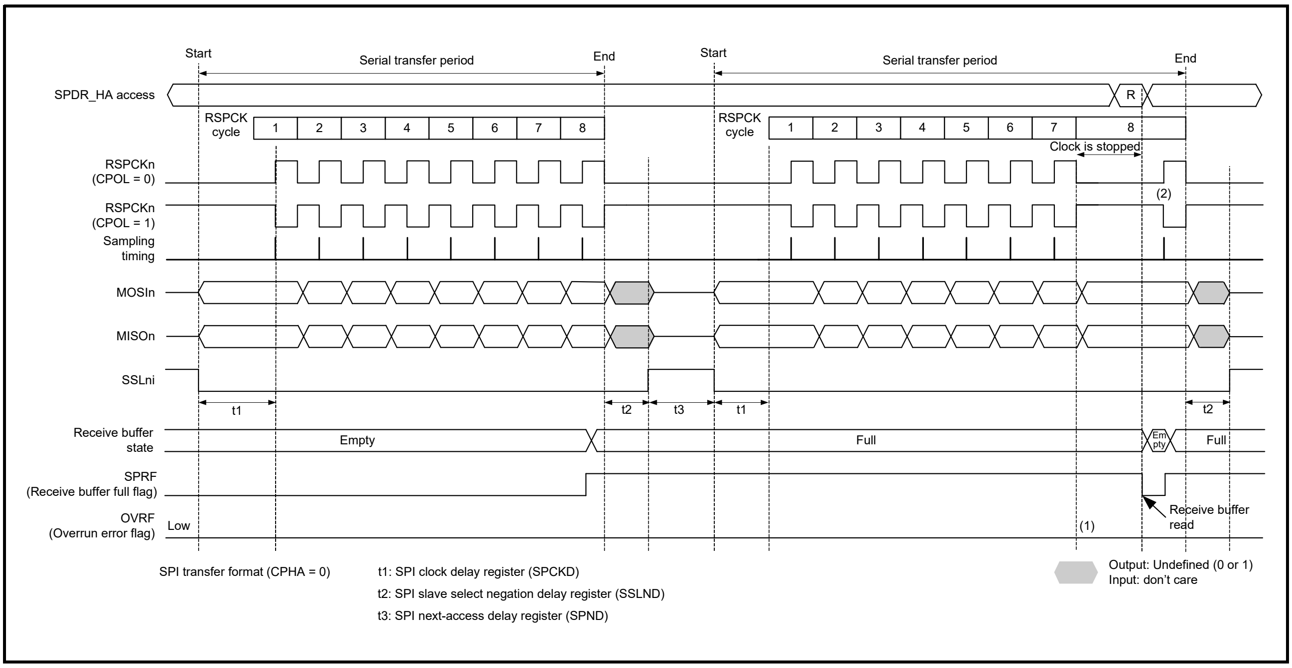

Start
Serial transfer period
End
Start
Serial transfer period
End
SPDR_HA access
R
RSPCK cycle
1 2 3 4 5 6 7 8
RSPCK cycle
1 2 3 4 5 6 7 8
Clock is stopped
RSPCKn (CPOL = 0)
RSPCKn (CPOL = 1)
(2)
Sampling timing
MOSIn
MISOn
SSLni
t1
t2
t3
t1
t2
Receive buffer state
Empty
Full
Empty
Full
SPRF (Receive buffer full flag)
OVRF (Overrun error flag)
Low
(1)
Receive buffer read
SPI transfer format (CPHA = 0)
t1: SPI clock delay register (SPCKD)
t2: SPI slave select negation delay register (SSLND)
t3: SPI next-access delay register (SPND)
Output: Undefined (0 or 1)
Input: don't care

Figure 32.34 **Clock stop waveform when serial transfer continues while receive buffer is full in master mode with CPHA = 0**

The operation of the flags at the timings shown in (1) and (2) in Figure 32.33 and Figure 32.34 is as follows:

(1) When the receive buffer is full, an overrun error does not occur because the RSPCK clock is stopped.

(2) If SPDR/SPDR_HA is read while the clock is stopped, data in the receive buffer can be read. The RSPCK clock restarts after reading the receive buffer (after SPSR.SPRF flag is set to 0).

32.3.8.2 Parity errors

When full-duplex synchronous serial communication is performed with the SPCR.TXMD bit set to 0 and the SPCR2.SPPE bit set to 1, the SPI checks for parity errors when serial transfer ends. On detecting a parity error in the received data, the SPI sets the SPSR.PERF flag to 1. Because the SPI does not copy data in the shift register to the receive buffer when the SPSR.OVRF flag is set to 1, parity error detection is not performed for the received data. To set the PERF flag to 0, write 0 to the PERF flag after the SPSR register is read with the PERF flag set to 1.

Figure 32.35 shows an example operation of OVRF and PERF flags. The SPSR access shown in Figure 32.35 indicates the condition of access to the SPSR register, where W denotes a write cycle, and R a read cycle. In the example in Figure 32.35, full-duplex synchronous serial communication 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.

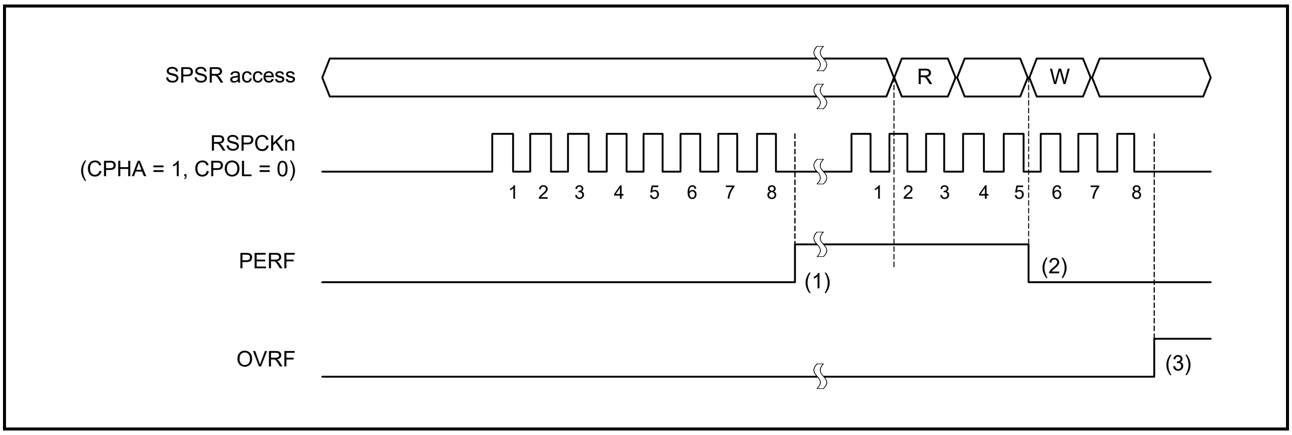


Figure 32.35 Operation example of the OVRF and PERF flags

The operation of the flags at the timing shown in (1) to (3) in Figure 32.35 is as follows:

(1) If a serial transfer terminates with the SPI not detecting an overrun error, the SPI copies data in the shift register to the receive buffer. The SPI checks the received data at this timing and sets the PERF flag to 1 if a parity error is detected. In master mode for SPI0, the SPI copies the value of the SPCMDm pointer 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, data in the shift register is not copied to the receive buffer. The SPI does not perform parity error detection at this time.

The occurrence of a parity error can be checked 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 errors. When the SPI is in master mode for SPI0, the value of the SPCMDm pointer at the error occurrence can be checked by reading the SPSSR.SPECM[2:0] bits.

32.3.8.3 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 an 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 for SPI0, the SPI copies the value of the SPCMDm pointer 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 32.3.9, Initializing the SPI). For multi-master configuration, detection of a mode fault error is used to stop the driving of output signals and the SPI function, which allows the master to be released.

The occurrence of a mode fault error can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. Detecting mode-fault errors without using the SPI error interrupt requires polling of SPSR. When using the SPI in master mode for SPI0, the value of the SPCMDm pointer at the error occurrence 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.

32.3.8.4 Underrun errors

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, the SPI detects an underrun error. The SPI then sets the SPSR.MODF and SPSR.UDRF flags to 1.

On detecting an underrun error, the SPI stops driving the output signals and clears the SPCR.SPE bit to 0 (see section 32.3.9, Initializing the SPI).

The occurrence of an underrun error can be checked 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.

32.3.9 Initializing the SPI

If 0 is written to the SPCR.SPE bit or if the SPI sets the SPE bit to 0 because it detected a mode fault error or an underrun error, the SPI disables the SPI function and initializes some of the module functions. When a system reset is generated, the SPI initializes all of the module functions. The following section describes initialization by clearing of the SPCR.SPE bit and by a system reset.

32.3.9.1 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 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 before initialization when the SPE bit is set to 1 again.

The SPRF, OVRF, MODF, PERF, and UDRF flags in the SPSR register are not initialized. The SPI Sequence Status Register (SPSSR) is not initialized for SPI0. 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. To disable any transmit buffer empty interrupts when the SPI is initialized, write 0 to the SPTIE bit simultaneously while writing 0 to the SPE bit.

32.3.9.2 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 32.3.9.1, Initialization by clearing the SPE bit.

32.3.10 SPI Operation

32.3.10.1 Master mode operation

The only difference between single-master mode and multi-master mode operation is the use of mode fault error detection (see section 32.3.8, Error Detection). In single-master mode, the SPI does not detect mode fault errors whereas in multi-master mode, it does. This section explains operations that are common to both modes.

(1) Starting serial transfer

The SPI updates the data in the transmit buffer (SPTX) when data is written to the SPI Data Register (SPDR/SPDR_HA) with the SPI transmit buffer empty, and 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 for SPI0, and the shift register is empty for SPI1, 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.

The polarity of the SSLni output pins depends on the SSLP register settings. For details on the SPI transfer format, see section 32.3.5, Transfer Format.

(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 = 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 the 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 32.3.5, Transfer Format.

(3) Sequence control

(a) SPI0

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- or 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 the SPI clock delay value (SPCKD), the SSL negation delay value (SSLND), and the next-access delay value (SPND).

According to the sequence length that is assigned to the SPSCR register, the SPI makes up a sequence comprised of a part or all of the 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 settings into the transfer format at the beginning of a 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 in the sequence, the SPI sets the pointer to SPCMD0 to execute the sequence repeatedly.

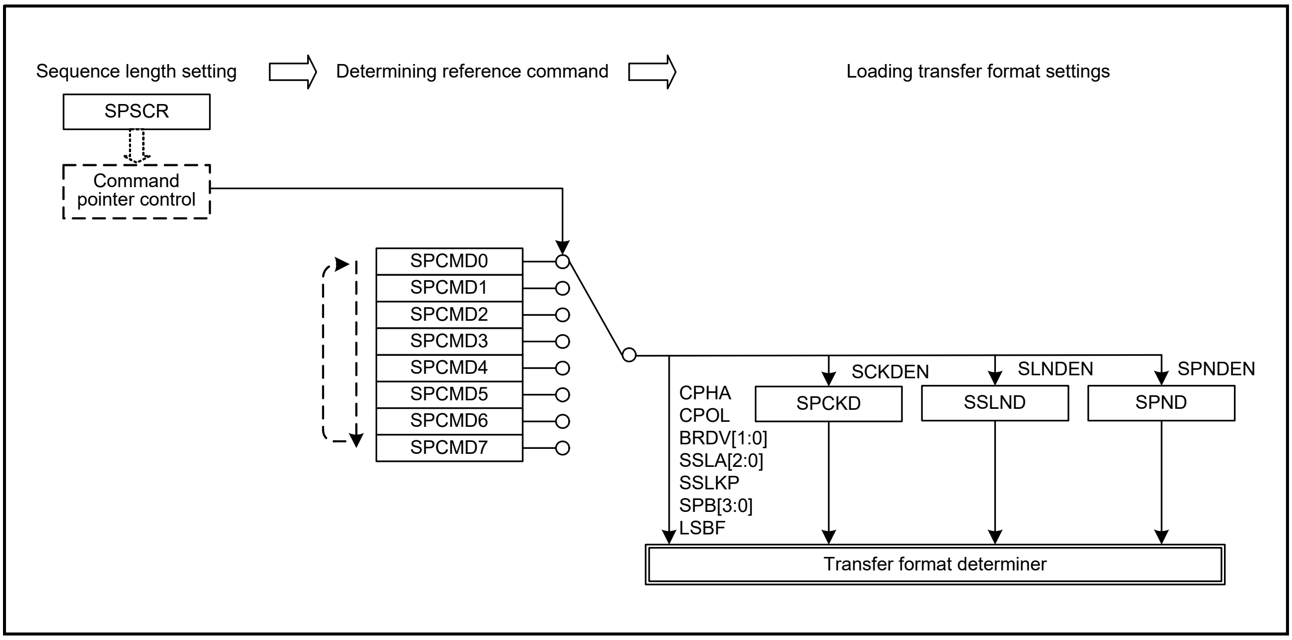

Sequence length setting
Determining reference command
Loading transfer format settings
SPSCR
Command pointer control
SPCMD0
SPCMD1
SPCMD2
SPCMD3
SPCMD4
SPCMD5
SPCMD6
SPCMD7
SCKDEN
SLNDEN
SPNDEN
CPHA
CPOL
BRDV[1:0]
SSLA[2:0]
SSLKP
SPB[3:0]
LSBF
SPCKD
SSLND
SPND
Transfer format determiner

Figure 32.36 Procedure for determining serial transfer format in master mode (SPI0)

In this section, a frame is the combination of the SPDR/SPDR_HA data and the SPCMDm settings.

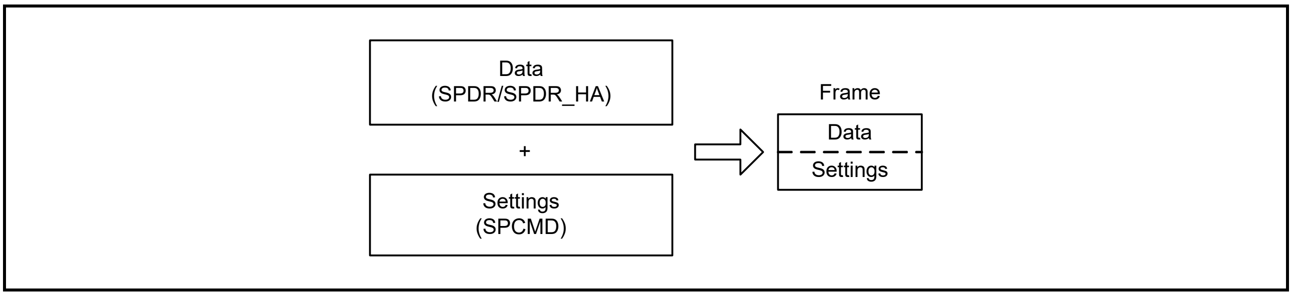

Data
(SPDR/SPDR_HA)
+
Settings
(SPCMD)
Frame
Data
Settings

Figure 32.37 Conceptual diagram of frames (SPI0)

Figure 32.38 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 32.4.

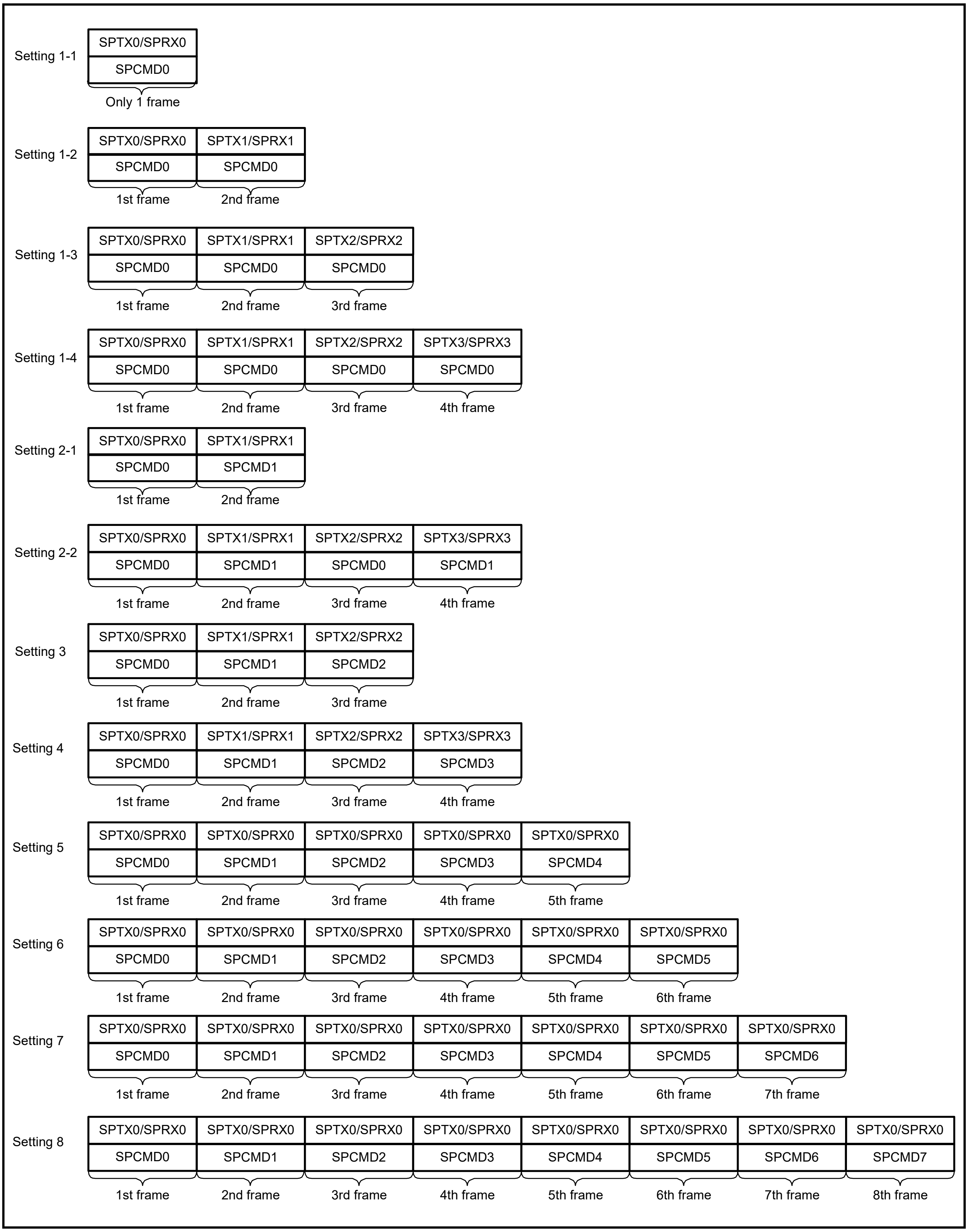

Setting 1-1
SPTX0/SPRX0
SPCMD0
Only 1 frame
Setting 1-2
SPTX0/SPRX0 SPTX1/SPRX1
SPCMD0 SPCMD0
1st frame 2nd frame
Setting 1-3
SPTX0/SPRX0 SPTX1/SPRX1 SPTX2/SPRX2
SPCMD0 SPCMD0 SPCMD0
1st frame 2nd frame 3rd frame
Setting 1-4
SPTX0/SPRX0 SPTX1/SPRX1 SPTX2/SPRX2 SPTX3/SPRX3
SPCMD0 SPCMD0 SPCMD0 SPCMD0
1st frame 2nd frame 3rd frame 4th frame
Setting 2-1
SPTX0/SPRX0 SPTX1/SPRX1
SPCMD0 SPCMD1
1st frame 2nd frame
Setting 2-2
SPTX0/SPRX0 SPTX1/SPRX1 SPTX2/SPRX2 SPTX3/SPRX3
SPCMD0 SPCMD1 SPCMD0 SPCMD1
1st frame 2nd frame 3rd frame 4th frame
Setting 3
SPTX0/SPRX0 SPTX1/SPRX1 SPTX2/SPRX2
SPCMD0 SPCMD1 SPCMD2
1st frame 2nd frame 3rd frame
Setting 4
SPTX0/SPRX0 SPTX1/SPRX1 SPTX2/SPRX2 SPTX3/SPRX3
SPCMD0 SPCMD1 SPCMD2 SPCMD3
1st frame 2nd frame 3rd frame 4th frame
Setting 5
SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0
SPCMD0 SPCMD1 SPCMD2 SPCMD3 SPCMD4
1st frame 2nd frame 3rd frame 4th frame 5th frame
Setting 6
SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0
SPCMD0 SPCMD1 SPCMD2 SPCMD3 SPCMD4 SPCMD5
1st frame 2nd frame 3rd frame 4th frame 5th frame 6th frame
Setting 7
SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0
SPCMD0 SPCMD1 SPCMD2 SPCMD3 SPCMD4 SPCMD5 SPCMD6
1st frame 2nd frame 3rd frame 4th frame 5th frame 6th frame 7th frame
Setting 8
SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0
SPCMD0 SPCMD1 SPCMD2 SPCMD3 SPCMD4 SPCMD5 SPCMD6 SPCMD7
1st frame 2nd frame 3rd frame 4th frame 5th frame 6th frame 7th frame 8th frame

Figure 32.38 Relationship between SPI Command Register (SPCMDm) and transmit and receive buffers in sequence operations (SPI0)

(b) SPI1

The transfer format in master mode is determined by the SPSCR, SPCMD0, SPBR, SPCKD, SSLND, and SPND registers.

The SPSCR register determines the sequence configuration for serial transfers that are executed by the SPI in master mode. The following are set in the SPCMD0 register:

- SSLni pin output signal value
- MSB- or 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 the SPI clock delay value (SPCKD), the SSL negation delay (SSLND), and the next-access delay value (SPND).

When the SPI function is enabled (SPCR.SPE = 1), 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.

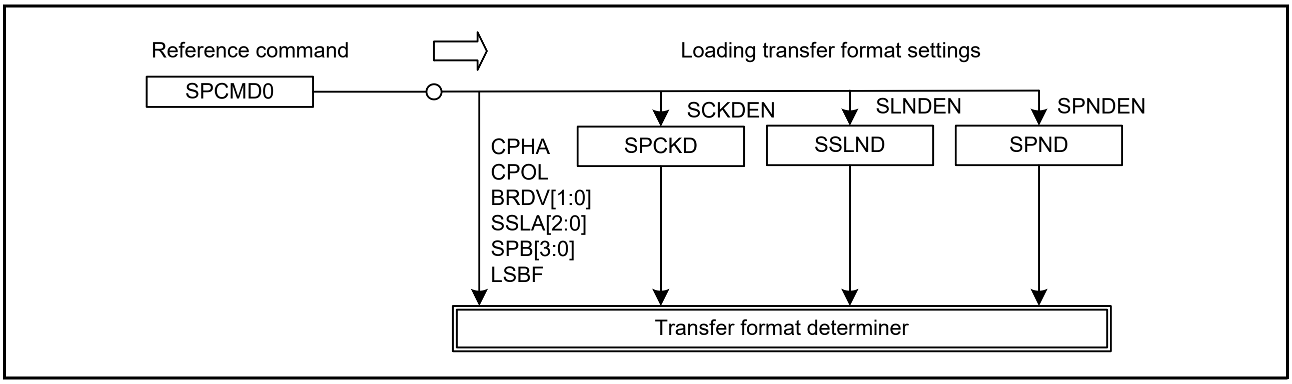

Reference command
Loading transfer format settings
SPCMD0
SCKDEN
SLNDEN
SPNDEN
CPHA
CPOL
BRDV[1:0]
SSLA[2:0]
SPB[3:0]
LSBF
SPCKD
SSLND
SPND
Transfer format determiner

Figure 32.39 Procedure for determining form of serial transfer in master mode (SPI1)

In this section, a frame is the combination of the SPDR/SPDR_HA data and the SPCMD0 settings.

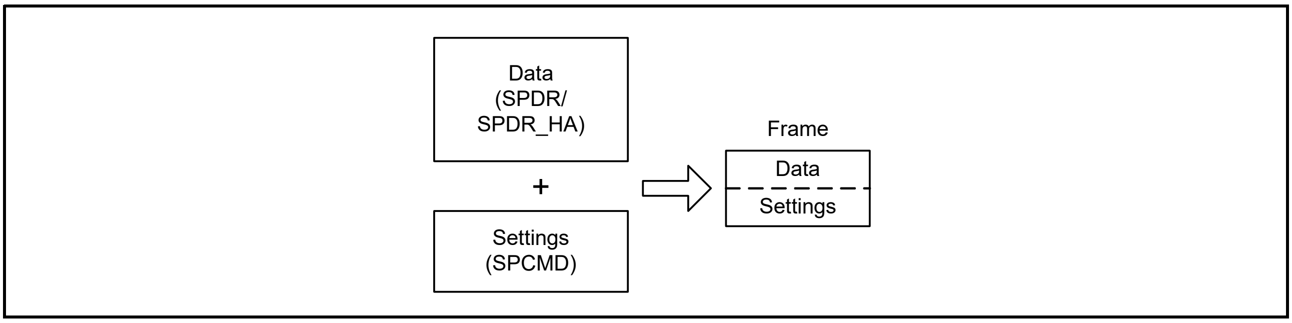

Data
(SPDR/
SPDR_HA)
+
Settings
(SPCMD)
Frame
Data
Settings

Figure 32.40 Conceptual diagram of frames (SPI1)

Figure 32.41 shows the relationship between the command and the transmit and receive buffers in the sequence of operations.

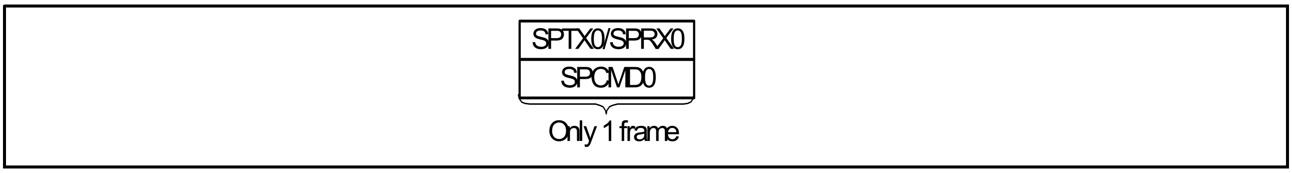


Figure 32.41 Relationship between the SPI Command Register and the transmit and receive buffers in sequence operations (SPI1)

(4) Burst transfer

- SPI0

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 32.42 shows an example of an SSLni signal operation for a burst transfer that is implemented using the SPCMD0 and SPCMD1 register settings. The following section explains the SPI operations (1) to (7) as shown in Figure 32.42.

Note: The polarity of the SSLni output signal depends on the SSLP register settings.

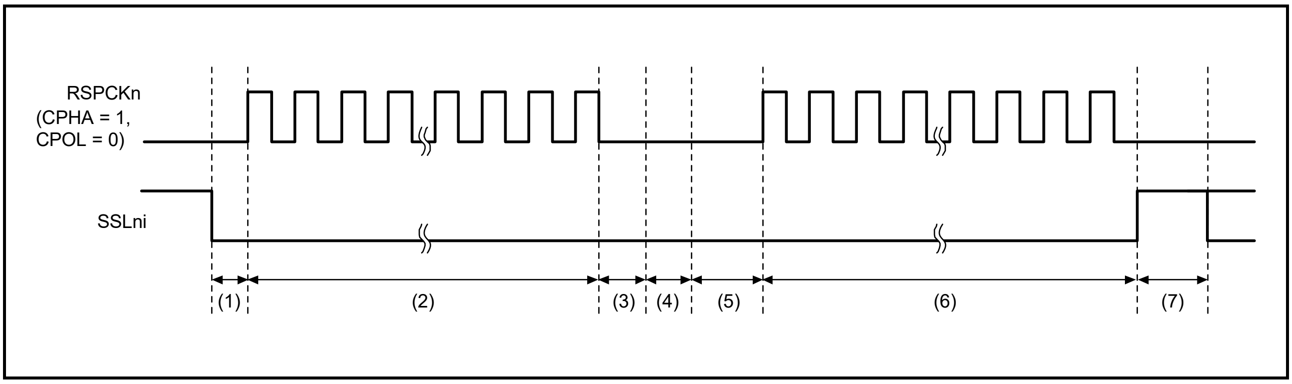


Figure 32.42 Example of burst transfer operation using the SSLKP bit (SPI0)

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 equal 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 32.42. 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.

In master mode, the SPI 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.

- SPI1

SPI does not support continuous serial transfer (burst transfer), keeping the SSL signal asserted. However, burst transfer can be implemented by controlling the SSL signal output in general-purpose ports.

(5) RSPCK delay (t1)

The RSPCK delay in master mode depends on the SPCMDm.SCKDEN bit setting and the SPCKD register setting. For SPI0, the SPI determines the SPCMDm register to be referenced by pointer control during a serial transfer, and determines an RSPCK delay using the SPCMDm.SCKDEN bit and SPCKD, as listed in Table 32.9. For SPI1, the SPI determines an RSPCK delay using the SPCMD0.SCKDEN bit and SPCKD, as listed in Table 32.9. For a definition of RSPCK delay, see section 32.3.5, Transfer Format.

Table 32.9 Relationship between the SCKDEN bit, SPCKD, and RSPCK delay

SPCMDm.SCKDEN bit	SPCKD.SCKDL[2:0] bits	RSPCK delay
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(6) SSL negation delay (t2)

The SSL negation delay in master mode depends on the SPCMDm.SLNDEN bit setting and the SSLND register setting. For SPI0, the SPI determines the SPCMDm register to be referenced by pointer control during serial transfer, and determines an SSL negation delay using the SPCMDm.SLNDEN bit and SSLND, as listed in Table 32.10. For SPI1, the SPI determines an SSL negation delay using the SPCMD0.SLNDEN bit and SSLND, as listed in Table 32.10. For a definition of SSL negation delay, see section 32.3.5, Transfer Format.

Table 32.10 Relationship between the 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)

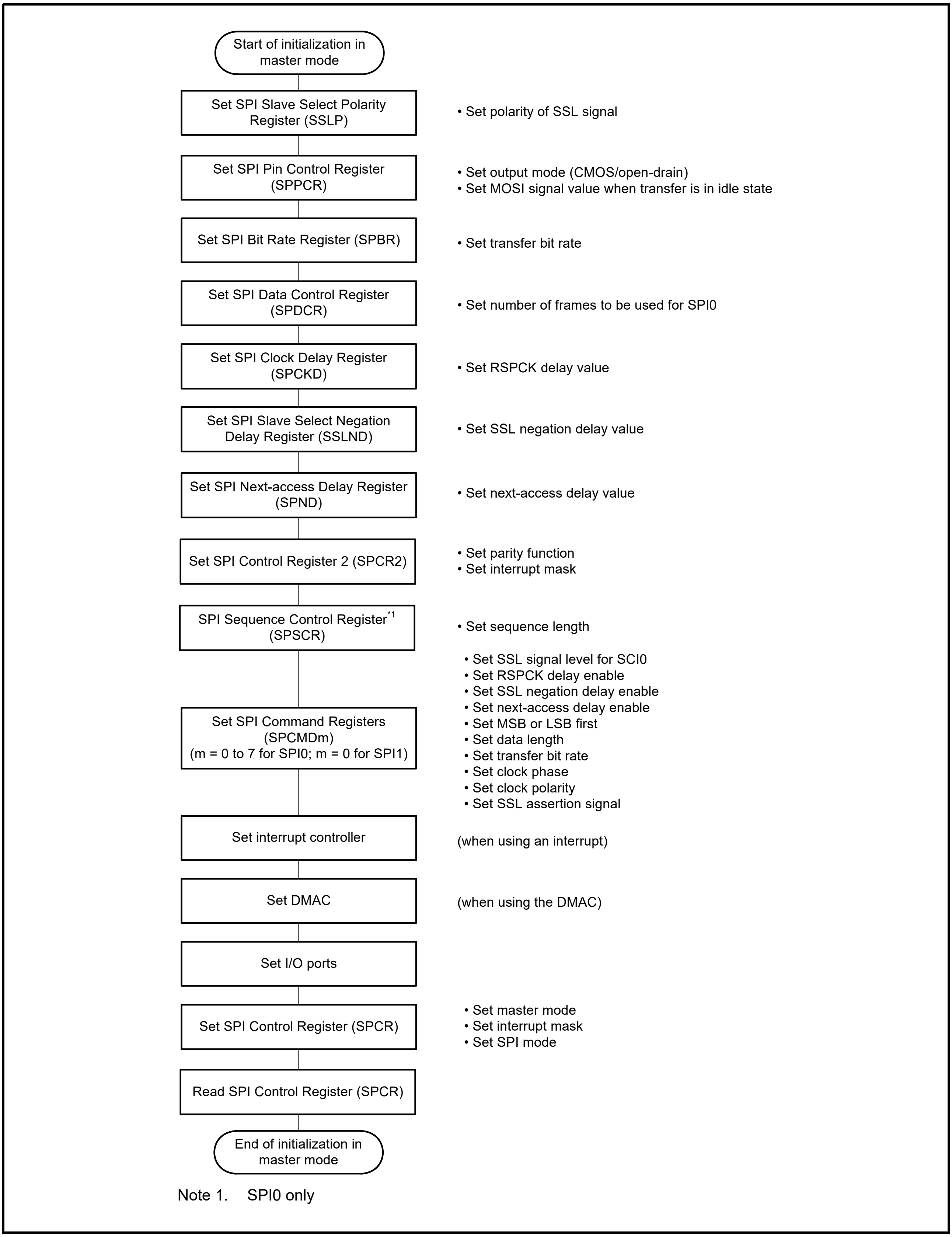
The next-access delay in master mode depends on the SPCMDm.SPNDEN bit setting and the SPND setting. For SPI0, the SPI determines the SPCMDm register to be referenced by pointer control during serial transfer, and determines a next-access delay during serial transfer using the SPCMDm.SPNDEN bit and SPND, as listed in Table 32.11. For SPI1, the SPI determines a next-access delay during serial transfer using the SPCMD0.SPNDEN bit and SPND, as listed in Table 32.11. For a definition of next-access delay, see section 32.3.5, Transfer Format.

Table 32.11 Relationship between SPNDEN bit, SPND, and next-access delay

SPCMDm.SPNDEN bit	SPND.SPNDL[2:0] bits	Next-access delay
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 32.43 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.


Start of initialization in master mode
Set SPI Slave Select Polarity Register (SSLP)
• Set polarity of SSL signal
Set SPI Pin Control Register (SPPCR)
• Set output mode (CMOS/open-drain)
• Set MOSI signal value when transfer is in idle state
Set SPI Bit Rate Register (SPBR)
• Set transfer bit rate
Set SPI Data Control Register (SPDCR)
• Set number of frames to be used for SPI0
Set SPI Clock Delay Register (SPCKD)
• Set RSPCK delay value
Set SPI Slave Select Negation Delay Register (SSLND)
• Set SSL negation delay value
Set SPI Next-access Delay Register (SPND)
• Set next-access delay value
Set SPI Control Register 2 (SPCR2)
• Set parity function
• Set interrupt mask
SPI Sequence Control Register*1 (SPSCR)
• Set sequence length
Set SPI Command Registers (SPCMDm) (m = 0 to 7 for SPI0; m = 0 for SPI1)
• Set SSL signal level for SCI0
• Set RSPCK delay enable
• Set SSL negation delay enable
• Set next-access delay enable
• Set MSB or LSB first
• Set data length
• Set transfer bit rate
• Set clock phase
• Set clock polarity
• Set SSL assertion signal
Set interrupt controller
(when using an interrupt)
Set DMAC
(when using the DMAC)
Set I/O ports
Set SPI Control Register (SPCR)
• Set master mode
• Set interrupt mask
• Set SPI mode
Read SPI Control Register (SPCR)
End of initialization in master mode

Note 1. SPI0 only

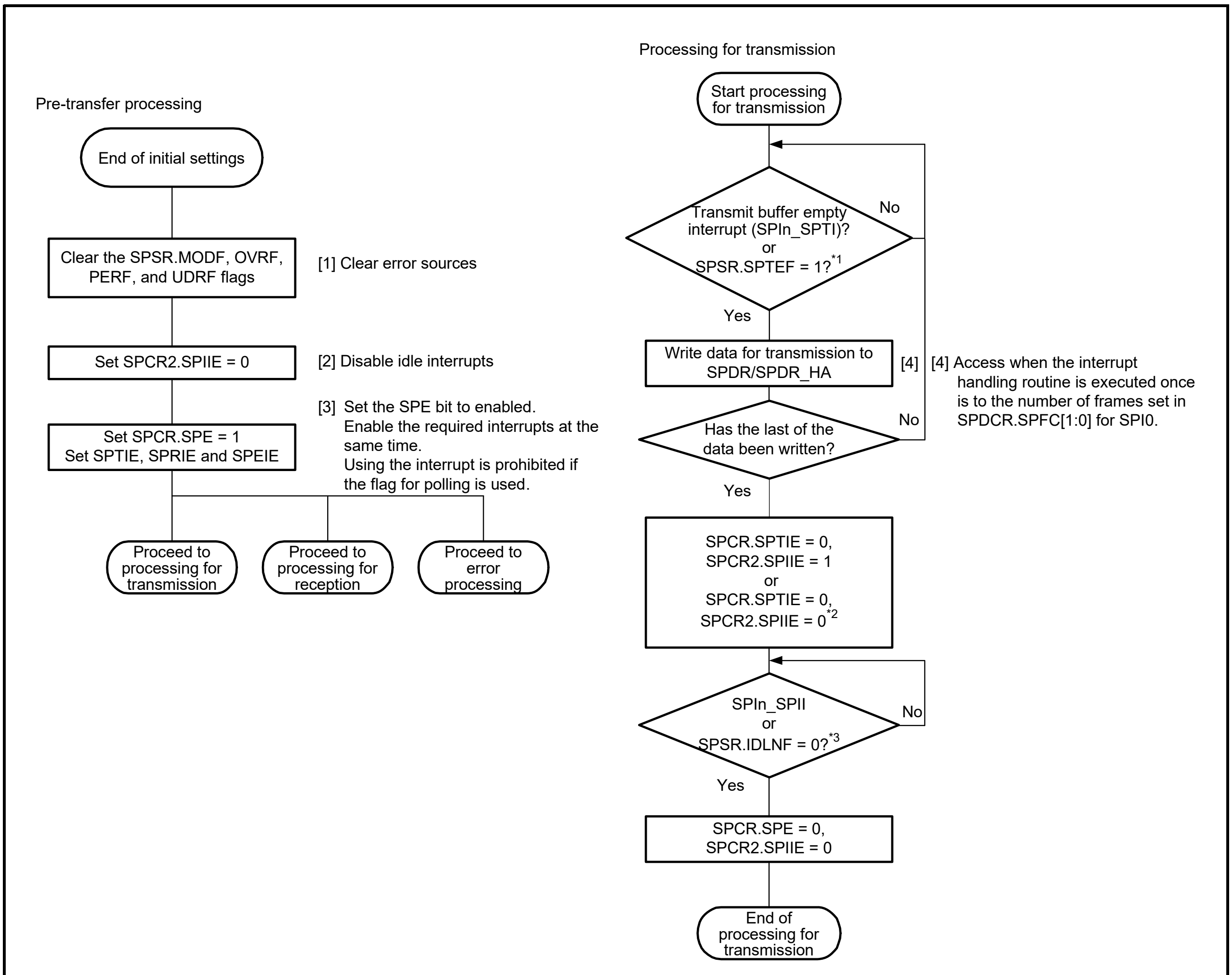
Figure 32.43 Example of initialization flow in master mode for SPI operation

(9) Software processing flow

Figure 32.44 to Figure 32.46 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.


Pre-transfer processing
End of initial settings
Clear the SPSR.MODF, OVRF, PERF, and UDRF flags
[1] Clear error sources
Set SPCR2.SPIIE = 0
[2] Disable idle interrupts
Set SPCR.SPE = 1
Set SPTIE, SPRIE and SPEIE
[3] Set the SPE bit to enabled. Enable the required interrupts at the same time. Using the interrupt is prohibited if the flag for polling is used.
Proceed to processing for transmission
Proceed to processing for reception
Proceed to error processing
Processing for transmission
Start processing for transmission
Transmit buffer empty interrupt (SPIn_SPTI)? or SPSR.SPTEF = 1?*1
No
Yes
Write data for transmission to SPDR/SPDR_HA
[4]
[4] Access when the interrupt handling routine is executed once is to the number of frames set in SPDCR.SPFC[1:0] for SPI0.
Has the last of the data been written?
No
Yes
SPCR.SPTIE = 0, SPCR2.SPIIE = 1 or SPCR.SPTIE = 0, SPCR2.SPIIE = 0*2
SPIn_SPII or SPSR.IDLNF = 0?*3
No
Yes
SPCR.SPE = 0, SPCR2.SPIIE = 0
End of processing for transmission

Note 1. Proceed to writing data for transmission to SPDR/SPDR_HA after checking the transmit buffer empty by reading the SPSR.SPTEF flag if the flag for polling is used.

Note 2. Setting the idle interrupt is prohibited (SPCR2.SPIIE = 0) if the flag for polling is used.

Note 3. Wait more than 1 PCLK after writing data for transmission to SPDE and before starting to poll PSR.IDLNF, if the flag for polling is used.

Figure 32.44 Transmission flow in master mode

(b) Receive processing flow

The SPI does not handle receive-only operations, therefore processing for transmission is required.

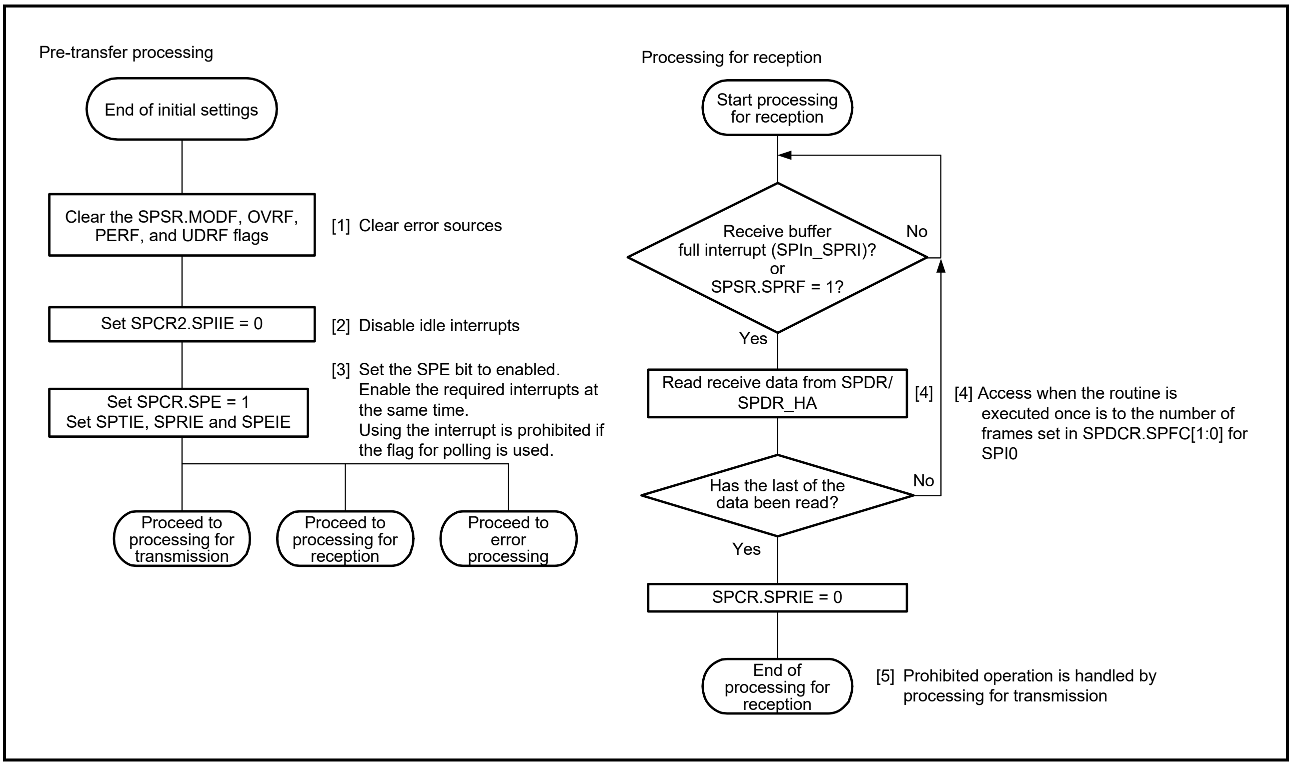

Pre-transfer processing
End of initial settings
Clear the SPSR.MODF, OVRF, PERF, and UDRF flags
[1] Clear error sources
Set SPCR2.SPIIE = 0
[2] Disable idle interrupts
Set SPCR.SPE = 1
Set SPTIE, SPRIE and SPEIE
[3] Set the SPE bit to enabled. Enable the required interrupts at the same time. Using the interrupt is prohibited if the flag for polling is used.
Proceed to processing for transmission
Proceed to processing for reception
Proceed to error processing
Processing for reception
Start processing for reception
Receive buffer full interrupt (SPIn_SPRI)? or SPSR.SPRF = 1?
No
Yes
Read receive data from SPDR/ SPDR_HA
[4]
[4] Access when the routine is executed once is to the number of frames set in SPDCR.SPFC[1:0] for SPI0
Has the last of the data been read?
No
Yes
SPCR.SPRIE = 0
End of processing for reception
[5] Prohibited operation is handled by processing for transmission

Figure 32.45 Reception flow in master mode

(c) Error processing flow

The SPI detects the following errors:

- Mode fault
- Underrun
- Overrun
- Parity.

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 for SPI0.

When an error is detected using an interrupt, clear the ICU.IELSRj.IR flag in the error processing routine. If this is not done, the ICU.IELSRj.IR flag might continue to indicate a transmit buffer empty or a receive buffer full interrupt request. If an SPIn_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

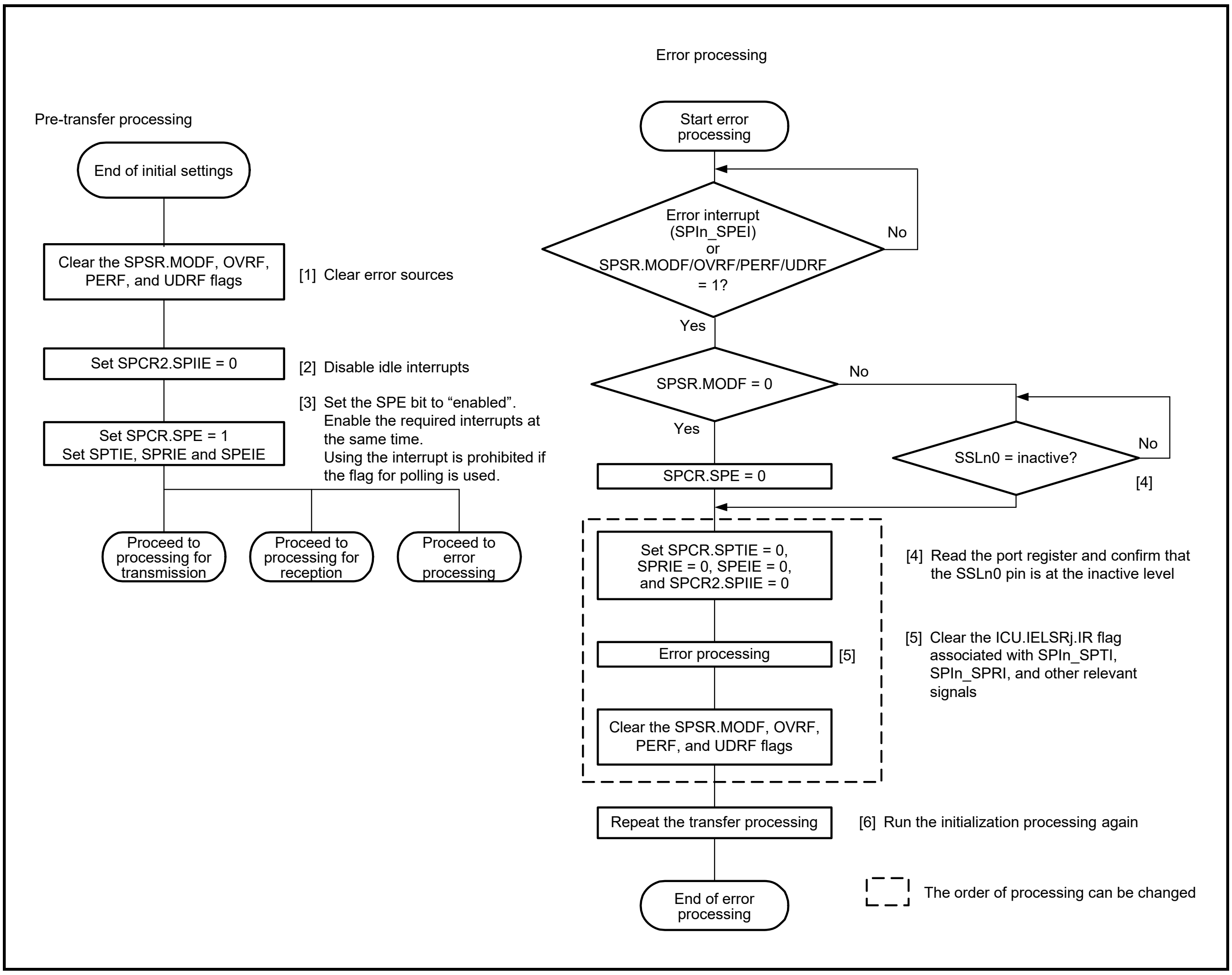

Pre-transfer processing
End of initial settings
Clear the SPSR.MODF, OVRF, PERF, and UDRF flags
[1] Clear error sources
Set SPCR2.SPIIE = 0
[2] Disable idle interrupts
Set SPCR.SPE = 1
Set SPTIE, SPRIE and SPEIE
[3] Set the SPE bit to "enabled". Enable the required interrupts at the same time. Using the interrupt is prohibited if the flag for polling is used.
Proceed to processing for transmission
Proceed to processing for reception
Proceed to error processing
Error processing
Start error processing
Error interrupt (SPIn_SPEI) or SPSR.MODF/OVRF/PERF/UDRF = 1?
No
Yes
SPSR.MODF = 0
No
Yes
SSLn0 = inactive?
No
[4]
SPCR.SPE = 0
Set SPCR.SPTIE = 0, SPRIE = 0, SPEIE = 0, and SPCR2.SPIIE = 0
[4] Read the port register and confirm that the SSLn0 pin is at the inactive level
Error processing
[5]
[5] Clear the ICU.IELSRj.IR flag associated with SPIn_SPTI, SPIn_SPRI, and other relevant signals
Clear the SPSR.MODF, OVRF, PERF, and UDRF flags
Repeat the transfer processing
[6] Run the initialization processing again
End of error processing
The order of processing can be changed

Figure 32.46 Error processing flow in master mode

32.3.10.2 Slave mode operation

(1) Starting serial transfer

When the SPCMD0.CPHA bit is 0, if the SPI detects an SSLn0 input signal assertion, it must drive valid data to the MISOn output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLn0 input signal triggers the start of a serial transfer.

When the CPHA bit is 1, if the SPI detects the first RSPCKn edge in an SSLn0 signal asserted condition, it must drive valid data to the MISOn output signal. For this reason, when the CPHA bit is 1, the first RSPCKn edge in an SSLn0 signal asserted condition triggers the start of a serial transfer.

Regardless of the CPHA bit setting, the SPI drives the MISOn output signal on SSLn0 signal assertion. The data that is output by the SPI is either valid or invalid, depending on the CPHA bit setting.

The polarity of the SSLn0 input signal depends on the setting of the SSLP.SSL0P bit. For details on the SPI transfer format, see section 32.3.5, Transfer Format.

(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 = 0), on termination of serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR/SPDR_HA register. On termination of a serial transfer, the SPI changes the status of the shift register to empty, regardless of the receive buffer state. A mode fault error occurs if the SPI detects an SSLn0 input signal negation from the beginning of the serial transfer to the end of the serial transfer (see section 32.3.8, 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 32.3.5, Transfer Format.

(3) Notes on single-slave operations

If the SPCMD0.CPHA bit is 0, the SPI starts serial transfers when it detects the assertion edge for an SSLn0 input signal. In the configuration example shown in Figure 32.10, if the SPI is in single-slave mode, the SSLn0 signal is fixed at an active state. Therefore, when the CPHA bit is set to 0, the SPI cannot correctly start a serial transfer. For the SPI to correctly execute transmit and receive operations in slave mode when the SSLn0 input signal is fixed at an active state, the CPHA bit must be set to 1. 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 an 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.

Burst transfer cannot be executed for SPI1.

(5) Initialization flow

Figure 32.47 shows an example of initialization flow for SPI operation when 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.

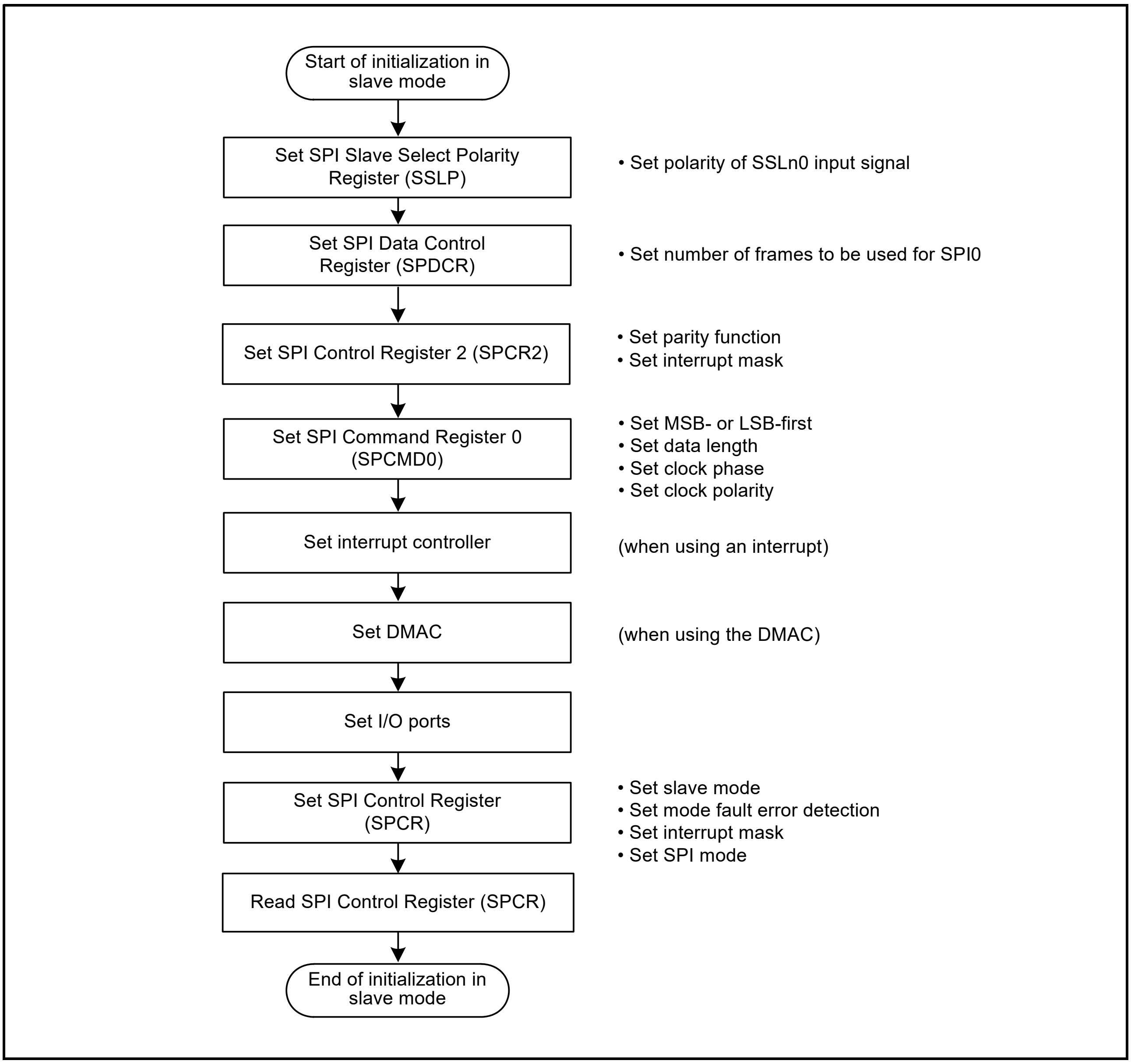
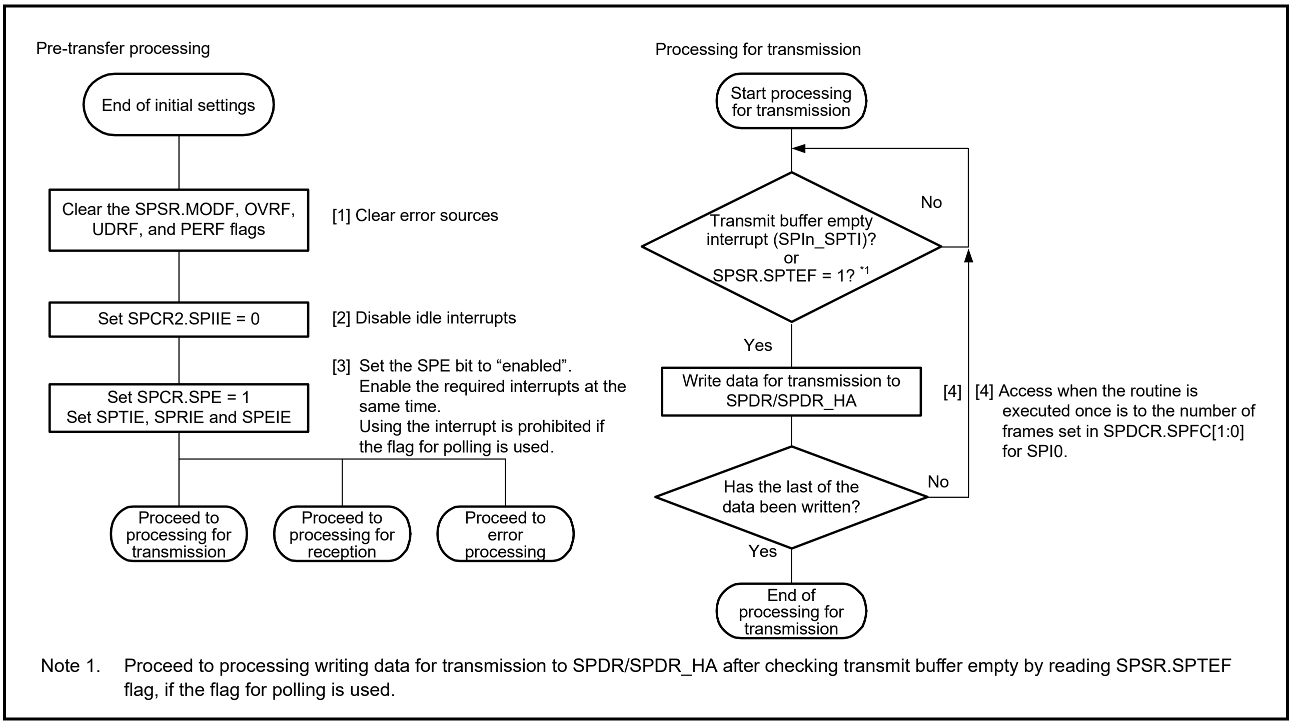

Start of initialization in slave mode
Set SPI Slave Select Polarity Register (SSLP)
• Set polarity of SSLn0 input signal
Set SPI Data Control Register (SPDCR)
• Set number of frames to be used for SPI0
Set SPI Control Register 2 (SPCR2)
• Set parity function
• Set interrupt mask
Set SPI Command Register 0 (SPCMD0)
• Set MSB- or LSB-first
• Set data length
• Set clock phase
• Set clock polarity
Set interrupt controller
(when using an interrupt)
Set DMAC
(when using the DMAC)
Set I/O ports
Set SPI Control Register (SPCR)
• Set slave mode
• Set mode fault error detection
• Set interrupt mask
• Set SPI mode
Read SPI Control Register (SPCR)
End of initialization in slave mode

Figure 32.47 Example initialization flow in slave mode for SPI operation

(6) Software processing flow

Figure 32.48 to Figure 32.50 show example flows of software processing.

(a) Transmit processing flow


Pre-transfer processing
End of initial settings
Clear the SPSR.MODF, OVRF, UDRF, and PERF flags
[1] Clear error sources
Set SPCR2.SPIIE = 0
[2] Disable idle interrupts
Set SPCR.SPE = 1
Set SPTIE, SPRIE and SPEIE
[3] Set the SPE bit to "enabled". Enable the required interrupts at the same time. Using the interrupt is prohibited if the flag for polling is used.
Proceed to processing for transmission
Proceed to processing for reception
Proceed to error processing
Processing for transmission
Start processing for transmission
Transmit buffer empty interrupt (SPIn_SPTI)? or SPSR.SPTEF = 1? *1
No
Yes
Write data for transmission to SPDR/SPDR_HA
[4]
[4] Access when the routine is executed once is to the number of frames set in SPDCR.SPFC[1:0] for SPI0.
Has the last of the data been written?
No
Yes
End of processing for transmission

Note 1. Proceed to processing writing data for transmission to SPDR/SPDR_HA after checking transmit buffer empty by reading SPSR.SPTEF flag, if the flag for polling is used.

Figure 32.48 Transmission flow in slave mode

(b) Receive processing flow

The SPI does not handle receive-only operation, therefore processing for transmission is required.

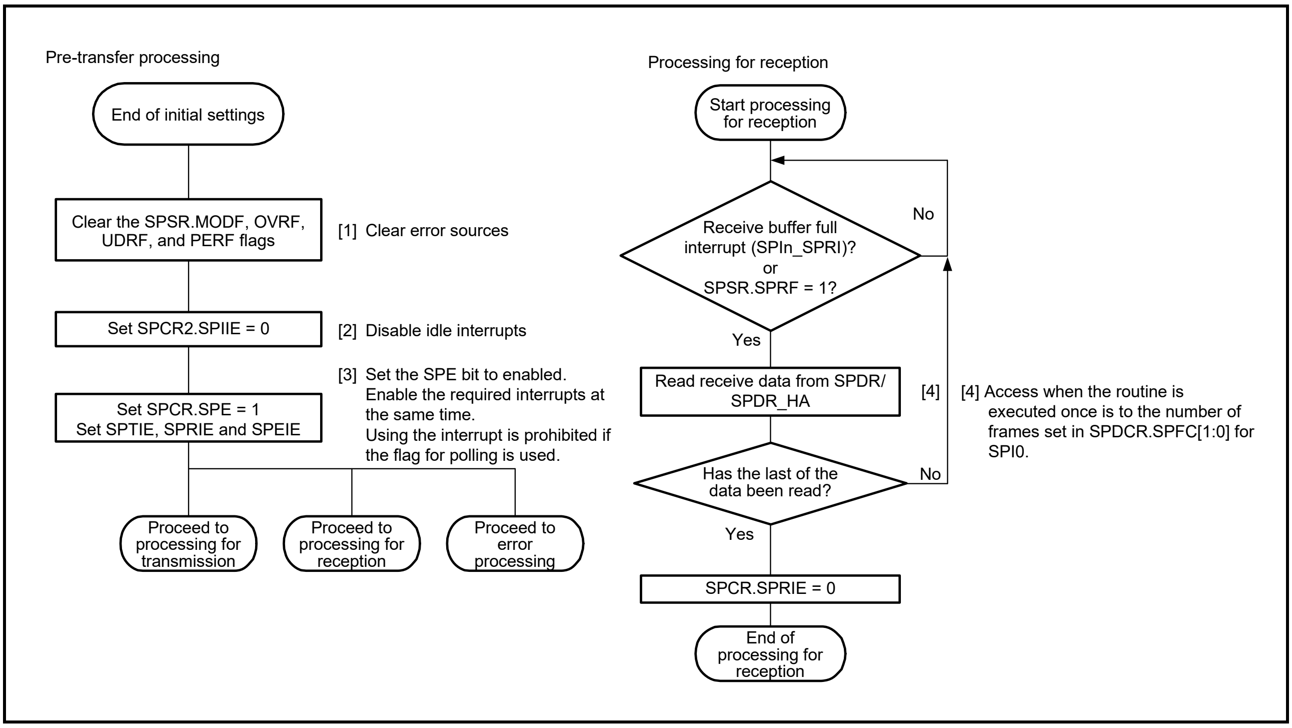

Pre-transfer processing
End of initial settings
Clear the SPSR.MODF, OVRF, UDRF, and PERF flags
[1] Clear error sources
Set SPCR2.SPIIE = 0
[2] Disable idle interrupts
Set SPCR.SPE = 1
Set SPTIE, SPRIE and SPEIE
[3] Set the SPE bit to enabled. Enable the required interrupts at the same time. Using the interrupt is prohibited if the flag for polling is used.
Proceed to processing for transmission
Proceed to processing for reception
Proceed to error processing
Processing for reception
Start processing for reception
Receive buffer full interrupt (SPIn_SPRI)? or SPSR.SPRF = 1?
No
Yes
Read receive data from SPDR/SPDR_HA
[4]
[4] Access when the routine is executed once is to the number of frames set in SPDCR.SPFC[1:0] for SPI0.
Has the last of the data been read?
No
Yes
SPCR.SPRIE = 0
End of processing for reception

Figure 32.49 Reception flow in slave mode

(c) Error processing flow

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.IELSRj.IR flag in the error processing routine. If this is not done, the ICU.IELSRj.IR flag might continue to indicate a transmit buffer empty or receive buffer full interrupt request. If a receive buffer full request is indicated, read the receive buffer and initialize the sequencer in the SPI.

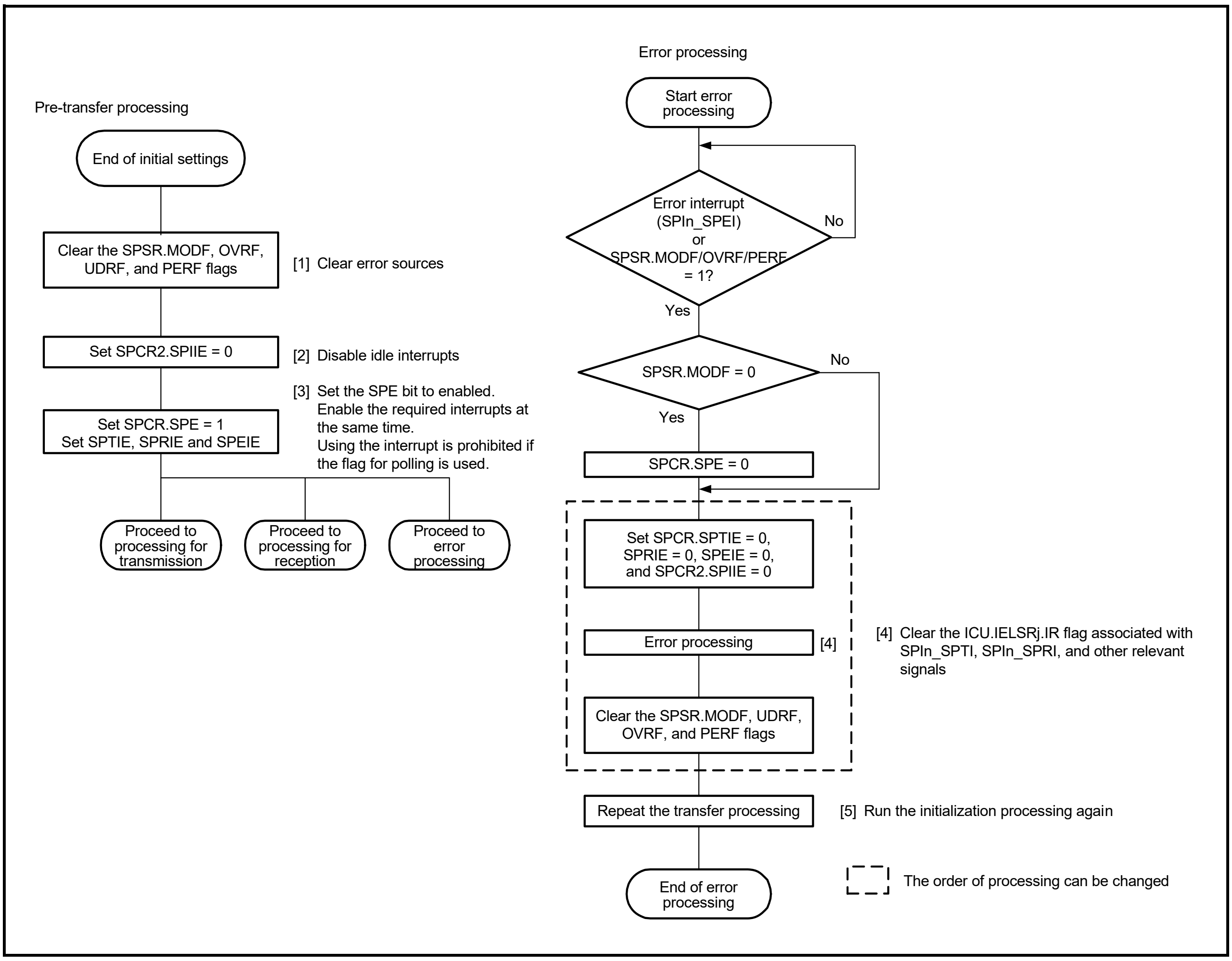

Pre-transfer processing
End of initial settings
Clear the SPSR.MODF, OVRF, UDRF, and PERF flags
[1] Clear error sources
Set SPCR2.SPIIE = 0
[2] Disable idle interrupts
Set SPCR.SPE = 1
Set SPTIE, SPRIE and SPEIE
[3] Set the SPE bit to enabled. Enable the required interrupts at the same time. Using the interrupt is prohibited if the flag for polling is used.
Proceed to processing for transmission
Proceed to processing for reception
Proceed to error processing
Error processing
Start error processing
Error interrupt (SPIn_SPEI) or SPSR.MODF/OVRF/PERF = 1?
No
Yes
SPSR.MODF = 0
No
Yes
SPCR.SPE = 0
Set SPCR.SPTIE = 0, SPRIE = 0, SPEIE = 0, and SPCR2.SPIIE = 0
Error processing
[4]
[4] Clear the ICU.IELSRj.IR flag associated with SPIn_SPTI, SPIn_SPRI, and other relevant signals
Clear the SPSR.MODF, UDRF, OVRF, and PERF flags
Repeat the transfer processing
[5] Run the initialization processing again
End of error processing
The order of processing can be changed

Figure 32.50 Error processing flow for slave mode

32.3.11 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the SPI. In clock synchronous operation, the SSLni pin is not used, and the RSPCKn, MOSIn, and MISOn pins handle communications. 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).

32.3.11.1 Master mode operation

(1) Starting serial transfer

The SPI updates the data in the transmit buffer (SPTX) of SPDR/SPDR_HA when data is written to the SPDR/

SPDR_HA register with the transmit buffer empty, 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 for SPI0 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.

Transfer in clock synchronous operation is conducted without the SSLn0 output signal. For details on the SPI transfer format, see section 32.3.5, Transfer Format.

(2) Terminating serial transfer

The SPI terminates the serial transfer after transmitting an RSPCKn edge associated with the sampling timing. When free space is available in the receive buffer (SPSR.SPRF = 0), on termination of serial transfer, the SPI copies data from the shift register to the receive buffer of the SPI data register (SPDR/SPDR_HA).

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMDm.SPB[3:0] bit setting. Transfer in clock synchronous operation is conducted without the SSLn0 output signal. For details on the SPI transfer format, see section 32.3.5, Transfer Format.

(3) Sequence control

(a) SPI0

The transfer format in master mode is determined by the 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 are set in SPCMDm register:

- SSLni output signal value
- MSB- 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.

The SPBR register holds some of the bit rate settings such as the SPI clock delay value (SPCKD), the SSL negation delay (SSLND), and the next-access delay value (SPND).

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 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 containing the sequence, the SPI sets the pointer to the SPCMD0 register to execute the sequence repeatedly.

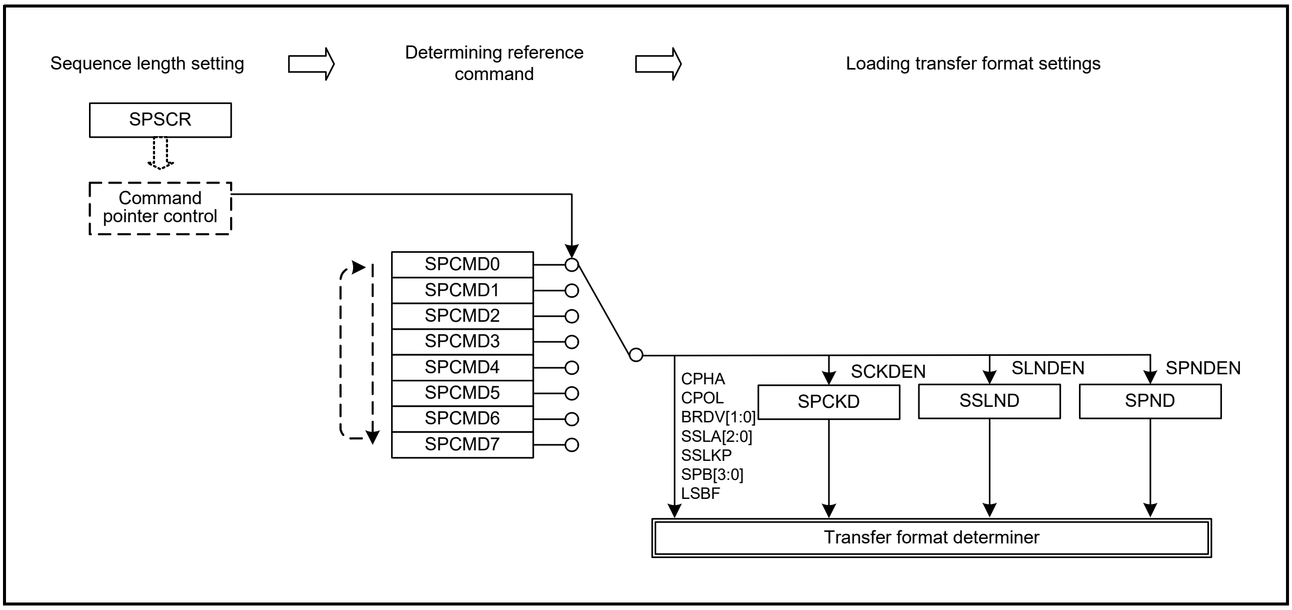

Sequence length setting
Determining reference command
Loading transfer format settings
SPSCR
Command pointer control
SPCMD0
SPCMD1
SPCMD2
SPCMD3
SPCMD4
SPCMD5
SPCMD6
SPCMD7
CPHA
CPOL
BRDV[1:0]
SSLA[2:0]
SSLKP
SPB[3:0]
LSBF
SCKDEN
SLNDEN
SPNDEN
SPCKD
SSLND
SPND
Transfer format determiner

Figure 32.51 Procedure to determine the format of serial transmission in master mode (SPI0)

In this section, a frame is the combination of the SPDR/SPDR_HA data and SPCMDm settings.

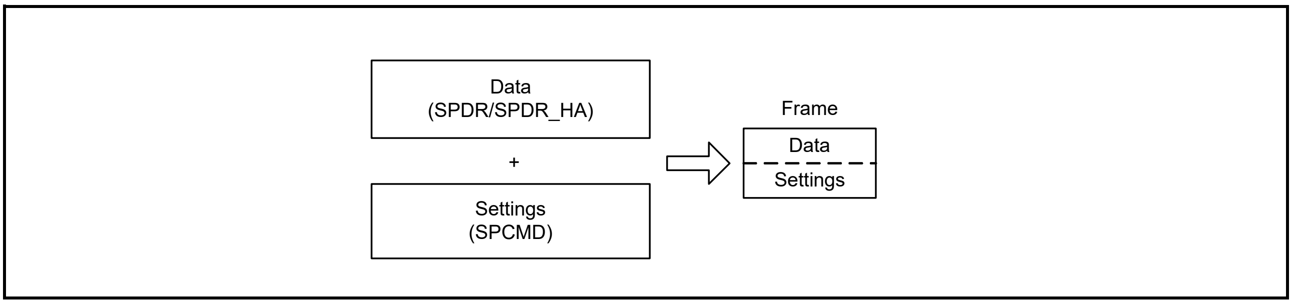

Data
(SPDR/SPDR_HA)
+
Settings
(SPCMD)
Frame
Data
Settings

Figure 32.52 Conceptual diagram of frames (SPI0)

Figure 32.53 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 32.4.

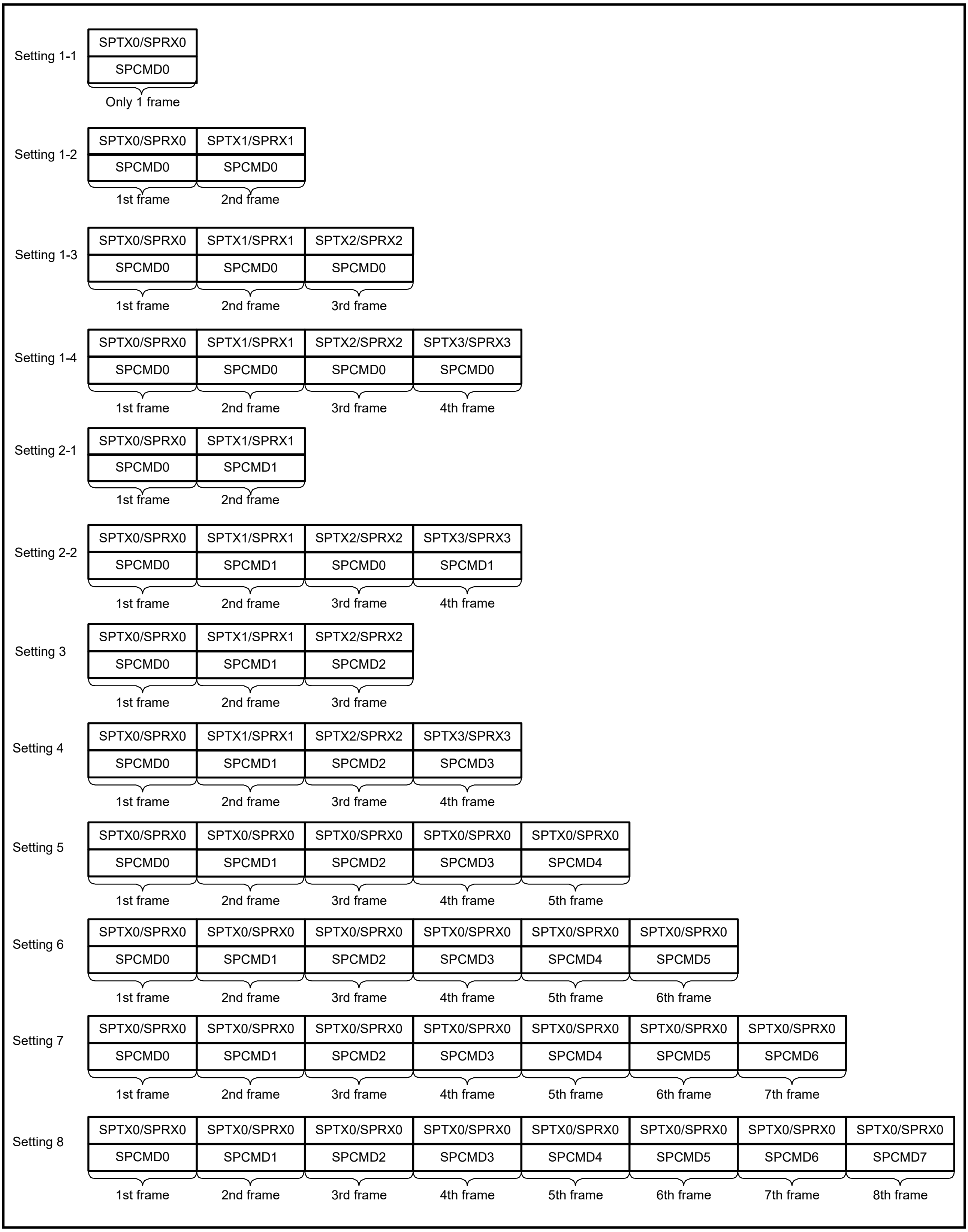

Setting 1-1
SPTX0/SPRX0
SPCMD0
Only 1 frame
Setting 1-2
SPTX0/SPRX0 SPTX1/SPRX1
SPCMD0 SPCMD0
1st frame 2nd frame
Setting 1-3
SPTX0/SPRX0 SPTX1/SPRX1 SPTX2/SPRX2
SPCMD0 SPCMD0 SPCMD0
1st frame 2nd frame 3rd frame
Setting 1-4
SPTX0/SPRX0 SPTX1/SPRX1 SPTX2/SPRX2 SPTX3/SPRX3
SPCMD0 SPCMD0 SPCMD0 SPCMD0
1st frame 2nd frame 3rd frame 4th frame
Setting 2-1
SPTX0/SPRX0 SPTX1/SPRX1
SPCMD0 SPCMD1
1st frame 2nd frame
Setting 2-2
SPTX0/SPRX0 SPTX1/SPRX1 SPTX2/SPRX2 SPTX3/SPRX3
SPCMD0 SPCMD1 SPCMD0 SPCMD1
1st frame 2nd frame 3rd frame 4th frame
Setting 3
SPTX0/SPRX0 SPTX1/SPRX1 SPTX2/SPRX2
SPCMD0 SPCMD1 SPCMD2
1st frame 2nd frame 3rd frame
Setting 4
SPTX0/SPRX0 SPTX1/SPRX1 SPTX2/SPRX2 SPTX3/SPRX3
SPCMD0 SPCMD1 SPCMD2 SPCMD3
1st frame 2nd frame 3rd frame 4th frame
Setting 5
SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0
SPCMD0 SPCMD1 SPCMD2 SPCMD3 SPCMD4
1st frame 2nd frame 3rd frame 4th frame 5th frame
Setting 6
SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0
SPCMD0 SPCMD1 SPCMD2 SPCMD3 SPCMD4 SPCMD5
1st frame 2nd frame 3rd frame 4th frame 5th frame 6th frame
Setting 7
SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0
SPCMD0 SPCMD1 SPCMD2 SPCMD3 SPCMD4 SPCMD5 SPCMD6
1st frame 2nd frame 3rd frame 4th frame 5th frame 6th frame 7th frame
Setting 8
SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0 SPTX0/SPRX0
SPCMD0 SPCMD1 SPCMD2 SPCMD3 SPCMD4 SPCMD5 SPCMD6 SPCMD7
1st frame 2nd frame 3rd frame 4th frame 5th frame 6th frame 7th frame 8th frame

Figure 32.53 Relationship between the SPI Command Register and the transmit and receive buffers in sequence operations (SPI0)

(b) SPI1

The transfer format in master mode is determined by the SPCMD0, SPBR, SPCKD, SSLND, and SPND registers. Although the SSLni signals are not output in clock synchronous operation, these settings are valid.

The following are set in the SPCMD0 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.

The SPBR register holds some of the bit rate settings such as the SPI clock delay value (SPCKD), the SSL negation delay (SSLND), and the next-access delay value (SPND).

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 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer.

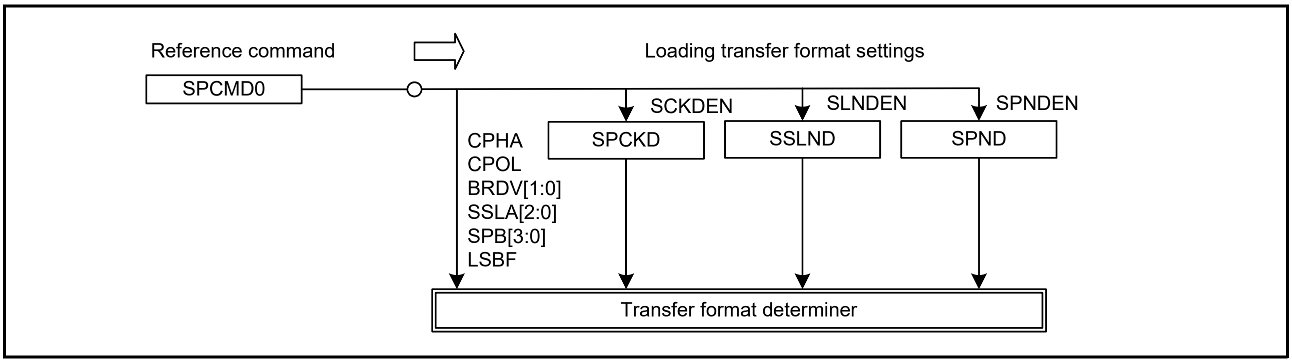

Reference command
SPCMD0
Loading transfer format settings
SCKDEN
SLNDEN
SPNDEN
SPCKD
SSLND
SPND
CPHA
CPOL
BRDV[1:0]
SSLA[2:0]
SPB[3:0]
LSBF
Transfer format determiner

Figure 32.54 Procedure to determine the format of serial transmission in master mode (SPI1)

In this section, a frame is the combination of the SPDR/SPDR_HA data and the SPCMD0 settings.

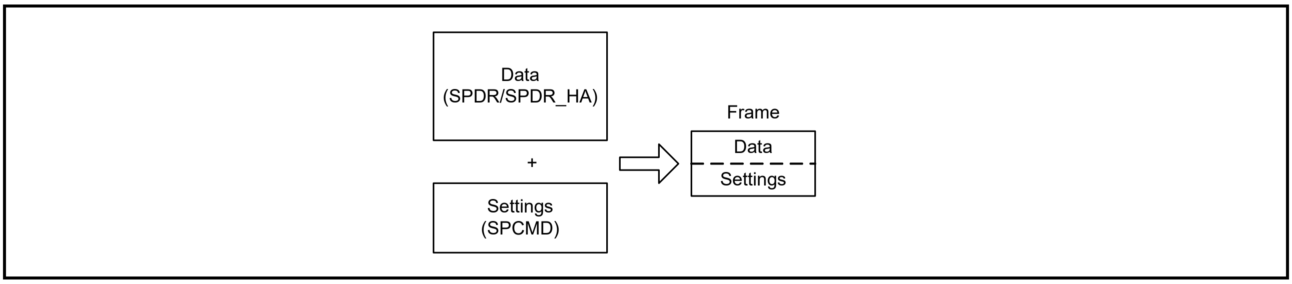

Data
(SPDR/SPDR_HA)
+
Settings
(SPCMD)
Frame
Data
Settings

Figure 32.55 Procedure to determine the format of serial transmission in master mode (SPI1)

Figure 32.56 shows the relationship between the command and the transmit and receive buffers in the sequence of operations.

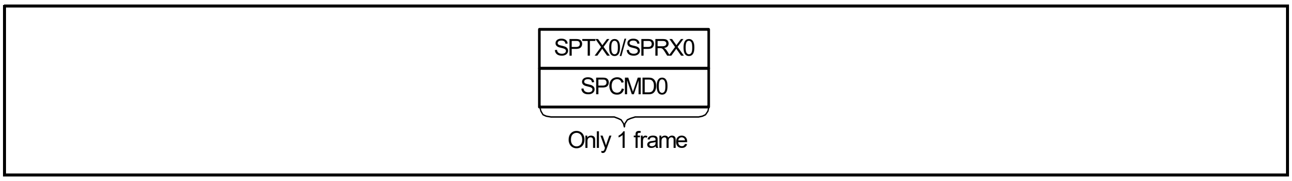


Figure 32.56 **Relationship between SPI Command Register and transmit and receive buffers in sequence operations (SPI1)**

(4) Initialization flow

Figure 32.57 shows an example of initialization flow for clock synchronous operation 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.

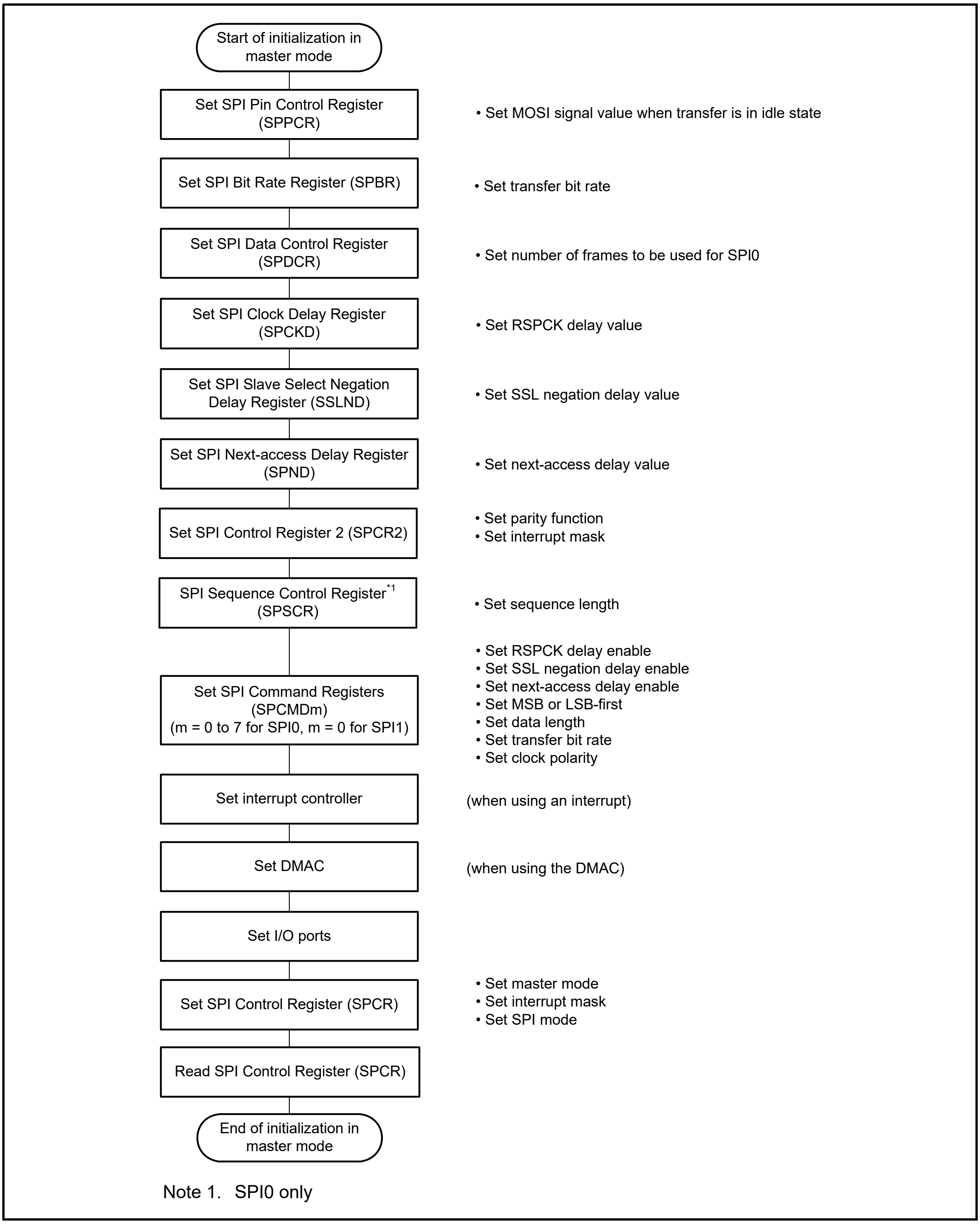

Start of initialization in master mode
Set SPI Pin Control Register (SPPCR)
• Set MOSI signal value when transfer is in idle state
Set SPI Bit Rate Register (SPBR)
• Set transfer bit rate
Set SPI Data Control Register (SPDCR)
• Set number of frames to be used for SPI0
Set SPI Clock Delay Register (SPCKD)
• Set RSPCK delay value
Set SPI Slave Select Negation Delay Register (SSLND)
• Set SSL negation delay value
Set SPI Next-access Delay Register (SPND)
• Set next-access delay value
Set SPI Control Register 2 (SPCR2)
• Set parity function
• Set interrupt mask
SPI Sequence Control Register*1 (SPSCR)
• Set sequence length
Set SPI Command Registers (SPCMDm) (m = 0 to 7 for SPI0, m = 0 for SPI1)
• Set RSPCK delay enable
• Set SSL negation delay enable
• Set next-access delay enable
• Set MSB or LSB-first
• Set data length
• Set transfer bit rate
• Set clock polarity
Set interrupt controller
(when using an interrupt)
Set DMAC
(when using the DMAC)
Set I/O ports
Set SPI Control Register (SPCR)
• Set master mode
• Set interrupt mask
• Set SPI mode
Read SPI Control Register (SPCR)
End of initialization in master mode
Note 1. SPI0 only

Figure 32.57 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 32.3.10.1, (9) Software processing flow.

Note: Mode fault errors are not generated in clock synchronous operation.

32.3.11.2 Slave mode operation

(1) Starting serial transfer

When the SPCR.SPMS bit is 1, the first RSPCKn edge triggers the start of a serial transfer in the SPI, and the SPI drives the MISOn output signal.

The SSLn0 input signal is not used in clock synchronous operation. For details on the SPI transfer format, see section 32.3.5, Transfer Format.

(2) Terminating serial transfer

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 = 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 32.3.5, Transfer Format.

(3) Initialization flow

Figure 32.58 shows an example of initialization flow for clock synchronous operation when 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.

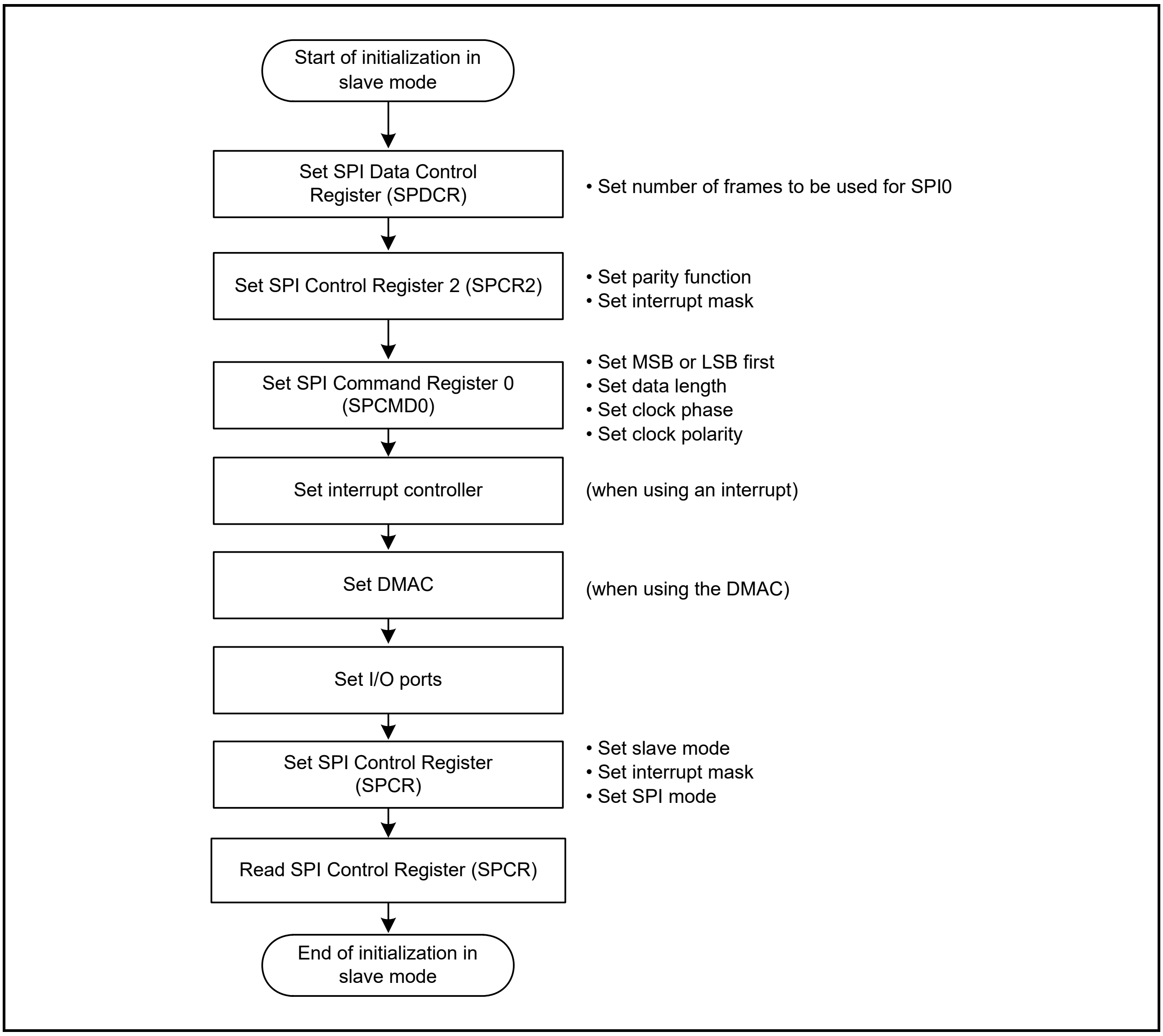

Start of initialization in slave mode
Set SPI Data Control Register (SPDCR)
• Set number of frames to be used for SPI0
Set SPI Control Register 2 (SPCR2)
• Set parity function
• Set interrupt mask
Set SPI Command Register 0 (SPCMD0)
• Set MSB or LSB first
• Set data length
• Set clock phase
• Set clock polarity
Set interrupt controller
(when using an interrupt)
Set DMAC
(when using the DMAC)
Set I/O ports
Set SPI Control Register (SPCR)
• Set slave mode
• Set interrupt mask
• Set SPI mode
Read SPI Control Register (SPCR)
End of initialization in slave mode

Figure 32.58 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 32.3.10.2, (6) Software processing flow.

Note 1. Mode fault errors are not generated in clock synchronous operation.

32.3.12 Loopback mode

When 1 is written to the SPLP2 or SPLP bit in the SPPCR register, 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, and connects the input path and output path of the shift register, establishing a loopback mode. The SPI does not shut off the path between the MOSIn pin and the shift register if the SPCR.MSTR bit is 1, and between the MISOn pin and the shift register if the SPCR.MSTR bit is 0. 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 32.12 lists the relationship between the SPLP2 and SPLP bits and the received data. Figure 32.59 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 32.12 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

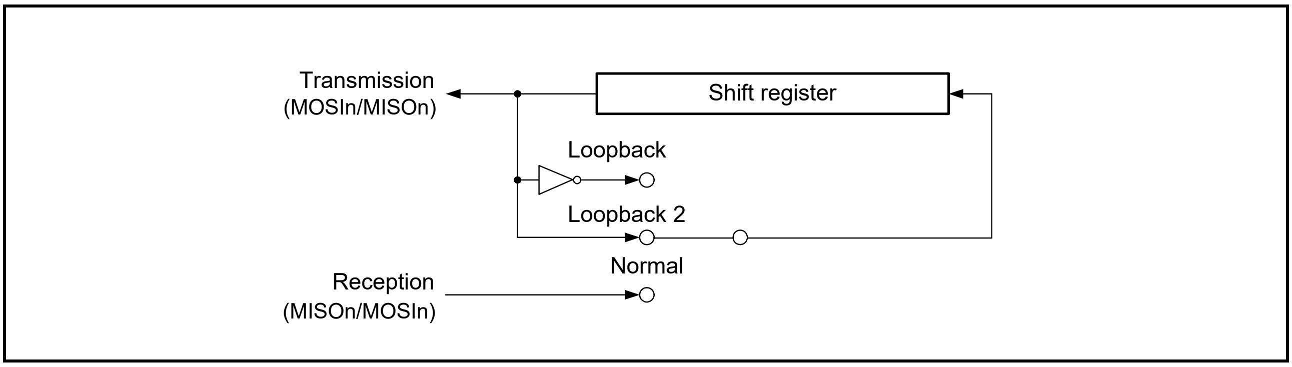

Transmission
(MOSIn/MISOn)
Shift register
Loopback
Loopback 2
Normal
Reception
(MISOn/MOSIn)

Figure 32.59 Configuration of shift register I/O paths in loopback mode for master mode

32.3.13 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for the transmit data and an error detecting unit used for the received data. To detect defects in these units, the parity circuit performs self-diagnosis as shown in Figure 32.60.

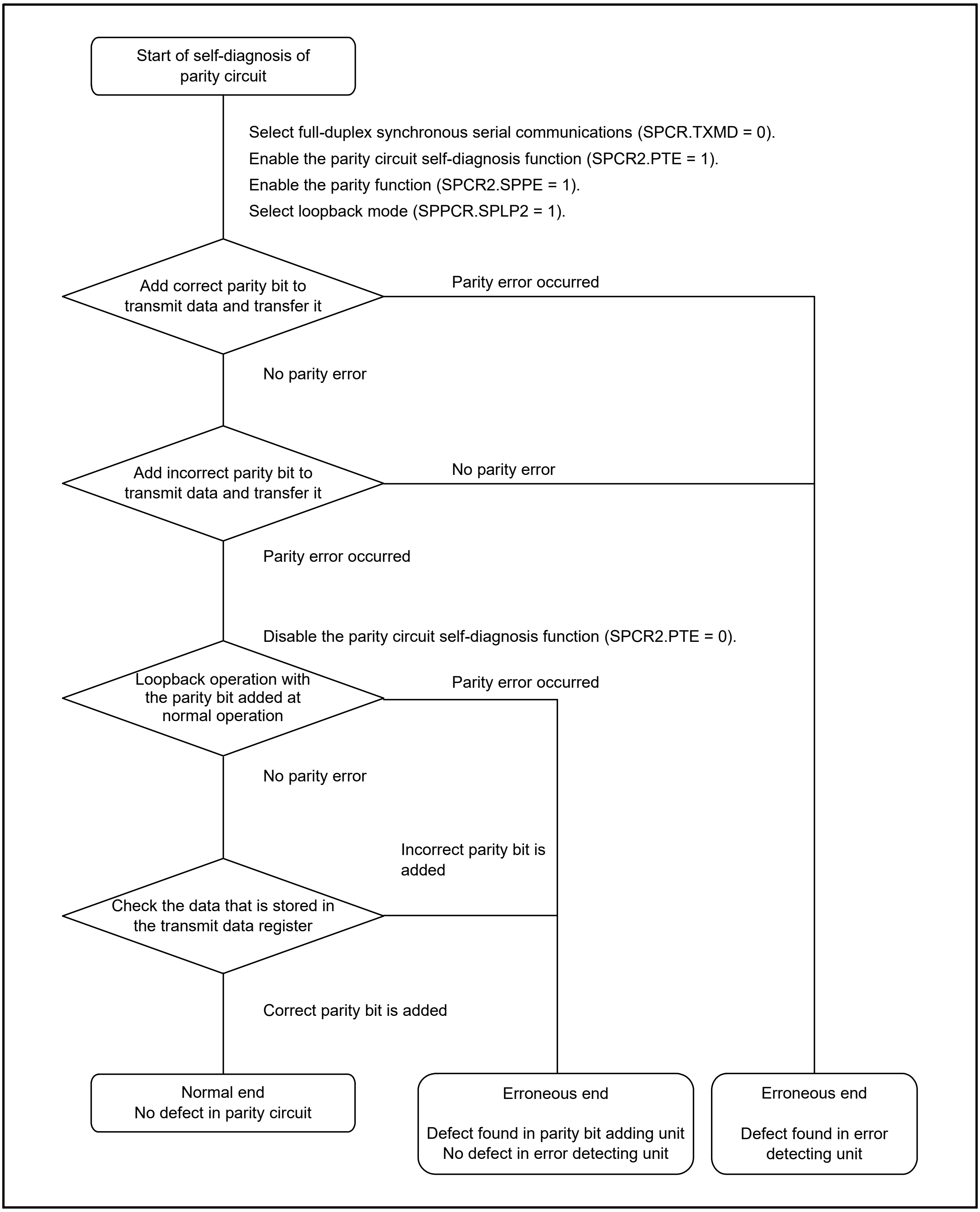

Start of self-diagnosis of parity circuit
Select full-duplex synchronous serial communications (SPCR.TXMD = 0).
Enable the parity circuit self-diagnosis function (SPCR2.PTE = 1).
Enable the parity function (SPCR2.SPPE = 1).
Select loopback mode (SPPCR.SPLP2 = 1).
Add correct parity bit to transmit data and transfer it
Parity error occurred
No parity error
Add incorrect parity bit to transmit data and transfer it
No parity error
Parity error occurred
Disable the parity circuit self-diagnosis function (SPCR2.PTE = 0).
Loopback operation with the parity bit added at normal operation
Parity error occurred
No parity error
Incorrect parity bit is added
Check the data that is stored in the transmit data register
Correct parity bit is added
Normal end
No defect in parity circuit
Erroneous end
Defect found in parity bit adding unit
No defect in error detecting unit
Erroneous end
Defect found in error detecting unit

Figure 32.60 Self-diagnosis flow for parity circuit

32.3.14 Interrupt Sources

The SPI has eight interrupt sources:

- Receive buffer full

- Transmit buffer empty
- Transmission-completed
- 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 32.13 lists the flags associated with the interrupt sources for the SPI. An interrupt is generated on satisfaction of an interrupt condition in Table 32.13. 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 a transfer-enabled status before setting the SPI. For information on how to set 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 occur while the ICU.IELSRj.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.IELSRj.IR flag becomes 0. A saved interrupt request is automatically discarded when it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally saved interrupt request can also be set to 0.

Table 32.13 SPI interrupt sources

Interrupt source	Symbol	Interrupt condition	DMAC/DTC activation
Receive buffer full	SPIn_SPRI	The receive buffer becomes full (SPSR.SPRF = 1) while the SPCR.SPRIE bit is 1	Possible
Transmit buffer empty	SPIn_SPTI	The transmit buffer becomes empty (SPSR.SPTEF = 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-completed	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 on conditions shown in Table 32.15.	Impossible

32.4 Event Link Operation

The Event Link Controller (ELC) can produce the following event output signals:

- Receive buffer full event output
- Transmit buffer empty event output
- Mode fault, underrun, overrun, or parity error event output
- SPI idle event output
- Transmission-completed event output.

The event link output signal is output regardless of the interrupt enable bit setting.

32.4.1 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 a serial transfer.

32.4.2 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.

32.4.3 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 32.5.4, Constraint on Mode Fault, Underrun, Overrun, or Parity Error Event Output if using this event signal.

(1) Mode fault

Table 32.14 lists the conditions for occurrence of a mode fault event.

Table 32.14 Conditions for occurrence of mode fault

Condition	SPCR.MODFEN bit	SSLn0 pin	Remark
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 the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1. Under these conditions, the MODF and UDRF flags are set to 1.

(3) Overrun

This event signal is output in response to an overrun when a serial transfer completes while the 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.

32.4.4 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).

32.4.5 Transmission-Completed Event Output

During both SPI and clock synchronous operations in master mode, an event is output when the IDLNF flag (SPI idle flag) changes from 1 to 0.

Table 32.15 Conditions for generation of transmission-completed event in slave mode

	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 mode or slave mode, an event is not output if 0 is written to the SPCR.SPE bit in transmission or the SPCR.SPE bit is cleared by the mode fault or underrun error.

32.5 Usage Notes

32.5.1 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.

32.5.2 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.

32.5.3 Constraint on Starting Transfer

If the ICU.IELSRj.IR flag is 1 at the start of transfer, an interrupt request is internally saved after transfer starts, and this can lead to unanticipated behavior of the ICU.IELSRj.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 = 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.IELSRj.IR flag to 0.

32.5.4 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 in multi-master mode (the SPCR.SPMS bit is 0, SPCR.MSTR bit is 1, and SPCR.MODFEN bit is 1).

32.5.5 Constraint on the SPRF and SPTEF Flags

If the polling flags, SPRF and SPTEF, are used, using an interrupt is prohibited, and you must set the SPCR.SPRIE and SPCR.SPTIE bits to 0. Either the interrupt or the flag can be used, but not both.

33. Cyclic Redundancy Check (CRC) Calculator

33.1 Overview

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 of 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.

Table 33.1 lists the specifications of the CRC calculator and Figure 33.1 shows a block diagram.

Table 33.1 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 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 to reduce power consumption	
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.

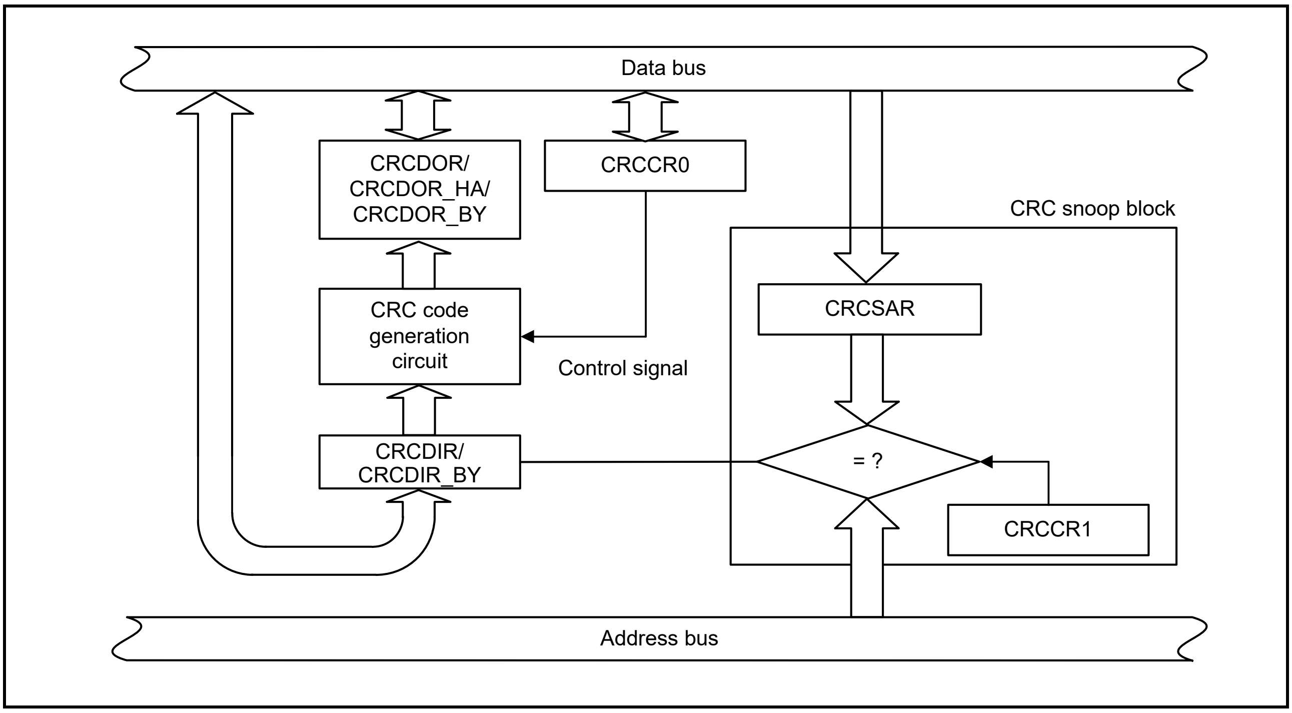


Figure 33.1 CRC calculator block diagram

33.2 Register Descriptions

33.2.1 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. Always set this bit to 1 when writing to this register.

GPS[2:0] bits (CRC Generating Polynomial Switching)

Set the GPS[2:0] bits to select the CRC Generating Polynomial.

LMS bit (CRC Calculation Switching)

Set this bit to select the bit order of the 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 33.3, Operation.

DORCLR bit (CRCDOR/CRCDOR_HA/CRCDOR_BY)

Write 1 to this bit to set the CRCDOR/CRCDOR_HA/CRCDOR_BY register to 0000 0000h. This bit is read as 0. Only 1 can be written to it.

33.2.2 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

Bit	Symbol	Bit name	Description	R/W
b7	CRCSEN	Snoop Enable	0: Disabled 1: Enabled.	R/W

CRCSWR bit (Snoop-On-Write/Read Switch)

The CRCSWR bit selects the direction of access in the address monitoring function.

When this bit is set to 0 (initial value), the CRC snoop operation to read a specific register address is valid. Similarly, when this bit is set to 1, the CRC snoop operation to write a specific register address is valid.

CRCSEN bit (Snoop Enable)

When the CRCSEN bit is set to 1, the CRC snoop operation is valid. When this bit is set to 0, the CRC snoop operation is invalid.

33.2.3 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.

33.2.4 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 calculation.

The CRCDOR_HA register is a read/write 16-bit register for CRC-16 or CRC-CCITT calculation.

The CRCDOR_BY register is a read/write 8-bit register for CRC-8 calculation.

Because its initial value is 0000 0000h, rewrite the CRCDOR/CRCDOR_HA/CRCDOR_BY register to performthe calculations using a value other than the initial value.

Data written to the CRCDIR/CRCDIR_BY register is CRC calculated and the result is stored in the CRCDOR/ CRCDOR_HA/CRCDOR_BY register. If the CRC code is calculated following 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.

33.2.5 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 the CRCSA[13:0] bits to the lower 14-bits of register address monitored by the CRC snoop operation.

Only the following addresses 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 0123h: SCI9.TDR, 4007 0125h: SCI9.RDR
- 4007 000Fh: SCI0.FTDRL,4007 0011h: SCI0.FRDRL
- 4007 002Fh: SCI1.FTDRL,4007 0031h: SCI1.FRDRL.

33.3 Operation

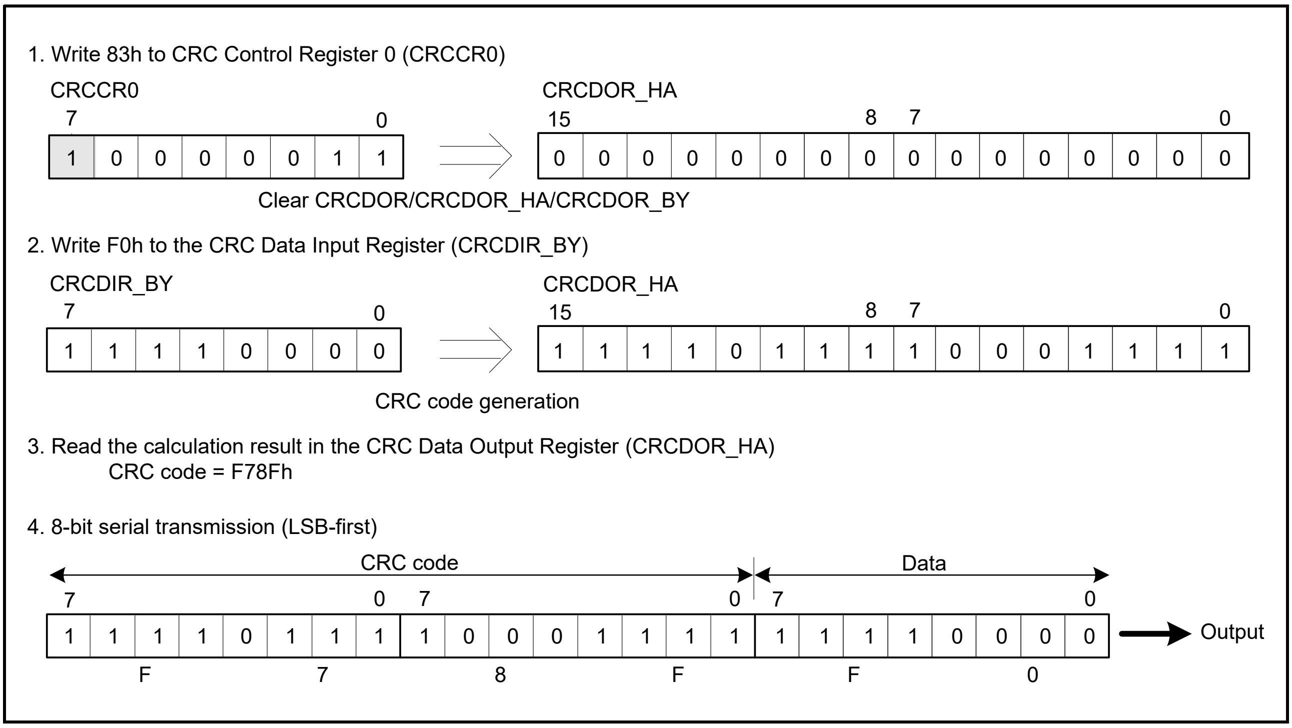
33.3.1 Basic Operation

The CRC calculator generates CRC codes for use in LSB-first or MSB-first transfer.

The following examples show CRC code generation for input data (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 a 32-bit CRC is in use, the valid bits of the CRC code are obtained in CRCDOR.

Figure 33.2 and Figure 33.3 show the LSB-first and MSB-first data transmission examples respectively. Figure 33.4 and Figure 33.5 show the LSB-first and MSB-first data reception examples.



1. Write 83h to CRC Control Register 0 (CRCCR0)

CRCCR0
7 0
1 0 0 0 0 0 1 1
CRCDOR_HA
15 8 7 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Clear CRCDOR/CRCDOR_HA/CRCDOR_BY

2. Write F0h to the CRC Data Input Register (CRCDIR_BY)

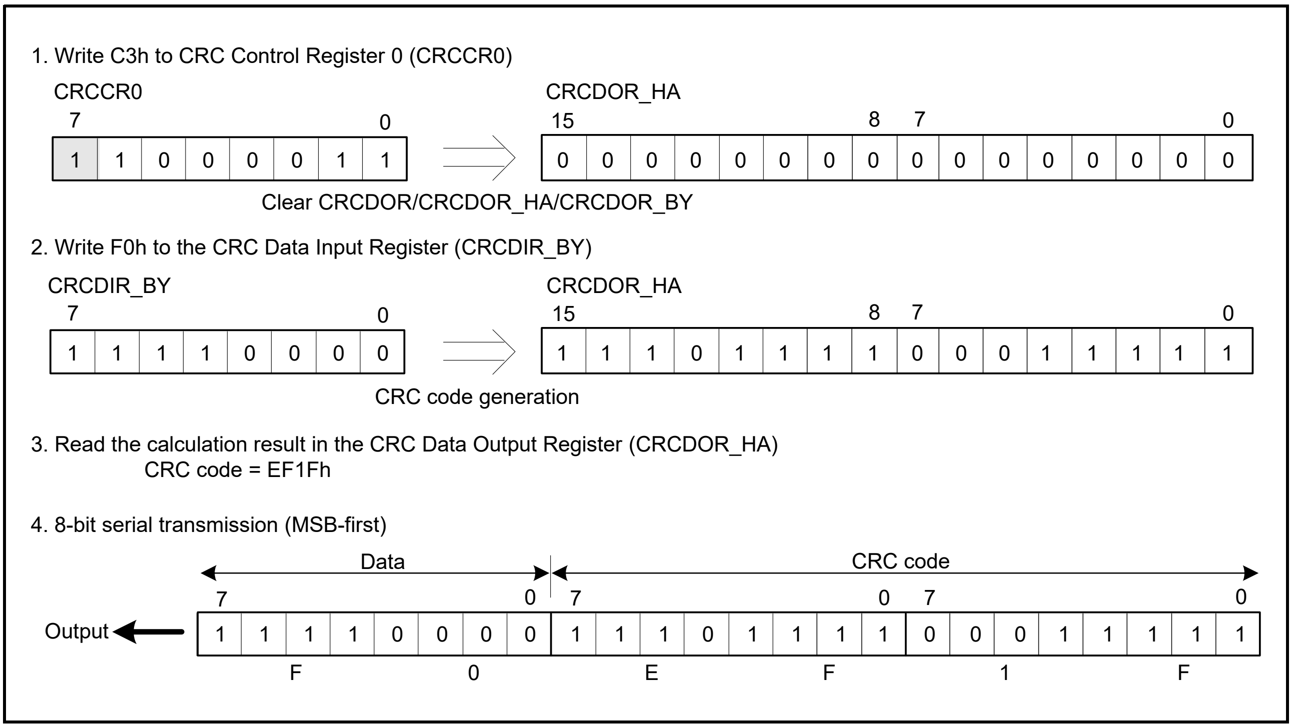
CRCDIR_BY
7 0
1 1 1 1 0 0 0 0
CRCDOR_HA
15 8 7 0
1 1 1 1 0 1 1 1 1 0 0 0 1 1 1 1
CRC code generation

3. Read the calculation result in the CRC Data Output Register (CRCDOR_HA)
CRC code = F78Fh

4. 8-bit serial transmission (LSB-first)

CRC code
Data
7 0 7 0 7 0
1 1 1 1 0 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0
Output
F 7 8 F F 0

Figure 33.2 LSB-first data transmission



1. Write C3h to CRC Control Register 0 (CRCCR0)

CRCCR0
7 0
1 1 0 0 0 0 1 1
CRCDOR_HA
15 8 7 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Clear CRCDOR/CRCDOR_HA/CRCDOR_BY

2. Write F0h to the CRC Data Input Register (CRCDIR_BY)

CRCDIR_BY
7 0
1 1 1 1 0 0 0 0
CRCDOR_HA
15 8 7 0
1 1 1 0 1 1 1 1 0 0 0 1 1 1 1 1
CRC code generation

3. Read the calculation result in the CRC Data Output Register (CRCDOR_HA)
CRC code = EF1Fh

4. 8-bit serial transmission (MSB-first)

Data
CRC code
7 0 7 0 7 0
Output
1 1 1 1 0 0 0 0 1 1 1 0 1 1 1 1 0 0 0 1 1 1 1 1
F 0 E F 1 F

Figure 33.3 MSB-first data transmission

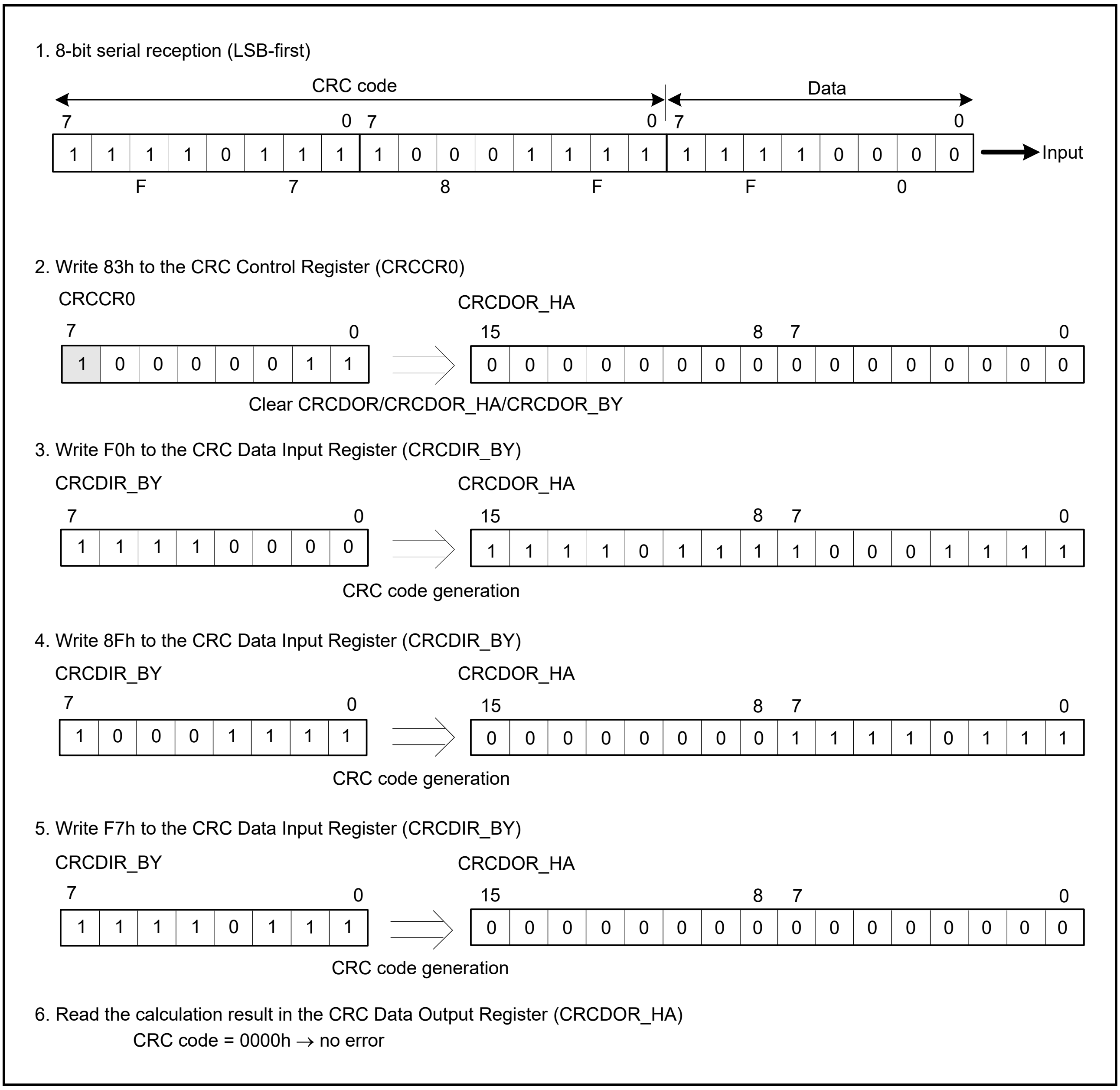

1. 8-bit serial reception (LSB-first)
CRC code
Data
7 0 7 0 7 0
1 1 1 1 0 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0
Input
F 7 8 F F 0
2. Write 83h to the CRC Control Register (CRCCR0)
CRCCR0
7 0
1 0 0 0 0 0 1 1
CRCDOR_HA
15 8 7 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Clear CRCDOR/CRCDOR_HA/CRCDOR_BY
3. Write F0h to the CRC Data Input Register (CRCDIR_BY)
CRCDIR_BY
7 0
1 1 1 1 0 0 0 0
CRCDOR_HA
15 8 7 0
1 1 1 1 0 1 1 1 1 0 0 0 1 1 1 1
CRC code generation
4. Write 8Fh to the CRC Data Input Register (CRCDIR_BY)
CRCDIR_BY
7 0
1 0 0 0 1 1 1 1
CRCDOR_HA
15 8 7 0
0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1
CRC code generation
5. Write F7h to the CRC Data Input Register (CRCDIR_BY)
CRCDIR_BY
7 0
1 1 1 1 0 1 1 1
CRCDOR_HA
15 8 7 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
CRC code generation
6. Read the calculation result in the CRC Data Output Register (CRCDOR_HA)
CRC code = 0000h → no error

Figure 33.4 LSB-first data reception

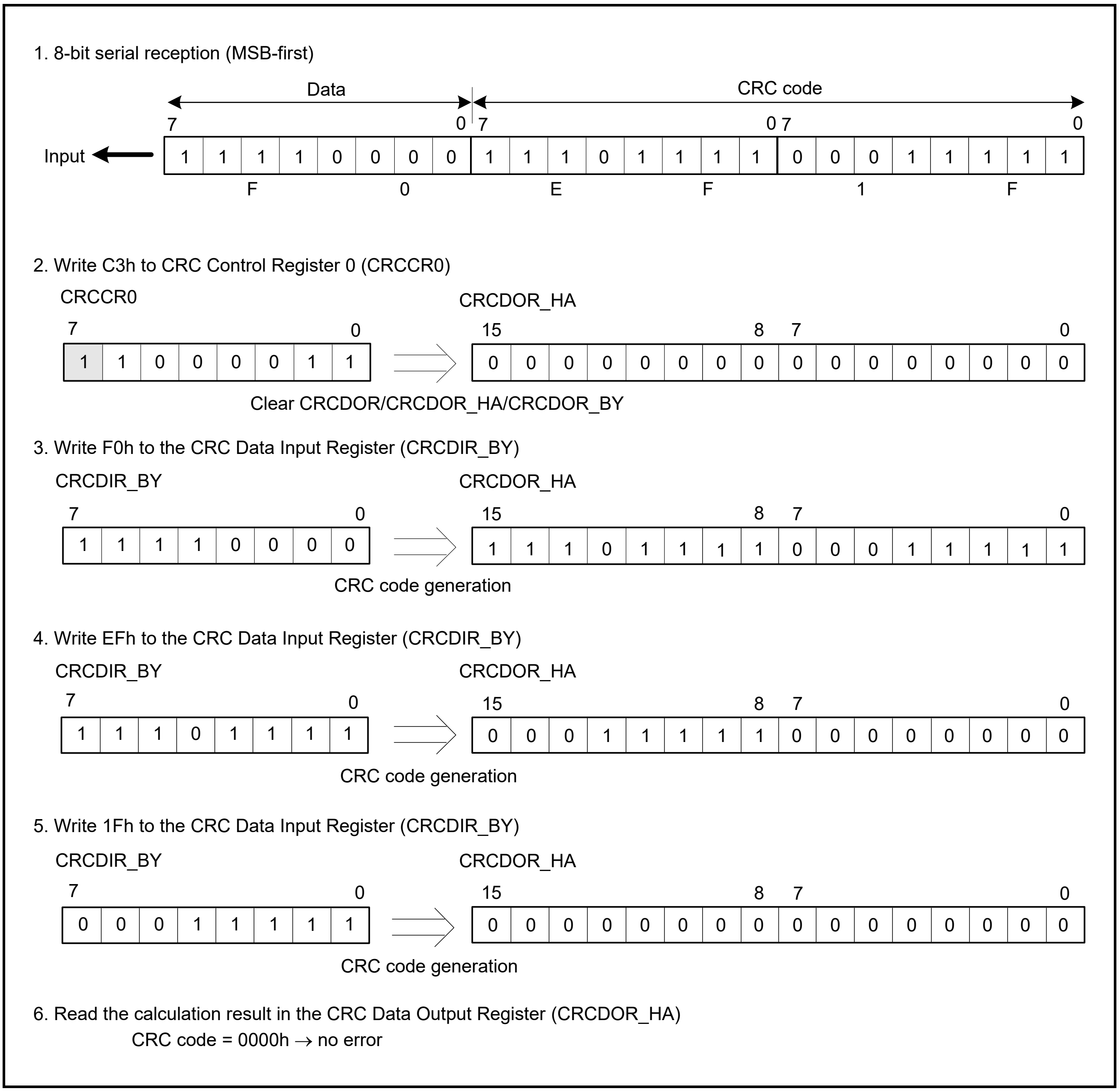

1. 8-bit serial reception (MSB-first)
Data
CRC code
7 0 7 0 7 0
Input
1 1 1 1 0 0 0 0 1 1 1 0 1 1 1 1 0 0 0 1 1 1 1 1
F 0 E F 1 F
2. Write C3h to CRC Control Register 0 (CRCCR0)
CRCCR0
7 0
1 1 0 0 0 0 1 1
CRCDOR_HA
15 8 7 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Clear CRCDOR/CRCDOR_HA/CRCDOR_BY
3. Write F0h to the CRC Data Input Register (CRCDIR_BY)
CRCDIR_BY
7 0
1 1 1 1 0 0 0 0
CRCDOR_HA
15 8 7 0
1 1 1 0 1 1 1 1 0 0 0 1 1 1 1 1
CRC code generation
4. Write EFh to the CRC Data Input Register (CRCDIR_BY)
CRCDIR_BY
7 0
1 1 1 0 1 1 1 1
CRCDOR_HA
15 8 7 0
0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0
CRC code generation
5. Write 1Fh to the CRC Data Input Register (CRCDIR_BY)
CRCDIR_BY
7 0
0 0 0 1 1 1 1 1
CRCDOR_HA
15 8 7 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
CRC code generation
6. Read the calculation result in the CRC Data Output Register (CRCDOR_HA)
CRC code = 0000h → no error

Figure 33.5 MSB-first data reception

33.3.2 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 calculations, 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 the CPU, DMA, and DTC, the CRC calculator stores the data in the CRCDIR_BY register and performs CRC calculations. 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 the CPU, DMA, and DTC, the CRC calculator stores the data in the CRCDIR_BY register and performs CRC calculations.

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.

33.4 Usage Notes

33.4.1 Settings for the Module-Stop State

The Module Stop Control Register C (MSTPCRC) can enable or disable CRC calculator operation. The CRC is stopped after a reset. The registers become accessible on releasing the module-stop state. For details, see section 11, Low Power Modes.

33.4.2 Notes on Transmission

The sequence of transmission for the CRC code differs based on whether the transmission is LSB-first or MSB-first. Figure 33.6 shows an LSB-first and MSB-first data transmission.

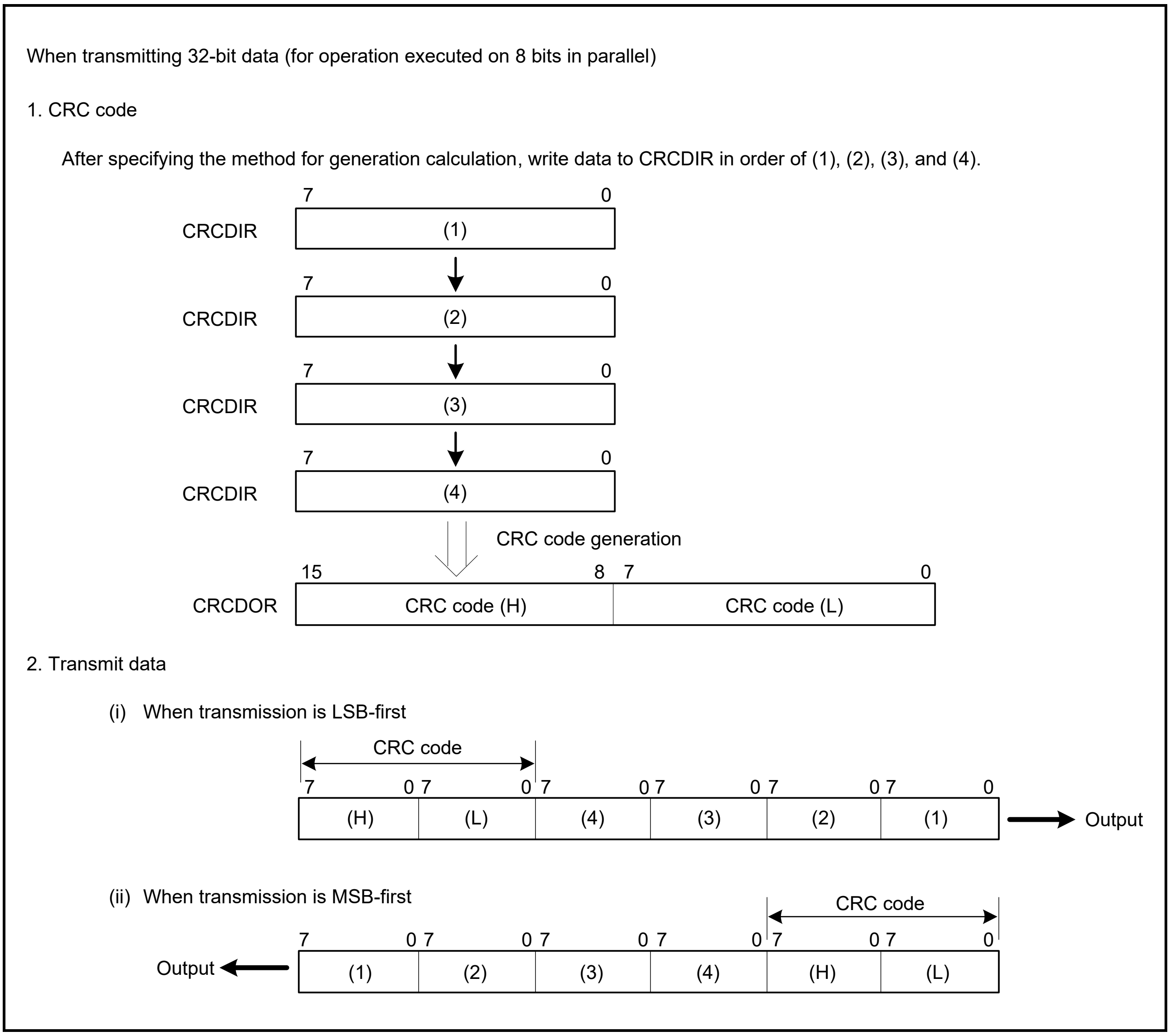

When transmitting 32-bit data (for operation executed on 8 bits in parallel)
1. CRC code
After specifying the method for generation calculation, write data to CRCDIR in order of (1), (2), (3), and (4).
CRCDIR
7
0
(1)
CRCDIR
(2)
CRCDIR
(3)
CRCDIR
(4)
CRC code generation
CRCDOR
15
8
7
0
CRC code (H)
CRC code (L)
2. Transmit data
(i) When transmission is LSB-first
CRC code
(H)
(L)
(4)
(3)
(2)
(1)
Output
(ii) When transmission is MSB-first
CRC code
Output
(1)
(2)
(3)
(4)
(H)
(L)

Figure 33.6 LSB-first and MSB-first data transmission

34. 14-Bit A/D Converter (ADC14)

34.1 Overview

The MCU provides a 14-bit successive approximation A/D converter (ADC14) unit. Up to 8 analog input channels are selectable. Temperature sensor output and internal reference voltage can be selected for conversion. The A/D conversion accuracy is 14-bit conversion, making it possible to optimize the trade-off between speed and resolution in generating a digital value.

The ADC14 supports the following operating modes:

- Single scan mode to convert the analog inputs of arbitrarily selected channels in ascending order of channel number
- Continuous scan mode to sequentially convert the analog inputs of arbitrarily selected channels continuously in ascending order of channel number
- 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, you can start group A and group B A/D conversion at different times by individually selecting their scan start conditions. In addition, when a priority control operation for group A is set, the ADC14 accepts group A scan start during group B A/D conversion, suspending group B conversion. This allows you to assign higher priority to A/D conversion start for group A.

In double trigger mode, the analog input of an arbitrarily selected channel is converted in single scan mode or group scan mode (group A), and the data converted by the first and second A/D conversion start triggers are stored in different registers, providing duplexing of A/D-converted data.

Self-diagnosis is performed once at the beginning of each scan, and one of the three voltage values generated in the ADC14 is A/D-converted.

The temperature sensor output and internal reference voltage cannot be selected for A/D conversion simultaneously. The temperature sensor output and the internal reference voltage are converted independently. 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), 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 ADC14 provides a compare function (window A and window B). This compare function specifies the upper reference value and lower 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 34.1 lists the ADC14 specifications, Table 34.2 lists the functions, and Figure 34.1 shows a block diagram. Table 34.2 lists the I/O pins.

Table 34.1 ADC14 specifications (1 of 3)

Parameter	Specifications
Number of units	One unit
Input channels	Up to 8 channels (AN004 to AN006, AN009, AN010, AN017, AN019, and AN020)
Extended analog function	Temperature sensor output, internal reference voltage
A/D conversion method	Successive approximation method
Resolution	14 bits, selectable to 14-bit or 12-bit conversion
Conversion time	0.79 μs/channel, when 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 34.1 ADC14 specifications (2 of 3)

Parameter	Specifications
Data registers	• 8 registers for analog input: - One register for A/D-converted data duplication in double trigger mode - Two registers for A/D-converted data duplication during extended operation in double trigger mode • One register for temperature sensor output • One register for internal reference voltage • One register for self-diagnosis • A/D conversion results are stored in A/D data registers • 12-bit and 14-bit accuracy output for A/D conversion results • A/D-converted value addition mode, in which the sum of all A/D conversion results are stored in the A/D data registers as the conversion accuracy bit count + 2 bits*4 • Double trigger mode, selectable in single scan and group scan modes: - The first unit of A/D-converted analog input data on one selected channel is stored in the data register for the channel, and the second unit is stored in the duplication register. • Extended operation in double trigger mode (available for specific triggers): - A/D-converted analog input data on one selected channel is stored in the duplication register provided for the associated trigger.
Operating modes	• Single scan mode: - A/D conversion is performed only once on the analog inputs of arbitrarily selected channels, on the temperature sensor output, and on the internal reference voltage • Continuous scan mode: - A/D conversion is performed repeatedly on the analog inputs of arbitrarily selected channels, on the temperature sensor output, and on the internal reference voltage • Group scan mode: - A/D conversion is performed only once on the analog inputs of arbitrarily selected channels divided into group A and group B - The scan start conditions can be independently selected for group A and group B, 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 stops and A/D conversion is performed on group A - Restart (rescan) of group B conversion after completion of group A conversion can be set.
Conditions for A/D conversion start	• Software trigger • Synchronous trigger from the Event Link Controller (ELC) • Asynchronous trigger from the external trigger pin, ADTRG0.
Function	• Variable sampling state count • Self-diagnosis of ADC14 • Selectable A/D-converted value addition mode or average mode • Analog input disconnection detection function (discharge and precharge functions) • Double trigger mode (duplication of A/D conversion data) • Switching function for 12-bit and 14-bit conversion*2 • Automatic clear function for A/D data registers • Digital comparison of values in the comparison and data registers, and between values in the data registers.
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 a single scan. - A compare interrupt request (ADC140_CMPAI/ADC140_CMPBI) can be generated in response to matches with a condition for digital comparison. - A window compare ELC event signal (ADC140_WCMPM) can be generated in response to matches with a condition for digital comparison. - A window compare ELC event signal (ADC140_WCMPUM) can be generated in response to mismatches with a condition for 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 channel 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 scans, 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

Table 34.1 ADC14 specifications (3 of 3)

Parameter	Specifications
Reference voltage	• VREFH0, AVCC0, or internal reference voltage is selectable as the high-potential reference voltage • VREFL0 or AVSS0 is selectable as the low-potential reference voltage.
Module-stop function	Module-stop state can be specified*3

Note: When selecting the temperature sensor output or the internal reference voltage, do not use continuous scan mode or group scan mode.

Note 1. Peripheral module clock PCLKB frequency is specified in the SCKDIVCR.PCKB[2:0] bits and A/D conversion clock ADCLK is specified in the SCKDIVCR.PCKC[2:0] bits. Maximum frequency of PCLKB is 32 MHz and maximum frequency of PCLKC (ADCLK) is 64 MHz.

Note 2. Changing the A/D conversion accuracy also changes the A/D conversion time. For details, see section 34.3.6, Analog Input Sampling and Scan Conversion Time.

Note 3. For details, see section 11, Low Power Modes.

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.

Table 34.2 ADC14 functions

Parameter			ADC140
Analog input channel			AN004 to AN006, AN009, AN010, AN017, AN019, and AN020 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. For details, see section 11, Low Power Modes.

Note 2. Wait for 1 µs or longer to start A/D conversion after release from the module-stop state.

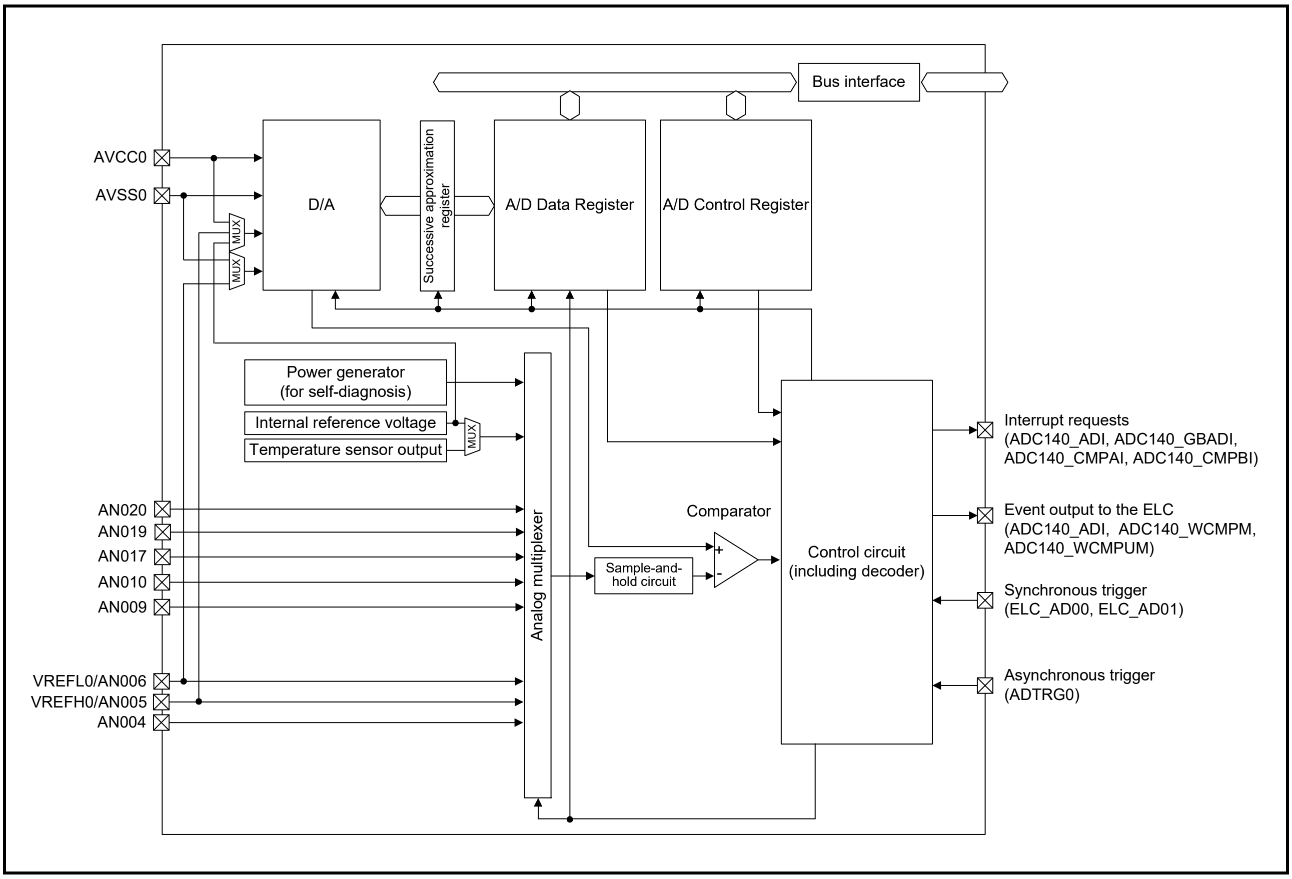

Bus interface
AVCC0
AVSS0
D/A
MUX
MUX
Successive approximation register
A/D Data Register
A/D Control Register
Power generator (for self-diagnosis)
Internal reference voltage
Temperature sensor output
MUX
Analog multiplexer
AN020
AN019
AN017
AN010
AN009
VREFL0/AN006
VREFH0/AN005
AN004
Sample-and-hold circuit
Comparator
Control circuit (including decoder)
Interrupt requests (ADC140_ADI, ADC140_GBADI, ADC140_CMPAI, ADC140_CMPBI)
Event output to the ELC (ADC140_ADI, ADC140_WCMPM, ADC140_WCMPUM)
Synchronous trigger (ELC_AD00, ELC_AD01)
Asynchronous trigger (ADTRG0)

Figure 34.1 ADC14 block diagram

Table 34.3 ADC14 I/O pins

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
	AN004 to AN006, AN009, AN010, AN017, AN019, and AN020	Input	Analog input pins 4 to 6, 9, 10, 17, 19, and 20
	ADTRG0	Input	External trigger input pin for starting A/D conversion

34.2 Register Descriptions

34.2.1 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 data registers include:

- ADDRy registers (y = 4 to 6, 9, 10, 17, 19, and 20): 16-bit read-only registers for storing the A/D conversion results
- ADDBLDR: 16-bit read-only register for storing the A/D conversion results in response to the second trigger in double trigger mode

- ADDBLDRA and ADDBLDRB: 16-bit read-only registers for storing the A/D conversion results in response to the respective triggers during extended operation in double trigger mode
- ADTSDR: 16-bit read-only register for storing the A/D conversion result of temperature sensor output
- ADOCDR: 16-bit read-only register for storing the A/D result of internal reference voltage.

The following conditions determine the formats for data in these registers:

- The setting in the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right setting)
- The setting in the A/D Conversion Accuracy Specify bits (ADCER.ADPRC[1:0]) (12-bit or 14-bit setting)
- The setting in the addition/average Count Select bits (ADADC.ADC[2:0]) (once, twice, thrice, four times, or 16 times setting)
- The setting in the Average Mode Enable bit (ADADC.AVEE) (addition or average setting).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

The data formats for each condition are as follows:

Settings for flush-right data with 14-bit accuracy

Address(es): ADC140.ADDR4 4005 C028h to ADC140.ADDR6 4005 C02Ch, ADC140.ADDR9 4005 C032h, ADC140.ADDR10 4005 C034h, ADC140.ADDR17 4005 C042h, ADC140.ADDR19 4005 C046h, ADC140.ADDR20 4005 C048h, 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

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

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

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

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value average mode. When the A/D-converted value average mode is selected, this register indicates the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting in the A/D Data Register Format Select bit in the same way as in normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit or 14-bit accuracy (ADPRC bit setting), 1, 2, 3, or 4 times can be selected for A/D -converted value addition. 16 times can also be selected, but only with 12-bit accuracy selected.

In addition mode, 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 based on the settings of the A/D Data Register Format Select bits.

When converting 1, 2, 3, or 4 times in addition mode with 12-bit or 14-bit accuracy, the conversion result is stored in the A/D data register as a 2-bit extended value of the specified accuracy.

When converting 16 times in addition mode with 12-bit accuracy, the A/D conversion result is stored in the A/D data register as a 4-bit-extended value of the specified accuracy.

The data formats for each given condition are shown below.

Settings for flush-right data with 14-bit accuracy in A/D-converted value addition mode

	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

Settings for flush-right data with 12-bit accuracy in A/D-converted value addition mode

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[15:0]*1															
	—	—	AD[13:0]*2													
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]*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.

Settings for flush-left data with 14-bit accuracy in A/D-converted value addition mode

	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

Settings for flush-left data with 12-bit accuracy in A/D-converted value addition mode

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[15:0]*1															
	AD[13:0]*2														—	—
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]*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 in addition mode.

34.2.2 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 ADC14. In addition to the AD[13:0] bits indicating the A/D-converted value, it includes the Self-Diagnosis Status bit (DIAGST).

The following conditions determine the format for data in this register:

- The setting in the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right setting)
- The setting in the A/D Conversion Accuracy Specify bits (ADCER.ADPRC[1:0]) (12-bit or 14-bit setting).

The A/D-converted value addition and average modes cannot be applied to the A/D self-diagnosis function. For details on self-diagnosis, see section 34.2.11, A/D Control Extended Register (ADCER).

This section describes the data formats for each condition.

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 34.2.11, A/D Control Extended Register (ADCER).	R

Note 1. Reference voltage refers to VREFH0.

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 34.2.11, A/D Control Extended Register (ADCER).	R

Note 1. Reference voltage refers to VREFH0.

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 34.2.11, 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.

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 on self-diagnosis, see section 34.2.11, 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.

34.2.3 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 valid in double trigger mode.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b6	GBADIE	Group B Scan End Interrupt Enable	0: Disable ADC140_GBADI interrupt generation on group B scan completion 1: Enable ADC140_GBADI interrupt generation on group B scan completion. Group B scan works only in group scan mode.	R/W
b7	DBLE	Double Trigger Mode Select	0: Deselect double trigger mode 1: Select 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 an asynchronous trigger (ADTRG0).	R/W
b9	TRGE	Trigger Start Enable	0: Disable A/D conversion to be started by a synchronous or asynchronous trigger 1: Enable A/D conversion to be started by a 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: Stop A/D conversion process 1: Start A/D conversion process.	R/W

Note 1. To start A/D conversion using an external pin (asynchronous trigger):
After a high-level signal is input to the external pin (ADTRG0), write 1 to both the TRGE and EXTRG bits in the ADCSR register and drive the ADTRG0 signal low. With these settings, the scan conversion process starts on detection of the falling edge of ADTRG0. For this configuration, the pulse width of the low-level input must be at least 1.5 clock PCLKB 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 from the specified analog input channel are stored in the A/D Data Register y when conversion is started by the first trigger, and in the A/D Data Duplexing Register when started by the second trigger. Table 34.4 shows selection of the channel for double-triggered operation.

In double trigger mode, the channels selected in the ADANSA0 and ADANSA1 registers are invalid, and the channel selected in the DBLANS[4:0] bits is A/D-converted instead.

When double trigger mode is used in group scan mode, double-trigger control is applied only to group A and not to group B. This means that multi-channel analog input can be selected for group B even in double trigger mode.

Only set the DBLANS[4:0] bits when the ADST bit is 0. Do not set these bits at the same time you write 1 to the ADST bit.

To enter A/D-converted value addition/average mode when in double trigger mode, select the channel using the DBLANS[4:0] bits in the ADANSA0 and ADANSA1 registers.

Table 34.4 Relationship between DBLANS bit settings and double-trigger enabled channels

DBLANS[4:0]	Duplication channel	DBLANS[4:0]	Duplication channel
00100	AN004	10001	AN017
00101	AN005	10011	AN019
00110	AN006	10100	AN020
01001	AN009		
01010	AN010		

Note: A/D-converted data from 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)

The DBLE bit selects or deselects double trigger mode. Double trigger mode can only be operated by the synchronous trigger (ELC) selected in the ADSTRGR.TRSA[5:0] bits.

Double trigger mode operates as follows:

- The ADC140_ADI interrupt is not output on completion of the first conversion but on completion of the second conversion
- The A/D conversion results from the duplication channel (selected in the DBLANS[4:0] bits) started by the first trigger are stored in the A/D Data Register y and those started by the second trigger are stored in 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 triggering cannot be used in double trigger mode. Always set the ADST bit to 0 before setting the DBLE bit. In other words, do not set this bit at the same time as writing 1 to the ADST bit.

EXTRG bit (Trigger Select)

The EXTRG bit selects the synchronous trigger or 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 and asynchronous triggers. Set this bit to 1 in group scan mode.

ADHSC bit (A/D Conversion Mode Select)

The ADHSC bit selects either the high-speed or low-current mode for A/D conversion.

For details on how to rewrite this bit, see section 34.8.8, ADHSC Bit Rewriting Procedure.

ADCS[1:0] bits (Scan Mode Select)

The ADCS[1:0] bits select the scan mode.

In single scan mode, A/D conversion is performed for the analog inputs selected in the ADANSA0 and ADANSA1 registers in ascending order of channel number, for a maximum of 8 channels. When 1 cycle of A/D conversion completes for all the selected channels, the scan conversion stops. When the temperature sensor output or internal reference voltage is selected, A/D conversion of the designated analog input channels is followed by A/D conversion of the temperature sensor output or internal reference voltage, in that order.

In continuous scan mode, when the ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs selected in the ADANSA0 and ADANSA1 registers in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion is repeated from the first channel. If the ADCSR.ADST bit is set to 0 during continuous scan, A/D conversion stops even when scanning is in progress. When the temperature sensor output or internal reference voltage is selected, A/D conversion of the designated analog input channels is followed by A/D conversion of the temperature sensor output or internal reference voltage, in that order.

In group scan mode:

- Group A scanning is started by the synchronous trigger (ELC) selected in the ADSTRGR.TRSA[5:0] bits. A/D conversion is performed on group A analog inputs, up to the maximum number of 8 channels selected in the ADANSA0 and ADANSA1 registers, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion stops.
- Group B scanning is started by the synchronous trigger (ELC) selected in the ADSTRGR.TRSB[5:0] bits. A/D conversion is performed on group B analog inputs, up to the maximum number of 8 channels selected in the ADANSB0 and ADANSB1 registers, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, conversion stops.

If the conversion processes in group A and B occur at the same time, those conversions 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, to assign priority to group A conversion.

In group scan mode, select different channels and triggers for group A and group B.

When the temperature sensor output or internal reference voltage is selected, A/D conversion of the designated analog input channels is followed by A/D conversion of the temperature sensor output and the internal reference voltage, in that order.

Set the ADST bit to 0 before setting the ADCS[1:0] bits. In other words, do not set both the ADCS[1:0] and ADST bits to 1 at the same time.

Table 34.5 Selectable targets for A/D conversion based on scan mode and double trigger mode settings

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 the A/D conversion process. Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and analog inputs to be converted.

[Setting conditions]

- On writing 1 through software
- When the synchronous trigger (ELC) selected in the ADSTRGR.TRSA[5:0] bits is detected when ADCSR.EXTRG is 0 and ADCSR.TRGE is 1
- When the synchronous trigger (ELC) selected in the ADSTRGR.TRSB[5:0] bits is detected when ADCSR.TRGE is set to 1 in group scan mode
- When the asynchronous trigger is detected while the ADCSR.TRGE and ADCSR.EXTRG bits are set to 1, and the ADSTRGR.TRSA[5:0] bits are set to 000000b
- When group A priority control operation mode is 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 starts.

[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 completes in single scan mode
- When group A scan completes in group scan mode
- When group B scan completes in group scan mode
- When group A priority control operation mode is 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 completes.

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), do not set the ADST bit to 0. When forcing A/D conversion to terminate, follow the procedure for clearing the ADST bit.

Note: If the single scan continuous function is used (ADGSPCR.GBRP = 1) when the group priority operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), the ADST bit is 1.

34.2.4 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
	—	—	—	—	—	ANSA10	ANSA09	—	—	ANSA06	ANSA05	ANSA04	—	—	—	—
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
b6 to b4	ANSA06 to ANSA04	A/D Conversion Channels Select	0: Associated input channel not selected 1: Associated input channel selected Bit [6] (ANSA06) corresponds to AN006 and bit [4] (ANSA04) corresponds to AN004.	R/W
b8, b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10, b9	ANSA10, ANSA09	A/D Conversion Channels Select	0: Associated input channel not selected 1: Associated input channel selected Bit [10] (ANSA10) corresponds to AN010 and bit [0] (ANSA09) corresponds to AN009.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ANSAn bits (n = 04 to 06, 09, and 10) (A/D Conversion Channels Select)

The ANSAn.ADANSA0 bits select the analog input channels for A/D conversion from AN004 to AN006, AN009, and AN010. The selected channels and their number can be set arbitrarily. The ANSA04 bit is associated with AN004 and the ANSA10 bit is associated with AN010.

When performing A/D conversion on the temperature sensor output or internal reference voltage, set the ADANSA0 register to 0000h to deselect all analog input channels.

In double trigger mode, the channel selected in the ADANSA0 register is invalid, and the channel specified in the ADCSR.DBLANS[4:0] bits is selected in group A instead.

In group scan mode, do not select the channels specified in A/D Channel Select Register B0 (ADANSB0) and A/D Channel Select Register B1 (ADANSB1).

Only set the ADANSA0 register when the ADCSR.ADST bit is 0.

34.2.5 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
	—	—	—	—	—	—	—	—	—	—	—	ANSA20	ANSA19	—	ANSA17	—
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
b1	ANSA17	A/D Conversion Channels Select	0: Associated input channel not selected 1: Associated input channel selected Bit [1] (ANSA17) corresponds to AN017.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4, b3	ANSA20, ANSA19	A/D Conversion Channels Select	0: Associated input channel not selected 1: Associated input channel selected Bit [4] (ANSA20) corresponds to AN020 and Bit [3] (ANSA19) corresponds to AN019.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ANSAn bits (n = 17, 19, and 20) (A/D Conversion Channels Select)

The ADANSA1.ANSAn bits select the analog input channels for A/D conversion from AN017, AN019, and AN020. The selected channels and their number can be set arbitrarily. The ANSA17 bit is associated with AN017 and the AN020 bit is associated with AN020.

When performing A/D conversion on the temperature sensor output or internal reference voltage, set the ADANSA1 register to 0000h to deselect all analog input channels.

In double trigger mode, the channel selected in the ADANSA1 register is invalid, and the channel selected in the ADCSR.DBLANS[4:0] bits is selected in group A instead.

In group scan mode, do not select the channels specified in A/D Channel Select Register B0 (ADANSB0) and A/D Channel Select Register B1 (ADANSB1).

Only set the ADANSA1 register when the ADCSR.ADST bit is 0.

34.2.6 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
	—	—	—	—	—	ANSB10	ANSB09	—	—	ANSB06	ANSB05	ANSB04	—	—	—	—
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
b6 to b4	ANSB06 to ANSB04	A/D Conversion Channels Select	0: Associated input channel not selected 1: Associated input channel selected Bit[6] (ANSB06) corresponds to AN006 and bit[4] (ANSB04) corresponds to AN004.	R/W
b8, b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b10, b9	ANSB10, ANSB09	A/D Conversion Channels Select	0: Associated input channel not selected 1: Associated input channel selected Bit [10] (ANSB10) corresponds to AN010 and Bit [0] (ANSB09) corresponds to AN009.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ANSBn bits (n = 04 to 06, 09, and 10) (A/D Conversion Channels Select)

The ADANSB0.ANSBn bits select the analog input channels for A/D conversion from AN004 to AN006, AN009, and AN010 in group B when group scan mode is selected. The ADANSB0 register is only used for group scan mode and not for any other modes.

Do not select the channels specified in group A as selected in the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits in double trigger mode.

The ANSB04 bit is associated with AN004, the ANSB06 bit is associated with AN006, and the ANSB10 bit is associated with AN010.

When performing A/D conversion on the temperature sensor output or internal reference voltage, set the ADANSB0 register to 0000h to deselect all analog input channels.

Only set the ADANSB register while the ADCSR.ADST bit is 0.

34.2.7 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
	—	—	—	—	—	—	—	—	—	—	—	ANSB20	ANSB19	—	ANSB17	—
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
b1	ANSB17	A/D Conversion Channels Select	0: Associated input channel not selected 1: Associated input channel selected Bit [1] (ANSB17) corresponds to AN017.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4, b3	ANSB20, ANSB19	A/D Conversion Channels Select	0: Associated input channel not selected 1: Associated input channel selected Bit [4] (ANSB20) corresponds to AN020 and Bit [3] (ANSB19) corresponds to AN019.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ANSBn bits (n = 17, 19, and 20) (A/D Conversion Channels Select)

The ADANSB1.ANSBn bits select the analog input channels for A/D conversion from AN017, AN019, and AN020 in group B when group scan mode is selected. The ADANSB1 register is only used for group scan mode and not for any other modes.

Do not select channels specified in group A as selected with the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits, in double trigger mode.

The ANSB17 bit is associated with AN017, the ANSB19 bit is associated with AN019, and the ANSB20 bit is associated with AN020.

When performing A/D conversion on the temperature sensor output or internal reference voltage, set the ADANSB1 register to 0000h to deselect all analog input channels.

Only set the ADANSB1 register bits when the ADST bit is 0.

34.2.8 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
	—	—	—	—	—	ADS10	ADS09	—	—	ADS06	ADS05	ADS04	—	—	—	—
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
b6 to b4	ADS06 to ADS04	A/D-Converted Value Addition/ Average Channel Select	0: Associated input channel not selected 1: Associated input channel selected Bit [6] (ADS06) corresponds to AN006 and Bit [4] (ADS04) corresponds to AN004.	R/W
b8, b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10, b9	ADS10, ADS09	A/D-Converted Value Addition/ Average Channel Select	0: Associated input channel not selected 1: Associated input channel selected Bit [10] (ADS10) corresponds to AN010 and Bit [0] (ADS09) corresponds to AN009.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSn bits (n = 04 to 06, 09, and 10) (A/D-Converted Value Addition/Average Channel Select)

The ADSn bits determine which A/D-converted channel selected in the ANSAn bits (n = 04 to 06, 09, and 10) in ADANSA0, or the ADCSR.DBLANS[4:0] bits is subject to A/D-converted value addition or averaging.

When the ANSBn bit (n = 04 to 06, 09, and 10) in ADANSB0 is set to 1, A/D conversion of the analog input of the respective channel is performed successively 1 to 16 times, as specified in the ADC[2:0] bits in the ADADC register.

When the ADADC.AVEE bit is 0, the value obtained by addition (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 channels on which the A/D conversion is performed and for which addition/average mode is not selected, a normal 1-time conversion is executed and the conversion result is stored in the A/D Data Register.

Only set the ADADS0 register bits when the ADCSR.ADST bit is 0.

34.2.9 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
	—	—	—	—	—	—	—	—	—	—	—	ADS20	ADS19	—	ADS17	—
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
b1	ADS17	A/D-Converted Value Addition/Average Channel Select	0: Associated input channel not selected 1: Associated input channel selected Bit [1] (ADS17) corresponds to AN017.	
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b4, b3	ADS20, ADS19	A/D-Converted Value Addition/Average Channel Select	0: Associated input channel not selected 1: Associated input channel selected Bit [4] (ADS20) corresponds to AN020 and Bit [3] (ADS19) corresponds to AN019.	
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSn bits (n = 17, 19, and 20) (A/D-Converted Value Addition/Average Channel Select)

The ADSn bits determine which A/D-converted channels selected in the ANSAn bits (n = 17, 19, and 20) in ADANSA1, or ADCSR.DBLANS[4:0] bits are subject to A/D-converted value addition or averaging. When ANSBn bit (n = 17, 19, and 20) in ADANSB1 is set to 1, A/D conversion of the analog input of the respective channel is performed successively 1 to 16 times as determined by the setting of the ADC[2:0] bits in the ADADC register.

When the ADADC.AVEE bit is 0, the value obtained by addition is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition is stored in the A/D data register. For channels on which the A/D conversion is performed and for which addition/average mode is not selected, a normal 1-time conversion is executed and the conversion result is stored in the A/D Data Register.

Only set the ADADS1 register when the ADCSR.ADST bit is 0.

Figure 34.2 shows a scanning operation sequence in which both the ADADS0.ADS06 and ADADS1.ADS19 bits are set to 1. For this example:

- Addition mode is selected (ADADS.AVEE = 0)
- The number of conversions is set to 4 (ADADC.ADC[1:0] = 11b)
- Channels AN004 to AN006, AN009, AN010, AN017, AN019, and AN020 are selected (ADANSA0.ANSA0[15:0] = 0670h, and ADANSA1.ANSA1[15:0] = 001Ah) in continuous scan mode (ADCSR.ADCS[1:0] = 10b).

The conversion process begins with AN004. The AN006 conversion is performed successively 4 times, and the added value is returned to A/D Data Register 6 (ADDR6). Next, the AN009 conversion process is started. The AN019 conversion is performed successively 4 times and the added value is returned to A/D Data Register 19 (ADDR19). After conversion of AN020, the conversion operation repeats in the same sequence starting from AN004.

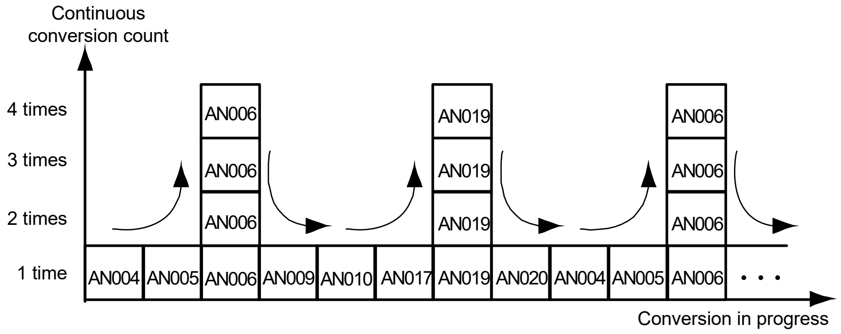


Figure 34.2 Scan conversion sequence with ADADC.ADC[2:0] = 011b, ADADS0.ADS06 = 1, and ADADS1.ADS19 = 1

34.2.10 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 (one addition) 0 1 0: 3-time conversion (two additions) 0 1 1: 4-time conversion (three additions) 1 0 1: 16-time conversion (15 additions). 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 deselected by setting the ADADC.AVEE bit to 0, set the addition count to 1, 2, 3, 4, or 16-time conversion. 16-time conversion can only be used with 12-bit accuracy.

Note 2. When average mode is selected by setting the ADADC.AVEE bit to 1, set the 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 for all channels for which A/D conversion and addition/average mode are selected, including the channels selected in double trigger mode with the ADCSR.DBLANS[4:0] bits. The count also applies to A/D conversion of the temperature sensor output and internal reference voltage.

The following restrictions apply to the setting of the ADC[2:0] bits:

- When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the count to 3-time conversion (ADADC.ADC[2:0] = 010b) or 16-time conversion (ADADC.ADC[2:0] = 101b) with a conversion accuracy setting of 14 bits (ADCER.ADPRC[1:0] = 11b)
- 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 is 14 bits (ADCER.ADPRC[1:0] = 11b), do not set the ADC[2:0] bits to 101b.

Only set the ADC[2:0] bits when the ADCSR.ADST bit is 0.

AVEE bit (Average Mode Enable)

The AVEE bit selects addition or average mode for the channels for which A/D conversion and A/D-converted value addition/average mode are selected, including the channel selected for double trigger mode in the ADCSR.DBLANS[4:0] bits, the temperature sensor output, and the 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).

Only set the AVEE bits while the ADCSR.ADST bit is 0.

34.2.11 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: 12-bit accuracy 0 1: Setting prohibited 1 0: Setting prohibited 1 1: 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: Automatic clearing disabled 1: Automatic clearing enabled.	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: Select 0 V 1 0: Select reference power supply voltage*[1] × 1/2 1 1: Select reference power supply voltage*[1].	R/W
b10	DIAGLD	Self-Diagnosis Mode Select	0: Select rotation mode for self-diagnosis voltage 1: Select fixed mode for self-diagnosis voltage.	R/W
b11	DIAGM	Self-Diagnosis Enable	0: ADC14 self-diagnosis disabled 1: ADC14 self-diagnosis enabled.	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 for the A/D data register format selected 1: Flush-left for the A/D data register format selected.	R/W

Note 1. Reference voltage refers to VREFH0.

ADPRC[1:0] bits (A/D Conversion Accuracy Specify)

The ADPRC[1:0] bits select the A/D conversion accuracy to 12-bits or 14-bits. Changing the A/D conversion accuracy also changes the bit width of valid data stored in the result register and the A/D conversion time.

For details, see section 34.3.6, Analog Input Sampling and Scan Conversion Time. Only set the ADPRC[1:0] bits 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 the ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, or ADOCDR register after any of these registers is read by the CPU, DTC, or DMAC.

DIAGVAL[1:0] bits (Self-Diagnosis Conversion Voltage Select)

The DIAGVAL[1:0] bits select the voltage value used in self-diagnosis fixed voltage mode. For details, see the ADCER.DIAGLD bit description.

Do not execute self-diagnosis 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 DIAGLD to 0 selects conversion of the voltages in rotation mode where 0 V, 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, self-diagnosis is executed from 0 V. The self-diagnosis voltage value does not return to 0 when scan conversion completes. When scan conversion is restarted, rotation starts at the voltage value following the previous value.

Setting DIAGLD to 1 selects fixed voltage, in which the fixed voltage specified by the ADCER.DIAGVAL[1:0] bits is converted. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value.

Only set the DIAGLD bit 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 ADC14. In self-diagnosis mode, one of the internally generated voltage values (0 V, the reference power supply × 1/2, or the reference power supply) is converted. When conversion completes, information on the converted voltage and the conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). ADRD can be read by software to determine whether the conversion result falls within the normal or abnormal range.

Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. In double trigger mode (ADCSR.DBLE = 1), self-diagnosis (DIAGM = 0) is deselected. When self-diagnosis is selected in group scan mode, self-diagnosis is executed separately for group A and B.

Only set the DIAGM bit when the ADCSR.ADST bit is 0.

ADRFMT bit (A/D Data Register Format Select)

The ADRFMT bit specifies flush-right or flush-left for the data to be stored in ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR, ADCMPDR0/1, ADWINLLB, ADWINULB, or ADRD.

Only set the ADRFMT bit when the ADCSR.ADST bit is 0.

34.2.12 A/D Conversion Start Trigger Select Register (ADSTRGR)

Address(es): ADC140.ADSTRGR 4005 C010h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	TRSA[5:0]						—	—	TRSB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	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 only used in group scan mode and not in any other scan mode. Software trigger or asynchronous trigger cannot be used as the scan conversion start trigger for group B. In group scan mode, set the TRSB[5:0] bits to a value other than 000000b and set the ADCSR.TRGE bit to 1.

When group A is given priority in group scan mode, setting the ADGSPCR.GBRP bit to 1 allows group B to continuously operate in single scan mode. When setting the ADGSPCR.GBRP bit to 1, set the TRSB[5:0] bits to 3Fh. The issuance period for a conversion trigger must be more than or equal to the actual scan conversion time (tSCAN). If the issuance period is less than tSCAN, A/D conversion by the trigger might have no effect.

When the GPT module is selected as an A/D conversion start trigger, a delay for synchronization processing occurs. For details, see section 34.3.6, Analog Input Sampling and Scan Conversion Time.

Table 34.6 lists the A/D conversion startup sources selected in the TRSB[5:0] bits.

Table 34.6 Selection of A/D conversion sources in the TRSB[5:0] bits

Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
Trigger source deselection 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

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 synchronous trigger (ELC) as the A/D conversion start source, set the TRGE bit in the ADCSR register to 1 and set the EXTRG bit in the ADCSR register to 0.

The software trigger (ADCSR.ADST) is enabled regardless of the setting in the ADCSR.TRGE bit, the ADCSR.EXTRG bit, or the TRSA[5:0] bits. The issuance period for a conversion trigger must be more than or equal to the actual scan conversion time (tSCAN). If the issuance period is less than tSCAN, A/D conversion by the trigger might have no effect. For details, see section 34.3.6, Analog Input Sampling and Scan Conversion Time.

Table 34.7 lists the A/D conversion start sources selected in the TRSA[5:0] bits.

Table 34.7 Selection of A/D activation sources in the TRSA[5:0] bits

Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
Trigger source deselection 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

34.2.13 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

Bit	Symbol	Bit name	Description	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 for the number of times specified in the ADC[2:0] bits in the ADADC register. The maximum addition count depends on the conversion accuracy, as seen in section 34.2.1. 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.

Only set the TSSAD bit while the ADCSR.ADST bit is 0.

OCSAD bit (Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select)

When the OCSAD bit is set to 1, A/D conversion of the internal reference voltage is selected and performed successively for the number of times specified by the ADC[2:0] bits in the ADADC register. The maximum addition count depends on the conversion accuracy as seen in section 34.2.1. When the ADADC.AVEE bit is 0, the value obtained by addition is returned to the A/D Internal Reference Voltage Data Register (ADOCDR). When the ADADC.AVEE bit is 1, the mean value is returned to ADOCDR.

Only set the OCSAD bit while the ADCSR.ADST bit is 0.

TSSA bit (Temperature Sensor Output A/D Conversion Select)

The TSSA bit selects A/D conversion of the temperature sensor output.

When executing the A/D conversion:

1. Set all the bits in the ADANSA0/1 and ADANSB0/1 registers, the ADCSR.DBLE bit, and ADEXICR.OCSA bit to 0.
2. Execute A/D conversion in single scan mode.

When executing the A/D conversion of the temperature sensor output, the ADDISCR register is set to 0Fh and the ADC14 executes discharge (15 ADCLK) before executing sampling. The minimum sampling time is 5 µs. The ADC14 executes discharge each time it executes A/D conversion of the temperature sensor output.

Only set the TSSA bit when the ADCSR.ADST bit is 0.

OCSA bit (Internal Reference Voltage A/D Conversion Select)

The OCSA bit selects A/D conversion of the internal reference voltage.

When executing the A/D conversion:

1. Set all the bits in the ADANSA0/1 and ADANSB0/1 registers, the ADCSR.DBLE bit, and ADEXICR.TSSA bit to 0.
2. Execute A/D conversion in single scan mode.

When executing the A/D conversion of the internal reference voltage, the ADDISCR register is set to 0Fh and the ADC14 executes discharge (15 ADCLK) before executing sampling.The minimum sampling time is 5 µs. The ADC14 executes discharge each time it executes A/D conversion of the internal reference voltage.

Only set the OCSA bit while the ADCSR.ADST bit is 0.

34.2.14 A/D Sampling State Register n (ADSSTRn) (n = 00, 04 to 06, 09, 10, L, T, O)

Address(es): ADC140.ADSSTR00 4005 C0E0h, ADC140.ADSSTR04 4005 C0E4h to ADC140.ADSSTR06 4005 C0E6h,
ADC140.ADSSTR09 4005 C0E9h, ADC140.ADSSTR10 4005 C0EAh, 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 1 ADCLK (A/D conversion clock) cycle and the ADCLK clock is 64 MHz, then one state is 15.625 ns. The initial value is 13 states.

The sampling time can be adjusted if the impedance of the analog input signal source is too high to secure sufficient sampling time, or if the ADCLK clock is slow.

Only set the SST[7:0] bits while the ADCSR.ADST bit is 0.

The lower limit of the sampling time setting depends on the frequency ratio:

- If the 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 of more than 5 states
- If the frequency ratio of PCLKB to PCLKC (ADCLK) = 1:2 or 1:4, the sampling time must be set to a value of more than 6 states.

Table 34.8 shows the relationship between the A/D Sampling State Register and the associated channels. For details, see section 34.3.6, Analog Input Sampling and Scan Conversion Time.

Table 34.8 Relationship between A/D Sampling State Register n and associated channels

Bit name	Associated channels
ADSSTR00.SST[7:0] bits	self-diagnosis function
ADSSTR04.SST[7:0] bits	AN004
ADSSTR05.SST[7:0] bits	AN005
ADSSTR06.SST[7:0] bits	AN006
ADSSTR09.SST[7:0] bits	AN009
ADSSTR10.SST[7:0] bits	AN010
ADSSTRL.SST[7:0] bits	AN017, AN019, AN020
ADSSTRT.SST[7:0] bits	Temperature sensor output*1
ADSSTRO.SST[7:0] bits	Internal reference voltage*1

Note 1. When the temperature sensor output or the internal reference voltage is converted, set the sampling time to more than 5 μs. Because the maximum SST[7:0] value is 255 states, the ADCLK frequency must be such that the resulting sampling time is at least 5 μs.

34.2.15 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 b0 0 0 0 0: The disconnection detection assist function is disabled 0 0 0 1: Setting prohibited Others: The number of states for the discharge or precharge period.	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 precharge or discharge for the A/D disconnection detection assist function. Only set the ADDISCR register when the ADCSR.ADST bit is 0.

When the temperature sensor output or internal reference voltage is converted, the A/D converter executes discharge automatically. This operation is achieved by setting the ADDISCR register to 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.

Disable the disconnection detection assist function if any of the following functions are used:

- Temperature sensor
- 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 detection assistance function is enabled.

PCHG bit (Precharge/discharge select)

The PCHG bit selects either precharge or discharge.

34.2.16 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: Operate without group A priority control 1: Operate with group A priority control.	R/W

Bit	Symbol	Bit name	Description	R/W
b1	GBRSCN	Group B Restart Setting	Enabled only when PGS = 1. Reserved when PGS = 0. 0: Do not restart group B scanning after it is stopped by group A priority control 1: Restart group B scanning after it is stopped by group A priority control.	R/W
b14 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
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 with priority control.	R/W

Note 1. The ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode) before setting PGS to 1. If these 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 GBRSCN bit.

PGS bit (Group A Priority Control Setting)

Set the PGS bit to 1 to give priority to operation on group A. The ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode) before setting the PGS bit to 1. If these bits are set to any other values, proper operation is not guaranteed.

When the PGS bit is set to 0, software must perform a clear operation as described in section 34.8.2, Notes on Stopping A/D Conversion. When the PGS bit is set to 1, use the settings as described in section 34.3.4.3, Operation with group A priority control.

GBRSCN bit (Group B Restart Setting)

The GBRSCN 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 group A conversion. 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 group A conversion.

When the GBRSCN bit is set to 0, triggers input during A/D conversion are ignored. Only set the GBRSCN bit while the ADCSR.ADST bit is 0.

The setting of the GBRSCN bit is valid when the PGS bit is 1.

GBRP bit (Group B Single Scan Continuous Start)

Set the GBRP bit to perform a single scan operation 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 starts automatically. If a group B conversion stops because of an operation on group A, the group A operation takes priority, and single scan on group B restarts automatically on completion of group A conversion.

Disable group B trigger input before setting the GBRP bit to 1. Setting the GBRP bit to 1 invalidates the setting in the GBRSCN bit. Only set the GBRP bit while the ADCSR.ADST is 0.

The setting in the GBRP bit is valid when the PGS bit is 1.

34.2.17 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: Output ADC140_WCMPM when window A OR window B comparison conditions are met. Otherwise, output ADC140_WCMPUM. 0 1: Output ADC140_WCMPM when window A EXOR window B comparison conditions are met. Otherwise, output ADC140_WCMPUM. 1 0: Output ADC140_WCMPM when window A AND window B comparison conditions are met. Otherwise, output ADC140_WCMPUM. 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 disabled. ADC140_WCMPM and ADC140_WCMPUM outputs are disabled. 1: Compare window B operation 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 disabled. ADC140_WCMPM and ADC140_WCMPUM outputs are disabled. 1: Compare window A operation 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 disabled when comparison conditions (window B) are met 1: ADC140_CMPBI interrupt enabled when comparison conditions (window B) are met.	R/W
b14	WCMPE	Window Function Setting	0: Window function 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 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)

The CMPAB[1:0] bits are valid when both window A and window B are enabled (CMPAE = 1 and CMPBE = 1) in single scan mode. These bits specify the compare function match/mismatch event output conditions and monitoring conditions of ADWINMON.MONCONB.

Only set the CMPAB[1:0] bits when the ADCSR.ADST bit is 0.

CMPBE bit (Compare Window B Operation Enable)

The CMPBE bit enables or disables the compare window B operation. Only set the CMPBE bit when the ADCSR.ADST bit is 0.

Set this bit to 0 before setting the following registers and bits:

- A/D Channel Select Registers A0/A1/B0/B1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1)
- OCSA or TSSA bit in the A/D Conversion Extended Input Control Register (ADEXICR)
- CMPCHB[5:0] bits in the Window B Channel Select Register (ADCMPBNSR).

CMPAE bit (Compare Window A Operation Enable)

The CMPAE bit enables or disables the compare window A operation. Only set the CMPAE bit when the ADCSR.ADST bit is 0.

Set this bit to 0 before setting the following registers and bits:

- A/D Channel Select Registers A0/A1/B0/B1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1)
- OCSA or TSSA bit in the A/D Conversion Extended Input Control Register (ADEXICR)
- Window A Channel Select Registers 0/1 (ADCMPANSR0, ADCMPANSR1)
- Window A Extended Input Select Register (ADCMPANSER).

CMPBIE bit (Compare B Interrupt Enable)

The CMPBIE bit enables or disables the ADC140_CMPBI interrupt output when the comparison conditions (window B) are met.

WCMPE bit (Window Function Setting)

The WCMPE bit enables or disables the window function. Only set the WCMPE bit when the ADCSR.ADST bit is 0.

CMPAIE bit (Compare A Interrupt Enable)

The CMPAIE bit enables or disables the ADC140_CMPAI interrupt output when the comparison conditions (window A) are met.

34.2.18 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
	—	—	—	—	—	CMPCHA10	CMPCHA09	—	—	CMPCHA06	CMPCHA05	CMPCHA04	—	—	—	—
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
b6 to b4	CMPCHA06 to CMPCHA04	Compare Window A Channel Select	0: Compare function disabled for the associated input channel 1: Compare function enabled for the associated input channel. Bit [6] (CMPCHA06) corresponds to AN006 and Bit [4] (CMPCHA04) corresponds to AN004.	R/W
b8, b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10, b9	CMPCHA10, CMPCHA09	Compare Window A Channel Select	0: Compare function disabled for the associated input channel 1: Compare function enabled for the associated input channel. Bit [10] (CMPCHA10) corresponds to AN010 and Bit [9] (CMPCHA09) corresponds to AN009.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPCHAn bits (n = 04 to 06, 09, and 10) (Compare Window A Channel Select)

The compare function is enabled by writing 1 to the CMPCHAn bit with the same number as the A/D conversion channel selected in the ADANSA0.ANSAn bits (n = 04 to 06, 09, and 10) and the ADANSB0.ANSBn bits (n = 04 to 06, 09, and 10).

Only set the CMPCHAn bits when the ADCSR.ADST bit is 0.

34.2.19 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
	—	—	—	—	—	—	—	—	—	—	—	CMPCHA20	CMPCHA19	—	CMPCHA17	—
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
b1	CMPCHA17	Compare Window A Channel Select	0: Compare function disabled for the associated input channel 1: Compare function enabled for the associated input channel Bit [1] (CMPCHA17) corresponds to AN017.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4, b3	CMPCHA20, CMPCHA19	Compare Window A Channel Select	0: Compare function disabled for the associated input channel 1: Compare function enabled for the associated input channel Bit [4] (CMPCHA20) corresponds to AN020 and Bit [3] (CMPCHA19) corresponds to AN019.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPCHAn bits (n = 17, 19, and 20) (Compare Window A Channel Select)

The compare function is enabled by writing 1 to the CMPCHAn bit with the same number as the A/D conversion channel selected in the ADANSA1.ANSAn bits (n = 17, 19, and 20) and the ADANSB1.ANSBn bits (n = 17, 19, and 20).

Only set the CMPCHAn bits while the ADCSR.ADST bit is 0.

34.2.20 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: Exclude the temperature sensor output from the compare window A target range 1: Include the temperature sensor output in the compare window A target range.	R/W
b1	CMPOCA	Internal Reference Voltage Compare Select	0: Exclude the internal reference voltage from the compare window A target range 1: Include the internal reference voltage in the compare window A target range.	R/W
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 when the ADEXICR.TSSA bit is 1.

Only set the CMPTSA bit when the ADCSR.ADST bit is 0.

CMPOCA bit (Internal Reference Voltage Compare Select)

The compare window A function is enabled by setting the CMPOCA bit to 1 when the ADEXICR.OCSA bit is 1.

Only set the CMPOCA bit when the ADCSR.ADST bit is 0.

34.2.21 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
	—	—	—	—	—	CMPLCHA10	CMPLCHA09	—	—	CMPLCHA06	CMPLCHA05	CMPLCHA04	—	—	—	—
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
b6 to b4	CMPLCHA06 to CMPLCHA04	Compare Window A Comparison Condition Select	These bits set comparison conditions for channels AN004 to AN006 to which window A comparison conditions are applied. Comparison conditions are shown in Figure 34.3. • When the window function is disabled (ADCMPCR.WCMPE = 0): 0: ADCMPDR0 value > A/D-converted value 1: ADCMPDR0 value < A/D-converted value. • When the window function is enabled (ADCMPCR.WCMPE = 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
b8, b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10, b9	CMPLCHA10, CMPLCHA09	Compare Window A Comparison Condition Select	These bits set comparison conditions for channels AN009, and AN010 to which window A comparison conditions are applied. Comparison conditions are shown in Figure 34.3. • When the window function is disabled (ADCMPCR.WCMPE = 0): 0: ADCMPDR0 value > A/D-converted value 1: ADCMPDR0 value < A/D-converted value. • When the window function is enabled (ADCMPCR.WCMPE = 1): 0: A/D-converted value < ACMPDR0 value, or ADCMPDR1 value < A/D-converted value 1: ADCMPDR0 value < A/D-converted value < ADCMPDR1 value.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPLCHAn bits (n = 04 to 06, 09, and 10) (Compare Window A Comparison Condition Select)

The CMPLCHAn bits specify the comparison conditions for channels AN004 to AN006, AN009, and AN010 to which window A comparison conditions are applied. These bits can be set for each analog input to be compared. CMPLCHA04, CMPLCHA06, CMPLCHA10 correspond to AN004, AN006, and AN010, 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.

Comparison conditions when the window function is disabled

CMPLCHAn = 0

ADCMPDR0 value ≤ A/D converted value	Not met
ADCMPDR0 value > A/D converted value	Met

CMPLCHAn = 1

ADCMPDR0 value < A/D converted value	Met
ADCMPDR0 value ≥ A/D converted value	Not met

Comparison conditions when the window function is enabled

CMPLCHAn = 0

ADCMPDR1 value < A/D converted value	Met
ADCMPDR0 value ≤ A/D converted value ≤ ADCMPDR1 value	Not met
A/D converted value < ADCMPDR0 value	Met

CMPLCHAn = 1

ADCMPDR1 value ≤ A/D converted value	Not met
ADCMPDR0 value < A/D converted value < ADCMPDR1 value	Met
A/D converted value ≤ ADCMPDR0 value	Not met

Figure 34.3 Comparison conditions for compare function window A

34.2.22 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
	—	—	—	—	—	—	—	—	—	—	—	CMPLCHA20	CMPLCHA19	—	CMPLCHA17	—
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

Bit	Symbol	Bit name	Description	R/W
b1	CMPLCHA17	Compare Window A Comparison Condition Select	This bit sets comparison conditions for channel AN017 to which window A comparison conditions are applied. Comparison conditions are shown in Figure 34.3. • When the window function is disabled (ADCMPCR.WCMPE = 0): 0: ADCMPDR0 value > A/D-converted value 1: ADCMPDR0 value < A/D-converted value. • When the window function is enabled (ADCMPCR.WCMPE = 1): 0: A/D-converted value < ACMPDR0 value, or ADCMPDR1 value < A/D-converted value 1: ADCMPDR0 value < A/D-converted value < ADCMPDR1 value.	
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4, b3	CMPLCHA20, CMPLCHA19	Compare Window A Comparison Condition Select	These bits set comparison conditions for channels AN019, and AN020 to which window A comparison conditions are applied. Comparison conditions are shown in Figure 34.3. • When the window function is disabled (ADCMPCR.WCMPE = 0): 0: ADCMPDR0 value > A/D-converted value 1: ADCMPDR0 value < A/D-converted value. • When the window function is enabled (ADCMPCR.WCMPE = 1): 0: A/D-converted value < ACMPDR0 value, or ADCMPDR1 value < A/D-converted value 1: ADCMPDR0 value < A/D-converted value < ADCMPDR1 value.	
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPLCHAn bits (n = 17, 19, and 20) (Compare Window A Comparison Condition Select)

The CMPLCHAn bits specify the comparison conditions for channels AN017, AN019, and AN020 to which window A comparison conditions are applied. These bits can be set for each analog input to be compared. CMPLCHA17, CMPLCHA19, and CMPLCHA20 correspond to AN017, AN019, and AN020, 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.

34.2.23 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 34.3. • When the window function is disabled (ADCMPCR.WCMPE = 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 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 34.3. • 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 A/D-converted value > ADCMPDR1 value 1: ADCMPDR0 value < A/D-converted value < ADCMPDR1 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)

The CMPLTSA bit specifies 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 flag is set to 1 and a compare interrupt (ADC140_CMPAI) is generated.

CMPLOCA bit (Compare Window A Internal Reference Voltage Comparison Condition Select)

The CMPLOCA bit specifies comparison conditions when the internal reference voltage is the target of the window A comparison condition. When the internal reference voltage comparison result meets the set condition, the ADCMPSER.CMPSTOCA flag is set to 1 and a compare interrupt (ADC140_CMPAI) is generated.

34.2.24 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 specifies the reference data when the compare window A function is used. ADCMPDR0 sets the lower reference for window A, and ADCMPDR1 sets the upper reference for window A.

The ADWINULB and ADWINLLB registers specify the reference data when the compare window B function is used. ADWINLLB sets the lower reference for window B, and ADWINULB sets the upper reference for window B.

ADCMPDRy, ADWINULB, and ADWINLLB are read/write registers.

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 reference is not less than the lower reference (ADCMPDR1 ≥ ADCMPDR0, ADWINULB ≥ ADWINLLB). ADCMPDR1 and ADWINULB are not used when the window function is disabled.

Note 1. The lower and the upper reference values are changed when each register is written. For example, when the upper reference value and the lower reference value are changed, the MCU compares the upper reference (after rewrite), and the lower reference (before rewrite) with the A/D conversion result. See Figure 34.4. If the comparison during the rewriting of these two references is erroneous, then rewrite these reference values when both ADCSR.ADST and the target Compare Window Operation Enable bit (ADCMPCR.CMPAE or ADCMPCR.CMPBE) is 0.

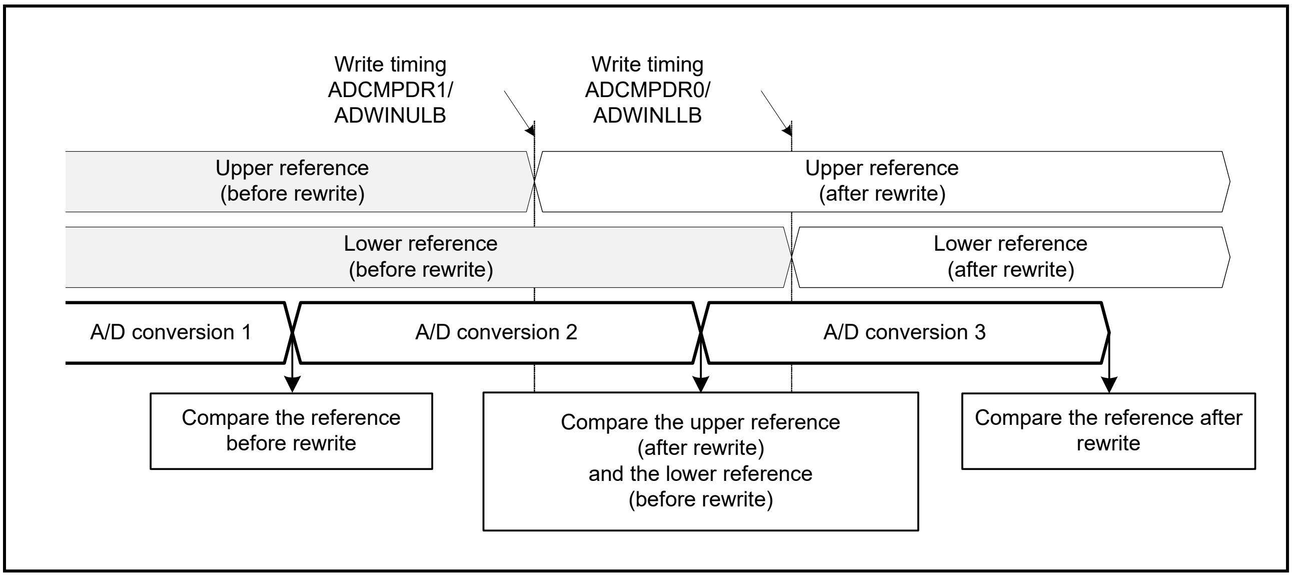

Write timing
ADCMPDR1/
ADWINULB
Write timing
ADCMPDR0/
ADWINLLB
Upper reference (before rewrite)
Upper reference (after rewrite)
Lower reference (before rewrite)
Lower reference (after rewrite)
A/D conversion 1
A/D conversion 2
A/D conversion 3
Compare the reference before rewrite
Compare the upper reference (after rewrite) and the lower reference (before rewrite)
Compare the reference after rewrite

Figure 34.4 Comparison between upper reference and lower 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 Specify bit (14-bit or 12-bit)
- The value of the A/D-Converted Value Addition/Average Channel Select bits (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 [13:0] are valid
- Flush-right data with 12-bit accuracy: Lower 12 bits [11:0] are valid
- Flush-left data with 14-bit accuracy: Upper 14 bits [15:2] are valid
- Flush-left data with 12-bit accuracy: Upper 12 bit [15:4] are valid.

(2) When A/D-converted value addition mode is selected

- Flush-right data with 14-bit accuracy: All bits [15:0] are valid
- Flush-right data with 12-bit accuracy: Lower 14 bits [13:0] are valid
- Flush-left data with 14-bit accuracy: All bits [15:0] are valid
- Flush-left data with 12-bit accuracy: Upper 14 bits [15:2] are valid.

34.2.25 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
	—	—	—	—	—	CMPST CHA10	CMPST CHA09	—	—	CMPST CHA06	CMPST CHA05	CMPST CHA04	—	—	—	—
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
b6 to b4	CMPSTCHA06 to CMPSTCHA04	Compare Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1b), these bits indicate the comparison result for channels AN004 to AN006, to which window A comparison conditions are applied: 0: Comparison conditions not met 1: Comparison conditions met.	R/W
b8, b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10, b9	CMPSTCHA10, CMPSTCHA09	Compare Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1b), these bits indicate the comparison result for channels AN009 and AN010, to which window A comparison conditions are applied: 0: Comparison conditions not met 1: Comparison conditions met.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPSTCHAn bits (n = 04 to 06, 09, and 10) (Compare Window A Flag)

The CMPSTCHAn bits are comparison result status flags for channels AN004 to AN006, AN009, and AN010 to which window A comparison conditions are applied. When the comparison condition set in 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 bit is set to 1. CMPSTCHA04, CMPSTCHA06, and CMPSTCHA10 correspond to AN004, AN006, and AN010, respectively.

Writing 1 to the CMPSTCHAn bits is invalid.

[Setting condition]

- The condition set in ADCMPLR0.CMPLCHAn is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

34.2.26 A/D Compare Function Window A Channel Status Register 1 (ADCMPSR1)

Address(es): ADC140.ADCMPSR1 4005 C0A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	CMPST CHA20	CMPST CHA19	—	CMPST CHA17	—
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
b1	CMPSTCHA17	Compare Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1b), this bit indicates the comparison result for channels AN017, to which window A comparison conditions are applied: 0: Comparison conditions not met 1: Comparison conditions met.	R/W

Bit	Symbol	Bit name	Description	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4, b3	CMPSTCHA20, CMPSTCHA19	Compare Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1b), these bits indicate the comparison result for channels AN019 and AN020, to which window A comparison conditions are applied: 0: Comparison conditions not met 1: Comparison conditions met.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPSTCHAn bits (n = 17, 19, and 20) (Compare Window A Flag)

The CMPSTCHAn bits are comparison result status flags for channels AN017, AN019, and AN020, to which window A comparison conditions are applied. When the comparison condition set in ADCMPLR1.CMPLCHAn is met at the end of A/D conversion, the associated CMPSTCHAn bit is set to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt (ADC140_CMPAI) request is generated when this bit is set to 1. CMPSTCHA17, CMPSTCHA19, and CMPSTCHA20 correspond to AN017, AN019, and AN020, respectively.

Writing 1 to the CMPSTCHAn bits is invalid.

[Setting condition]

- The condition set in ADCMPLR1.CMPLCHAn is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

34.2.27 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
	—	—	—	—	—	—	CMPST OCA	CMPST TSA
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 not met 1: Comparison conditions 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 not met 1: Comparison conditions 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)

The CMPSTTSA bit is a status flag that indicates the temperature sensor output comparison result. When the comparison condition set in ADCMPLER.CMPLTSA is met at the end of A/D conversion, this bit is set to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt (ADC140_CMPAI) request is generated when this bit is set to 1.

Writing 1 to the CMPSTTSA bit is invalid.

[Setting condition]

- The condition set in ADCMPLER.CMPLTSA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

CMPSTOCA bit (Compare Window A Internal Reference Voltage Compare Flag)

The CMPSTOCA bit is a status flag that indicates the internal reference voltage comparison result. When the comparison condition set in ADCMPLER.CMPLOCA is met at the end of A/D conversion, this bit is set to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt (ADC140_CMPAI) request is generated when this bit is set to 1.

Writing 1 to the CMPSTOCA bit is invalid.

[Setting condition]

- The condition set in ADCMPLER.CMPLOCA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

34.2.28 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 1 0 0: AN004 0 0 0 1 0 1: AN005 0 0 0 1 1 0: AN006 0 0 1 0 0 1: AN009 0 0 1 0 1 0: AN010 0 1 0 0 0 1: AN017 0 1 0 0 1 1: AN019 0 1 0 1 0 0: AN020 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 for channels for window B. The comparison conditions are shown in Figure 34.5. • When the window function is disabled (ADCMPCR.WCMPE = 0): 0: ADWINLLB value > A/D-converted value 1: ADWINLLB value < A/D-converted value. • When the window function is enabled (ADCMPCR.WCMPE = 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)

The CMPCHB[5:0] bits select channels to be compared with the compare window B conditions AN004 to AN006, AN009, AN010, AN017, AN019, and AN020, 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 in the following bits:

- ADANSA0.ANSAn bits (n = 04 to 06, 09, and 10)

- ADANSA1.ANSAn bits (n = 17, 19, and 20)
- ADANSB0.ANSBn bits (n = 04 to 06, 09, and 10)
- ADANSB1.ANSBn bits (n = 17, 19, and 20).

Set CMPCHB[5:0] bits while the ADCSR.ADST bit is 0.

CMPLB bit (Compare Window B Comparison Condition Setting)

The CMPLB bit selects the comparison conditions for 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.

Compare conditions when the window function is disabled

CMPLB = 0

ADWINLLB value ≤ A/D converted value	Not met
ADWINLLB value > A/D converted value	Met

CMPLB = 1

ADWINLLB value < A/D converted value	Met
ADWINLLB value ≥ A/D converted value	Not met

Compare conditions when the window function is enabled

CMPLB = 0

A/D converted value > ADWINULB value	Met
ADWINLLB value ≤ A/D converted value ≤ ADWINULB value	Not met
A/D converted value < ADWINLLB value	Met

CMPLB = 1

A/D converted value ≥ ADWINULB value	Not met
ADWINLLB value < A/D converted value < ADWINULB value	Met
A/D converted value ≤ ADWINLLB value	Not met

Figure 34.5 Compare conditions for compare function window B

34.2.29 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 for channels AN004 to AN006, AN009, AN010, AN017, AN019, and AN020, temperature sensor output, and internal reference voltage, to which window B comparison conditions are applied: 0: Comparison conditions not met 1: Comparison conditions 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)

The CMPSTB bit is a status flag that indicates the comparison result for channels AN004 to AN006, AN009, AN010, AN017, AN019, and AN020, the temperature sensor, and the internal reference voltage, to which window B comparison conditions are applied. When the comparison condition set in 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 bit is set to 1.

Writing 1 to the CMPSTB bit is invalid.

[Setting condition]

- The condition set in ADCMPBNSR.CMPLB is met when ADCMPCR.CMPBE = 1.

[Clearing condition]

- Writing 0 after reading 1.

34.2.30 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. This bit is valid when both window A and window B operations are enabled: 0: Window A/window B composite conditions not met 1: Window A/window B composite conditions met.	R
b3 to b1	—	Reserved	These bits are read as 0.	R
b4	MONCMPA	Comparison Result Monitor A	0: Window A comparison conditions not met 1: Window A comparison conditions met.	R
b5	MONCMPB	Comparison Result Monitor B	0: Window B comparison conditions not met 1: Window B comparison conditions met.	R
b7, b6	—	Reserved	These bits are read as 0.	R

MONCOMB bit (Combination Result Monitor)

The read-only MONCOMB bit indicates the combined result of comparison condition results A and B with the combination condition set in the ADCMPCR.CMPAB[1:0] bits.

[Setting condition]

- The combined result meets the combination condition set in the ADCMPCR.CMPAB[1:0] bits when ADCMPCR.CMPAE = 1 and ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The combined result does not meet the combination condition set in the ADCMPCR.CMPAB[1:0] bits
- ADCMPCR.CMPAE = 0 or ADCMPCR.CMPBE = 0.

MONCMPA bit (Comparison Result Monitor A)

The read-only MONCMPA bit is read as 1 when the A/D-converted value of the window A target channel meets the condition set in ADCMPLR0/ADCMPLR1 and ADCMPLER. Otherwise, it is read as 0.

[Setting condition]

- The A/D-converted value meets the condition set in ADCMPLR0.CMPLCHAn when ADCMPCR.CMPAE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set in 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)

The read-only MONCMPB bit is read as 1 when the A/D-converted value of the window B target channel meets the condition set in the ADCMPBNSR.CMPLB bit. Otherwise, it is read as 0.

[Setting condition]

- The A/D-converted value meets the condition set in ADCMPBNSR.CMPLB when ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set in ADCMPBNSR.CMPLB when ADCMPCR.CMPBE = 1
- ADCMPCR.CMPBE = 0 (automatically cleared when the ADCMPCR.CMPBE value changes from 1 to 0).

34.2.31 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

Bit	Symbol	Bit name	Description	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)

The HVSEL[1:0] bits specify the high-potential reference voltage as AVCC0, VREFH0, or the internal reference voltage (1.45 V).

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 completes, set HVSEL[1:0] = 10b and start the A/D conversion.

When the internal reference voltage is selected as the high-potential reference voltage (HVSEL[1:0] = 10b), A/D conversion is possible for channels AN004 to AN006, AN009, AN010, AN017, AN019, and AN020, but A/D conversion of the internal reference voltage and the temperature sensor output is prohibited.

LVSEL bit (Low-Potential Reference Voltage Select)

The LVSEL bit specifies the low-potential reference voltage as AVSS0 or VREFL0.

ADSLP bit (Sleep)

The ADSLP bit transitions the ADC14 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 it to 0. Also, 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 34.8.8, ADHSC Bit Rewriting Procedure.

34.3 Operation

34.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.

Scan conversion is performed in any of the three operating modes and two conversion modes.

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 software sets the ADCSR.ADST bit to 0 from 1. In group scan mode, the selected channels of group A and the selected channels of group B are scanned once after scan starts in response to the respective synchronous trigger (ELC).

In single scan mode and continuous scan mode, A/D conversion is performed on the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n. In group scan mode, A/D conversion is performed on the ANn channels in group A selected in the ADANSA0 and ADANSA1 registers first, and on the ANn channels in group B selected in the ADANSB0 and ADANSB1 registers, respectively, starting from the channel with the smallest number n.

When self-diagnosis is selected, it is executed once at the beginning of each scan, and one of the three voltages internally generated in the ADC14 is converted.

Simultaneous selection of both temperature sensor output and internal reference voltage is prohibited. 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 can 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 in the ADCSR.DBLANS[4:0] bits is duplicated only if the conversion is started by the synchronous trigger (ELC) selected in the ADSTRGR.TRSA[5:0] bits. In group scan mode, only group A can use the double trigger mode.

In the extended operation of double trigger mode, the A/D conversion operation is generated from the synchronous trigger combination selected in the ADSTRGR.TRSA[5:0] bits. In addition to normal double trigger mode operation, A/D conversion data with odd number triggers (ELC_AD00) is stored into the A/D Data Duplexing Register A (ADDBLDRA), and A/D conversion data with even number triggers (ELC_AD01) is stored into the A/D Data Duplexing Register B (ADDBLDRB). In the extended operation of double trigger mode, when a combination of triggers occurs at the same time, data duplexing register settings for the specified triggers do not work, and A/D conversion data is stored into the A/D Data Duplexing Register B (ADDBLDRB). The ADC14 ignores a synchronous trigger that occurs during the A/D conversion started by another synchronous trigger.

34.3.2 Single scan Mode

34.3.2.1 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 ADCSR.ADST bit is set to 1 (starting A/D conversion) by a software trigger, a synchronous trigger input (ELC), or an asynchronous trigger input, A/D conversion is performed on the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
2. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D data register (ADDRy).
3. When A/D conversion of all the selected channels completes, an ADC140_ADI interrupt request is generated.
4. The ADST bit remains 1 (starting A/D conversion) during A/D conversion, and is automatically set to 0 when A/D conversion of all the selected channels completes. The ADC14 then enters a wait state.

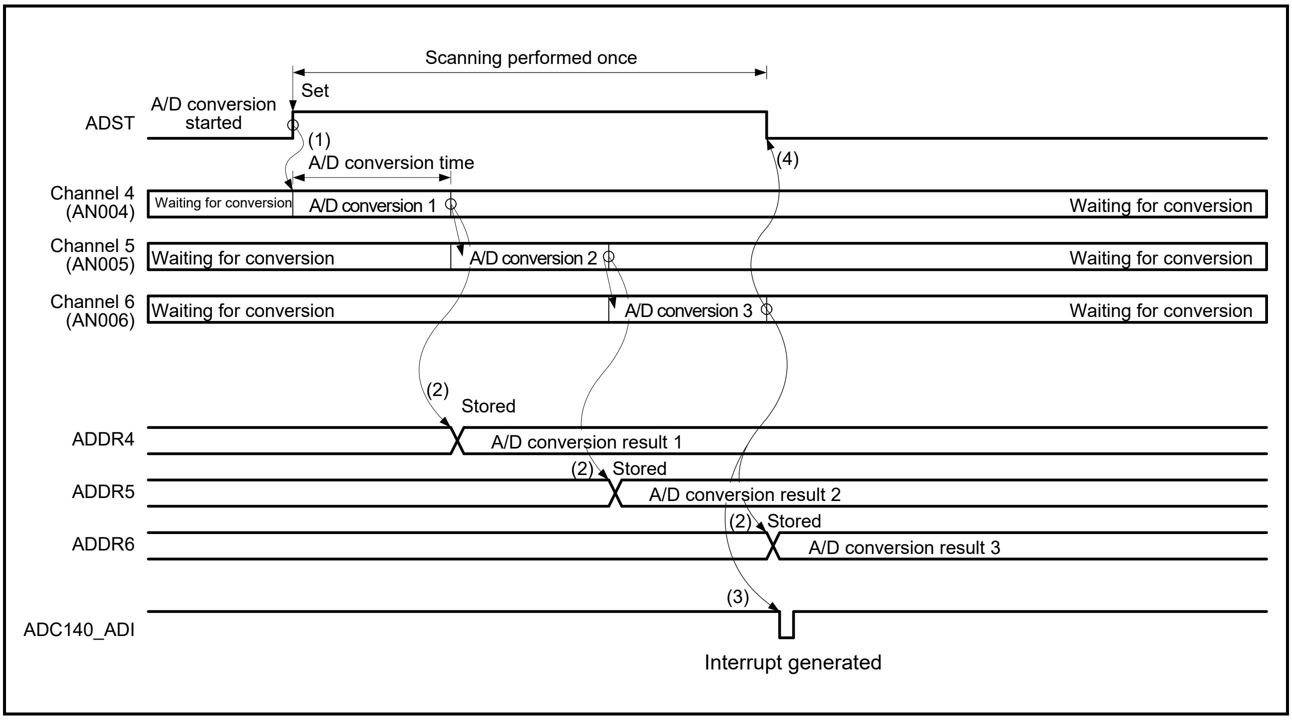

Scanning performed once
Set
A/D conversion started
ADST
(1)
A/D conversion time
(4)
Channel 4 (AN004)
Waiting for conversion
A/D conversion 1
Waiting for conversion
Channel 5 (AN005)
Waiting for conversion
A/D conversion 2
Waiting for conversion
Channel 6 (AN006)
Waiting for conversion
A/D conversion 3
Waiting for conversion
(2)
Stored
ADDR4
A/D conversion result 1
(2)
Stored
ADDR5
A/D conversion result 2
(2)
Stored
ADDR6
A/D conversion result 3
(3)
ADC140_ADI
Interrupt generated

Figure 34.6 Example of basic operation in single scan mode when AN004 to AN006 are selected

34.3.2.2 Channel selection and self-diagnosis

When channels and self-diagnosis are selected, A/D conversion is first performed on the reference voltage VREFH0 (×0, ×1/2, or ×1) supplied to ADC14. A/D conversion is then performed once on the analog input of the selected channels as follows:

1. A/D conversion for self-diagnosis is first started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (ELC), or an asynchronous trigger input.
2. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed on the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register (ADDRy).
4. When A/D conversion of all the selected channels completes, an ADC140_ADI interrupt request is generated.
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 completes. The ADC14 then enters a wait state.

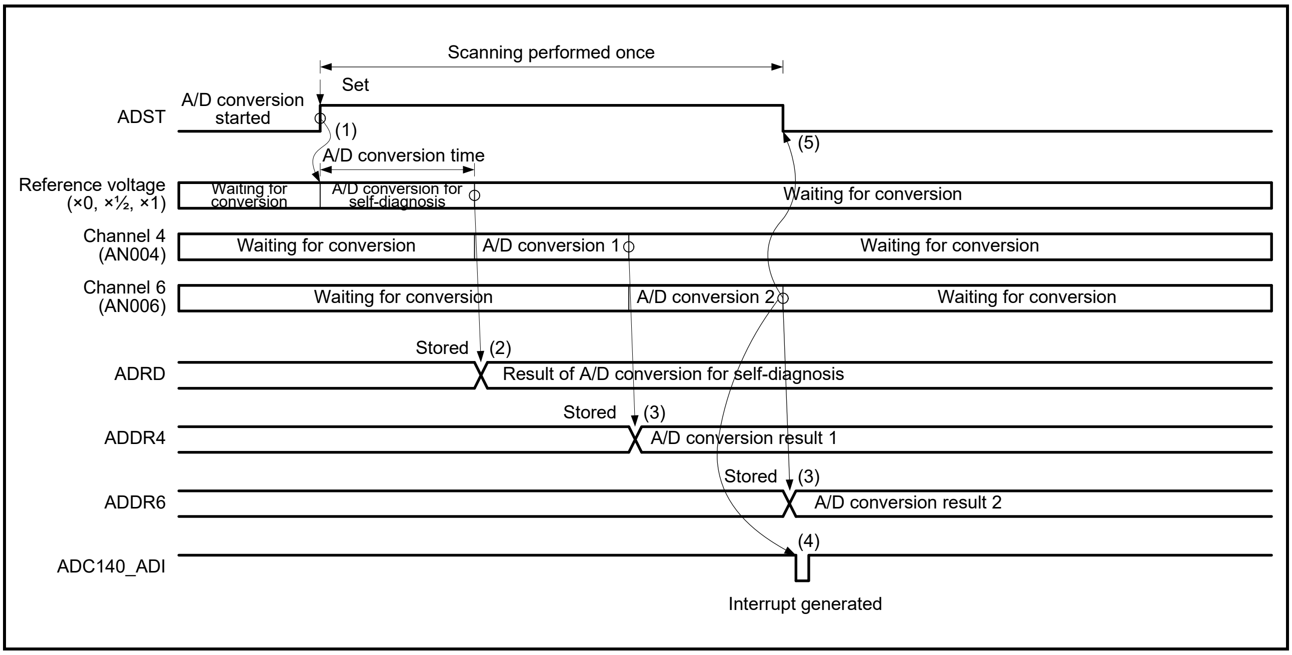

Scanning performed once
Set
A/D conversion started
ADST
(1)
(5)
A/D conversion time
Reference voltage (×0, ×½, ×1)
Waiting for conversion
A/D conversion for self-diagnosis
Waiting for conversion
Channel 4 (AN004)
Waiting for conversion
A/D conversion 1
Waiting for conversion
Channel 6 (AN006)
Waiting for conversion
A/D conversion 2
Waiting for conversion
Stored
(2)
ADRD
Result of A/D conversion for self-diagnosis
Stored
(3)
ADDR4
A/D conversion result 1
Stored
(3)
ADDR6
A/D conversion result 2
(4)
ADC140_ADI
Interrupt generated

Figure 34.7 Example of basic operation in single scan mode when AN004 and AN006 selected with self-diagnosis

34.3.2.3 A/D conversion of temperature sensor output or internal reference voltage

A/D conversion is performed on the temperature sensor output or the internal reference voltage in single scan mode as described in this section.

When selecting A/D conversion of the temperature sensor output or the internal reference voltage, deselect all channels by setting the ADANSA0 and ADANSA01 registers to all 0’s, and the ADCSR.DBLE bit to 0.

When selecting A/D conversion of temperature sensor output, set the Internal Reference Voltage A/D Conversion Select bit (ADEXICR.OCSA) to 0 (deselected). When selecting A/D conversion of internal reference voltage, set the Temperature Sensor Output A/D Conversion Select bit (ADEXICR.TSSA) to 0 (deselected).

The operation is as follows:

1. Set the sampling time to 5 µs or longer. Take note of the A/D Sampling State Register T and A/D Sampling State Register O (ADSSTRT/ADSSTRO) settings, and ADCLK frequency.
2. After switching to A/D conversion of internal reference voltage or temperature sensor output, set the ADST bit to 1 to start conversion.
3. On completion of A/D conversion, the result is stored in the Temperature Sensor Data Register (ADTSDR) or the A/D Internal Reference Voltage Data Register (ADOCDR), and an ADC140_ADI interrupt request is generated.
4. The ADST bit remains 1 during A/D conversion and is automatically set to 0 on completion of the A/D conversion. The ADC14 then enters a wait state.

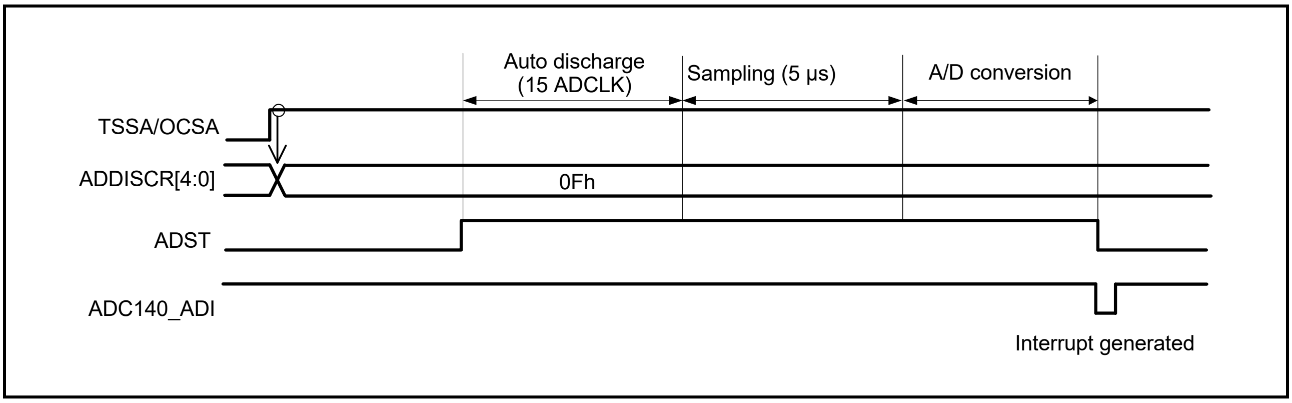


Figure 34.8 **Example of basic operation in single scan mode when AN004 and temperature sensor output or internal reference voltage is selected**

34.3.2.4 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 in sequence as described in this section.

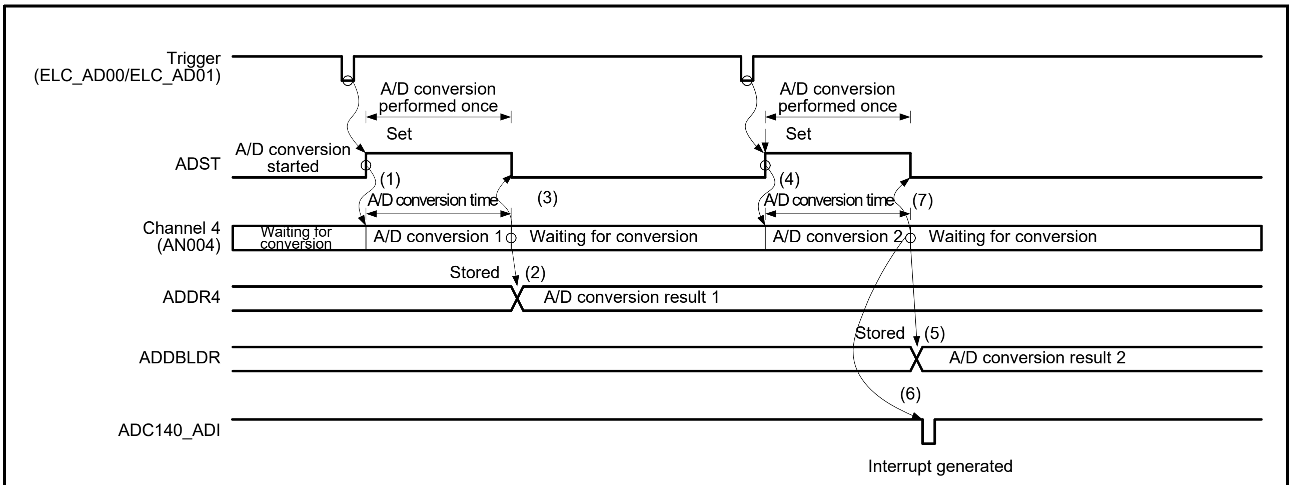
Deselect self-diagnosis, and set the Temperature Sensor Output A/D Conversion Select bit (ADEXICR.TSSA) and the Internal Reference Voltage A/D Conversion Select bit (ADEXICR.OCSA) to 0.

Duplication of A/D conversion data is enabled by setting the channel number to be duplicated in the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid.

In double trigger mode, select a synchronous trigger (ELC) with the ADSTRGR.TRSA[5:0] bits. Additionally, set the ADCSR.EXTRG bit to 0 and the ADCSR.TRGE bit to 1. Do not use a software trigger.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a synchronous trigger input (ELC), A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
2. When A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register (ADDRy).
3. The ADST bit is automatically set to 0 and the ADC14 enters a wait state. An ADC140_ADI interrupt request is not generated.
4. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger input, A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
5. When A/D conversion completes, the A/D conversion result is stored into the A/D Data Duplexing Register (ADDBLDR), which is only used in double trigger mode.
6. An ADC140_ADI interrupt request is generated.
7. The ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion completes. The ADC14 then enters a wait state.


Trigger
(ELC_AD00/ELC_AD01)
A/D conversion performed once
Set
ADST
A/D conversion started
(1)
A/D conversion time
(3)
(4)
(7)
Channel 4 (AN004)
Waiting for conversion
A/D conversion 1
Waiting for conversion
A/D conversion 2
Waiting for conversion
Stored
(2)
ADDR4
A/D conversion result 1
Stored
(5)
ADDBLDR
A/D conversion result 2
(6)
ADC140_ADI
Interrupt generated

Note: In the figure, AN004 is set to be duplicated and the ELC_AD00/ELC_AD01 trigger is selected.

Figure 34.9 Example of operation in single scan mode with double trigger mode selected when AN004 is duplicated

34.3.2.5 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.

Deselect self-diagnosis, and set the Temperature Sensor Output A/D Conversion Select bit (ADEXICR.TSSA) and the Internal Reference Voltage A/D Conversion Select bit (ADEXICR.OCSA) to 0.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid.

In extended double trigger mode, select a synchronous trigger ELC_AD00/ELC_AD01 by setting the ADSTRGR.TRSA[5:0] bits to 0Bh, set the ADCSR.EXTRG bit to 0, and set the ADCSR.TRGE bit to 1. Do not use a software trigger.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a synchronous trigger input (ELC_AD00/ELC_AD01), A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
2. When A/D conversion of a single channel completes, the conversion result is stored in the associated A/D Data Register (ADDRy) and in A/D Data Duplexing Register A (ADDBLDRA) or A/D Data Duplexing Register B (ADDBLDRB) when the trigger of ELC_AD00 or ELC_AD01 is input, respectively.
3. The ADCSR.ADST bit is automatically set to 0 and the ADC14 enters a wait state. An ADC140_ADI interrupt is not generated.
4. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger input (ELC_AD00/ELC_AD01), A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
5. When A/D conversion completes, the 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.
7. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when the A/D conversion completes. The ADC14 then enters a wait state.

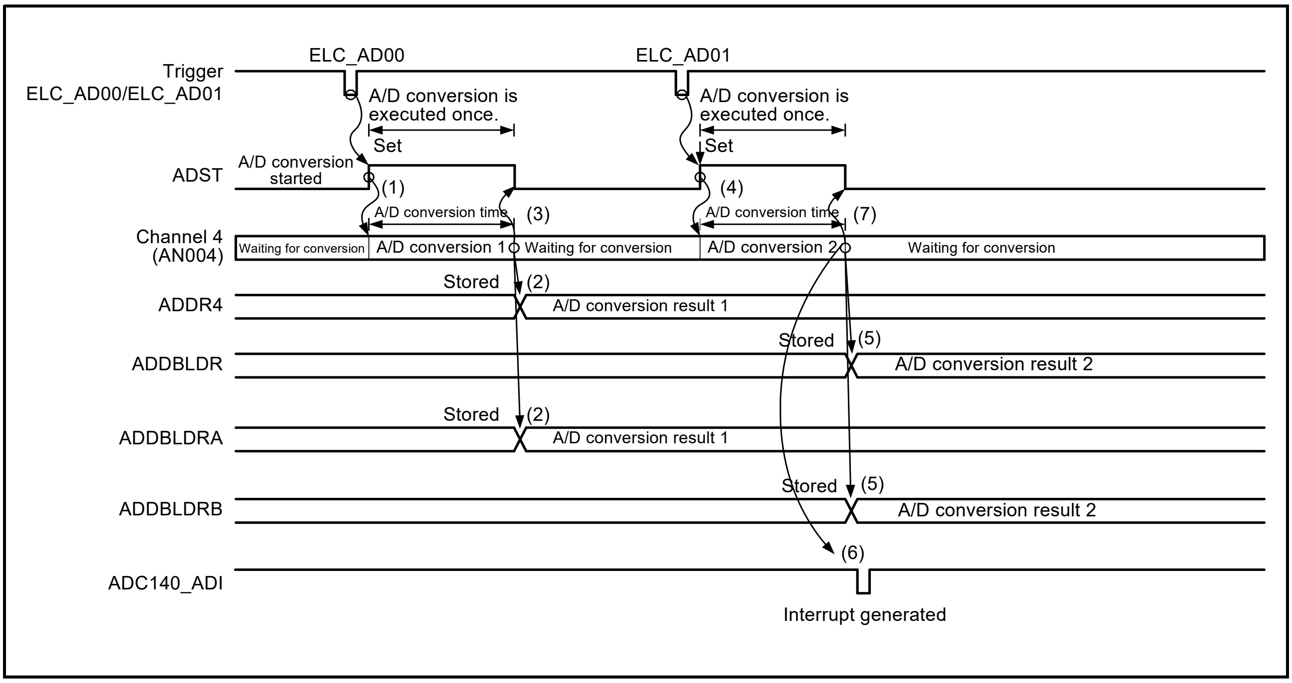

ELC_AD00
ELC_AD01
Trigger ELC_AD00/ELC_AD01
A/D conversion is executed once.
Set
ADST
A/D conversion started
(1)
(4)
A/D conversion time
(3)
(7)
Channel 4 (AN004)
Waiting for conversion
A/D conversion 1
A/D conversion 2
Stored
(2)
ADDR4
A/D conversion result 1
ADDBLDR
(5)
A/D conversion result 2
ADDBLDRA
ADDBLDRB
(6)
ADC140_ADI
Interrupt generated

Figure 34.10 Example of extended operation in double trigger mode (1) with duplication selected for AN004, and ELC_AD00/ELC_AD01 selected

34.3.3 Continuous Scan Mode

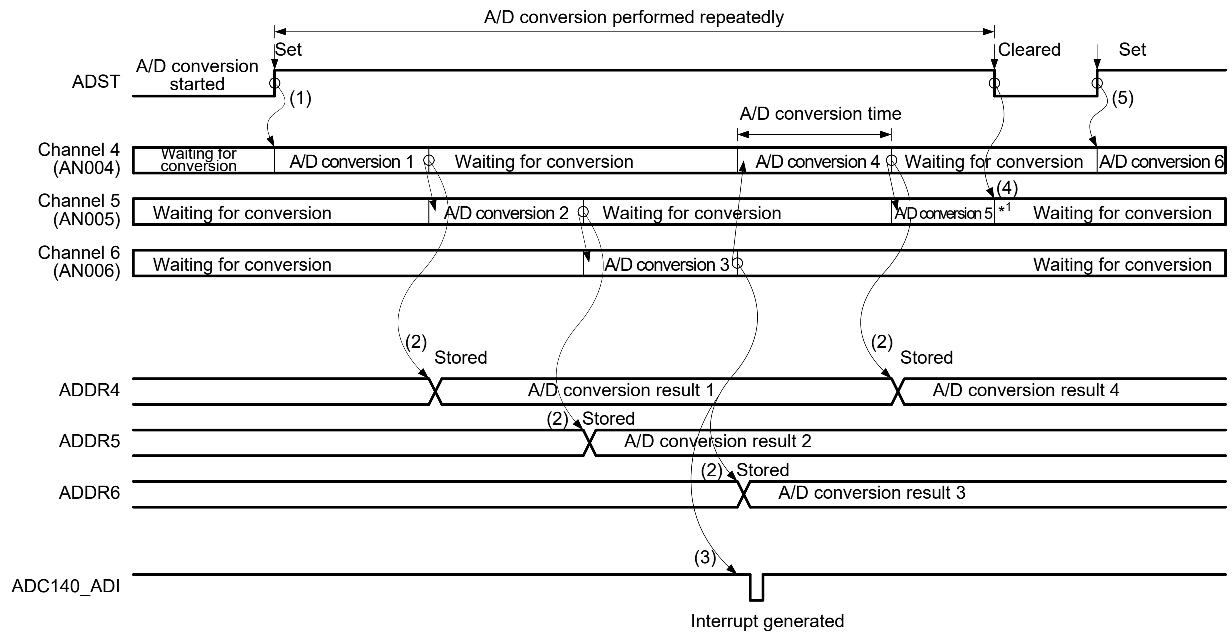
34.3.3.1 Basic operation

In continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels as described in this section.

In this mode, deselect the temperature sensor output A/D conversion and the internal reference voltage A/D conversion by setting the ADEXICR.TSSA and ADEXICR.OCSA bits to 0.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (ELC), or an asynchronous trigger input, A/D conversion is performed on the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
2. When A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register (ADDRy).
3. When A/D conversion of all the selected channels completes, an ADC140_ADI interrupt request is generated without register setting. The ADC14 sequentially starts A/D conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
4. The ADCSR.ADST bit is not automatically cleared and steps 2. and 3. are repeated as long as 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 ADC14 enters a wait state.
5. When the ADST bit is then set to 1 (A/D conversion start), A/D conversion starts again for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.


A/D conversion performed repeatedly
Set
Cleared
Set
ADST
A/D conversion started
(1)
(5)
A/D conversion time
Channel 4 (AN004)
Waiting for conversion
A/D conversion 1
Waiting for conversion
A/D conversion 4
Waiting for conversion
A/D conversion 6
(4)
Channel 5 (AN005)
Waiting for conversion
A/D conversion 2
Waiting for conversion
A/D conversion 5
*1
Waiting for conversion
Channel 6 (AN006)
Waiting for conversion
A/D conversion 3
Waiting for conversion
(2)
Stored
(2)
Stored
ADDR4
A/D conversion result 1
A/D conversion result 4
(2)
Stored
ADDR5
A/D conversion result 2
(2)
Stored
ADDR6
A/D conversion result 3
(3)
ADC140_ADI
Interrupt generated

Note 1. Data for A/D conversion 5 is ignored.

Figure 34.11 Example of basic operation in continuous scan mode with AN004 to AN006 selected

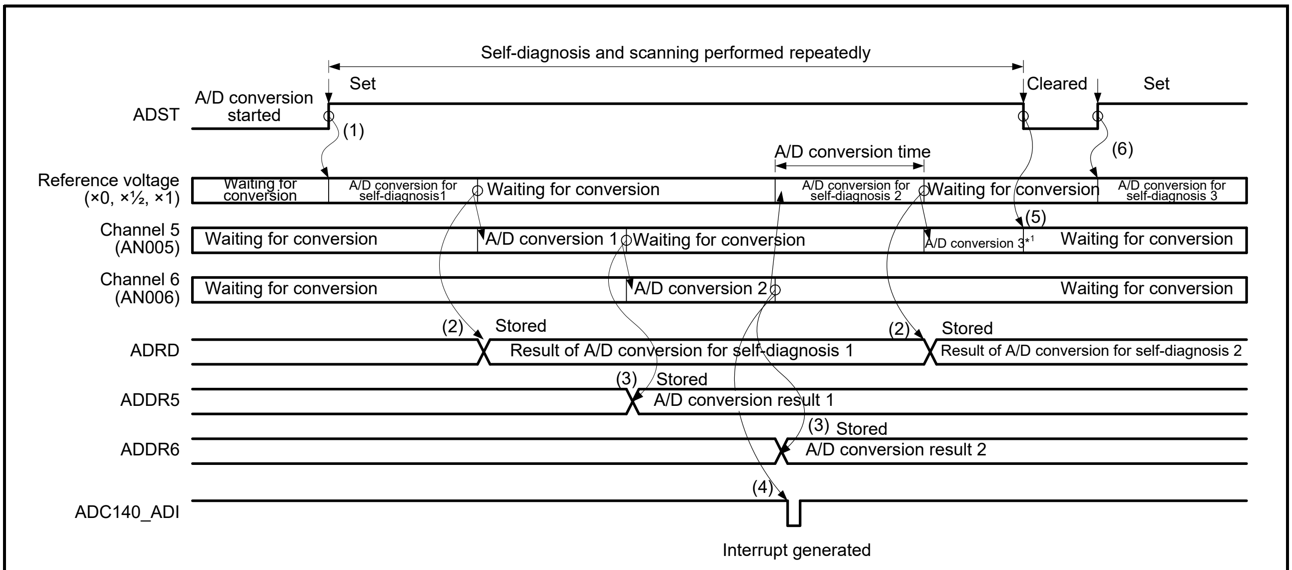
34.3.3.2 Channel selection and self-diagnosis

When channels are selected together with self-diagnosis, A/D conversion is first performed on the reference voltage VREFH0 (×0, ×1/2, or ×1) supplied to the ADC14, then A/D conversion is performed on the analog input of the selected channels. This sequence is repeated as described in this section.

In continuous scan mode, deselect the temperature sensor output A/D conversion and the internal reference voltage A/D conversion by setting the ADEXICR.TSSA and ADEXICR.OCSA bits to 0.

The operation is as follows:

1. A/D conversion for self-diagnosis is first started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (ELC), or an asynchronous trigger input.
2. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored into the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed on the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel completes, the A/D conversion result is stored into the associated A/D Data Register (ADDRy).
4. When A/D conversion of all the selected channels completes, an ADC140_ADI interrupt request is generated (without register setting). At the same time, the ADC14 starts A/D conversion for self-diagnosis and then on the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
5. The 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 ADC14 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 starts again.


Self-diagnosis and scanning performed repeatedly
Set
Cleared
Set
ADST
A/D conversion started
(1)
A/D conversion time
(6)
Reference voltage (×0, ×½, ×1)
Waiting for conversion
A/D conversion for self-diagnosis1
Waiting for conversion
A/D conversion for self-diagnosis 2
Waiting for conversion
A/D conversion for self-diagnosis 3
Channel 5 (AN005)
Waiting for conversion
A/D conversion 1
Waiting for conversion
(5)
A/D conversion 3*1
Waiting for conversion
Channel 6 (AN006)
Waiting for conversion
A/D conversion 2
Waiting for conversion
(2) Stored
(2) Stored
ADRD
Result of A/D conversion for self-diagnosis 1
Result of A/D conversion for self-diagnosis 2
(3) Stored
ADDR5
A/D conversion result 1
(3) Stored
ADDR6
A/D conversion result 2
(4)
ADC140_ADI
Interrupt generated

Note 1. Data for A/D conversion 3 is ignored.

Figure 34.12 Example of basic operation in continuous scan mode when AN005 and AN006 are selected with self-diagnosis

34.3.4 Group Scan Mode

34.3.4.1 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 described in this section. The scan operation of each group is similar to the scan operation in single scan mode.

The synchronous triggers for group A and group B can be selected with the ADSTRGR.TRSA[5:0] bits for group A and with the ADSTRGR.TRSB[5:0] bits for group B. Use different triggers for group A and group B to prevent simultaneous A/D conversion of the two groups. Do not use a software trigger.

The group A channels to be A/D-converted are selected using the ADANSA0 and ADANSA1 registers, 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, set the Temperature Sensor Output A/D Conversion Select bit (ADEXICR.TSSA) and the Internal Reference Voltage A/D Conversion Select bit (ADEXICR.OCSA) to 0. When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed for group A and group B.

The following sequence describes operation in group scan mode using a synchronous trigger from the ELC. In this example, the ELC_AD00 trigger from the ELC is used to start conversion of group A and the ELC_AD01 trigger from the ELC is used to start conversion of group B. Also, the ELC_AD00 and ELC_AD01 are selected for the GPT event in the associated ELC.ELSRn registers.

The operation is as follows:

1. Scanning of group A is started by ELC_AD00.
2. When group A scanning completes, an ADC140_ADI interrupt is generated without register setting.
3. Scanning of group B is started by ELC_AD01.
4. When group B scanning completes, an ADC140_GBADI interrupt is generated if the ADCSR.GBADIE bit is 1 (ADC140_GBADI interrupt is enabled).

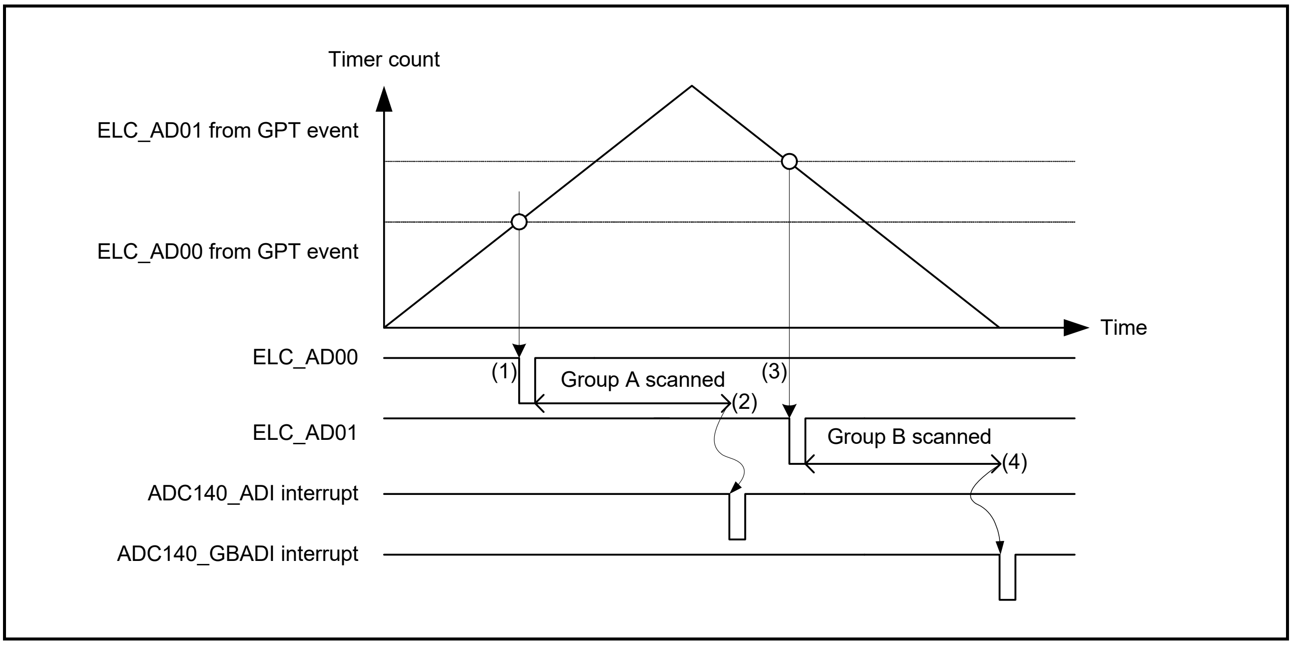


Figure 34.13 Example of basic operation in group scan mode with synchronous triggers from ELC

34.3.4.2 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 in sequence for group A. For group B, single scan operation started by a synchronous trigger (ELC) is performed once.

In group scan mode, select synchronous triggers for group A and B with the ADSTRGR.TRSA[5:0] bits for group A and the ADSTRGR.TRSB[5:0] bits for group B. Use different triggers for group A and group B to prevent simultaneous A/D conversion of the two groups. Do not use a software trigger or an asynchronous trigger (ADTRG0).

When ELC_AD00/ELC_AD01 is selected as the group A synchronous trigger by setting the ADSTRGR.TRSA[5:0] bits to 0Bh, operation proceeds in extended double trigger mode.

The group A channel to be A/D-converted is selected with the ADCSR.DBLANS[4:0] bits and the group B channel to be A/D converted is selected in the ADANSB0 and ADANSB1 registers. Group A and group B cannot use the same channels.

In group scan mode, deselect the temperature sensor output A/D conversion and the internal reference voltage A/D conversion by setting the ADEXICR.TSSA and ADEXICR.OCSA bits to 0.

When 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 ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE to 1.

The following sequence describes operation in group scan mode with double trigger mode using synchronous triggers from the ELC. In this example, the ELC_AD00 trigger is used to start conversion of group A and the ELC_AD01 trigger is used to start conversion of group B. In addition, ELC_AD00 and ELC_AD01 are selected for the GPT event by the associated ELC.ELSRn registers.

The operation is as follows:

1. Scanning of group B is started by the ELC_AD00 trigger from the ELC.
2. When group B scanning completes, an ADC140_GBADI interrupt is generated if the GBADIE bit in the ADCSR register is 1 (group B scan end interrupt is enabled).
3. The first scan of group A is started by the first ELC_AD01 trigger.
4. When the first scan of group A completes, the conversion result is stored in the associated A/D Data Register (ADDRy). An ADC140_ADI interrupt request is not generated.

5. The second scan of group A is started by the second ELC_AD01 trigger.
6. When the second scan of group A completes, the conversion result is stored in the ADDBLDR register. An ADC140_ADI interrupt is generated without register setting.

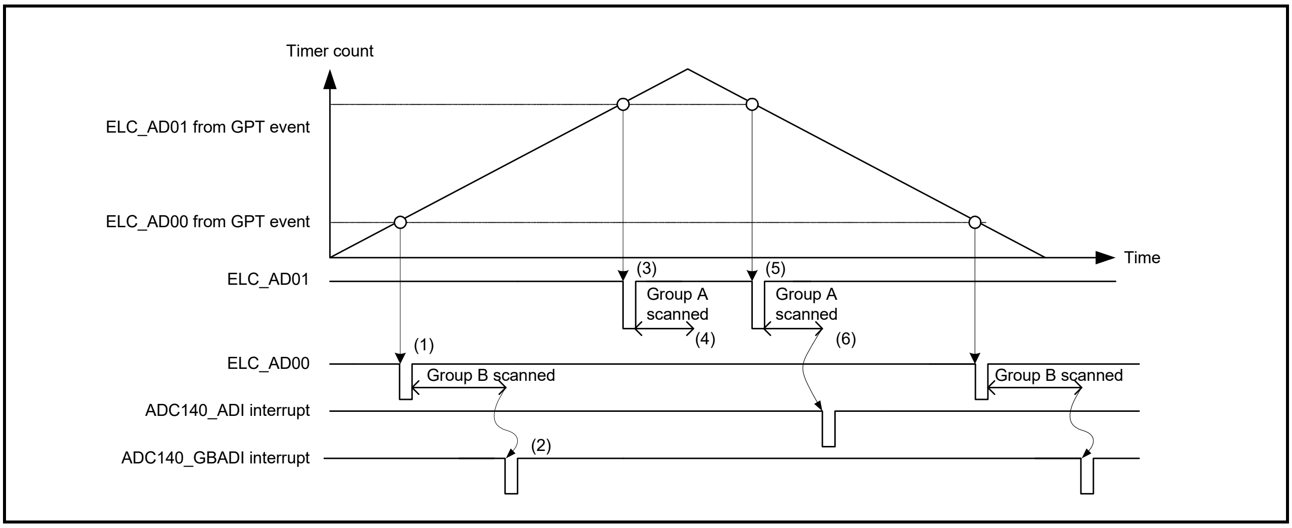


Figure 34.14 Example of operation in group scan mode with double trigger mode using synchronous triggers from the ELC

34.3.4.3 Operation with group A priority control

Setting the ADGSPCR.PGS bit to 1 in group scan mode makes the operation proceed with group A priority control. When setting the PGS bit in the ADGSPCR register to 1, follow the procedure described in Figure 34.15. If the procedure is not followed, A/D conversion operation and stored data are not guaranteed.

In basic group scan mode, while A/D conversion is in progress for group A or group B, input of the trigger for A/D conversion for the other group is ignored. With 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 ADGSPCR.GBRSCN bit is 0, the ADC14 enters wait state on completion of the A/D conversion for group A. If ADGSPCR.GBRSCN bit is 1, the ADC14 automatically restarts group B scanning from the head of the group after completion of the A/D conversion for group A. Table 34.9 summarizes operations in response to the input of a trigger during A/D conversion with 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 when 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.

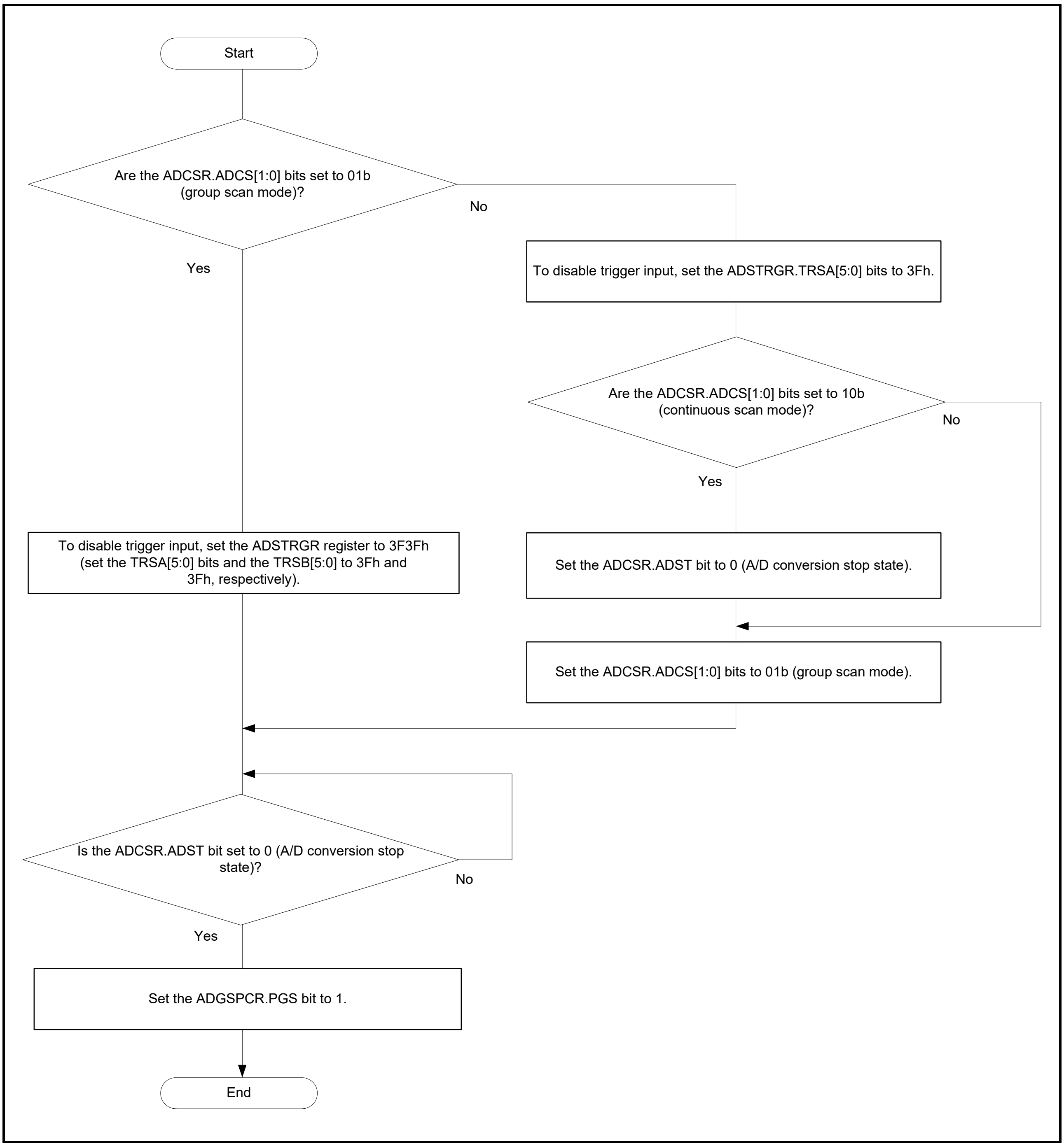

Start
Are the ADCSR.ADCS[1:0] bits set to 01b (group scan mode)?
No
Yes
To disable trigger input, set the ADSTRGR.TRSA[5:0] bits to 3Fh.
Are the ADCSR.ADCS[1:0] bits set to 10b (continuous scan mode)?
No
Yes
To disable trigger input, set the ADSTRGR register to 3F3Fh (set the TRSA[5:0] bits and the TRSB[5:0] to 3Fh and 3Fh, respectively).
Set the ADCSR.ADST bit to 0 (A/D conversion stop state).
Set the ADCSR.ADCS[1:0] bits to 01b (group scan mode).
Is the ADCSR.ADST bit set to 0 (A/D conversion stop state)?
No
Yes
Set the ADGSPCR.PGS bit to 1.
End

Figure 34.15 Flow for ADGSPCR.PGS bit setting

Table 34.9 Control of A/D conversion operations based on the ADGSPCR.GBRSCN bit settings (1 of 2)

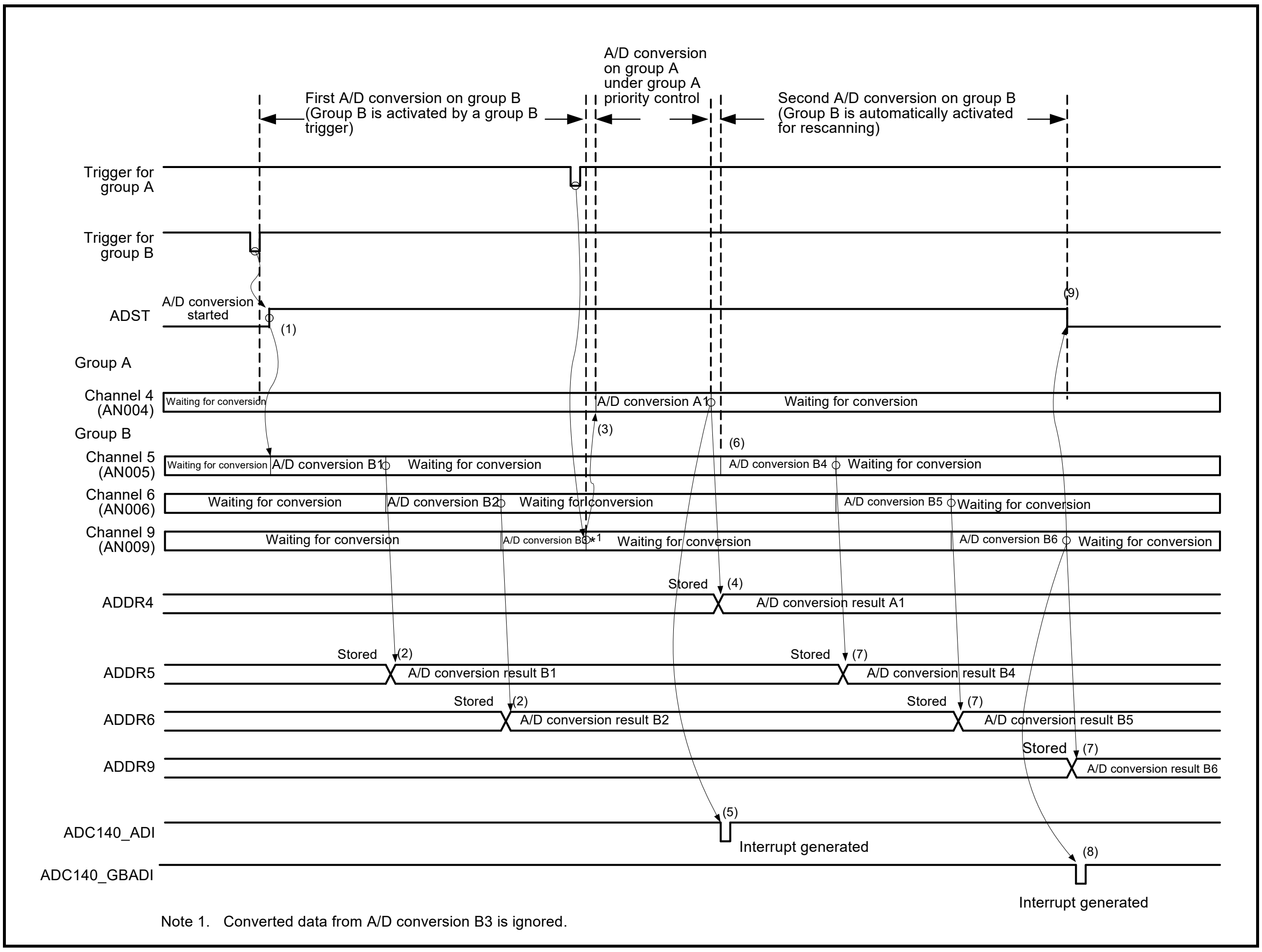
A/D conversion operation	Trigger input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group A is in progress	Input of trigger for group A	Trigger input is ignored	Trigger input is ignored
	Input of trigger for group B	Trigger input is ignored	A/D conversion is performed on group B after A/D conversion on group A completes

Table 34.9 Control of A/D conversion operations based on the ADGSPCR.GBRSCN bit settings (2 of 2)

A/D conversion operation	Trigger input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group B is in progress	Input of trigger for group A	Group B conversion stops and group A conversion starts	• Group B conversion stops and group A conversion starts • Group B conversion starts after group A conversion completes.
	Input of trigger for group B	Trigger input is ignored	Trigger input is ignored

The following sequence describes operations in group scan mode with group A priority control (for example, ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0) when channel 4 is selected for group A and channel 5, 6, and 9 are selected for group B.

1. When input of a trigger for group B sets the ADCSR.ADST bit to 1 (A/D conversion start), 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 for each group B channel, the result is stored in the associated A/D Data Register y (ADDRy).
3. When a group A trigger is input while A/D conversion for group B is in progress, and A/D conversion for group B is discontinued with the ADCSR.ADST bit remains 1, A/D conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the smallest number n. If A/D conversion is not complete when the conversion of group B is interrupted, the A/D conversion result is not stored in the A/D Data Register (ADDRy).
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. A/D conversion for the ANn channels in group B selected in the ADANSB0 and ADANSB1 registers restarts in order from the channel with the smallest number n with the ADCSR.ADST bit remains 1.
7. On completion of A/D conversion on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
8. On completion of A/D conversion of all group B channels, an ADC140_GBADI interrupt request is generated if the ADCSR.GBADIE bit is 1 (ADC140_GBADI interrupt on group B scan end is enabled).
9. The ADCSR.ADST bit is automatically cleared and the ADC14 enters the wait state when A/D conversion is complete.


A/D conversion on group A under group A priority control
First A/D conversion on group B (Group B is activated by a group B trigger)
Second A/D conversion on group B (Group B is automatically activated for rescanning)
Trigger for group A
Trigger for group B
A/D conversion started
ADST
Group A
Channel 4 (AN004)
Group B
Channel 5 (AN005)
Channel 6 (AN006)
Channel 9 (AN009)
Waiting for conversion
A/D conversion A1
A/D conversion B1
A/D conversion B2
A/D conversion B3
*1
A/D conversion B4
A/D conversion B5
A/D conversion B6
ADDR4
ADDR5
ADDR6
ADDR9
Stored
A/D conversion result A1
A/D conversion result B1
A/D conversion result B4
A/D conversion result B2
A/D conversion result B5
A/D conversion result B6
ADC140_ADI
ADC140_GBADI
Interrupt generated
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)

Note 1. Converted data from A/D conversion B3 is ignored.

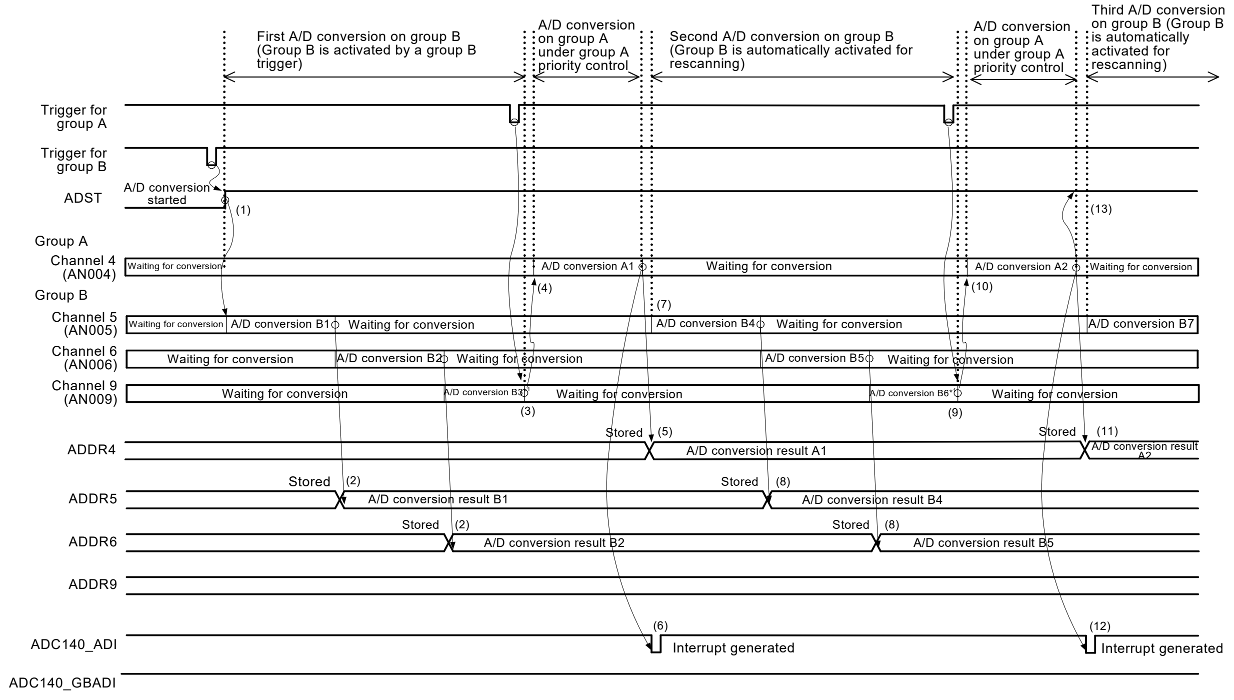
Figure 34.16 Example operation with group A priority control (1), when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0

The following sequence is an example operation when a group A trigger is input again during rescanning operation on group B. In this example, channel 4 is selected for group A and channels 5, 6, and 9 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 (A/D conversion start), 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 associated A/D Data Register (ADDRy).
3. When a group A trigger is input during A/D conversion for group B is in progress, and A/D conversion for group B is discontinued with the ADCSR.ADST bit remains 1. If A/D conversion is not complete when the AD conversion of group B is interrupted, A/D conversion result is not stored in the A/D Data Register (ADDRy).
4. 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 associated A/D Data Register y (ADDRy).
6. An ADC140_ADI interrupt request is generated without register setting.
7. If the ADGSPCR.GBRSCN bit is 1, when the A/D conversion of group A is complete, the ADCSR.ADST bit remains 1 and group B is rescanned. 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 associated A/D Data Register y

(ADDRy).

9. If a group A trigger is input during A/D conversion on group B for rescanning, the ADCSR.ADST bit remains 1 and the ongoing A/D conversion on group B is discontinued.
10. 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 associated A/D Data Register y (ADDRy).
12. An ADC140_ADI interrupt request is generated without register setting.
13. If the ADGSPCR.GBRSCN bit is 1, when A/D conversion of group A is complete, the ADCSR.ADST bit remains 1 and group B is rescanned. 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 ADC14 enters a wait state.


First A/D conversion on group B (Group B is activated by a group B trigger)
A/D conversion on group A under group A priority control
Second A/D conversion on group B (Group B is automatically activated for rescanning)
A/D conversion on group A under group A priority control
Third A/D conversion on group B (Group B is automatically activated for rescanning)
Trigger for group A
Trigger for group B
ADST
A/D conversion started
(1)
(13)
Group A
Channel 4 (AN004)
Waiting for conversion
A/D conversion A1
Waiting for conversion
A/D conversion A2
Waiting for conversion
(4)
(10)
Group B
Channel 5 (AN005)
Waiting for conversion
A/D conversion B1
Waiting for conversion
(7)
A/D conversion B4
Waiting for conversion
A/D conversion B7
Channel 6 (AN006)
Waiting for conversion
A/D conversion B2
Waiting for conversion
A/D conversion B5
Waiting for conversion
Channel 9 (AN009)
Waiting for conversion
A/D conversion B3*1
Waiting for conversion
A/D conversion B6*1
Waiting for conversion
(3)
(9)
ADDR4
Stored
(5)
A/D conversion result A1
Stored
(11)
A/D conversion result A2
ADDR5
Stored
(2)
A/D conversion result B1
Stored
(8)
A/D conversion result B4
ADDR6
Stored
(2)
A/D conversion result B2
Stored
(8)
A/D conversion result B5
ADDR9
ADC140_ADI
(6)
Interrupt generated
(12)
Interrupt generated
ADC140_GBADI

Note 1. Converted data from A/D conversion B3 and A/D conversion B6 are ignored.

Figure 34.17 Example operation with group A priority control (2), when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0

The following sequence 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 5, 6, and 9 are selected for group A and channel 4 is selected for group B when operation on group A is given priority (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0).

1. When input of a group A trigger sets the ADCSR.ADST bit to 1 (A/D conversion start), 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 y (ADDRy).
3. If a group B trigger is input during A/D conversion on group A, group B conversion can be performed after the group A conversion completes. 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 group A conversion, an ADC140_ADI interrupt request is generated without the register setting.
5. On completion of group A conversion, the ADCSR.ADST bit remains 1 and group B is rescanned. Then, A/D 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 y (ADDRy).
7. On completion of the rescanning operation on group B, an ADC140_GBADI interrupt request is generated if the ADCSR.GBADIE bit is 1 (ADC140_GBADI scan end interrupt is enabled).
8. The ADCSR.ADST bit is automatically cleared and the ADC14 enters the wait state when A/D conversion is complete.

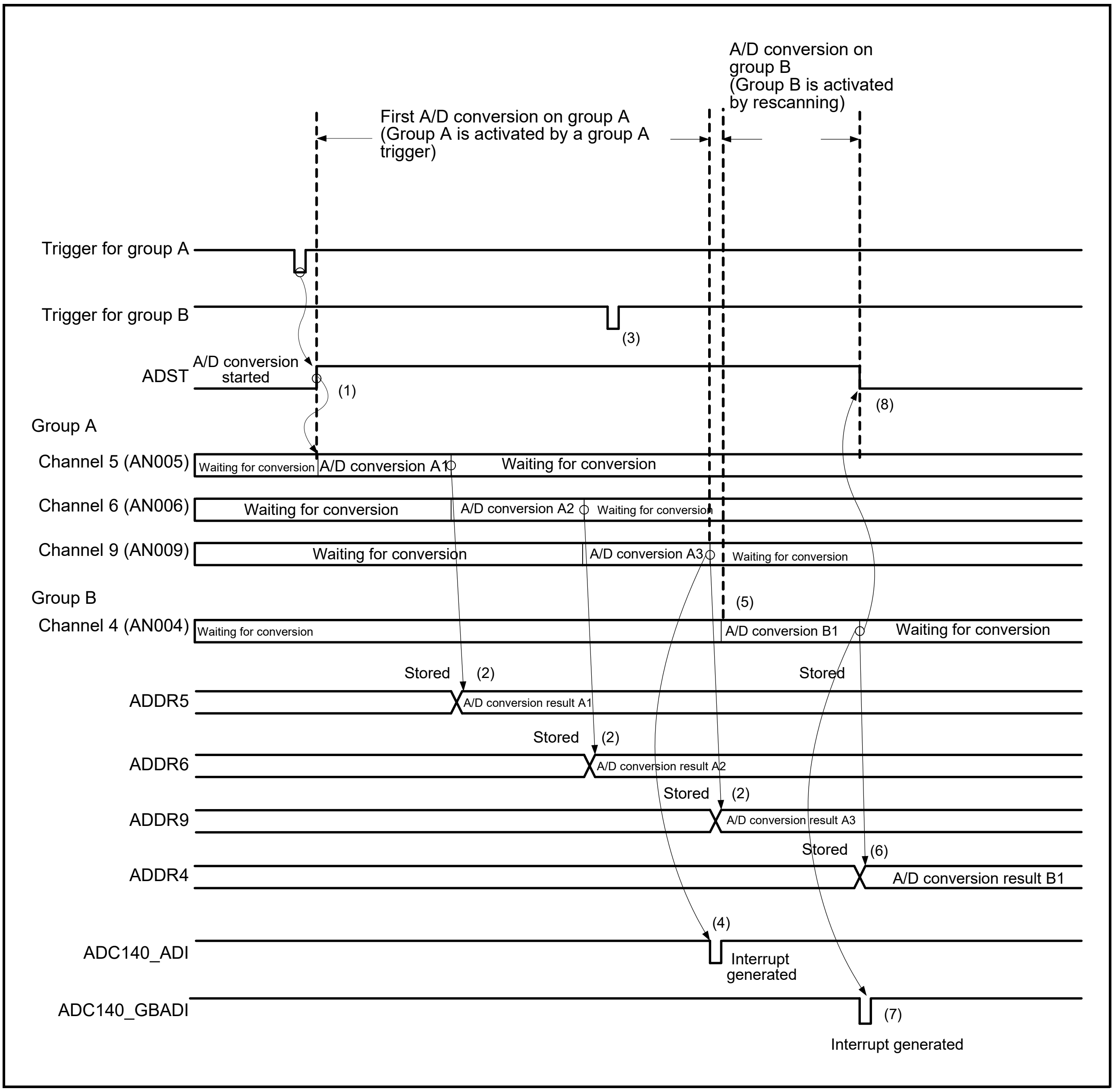
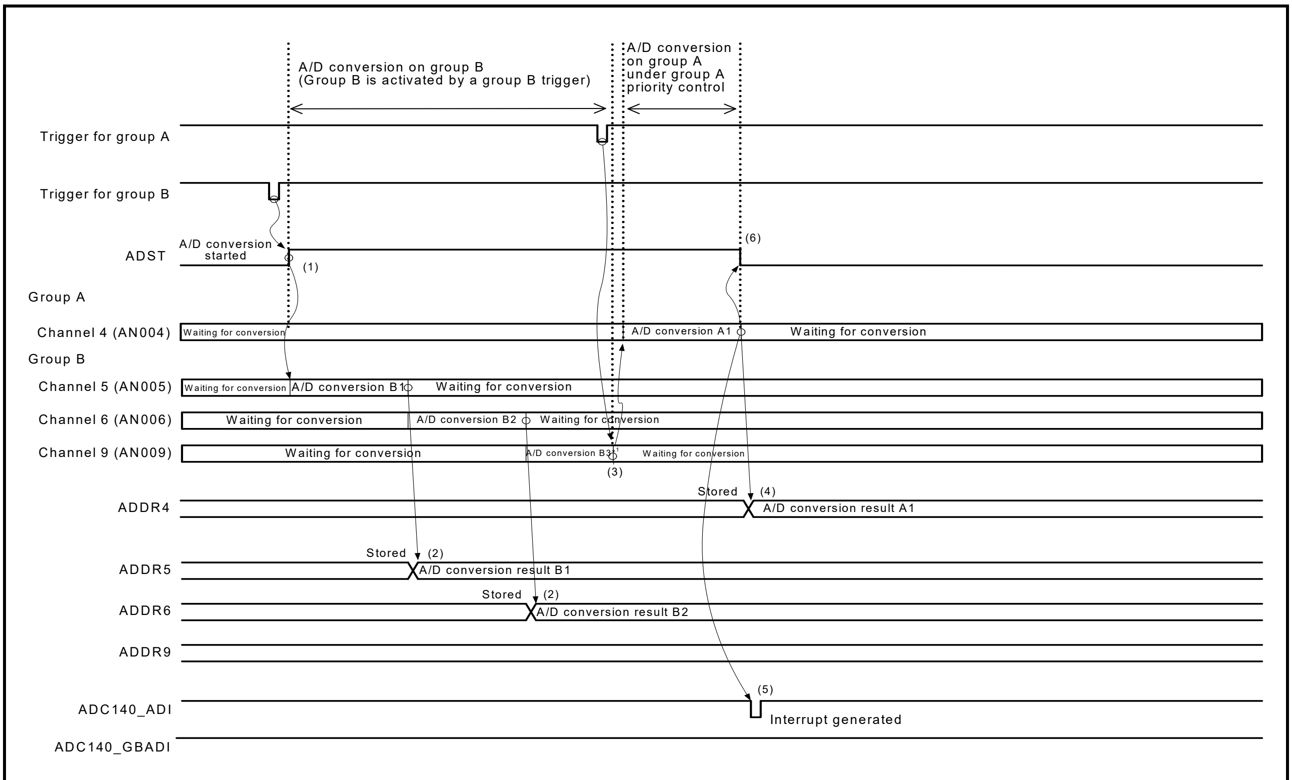

A/D conversion on group B (Group B is activated by rescanning)
First A/D conversion on group A (Group A is activated by a group A trigger)
Trigger for group A
Trigger for group B
(3)
A/D conversion started
ADST
(1)
(8)
Group A
Channel 5 (AN005)
Waiting for conversion
A/D conversion A1
Waiting for conversion
Channel 6 (AN006)
Waiting for conversion
A/D conversion A2
Waiting for conversion
Channel 9 (AN009)
Waiting for conversion
A/D conversion A3
Waiting for conversion
Group B
(5)
Channel 4 (AN004)
Waiting for conversion
A/D conversion B1
Waiting for conversion
Stored
(2)
Stored
ADDR5
A/D conversion result A1
Stored
(2)
ADDR6
A/D conversion result A2
Stored
(2)
ADDR9
A/D conversion result A3
Stored
(6)
ADDR4
A/D conversion result B1
(4)
ADC140_ADI
Interrupt generated
ADC140_GBADI
(7)
Interrupt generated

Figure 34.18 Example operation with group A priority control (3), when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0

The following sequence is an example of operation with group A priority control in which channel 4 is selected for group

A and channels 5, 6, and 9 are selected for group B (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0):

1. When input of a group B trigger sets the ADCSR.ADST bit to 1 (A/D conversion start), 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 y (ADDRy).
3. If a group A trigger is input while A/D conversion for group B is in progress, and A/D conversion for group B is discontinued with the ADCSR.ADST bit remains 1. Then, A/D 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 y (ADDRy).
5. An ADC140_ADI interrupt request is generated without register setting.
6. The ADCSR.ADST bit is automatically cleared and the ADC14 enters the wait state when A/D conversion is complete.


A/D conversion on group B
(Group B is activated by a group B trigger)
A/D conversion on group A under group A priority control
Trigger for group A
Trigger for group B
ADST
A/D conversion started
(1)
(6)
Group A
Channel 4 (AN004)
Waiting for conversion
A/D conversion A1
Waiting for conversion
Group B
Channel 5 (AN005)
Waiting for conversion
A/D conversion B1
Waiting for conversion
Channel 6 (AN006)
Waiting for conversion
A/D conversion B2
Waiting for conversion
Channel 9 (AN009)
Waiting for conversion
A/D conversion B3*1
Waiting for conversion
(3)
ADDR4
Stored
(4)
A/D conversion result A1
ADDR5
Stored
(2)
A/D conversion result B1
ADDR6
Stored
(2)
A/D conversion result B2
ADDR9
ADC140_ADI
(5)
Interrupt generated
ADC140_GBADI

Note 1. The converted data of A/D conversion B3 is ignored.

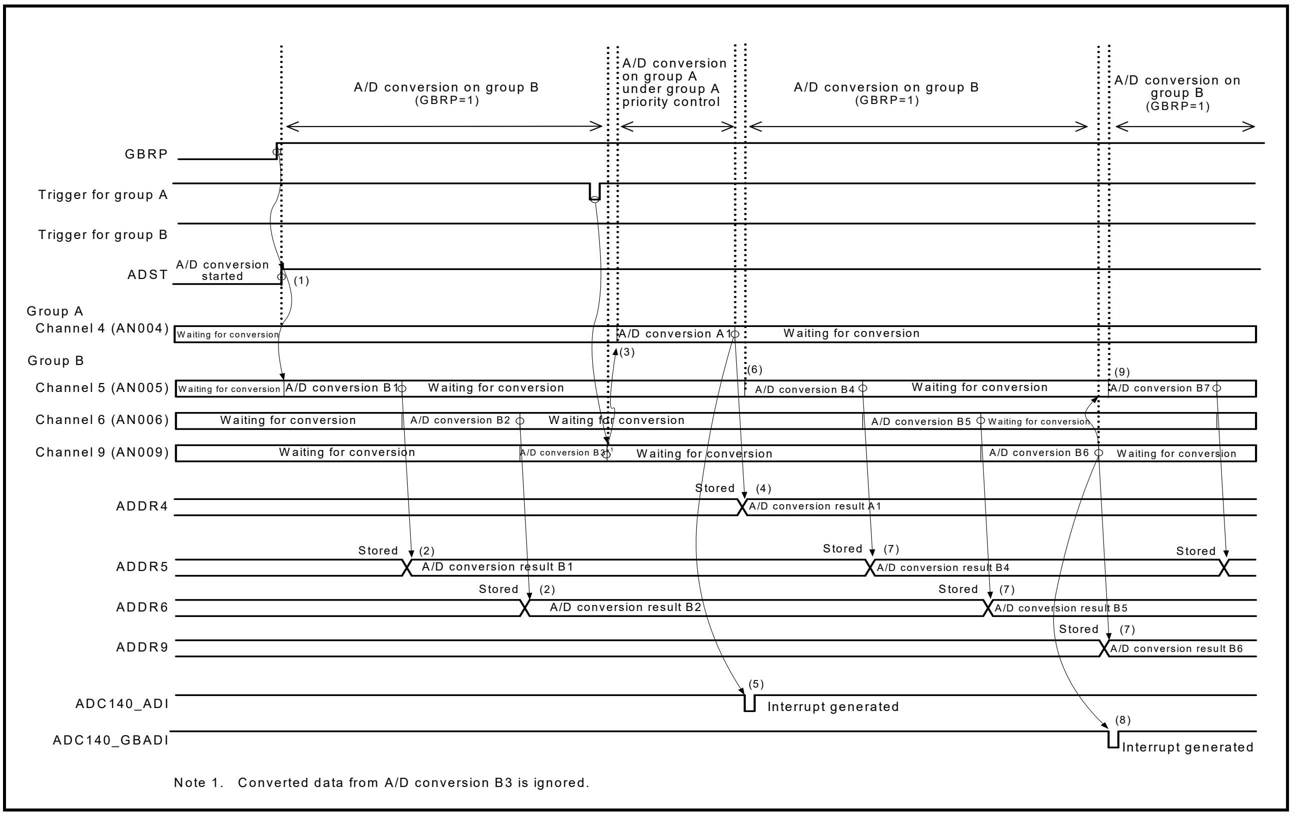
Figure 34.19 Example operation with group A priority control (4), when ADGSPCR.GBRSCN = 0 and ADGSPCR.GBRP = 0

The following sequence is an example of operation with group A priority control in which channel 4 is selected for group A and channels 5, 6, and 9 are selected for group B (ADGSPCR.GBRP = 1):

1. The ADCSR.ADST bit is set to 1 (A/D conversion start) 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 associated A/D Data Register y (ADDRy).
3. If a group A trigger is input while A/D conversion for group B is in progress, and A/D conversion for group B is discontinued with the ADCSR.ADST bit remains 1. Then, A/D 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 y (ADDRy).
5. An ADC140_ADI interrupt request is generated without the register setting.
6. A/D conversion for the ANn channels in group B selected in the ADANSB0 and ADANSB1 registers restarts in order from the channel with the smallest number n with the ADCSR.ADST bit remains 1.
7. On completion of A/D conversion on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
8. An ADC140_GBADI interrupt request is generated if the ADCSR.GBADIE bit is 1.
9. A/D 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. Setting the ADCSR.ADST bit to 0 is prohibited while the ADGSPCR.GBRP bit is set to 1. To forcibly stop A/D conversion while ADGSPCR.GBRP = 1, follow the procedure for clearing the ADCSR.ADST bit operation by software, shown in Figure 34.31.


A/D conversion on group B (GBRP=1)
A/D conversion on group A under group A priority control
A/D conversion on group B (GBRP=1)
A/D conversion on group B (GBRP=1)
GBRP
Trigger for group A
Trigger for group B
ADST
A/D conversion started
(1)
Group A
Channel 4 (AN004)
Waiting for conversion
A/D conversion A1
(3)
Waiting for conversion
Group B
Channel 5 (AN005)
Waiting for conversion
A/D conversion B1
Waiting for conversion
(6)
A/D conversion B4
Waiting for conversion
(9)
A/D conversion B7
Channel 6 (AN006)
Waiting for conversion
A/D conversion B2
Waiting for conversion
A/D conversion B5
Waiting for conversion
Channel 9 (AN009)
Waiting for conversion
A/D conversion B3
Waiting for conversion
A/D conversion B6
Waiting for conversion
ADDR4
Stored
(4)
A/D conversion result A1
ADDR5
Stored
(2)
A/D conversion result B1
Stored
(7)
A/D conversion result B4
Stored
ADDR6
Stored
(2)
A/D conversion result B2
Stored
(7)
A/D conversion result B5
ADDR9
Stored
(7)
A/D conversion result B6
ADC140_ADI
(5)
Interrupt generated
ADC140_GBADI
(8)
Interrupt generated

Note 1. Converted data from A/D conversion B3 is ignored.

Figure 34.20 Example operation with group A priority control (5) when ADGSPCR.GBRP = 1

34.3.5 Compare Function for Window A and Window B

34.3.5.1 Compare function

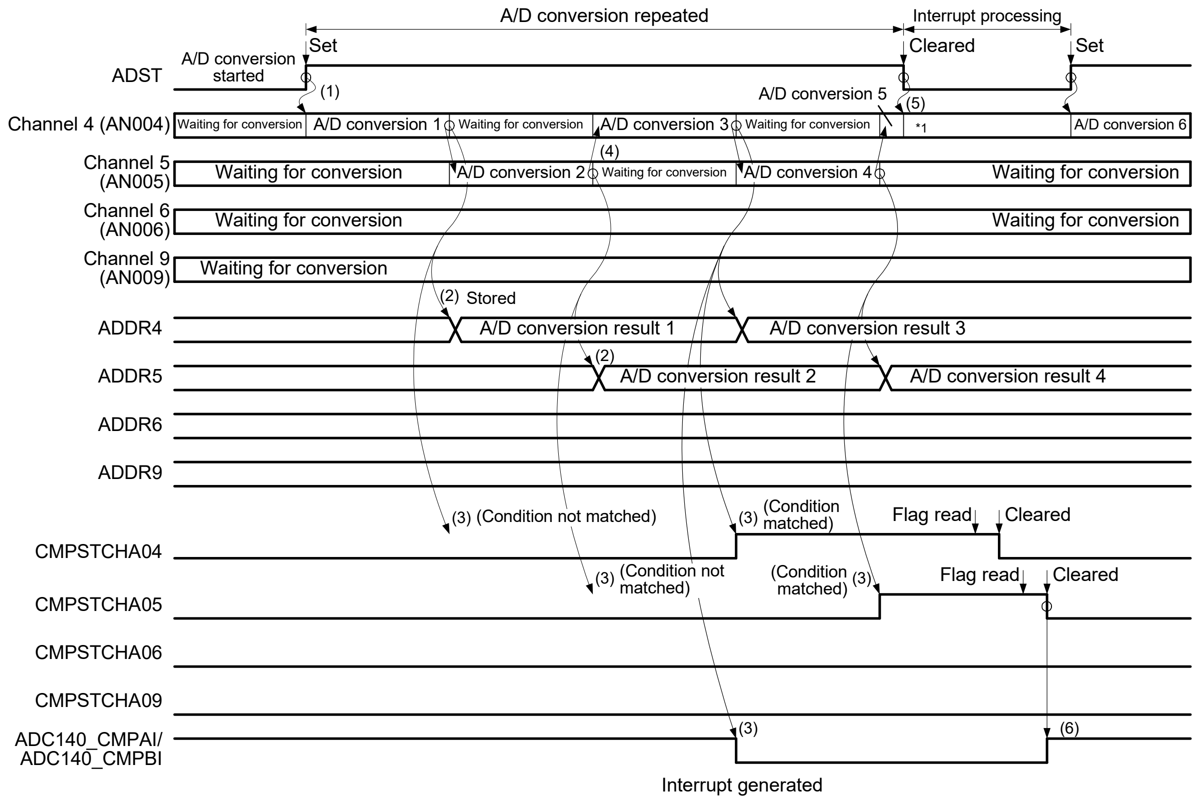
The compare function compares a reference value with the A/D conversion result. The reference value can be set for window A and window B independently. When the compare function is in use, the self-diagnosis function and double trigger mode cannot be used. The main differences between window A and window B are their different interrupt output signals and the restriction on window B to select only one channel.

The following sequence describes an example operation that combines continuous scan mode and the compare function.

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, a synchronous trigger (ELC), or an asynchronous trigger, A/D conversion starts for the selected channel. Do not select the temperature sensor and

internal reference voltage 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 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 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. On 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 on 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 canceled. To perform comparison again, restart the A/D conversion.


A/D conversion repeated
Interrupt processing
A/D conversion started
Set
Cleared
Set
ADST
(1)
A/D conversion 5
(5)
Channel 4 (AN004)
Waiting for conversion
A/D conversion 1
Waiting for conversion
A/D conversion 3
Waiting for conversion
*1
A/D conversion 6
(4)
Channel 5 (AN005)
Waiting for conversion
A/D conversion 2
Waiting for conversion
A/D conversion 4
Waiting for conversion
Channel 6 (AN006)
Waiting for conversion
Waiting for conversion
Channel 9 (AN009)
Waiting for conversion
(2) Stored
ADDR4
A/D conversion result 1
A/D conversion result 3
(2)
ADDR5
A/D conversion result 2
A/D conversion result 4
ADDR6
ADDR9
(3) (Condition not matched)
(3) (Condition matched)
Flag read
Cleared
CMPSTCHA04
(3) (Condition not matched)
(Condition matched) (3)
Flag read
Cleared
CMPSTCHA05
CMPSTCHA06
CMPSTCHA09
ADC140_CMPAI/ ADC140_CMPBI
(3)
(6)
Interrupt generated

Note 1. Data on conversion is ignored.

Figure 34.21 Example of compare function operation, when AN004 to AN006, and AN009 are compared

34.3.5.2 Event output of compare function

The event output of the compare function specifies the upper reference voltage value for window A and the lower reference voltage value for window B, compares the A/D-converted value of the selected channel with the upper and lower side reference voltage values, and outputs the ADC140_WCMPM/ADC140_WCMPUM events according to event conditions (A OR B, A AND B, A XOR B) and comparison result of window A and window B.

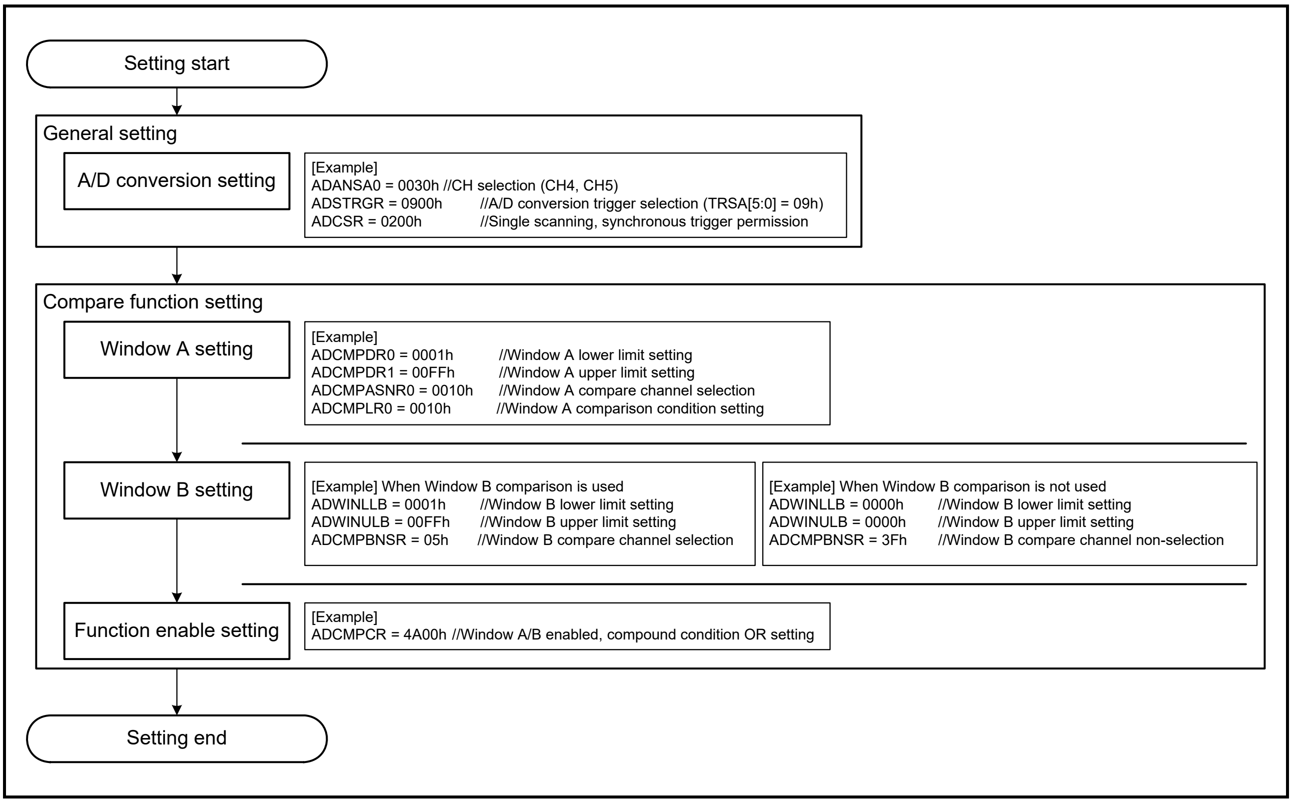
If more than one channel is selected for window A, and even one channel in window A meets the comparison condition, the comparison result of window A is met. When using this function, perform A/D conversion in single scan mode.

Any channels from AN004 to AN006, AN009, AN010, AN017, AN019, and A020, internal reference voltage, and temperature sensor output are selectable for window A. However, neither the internal reference voltage nor the temperature sensor output can be selected together with any other channel. In addition, if the internal reference voltage is selected as the high-potential reference voltage of the ADC14, the internal reference voltage or the temperature sensor output cannot be A/D-converted.

A single channel from AN004 to AN006, AN009, AN010, AN017, AN019, and A020, internal reference voltage, and temperature sensor output is selectable for window B. However, neither the internal reference voltage nor the temperature sensor output can be selected together with any other channel. In addition, if the internal reference voltage is selected as the high-potential reference voltage, the internal reference voltage or the temperature sensor output cannot be A/D-converted.

The following sequence describes the setting procedure and example when using event output of the compare function:

1. Confirm that the ADCSR.ADCS[1:0] bits are 00b (single scan mode).
2. Select the channel for window A in ADCMPANSR0/1 and ADCMPANSER. Set the window comparison conditions in ADCMPLR0/1, ADCMPLER registers. Set the upper and lower reference values in the ADCMPDR0/1 registers.
3. Select the channel and comparison conditions for window B in the ADCMPBNSR register, and set the upper and lower reference values in the ADWINULB and ADWINLLB registers.
4. Set the composite conditions for window A/B, window A/B operation enable, and interrupt output enable in the ADCMPCR register.



```
[Example]
ADANSA0 = 0030h //CH selection (CH4, CH5)
ADSTRGR = 0900h        //A/D conversion trigger selection (TRSA[5:0] = 09h)
ADCSR = 0200h          //Single scanning, synchronous trigger permission
```

```
[Example]
ADCMPDR0 = 0001h       //Window A lower limit setting
ADCMPDR1 = 00FFh       //Window A upper limit setting
ADCMPASNR0 = 0010h     //Window A compare channel selection
ADCMPLR0 = 0010h       //Window A comparison condition setting
```

```
[Example] When Window B comparison is used
ADWINLLB = 0001h       //Window B lower limit setting
ADWINULB = 00FFh       //Window B upper limit setting
ADCMPBNSR = 05h        //Window B compare channel selection
```

```
[Example] When Window B comparison is not used
ADWINLLB = 0000h       //Window B lower limit setting
ADWINULB = 0000h       //Window B upper limit setting
ADCMPBNSR = 3Fh        //Window B compare channel non-selection
```

```
[Example]
ADCMPCR = 4A00h //Window A/B enabled, compound condition OR setting
```

Figure 34.22 Setting example when using event output of the compare function

For event output usage when using only window A for the compare function, note the following:

- Enable both window A and window B (ADCMPCR.CMPAE = 1, ADCMPCR.CMPBE = 1)

- Set the compound condition of window A and B to OR condition (ADCMPCR.CMPAB[1:0] = 00b)
- Set the compared channel of window B to “Do not select” (ADCMPBNSR.CMPCHB[5:0] = 111111b)
- Set the compare condition of window B to 0 < results < 0 means mismatch always (ADCMPCR.WCMPE = 1, ADWINLLB[15:0] = ADWINULB[15:0] = 0000h, ADCMPBNSR.CMPLB = 1).

Figure 34.23 shows an example event output operation of the compare function.

A scan end event (ADC140_ADI) is output at the same time as a one-time single scan completion. A match or mismatch event (ADC140_WCMPM/ADC140_WCMPUM) is output with a clock delay of 1 PCLKB cycle set in ADCMPCR.CMPAB[1:0].

Note: The match and mismatch events are exclusive, so both events do not output simultaneously.

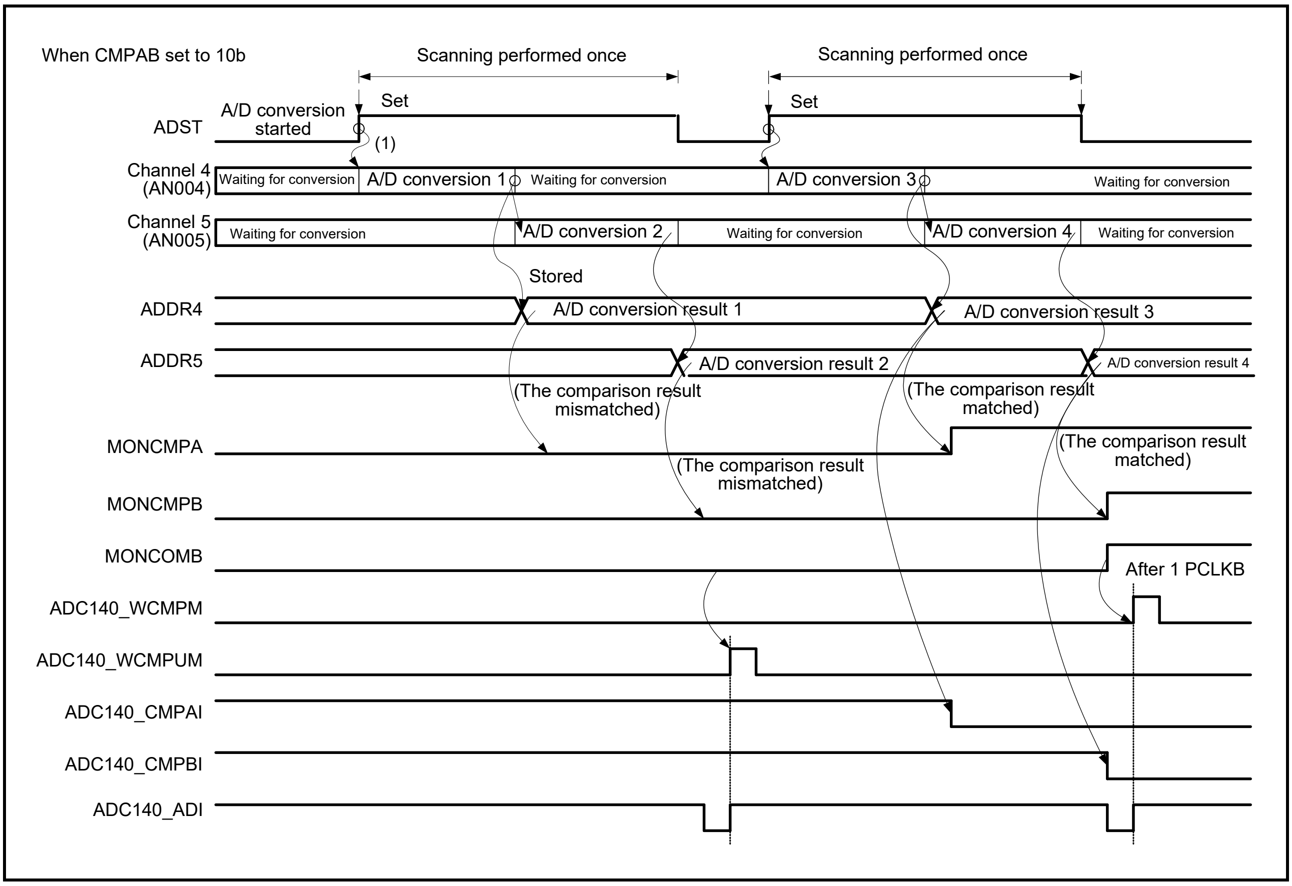

When CMPAB set to 10b
Scanning performed once
Scanning performed once
A/D conversion started
Set
Set
(1)
ADST
Channel 4 (AN004)
Waiting for conversion
A/D conversion 1
Waiting for conversion
A/D conversion 3
Waiting for conversion
Channel 5 (AN005)
Waiting for conversion
A/D conversion 2
Waiting for conversion
A/D conversion 4
Waiting for conversion
Stored
ADDR4
A/D conversion result 1
A/D conversion result 3
ADDR5
A/D conversion result 2
A/D conversion result 4
(The comparison result mismatched)
(The comparison result matched)
MONCMPA
(The comparison result mismatched)
(The comparison result matched)
MONCMPB
MONCOMB
After 1 PCLKB
ADC140_WCMPM
ADC140_WCMPUM
ADC140_CMPAI
ADC140_CMPBI
ADC140_ADI

Figure 34.23 Event output operation example of compare function when AN004 to AN005 are compared

Note: Event output of the compare function outputs match/mismatch from the comparison results of window A and window B, as set in ADCMPCR.CMPAB[1:0].

Note: The comparison result of window A is the logical addition of the comparison results of the comparison target channels of window A. The comparison results of window A and 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.

34.3.5.3 Restrictions on the compare function

The following restrictions apply to the compare function:

- The compare function cannot be used together with the self-diagnosis function or double trigger mode. The compare function is not available for ADRD, ADDBLDR, ADDBLDRA, and ADDBLDRB.
- Specify single scan mode when using match/mismatch event outputs

- When the temperature sensor or internal reference voltage is selected for window A, window B operations are disabled
- When the temperature sensor or internal reference voltage is selected for window B, window A operations are disabled
- Setting the same channel for window A and window B is prohibited
- Set the reference voltage values so that the high-potential reference voltage value is equal to or larger than the low-potential reference voltage value.

34.3.6 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 34.24 shows the scan conversion timing, in which scan conversion is activated by a software trigger or a synchronous trigger (ELC). Figure 34.25 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 ADC14. If the sampling time is not sufficient because of 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 34.10 shows the times for conversion during scanning.

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 first 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 the sum of tSPL and tSAM for each selected channel.

Table 34.10 Times for conversion during scanning (in number of ADCLK and PCLKB cycles)

Parameter			Symbol		Type/Conditions			Unit
					Synchronous trigger*6	**Asynchronous trigger**	**Software trigger**	
Scan start processing time*1, *2	A/D conversion on group A with 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 from group A.)	t_D		3 PCLKB + 6 ADCLK, 5 PCLKB + 3 ADCLK*5	—	—	Cycle
		Group B is not to be stopped. (Activation by an A/D conversion source from group A.)			2 PCLKB + 4 ADCLK	—	—	
	A/D conversion when self-diagnosis is enabled	A/D conversion for self-diagnosis is to be started.			2 PCLKB + 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			
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) x 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*5			

Note 1. See Figure 34.24 and Figure 34.25 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 starting A/D conversion.

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 required sampling time (ns) is specified according to the voltage conditions. The sampling time setting must satisfy the electrical characteristics.

Note 5. If ADCLK is faster than PCLKB (PCLKB to ADCLK frequency ratio = 1:2 or 1:4).

Note 6. This does not include the time consumed in the path from timer output to trigger input.

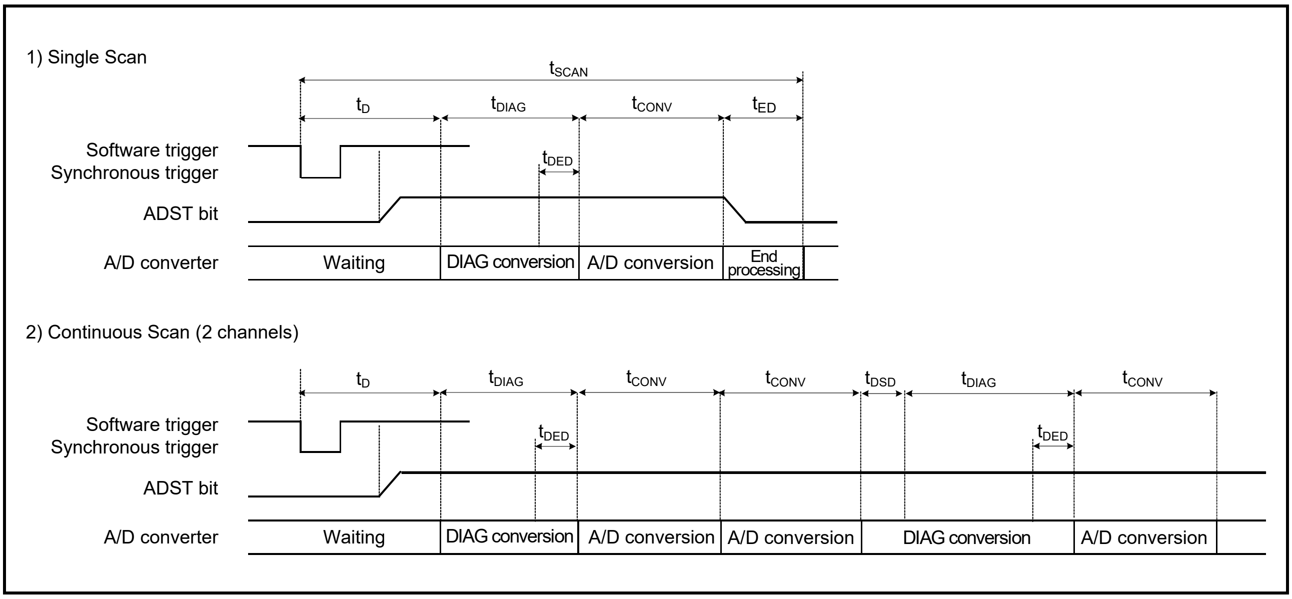

1) Single Scan
tSCAN
tD
tDIAG
tCONV
tED
tDED
Software trigger
Synchronous trigger
ADST bit
A/D converter
Waiting
DIAG conversion
A/D conversion
End processing
2) Continuous Scan (2 channels)
tD
tDIAG
tCONV
tCONV
tDSD
tDIAG
tCONV
tDED
tDED
Software trigger
Synchronous trigger
ADST bit
A/D converter
Waiting
DIAG conversion
A/D conversion
A/D conversion
DIAG conversion
A/D conversion

Figure 34.24 Scan conversion timing when activated by software or synchronous trigger input (ELC)

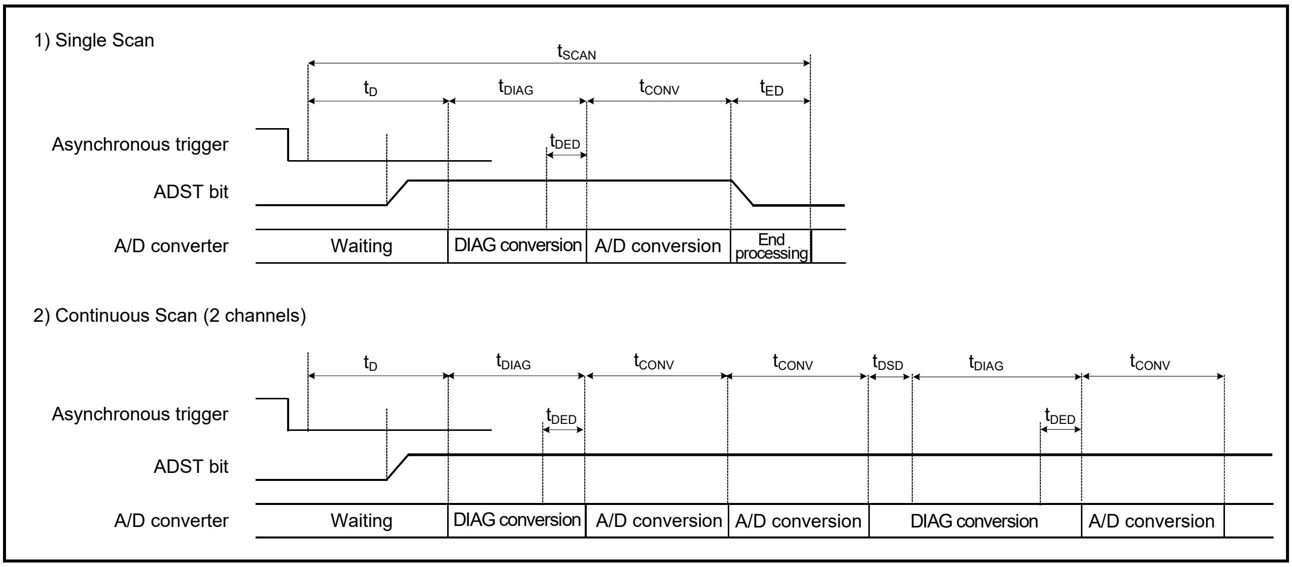

1) Single Scan
tSCAN
tD
tDIAG
tCONV
tED
tDED
Asynchronous trigger
ADST bit
A/D converter
Waiting
DIAG conversion
A/D conversion
End processing
2) Continuous Scan (2 channels)
tD
tDIAG
tCONV
tCONV
tDSD
tDIAG
tCONV
tDED
tDED
Asynchronous trigger
ADST bit
A/D converter
Waiting
DIAG conversion
A/D conversion
A/D conversion
DIAG conversion
A/D conversion

Figure 34.25 Scan conversion timing when activated by asynchronous trigger input (ADTRG0)

34.3.7 Usage Example of A/D Data Register Automatic Clearing Function

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). In the following examples, the function to automatically clear the ADDRy register is enabled and disabled:

- When the ACE bit in ADCER is 0 (automatic clearing disabled) and, for some reason, if the A/D conversion result (0222h) is not written to the ADDRy register, the ADDRy value retains 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 in a general-purpose register.
- When the ACE bit in ADCER is 1 (automatic clearing enabled), if ADDRy = 0111h is read by the CPU, DTC, or DMAC, ADDRy is automatically set to 0000h. If the A/D conversion result of 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, 0000h is saved in the general-purpose register. Occurrence of an ADDRy update failure can be determined by checking that the read data value is 0000h.

34.3.8 A/D-Converted Value Addition/Average Mode

A/D-converted value addition/average mode can be used when A/D conversion of the analog input of the selected channels, A/D conversion of the temperature sensor output, or A/D conversion of the internal reference voltage is selected.

In A/D-converted value addition mode, the same channel is A/D-converted 1, 2, 3, 4, or 16*[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. However, this function cannot always guarantee an improvement in A/D conversion accuracy.

The A/D-converted value addition or average mode can be specified for A/D conversion of the channel select analog input, temperature sensor output, or internal reference voltage.

Note 1. The addition count can be set to 16 only when 12-bit accuracy is selected.

34.3.9 Disconnection Detection Assist Function

The ADC14 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.

If any of the following functions are used, the disconnection detection assist function must be disabled:

- The temperature sensor
- The internal reference voltage
- The A/D self-diagnosis.

Figure 34.26 shows the A/D conversion operation when the disconnection detection assist function is used. Figure 34.27 shows an example of disconnection detection when precharge is selected. Figure 34.28 shows an example of disconnection detection when discharge is selected.

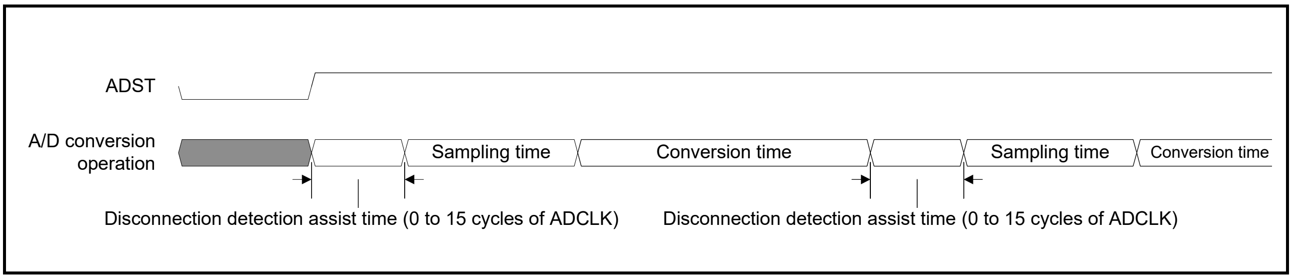
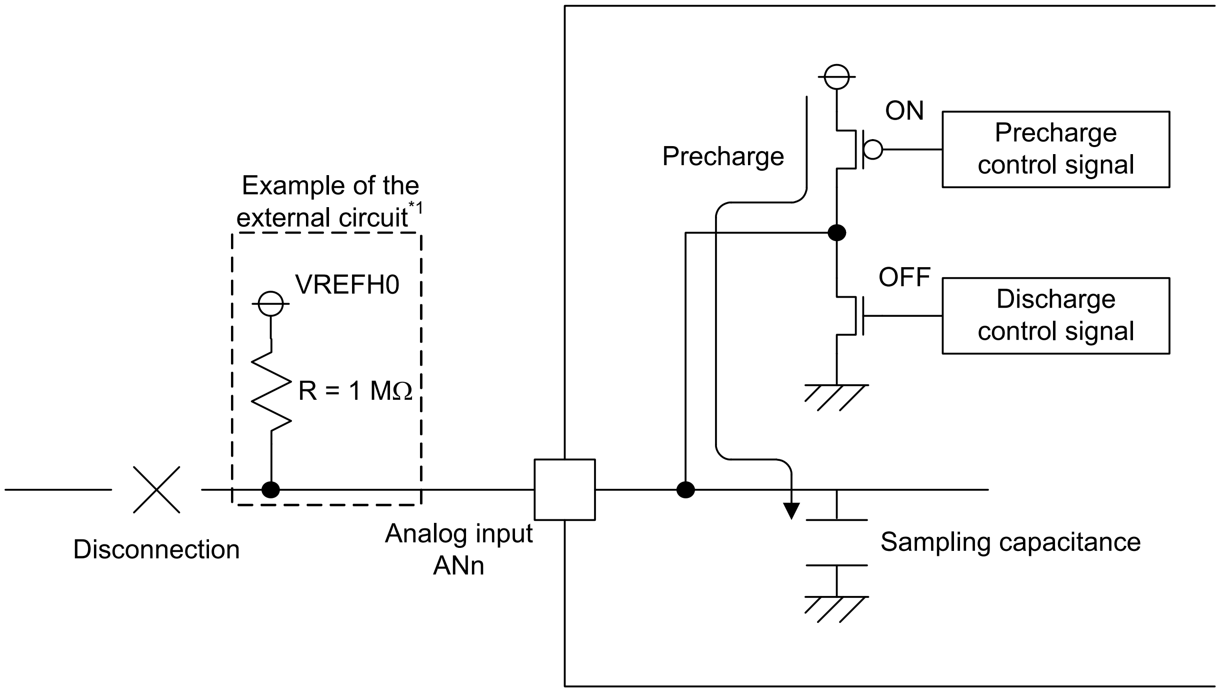
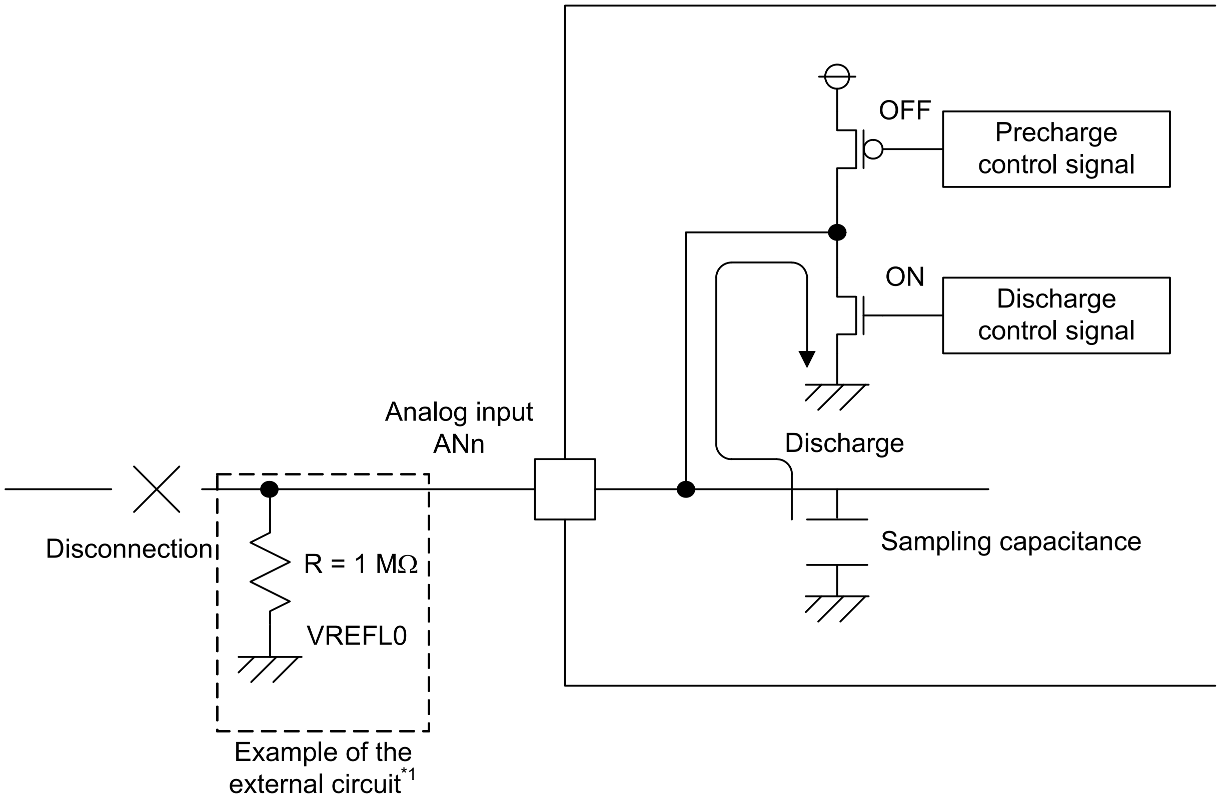

ADST
A/D conversion operation
Sampling time
Conversion time
Sampling time
Conversion time
Disconnection detection assist time (0 to 15 cycles of ADCLK)
Disconnection detection assist time (0 to 15 cycles of ADCLK)

Figure 34.26 A/D conversion operation when the disconnection detection assist function is used


Example of the
external circuit*1
VREFH0
R = 1 MΩ
Disconnection
Analog input
ANn
Precharge
ON
Precharge
control signal
OFF
Discharge
control signal
Sampling capacitance

Note 1. The converted result should be used after full evaluation because the resulting data on disconnection varies depending on the external circuit.

Figure 34.27 Example of disconnection detection when precharge is selected


OFF
Precharge control signal
ON
Discharge control signal
Analog input ANn
Discharge
Disconnection
R = 1 MΩ
VREFL0
Sampling capacitance
Example of the external circuit*1

Note 1. The converted result should be used after full evaluation because the resulting data on disconnection varies depending on the external circuit.

Figure 34.28 Example of disconnection detection when discharge is selected

34.3.10 Starting A/D Conversion with an Asynchronous Trigger

The A/D conversion can be started by the input of an asynchronous trigger. To start the A/D conversion by an asynchronous trigger:

1. Set the pin function in the PmnPFS register.
2. Set the A/D Conversion Start Trigger Select bits (ADSTRGR.TRSA[5:0]) to 000000b.
3. Input a high-level signal to the asynchronous trigger (ADTRG0 pin).
4. Set the ADCSR.TRGE and ADCSR.EXTRG bits to 1.

Figure 34.29 shows the timing of the asynchronous trigger input.

An asynchronous trigger cannot be selected in the A/D Conversion Start Trigger Select bits (ADSTRGR.TRSB[5:0]) for group B in group scan mode. For details on setting the pin function, see section 20, I/O Ports.

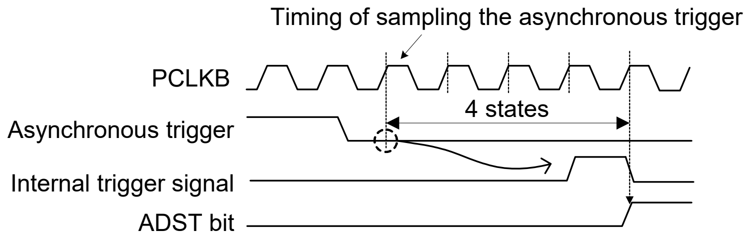

Timing of sampling the asynchronous trigger
PCLKB
4 states
Asynchronous trigger
Internal trigger signal
ADST bit

Figure 34.29 Asynchronous trigger input timing

34.3.11 Starting A/D Conversion with a 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. Set the ADCSR.EXTRG bit to 0.
3. Select the relevant sources in the ADSTRGR.TRSA[5:0] and ADSTRGR.TRSB[5:0] bits.

34.4 Interrupt Sources and DTC or DMAC Transfer Requests

34.4.1 Interrupt Requests

The ADC14 can send scan end interrupt requests, ADC140_ADI and ADC140_GBADI, to the CPU. The ADC14 also generates the ADC140_CMPAI and ADC140_CMPBI interrupts to the CPU in response to matches with a comparison condition.

An ADC140_ADI interrupt is always generated. An ADC140_GBADI interrupt can be generated by setting the ADCSR.GBADIE bit to 1. Similarly, ADC140_CMPAI and ADC140_CMPBI interrupts can be generated by setting the ADCMPCR.CMPAIE and ADCMPCR.CMPBIE bits to 1.

In addition, the DTC or DMAC can be started when an ADC140_ADI or ADC140_GBADI interrupt is generated. Using these interrupts to activate the DTC or DMAC to read the converted data enables continuous conversion without burdening software.

For details on DTC settings, see section 18, Data Transfer Controller (DTC), and for details on DMAC settings, see section 17, DMA Controller (DMAC).

Table 34.11 describes the interrupt sources and ELC events available for the ADC14.

Table 34.11 ADC14 interrupt sources and ELC events (1 of 2)

Operation							
Scan mode	**Double-trigger mode**	**Compare function window A and window B**	**Interrupt request or ELC event**	**Interrupt request**	**DTC or DMAC activation**	**ELC event request**	**Function**
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 the even-numbered times
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

Table 34.11 ADC14 interrupt sources and ELC events (2 of 2)

Operation							
Scan mode	Double-trigger mode	Compare function window A and window B	Interrupt request or ELC event	Interrupt request	DTC or DMAC activation	ELC event request	Function
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 the even-numbered times
			ADC140_GBADI	✓	✓	×	ADC140_GBADI dedicated to group B is generated at the end of group B scan

✓: available, ×: unavailable.

34.5 Event Link Function

34.5.1 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 and ADC140_CMPAI/ADC140_CMPBI interrupts cannot be used as event signals. For details, see Table 34.11, ADC14 interrupt sources and ELC events.

34.5.2 ADC14 Operation through an Event from the ELC

The ADC14 can start A/D conversion by the preset event specified in the ELSRn settings of the ELC as follows:

- Select the ELC_AD00 signal in the ELC.ELSR8 register
- Select the ELC_AD01 signal in the ELC.ELSR9 register.

If an ELC_AD00 or ELC_AD01 event occurs during A/D conversion, the event is disabled.

34.6 Selecting Reference Voltage

The ADC14 can select VREFH0 or AVCC0 as the high-potential reference voltage, and 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 ADHVREFCNT register description.

34.7 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 AN004 to AN006, AN009, AN010, AN017, AN019, and AN020, but A/D conversion of the internal reference voltage and the temperature sensor output is prohibited.

The A/D conversion procedure is as follows:

1. Set ADHVREFCNT.HVSEL[1:0] to 11b to discharge the high-potential reference voltage path in ADC14.
2. Wait for a 1 µs discharge period in software.
3. Set ADHVREFCNT.HVSEL[1:0] to 10b to select internal reference voltage as the high-potential reference voltage.

Note: The ADC14 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 after 1 µs.

4. Wait until the internal reference voltage is stabilized (for 5 µs) in software, and then perform A/D conversion.

Figure 34.30 shows a waveform for the procedure to select internal reference voltage as the high-potential reference voltage.

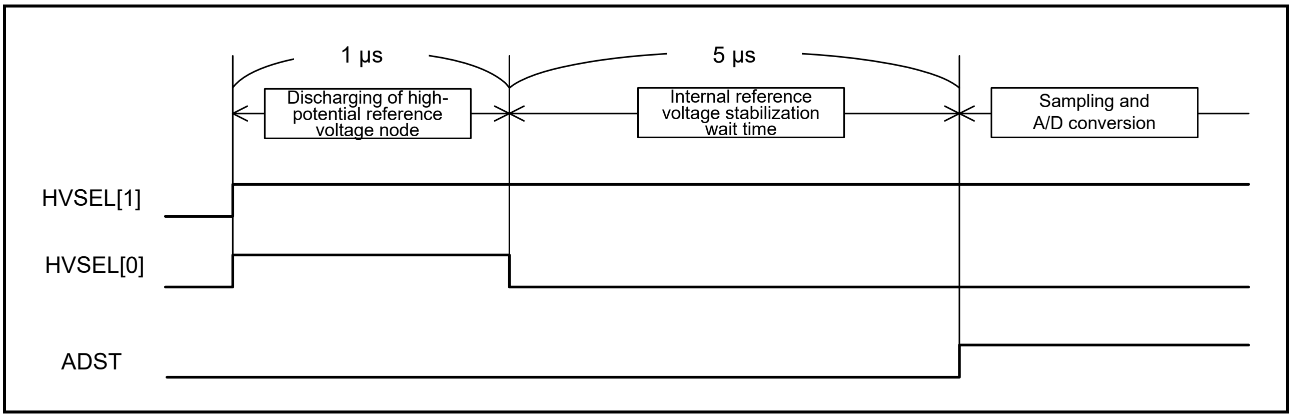

1 µs
5 µs
Discharging of high-potential reference voltage node
Internal reference voltage stabilization wait time
Sampling and A/D conversion
HVSEL[1]
HVSEL[0]
ADST

Figure 34.30 Procedure to select internal reference voltage as the high-potential reference voltage

34.8 Usage Notes

34.8.1 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 initially read might conflict with the subsequent A/D-converted value read. To prevent this, do not read the data registers in byte units.

34.8.2 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 34.31.

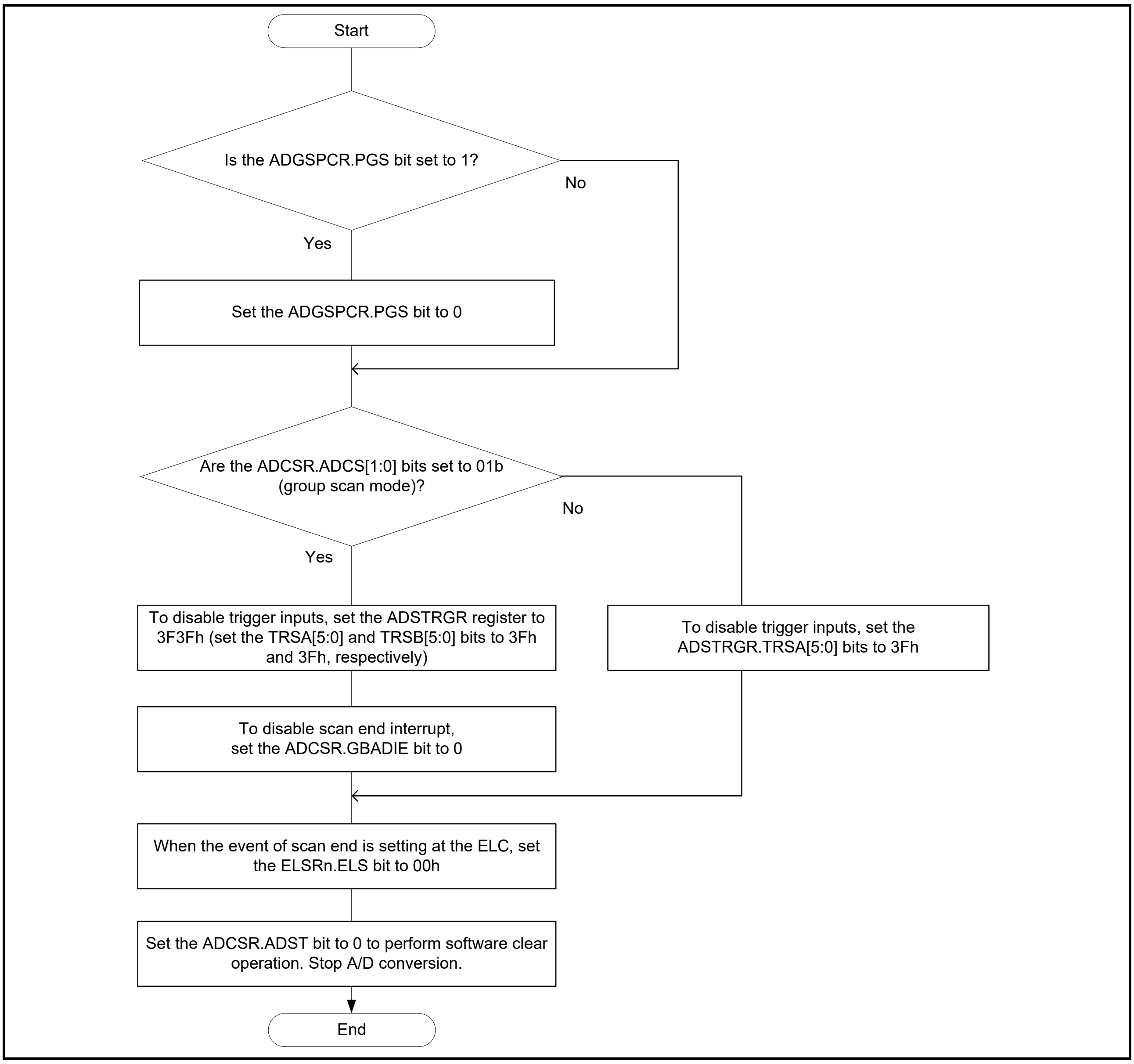

Start
Is the ADGSPCR.PGS bit set to 1?
No
Yes
Set the ADGSPCR.PGS bit to 0
Are the ADCSR.ADCS[1:0] bits set to 01b (group scan mode)?
No
Yes
To disable trigger inputs, set the ADSTRGR register to 3F3Fh (set the TRSA[5:0] and TRSB[5:0] bits to 3Fh and 3Fh, respectively)
To disable trigger inputs, set the ADSTRGR.TRSA[5:0] bits to 3Fh
To disable scan end interrupt, set the ADCSR.GBADIE bit to 0
When the event of scan end is setting at the ELC, set the ELSRn.ELS bit to 00h
Set the ADCSR.ADST bit to 0 to perform software clear operation. Stop A/D conversion.
End

Figure 34.31 Procedure for clearing the ADCSR.ADST bit through software

34.8.3 A/D Conversion Restarting Timing and Termination Timing

A maximum of 6 ADCLK cycles is required for the idle analog unit in the ADC14 to restart on setting the ADCSR.ADST bit to 1. A maximum of 3 ADCLK cycles is required for the operating analog unit in the ADC14 to terminate on setting the ADCSR.ADST bit to 0.

34.8.4 Restrictions on Scan End Interrupt Handling

When scanning the same analog input twice using any trigger, the first A/D-converted data is overwritten with the second A/D-converted data. This occurs when the CPU does not complete the reading of the A/D-converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated.

34.8.5 Settings for the Module-Stop State

The Module Stop Control Register can enable or disable the ADC14 operation. The ADC14 is initially stopped after a reset. 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.

34.8.6 Restrictions on Entering Low Power States

Before entering the module-stop state or Software Standby mode, be sure to stop A/D conversion. Set the ADCSR.ADST bit to 0 and secure a period of time until the analog unit of the ADC14 stops. Follow the procedure shown in Figure 34.31 to clear the ADCSR.ADST bit through software. Then, wait for 3 clock cycles of ADCLK before entering the module-stop state or Software Standby mode.

34.8.7 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 ADC14. This error arises because an erroneous voltage is input to the analog input pins because of 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 using the following formula.

Maximum error in absolute accuracy (LSB) = 4095 × Rs / (Rs + Rp)

Only use disconnection detection assistance after thorough evaluation.

34.8.8 ADHSC Bit Rewriting Procedure

Before changing the A/D Conversion Select bit (ADCSR.ADHSC) from 0 to 1 or from 1 to 0, the ADC14 must be in the standby state. Use the following procedure 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.

The procedure to modify the ADCSR.ADHSC bit is as follows:

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 set the Sleep bit (ADHVREFCNT.ADSLP) to 0.

Note: Setting the Sleep bit (ADHVREFCNT.ADSLP) to 1 is prohibited except when modifying the A/D Conversion Select bit (ADCSR.ADHSC).

Note: Do not reset the Sleep bit when the A/D Conversion Select bit (ADCSR.ADHSC) is 1. After this bit is set to 0 or the operating mode transitions to module-stop mode, reset the Sleep bit based on the ADCSR.ADHSC bit rewriting procedure.

34.8.9 Notes on Operating Modes and Status Bits

Initialize or set again individually, if necessary, the voltage values in self-diagnosis, the determination of the first scan or second scan in double trigger mode, the data buffer pointer, and status monitor in the compare function.

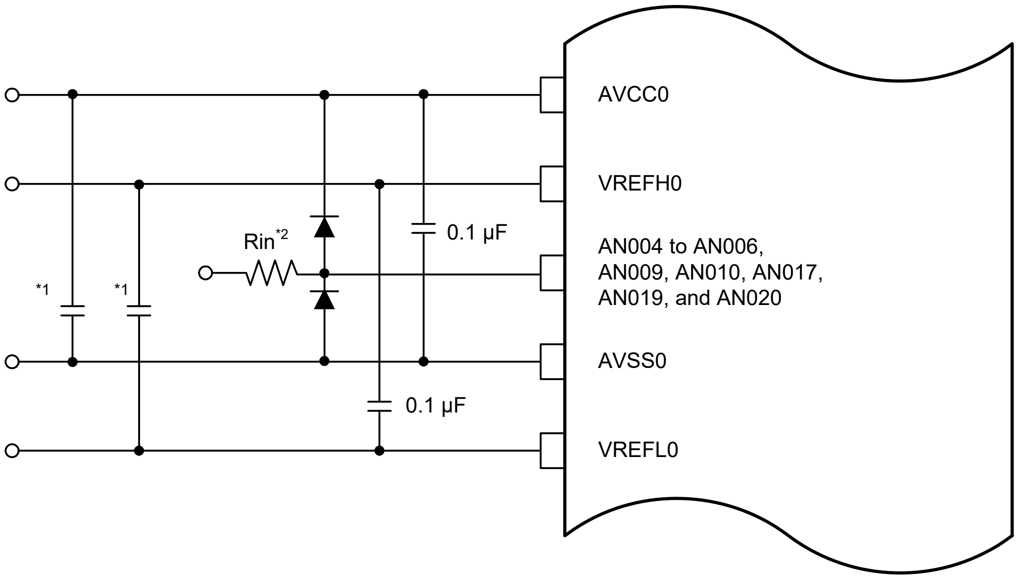
- Select the voltage values in self-diagnosis (ADCER.DIAGVAL[1:0]) after setting ADCER.DIAGLD to 1
- Double trigger mode operates as the first scan after setting ADCSR.DBLE to 1 from 0
- The status monitor bits (MONCMPA, MONCMPB, MONCOMB) in the compare function are initialized after the ADCMPCR.CMPAE and ADCMPCR.CMPBE bits are set to 0.

34.8.10 Notes on Board Design

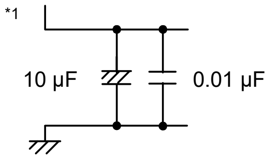
The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or be placed near each other. If these rules are not followed, noise can occur on analog signals and A/D conversion accuracy is affected. The analog input pins (AN004 to AN006, AN009, AN010, AN017, AN019, and AN020), reference power supply pin (VREFH0), reference ground pin (VREFL0), and analog power supply (AVCC0) should be separated from digital circuits using the analog ground (AVSS0). The analog ground (AVSS0) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).

34.8.11 Notes on Noise Reduction

To prevent the analog input pins (AN004 to AN006, AN009, AN010, AN017, AN019, and AN020) from being destroyed by abnormal voltage such as excessive surges, insert a capacitor between AVCC0 and AVSS0 and between VREFH0 and VREFL0. Additionally, connect a protection circuit to protect the analog input pins (AN004 to AN006, AN009, AN010, AN017, AN019, and AN020) as shown in Figure 34.32.



Note 1. The values shown here are reference values.



Note 2. Rin: Signal source impedance.

Figure 34.32 Example protection circuit for analog inputs

34.8.12 Port Settings when Using the 14-bit A/D Converter Input

When using the high precision channels, do not use PORT0 as general I/O, IRQ2, IRQ3 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.

34.8.13 Relationship between the ADC14, OPAMP, and ACMPLP

Table 34.12 lists the A/D conversion targets that should not be selected as an OPAMP and ACMPLP input during A/D conversion.

Table 34.12 OPAMP and ACMPLP pins that should not be selected during A/D conversion

Target of 14-bit A/D conversion	OPAMP	ACMPLP
AN005	AMP2-	-
AN006	AMP2+	-
AN017	-	CMPIN1
AN019	-	CMPREF1
AN020	-	CMPIN1

34.8.14 Notes on Canceling Software Standby Mode

After transitioning from Software Standby mode to Normal mode, wait for 1 µs before starting A/D conversion.

35. 12-Bit D/A Converter (DAC12)

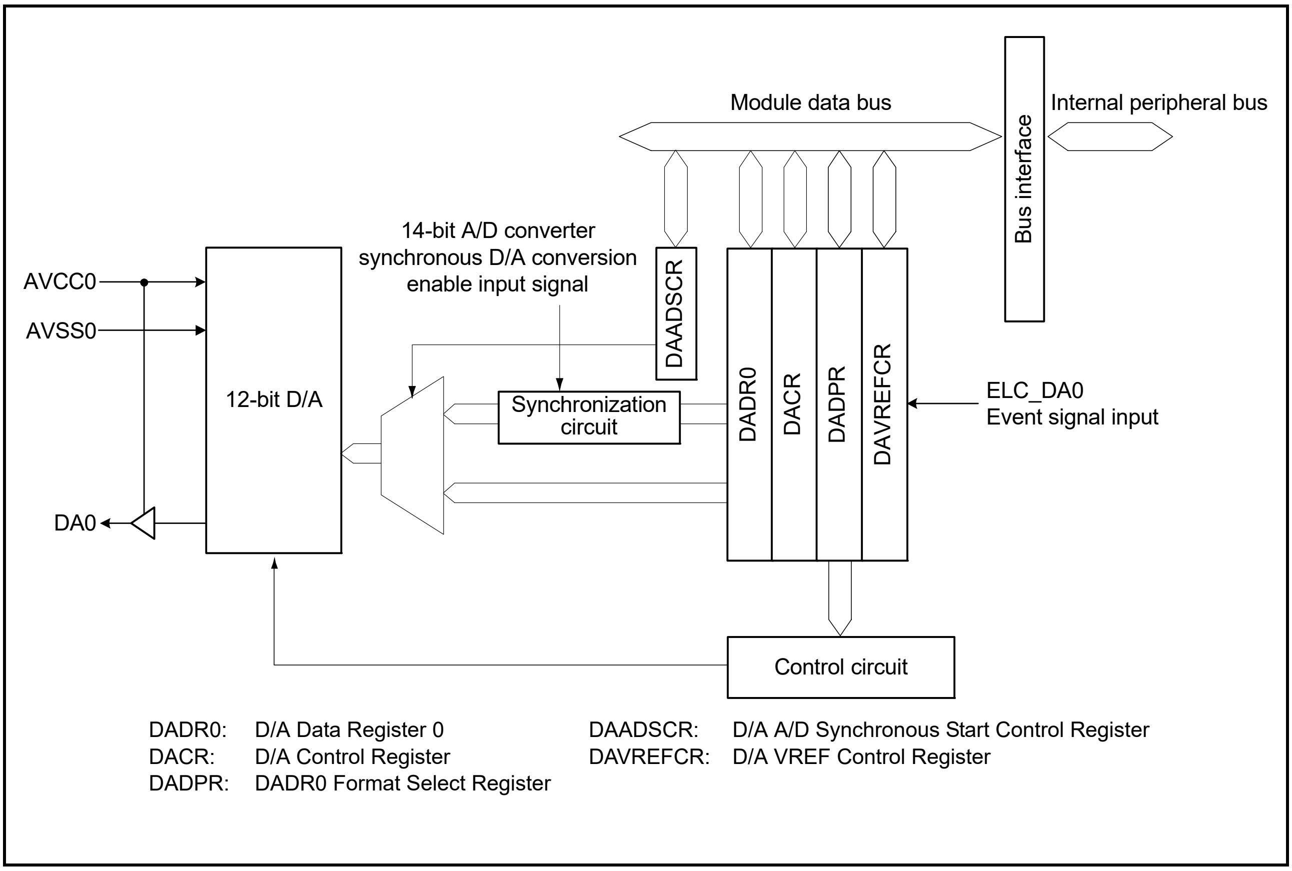
35.1 Overview

The MCU provides a 12-bit D/A Converter (DAC12).

Table 35.1 lists the DAC12 specifications, Figure 35.1 shows the block diagram, and Table 35.2 lists the I/O pins.

Table 35.1 DAC12 specifications

Parameter	Specifications
Resolution	12 bits
Output channels	1 channel
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 conversion can be started on input of an event signal


Module data bus
Internal peripheral bus
Bus interface
14-bit A/D converter synchronous D/A conversion enable input signal
AVCC0
AVSS0
12-bit D/A
DAADSCR
Synchronization circuit
DADR0
DACR
DADPR
DAVREFCR
ELC_DA0
Event signal input
DA0
Control circuit

DADR0: D/A Data Register 0
DACR: D/A Control Register
DADPR: DADR0 Format Select Register
DAADSCR: D/A A/D Synchronous Start Control Register
DAVREFCR: D/A VREF Control Register

Figure 35.1 DAC12 block diagram

Table 35.2 DAC12 pin configuration (1 of 2)

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.

Table 35.2 DAC12 pin configuration (2 of 2)

Pin name	I/O	Function
DA0	Output	Channel 0 analog output pin

35.2 Register Descriptions

35.2.1 D/A Data Register 0 (DADR0)

Address(es): DAC12.DADR0 4005 E000h

	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 DADR0 register is a 16-bit read/write register that stores data for D/A conversion. When an analog output is enabled, the values in DADR0 are converted and output to the analog output pins.

The 12-bit data can be formatted as left- or right-justified by setting the DADPR.DPSEL bit. In right-justified format (DADPR.DPSEL = 0), the lower 12 bits, [11:0], are valid. In left-justified format (DADPR.DPSEL = 1), the upper 12 bits, [15:4], are valid.

35.2.2 D/A Control Register (DACR)

Address(es): DAC12.DACR 4005 E004h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	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 of channel 0 (DA0).	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Only set this register while the ADC14 is halted when the DAADSCR.DAADST bit is 1 (interference reduction between D/A and A/D conversions is enabled). Only set DACR while the ADCSR.ADST bit is 0, and after selecting the software trigger as the ADC14 trigger to securely stop the ADC14.

DAOE0 bit (D/A Output Enable 0)

The DAOE0 bit controls D/A conversion and analog output.

When interference reduction between D/A and A/D conversions is enabled (DAADSCR.DAADST = 1), only set the DAOE0 bit while the ADC14 is halted (ADCSR.ADST = 0), and the software trigger is selected as the ADC14 trigger to securely stop the ADC14.

The event link function can be used to set the DAOE0 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.

35.2.3 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	DADR0 Format Select	0: Right-justified format 1: Left-justified format.	R/W

35.2.4 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 operation with ADC14 operation (disable interference reduction between D/A and A/D conversion) 1: Synchronize DAC12 operation 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 on or off the synchronization of the D/A conversion with the synchronous D/A conversion enable input signal from the ADC14 trigger.

Only set this register while the ADC14 is halted (ADCSR.ADST = 0) and the software trigger is selected as the ADC14 trigger.

Select unit 1 as the target ADC14 unit before setting the DAADST bit to 1.

DAADST bit (D/A A/D Synchronous Conversion)

Setting the DAADST bit to 0 allows the DADR0 register value to be converted into analog data at any time. Setting the DAADST bit to 1 allows synchronous D/A conversion with the synchronous D/A conversion enable input signal from ADC14. When the DADR0 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 bit is set to 0) and the software trigger is selected as the ADC14 trigger to securely stop the ADC14. The event link function cannot be used when the DAADST bit is set to 1. Stop the event link function by setting the ELSR12 register of the ELC. The setting of the DAADST bit is shared by channels 0 and 1 of the DAC12.

35.2.5 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 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. When changing the value of these bits, write 000b to these 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 register 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 35.3.2, 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 turn off even when the MCU enters Software Standby mode with the internal reference voltage selected.

35.3 Operation

When the DAOE0 bit 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 35.2 shows the timing of this operation.

1. Set the data for D/A conversion in the DADPR.DPSEL bit and the DADR0 register.
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 (reference) is expressed by the following formula:

$$\frac{\text{Setting value of DADR0}}{4096} \times \text{Reference voltage}$$

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 than one A/D conversion time might be required.
4. To disable analog input, set the DAOE0 bit to 0.

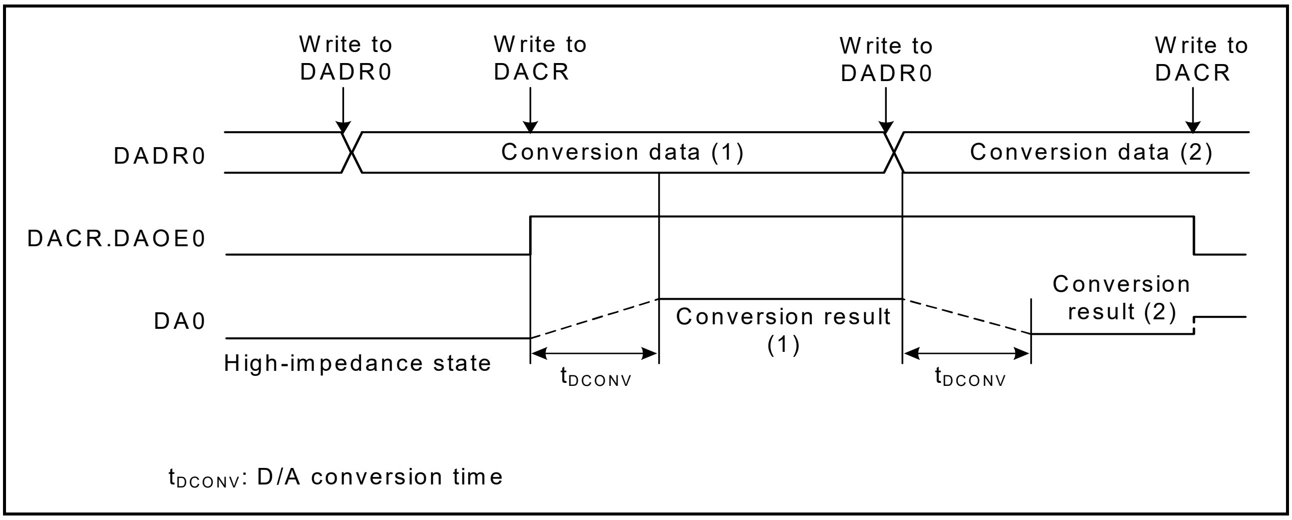

Write to DADR0
Write to DACR
Write to DADR0
Write to DACR
DADR0
Conversion data (1)
Conversion data (2)
DACR.DAOE0
DA0
Conversion result (1)
Conversion result (2)
High-impedance state
t_{DCONV}
t_{DCONV}
t_{DCONV}: D/A conversion time

Figure 35.2 Example DAC12 operation

35.3.1 Reducing 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 ADC14 operation.

While the DAADSCR.DAADST bit is 1, D/A conversion does not start immediately on updating the DADR0 register. Instead:

- If the DADR0 register data is modified while the ADC14 is halted, D/A conversion starts in 1 PCLKB cycle
- If the DADR0 register data is modified when the ADC14 is in progress, D/A conversion starts on A/D conversion completion. Therefore, it takes up to one A/D conversion time period for the DADR0 register data update to reflect as the D/A conversion circuit output. Until the D/A conversion completes, the DADR0 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 DADR0 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. Figure 35.3 shows the timing of this operation.

To perform D/A conversion in synchronization 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 1 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.

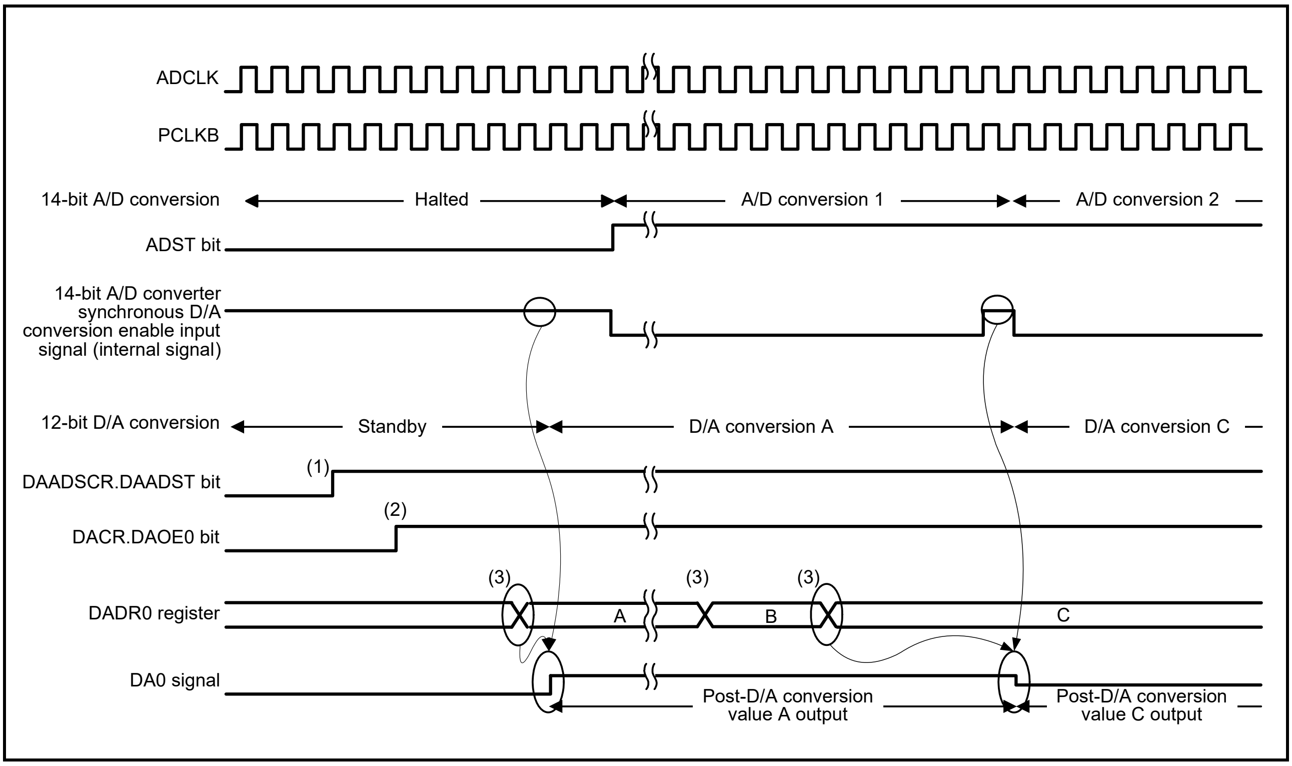

ADCLK
PCLKB
14-bit A/D conversion
Halted
A/D conversion 1
A/D conversion 2
ADST bit
14-bit A/D converter synchronous D/A conversion enable input signal (internal signal)
12-bit D/A conversion
Standby
D/A conversion A
D/A conversion C
DAADSCR.DAADST bit
(1)
DACR.DAOE0 bit
(2)
(3)
(3)
(3)
DADR0 register
A
B
C
DA0 signal
Post-D/A conversion value A output
Post-D/A conversion value C output

Figure 35.3 Example of 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 1 ADCLK output cycle between A/D conversion 1 and A/D conversion 2, as shown in Figure 35.4. In this case, post-D/A conversion value A is continuously output as the DA0 signal.

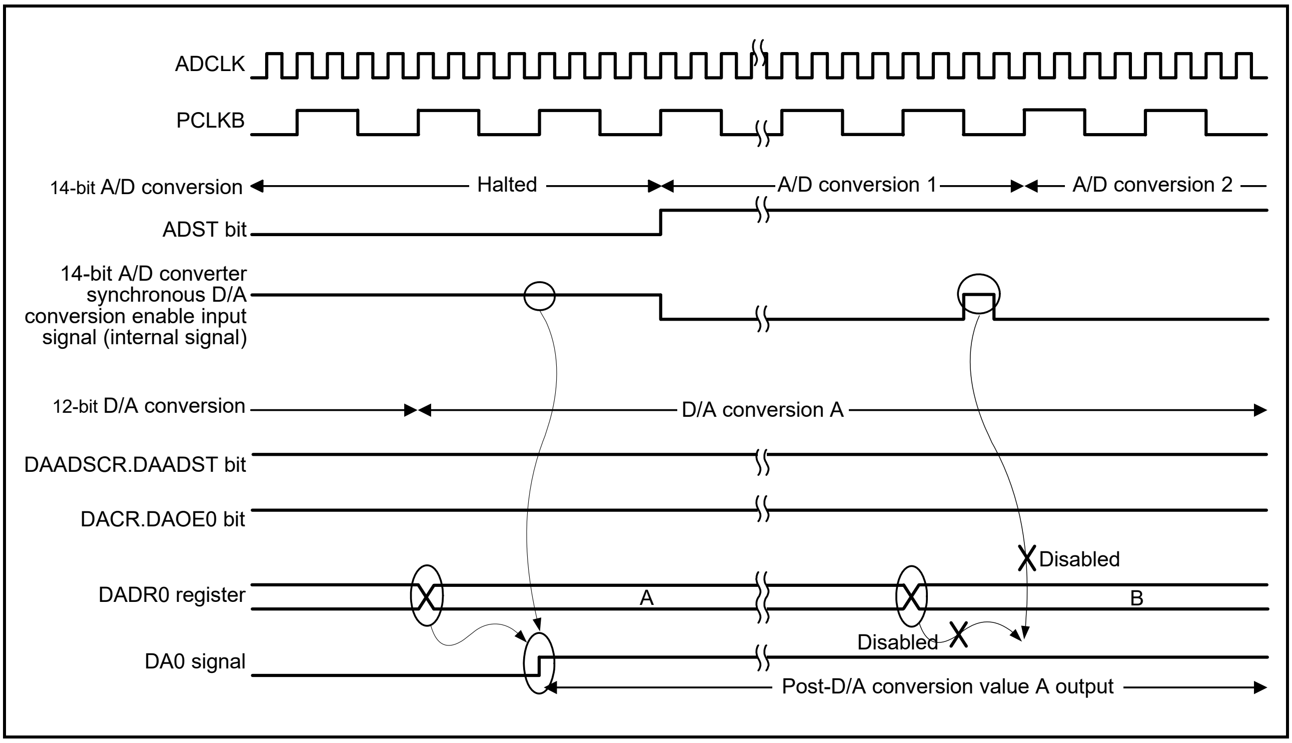

ADCLK
PCLKB
14-bit A/D conversion
Halted
A/D conversion 1
A/D conversion 2
ADST bit
14-bit A/D converter synchronous D/A conversion enable input signal (internal signal)
12-bit D/A conversion
D/A conversion A
DAADSCR.DAADST bit
DACR.DAOE0 bit
Disabled
DADR0 register
A
B
Disabled
DA0 signal
Post-D/A conversion value A output

Figure 35.4 Example when DAC12 cannot capture ADC14 synchronous D/A conversion enable input signal

35.3.2 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 shows the discharging procedure:

1. Write 000b to the REF[2:0] bits.
2. Set the DADR0 register to 0000h.
3. Keep the state of (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.DAOE0 bit to 1 and wait 5 µs, the stabilization wait time of the internal reference voltage.
6. Write data to the DADR0 register and start D/A conversion.

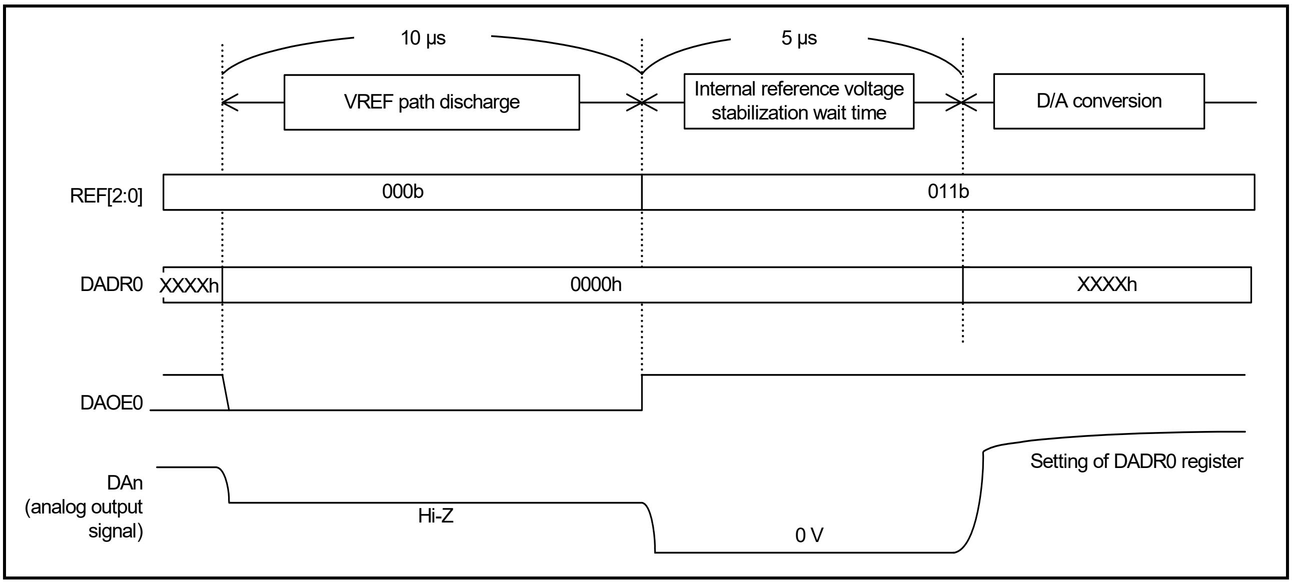


Figure 35.5 Procedure for selecting the internal reference voltage as the reference voltage

35.4 Event Link Operation Setting Procedure

To set up an event link operation (for DA0):

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 enables event link operation for all modules with the event link function selected.
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 starts on channel 0.
5. Set the ELSR12.ELS[7:0] bits to 00h 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.

35.5 Usage Notes on Event Link Operation

- When the event specified by the ELC_DA0 event signal is generated while a write cycle to the DACR.DAOE0 bit is processed, 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.

35.6 Usage Notes

35.6.1 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.

35.6.2 DAC12 Operation in Module-Stop State

When the MCU enters the module-stop state with D/A conversion enabled, the D/A output is saved, 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.DAOE0 bit to 0.

35.6.3 DAC12 Operation in Software Standby Mode

When the MCU enters Software Standby mode with D/A conversion enabled, the D/A output is saved, 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.DAOE0 bit to 0.

35.6.4 Restriction 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.

36. Temperature Sensor (TSN)

36.1 Overview

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 also be used by the end application.

Table 36.1 lists the temperature sensor specifications, and Figure 36.1 shows a block diagram.

Table 36.1 Temperature sensor specifications

Parameter	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the ADC14

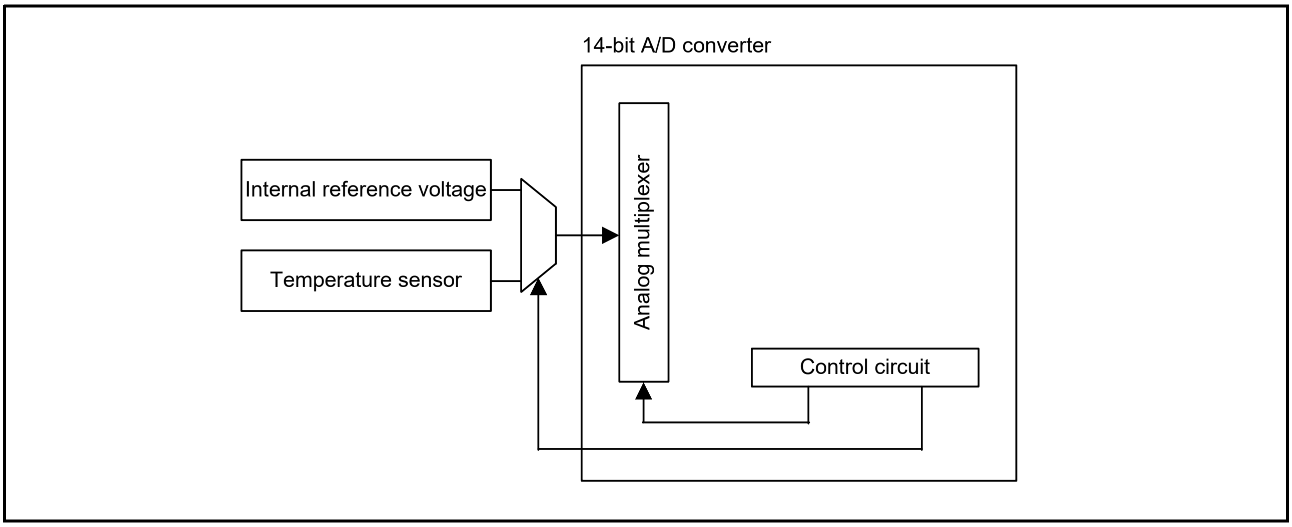


Figure 36.1 Temperature sensor block diagram

36.2 Register Descriptions

36.2.1 Temperature Sensor Calibration Data Register H (TSCDRH)

Address(es): TSN.TSCDRH 407E C229h

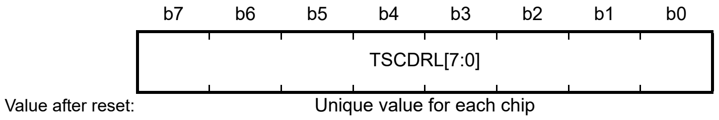
b7	b6	b5	b4	b3	b2	b1	b0
TSCDRH[7:0]							

Value after reset: Unique value for each chip

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

36.2.2 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

The TSCDRH and TSCDRL registers store temperature sensor calibration data measured for each chip at factory shipment.

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.

36.3 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.

36.3.1 Preparation for Using Temperature Sensor

The temperature (T) is proportional to the sensor voltage output (Vs), so temperature is calculated with the following formula:

$T = (Vs - V1)/\text{slope} + T1$

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. Use the ADC14 to measure the voltage V2 output by the temperature sensor at a different temperature T2. Obtain the temperature gradient (slope = (V2 - V1)/(T2 - T1)) from these results.
3. 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 48, Electrical Characteristics, only one experimental measurement is required to determine V1 and T1. However, this method gives less accurate temperature results than measurement at two points.

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 = TSCDRH register value << 8 + TSCDRL register value

V1 is calculated from CAL125 as follows:

$V1 = 3.3 \times CAL125/4096$ [V]

Using this value, the measured temperature can be calculated according to the following formula:

$T = (Vs - V1)/Slope + 125$ [°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 36.2 shows the error in the measured temperature. The variation range is 3σ.

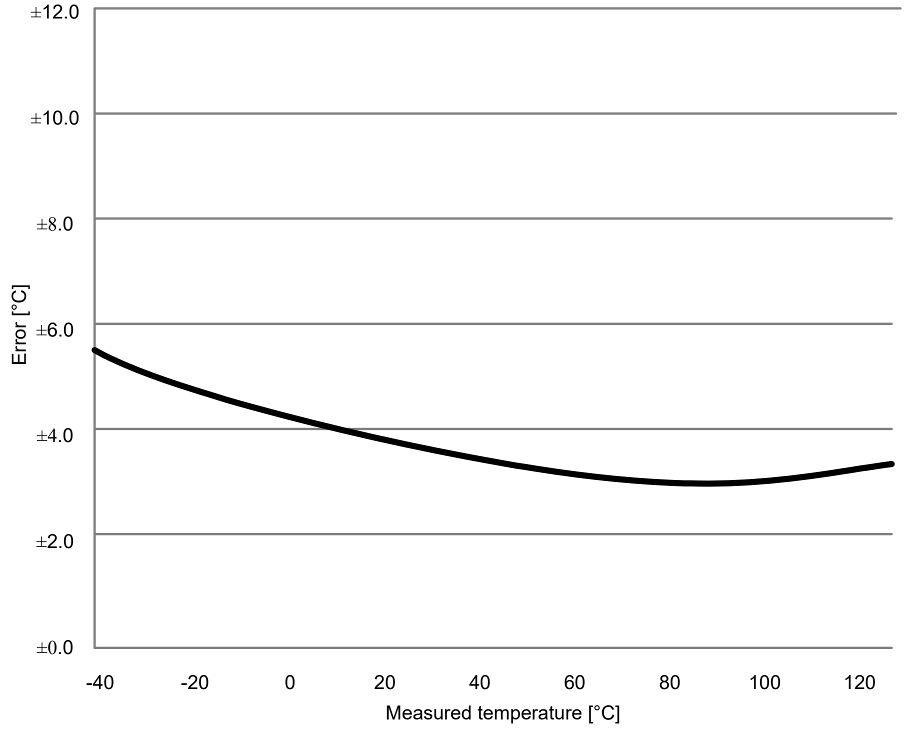


Figure 36.2 Error in the measured temperature (designed values)

36.3.2 Procedure for Using the Temperature Sensor

For details, see section 34, 14-Bit A/D Converter (ADC14).

37. Operational Amplifier (OPAMP)

37.1 Overview

Operational amplifiers can be used to amplify small analog input voltages and output the amplified voltages. The MCU has an operational amplifier unit with two input pins and one output pin.

The operational amplifiers have the following functions:

- The output signal from a unit can be used for the input signal 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.

Figure 37.1 shows a block diagram of the operational amplifier, and Table 37.1 lists the unit configuration.

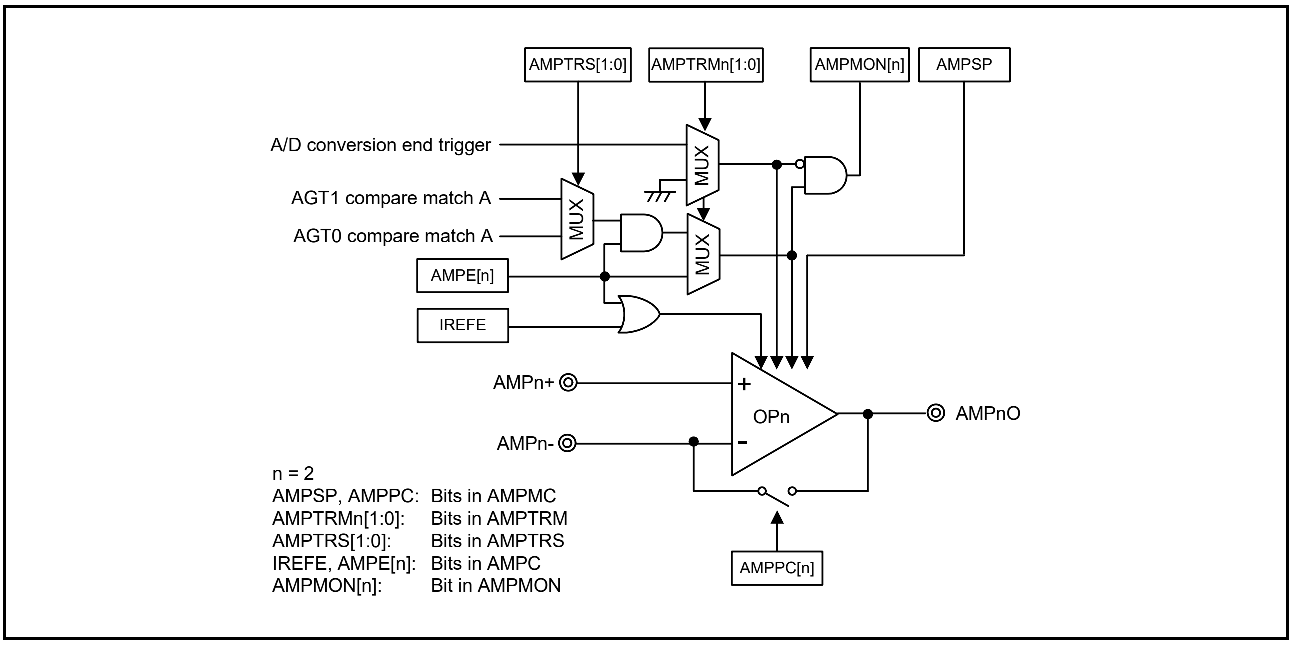

AMPTRS[1:0]
AMPTRMn[1:0]
AMPMON[n]
AMPSP
A/D conversion end trigger
MUX
AGT1 compare match A
AGT0 compare match A
MUX
MUX
AMPE[n]
IREFE
AMPn+
+
OPn
-
AMPnO
AMPn-
AMPPC[n]
n = 2
AMPSP, AMPPC: Bits in AMPMC
AMPTRMn[1:0]: Bits in AMPTRM
AMPTRS[1:0]: Bits in AMPTRS
IREFE, AMPE[n]: Bits in AMPC
AMPMON[n]: Bit in AMPMON

Figure 37.1 Operational amplifier block diagram

Table 37.1 OPAMP unit configuration

Unit	I/O pin	I/O	Function
Unit 2 (Operational amplifier 2)	AMP2+, AMP2-	Input	Input pin of operational amplifier 2 (+, -)
	AMP2O	Output	Output pin of operational amplifier 2

37.2 Register Descriptions

37.2.1 Operational Amplifier Mode Control Register (AMPMC)

Address(es): OPAMP.AMPMC 4008 6008h

	b7	b6	b5	b4	b3	b2	b1	b0
	AMPSP	—	—	—	—	AMPPC [2]	—	—
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	AMPPC[2]	Operational Amplifier Precharge Control	0: Precharging of operational amplifier 2 is stopped. 1: Precharging of operational amplifier 2 is enabled.	R/W
b6 to b3	—	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 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.

37.2.2 Operational Amplifier Trigger Mode Control Register (AMPTRM)

Address(es): OPAMP.AMPTRM 4008 6009h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AMPTRM2[1:0]		—	—	—	—
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
b5, b4	AMPTRM2[1:0]	OPAMP Function Activation/ Stop Trigger Control *2	AMPTRM2[1] AMPTRM2[0] 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
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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 OPAMP Operation Control bit to be activated to 1 (operational amplifier wait state is enabled).

Note 2. When changing the set values of AMPTRM2[1:0], make sure that the AMPE[2] bit in the AMPC register is 0 (operation amplifier is stopped).

37.2.3 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 2: Operational amplifier activation trigger 2 0 1: Operational amplifier 2: Operational amplifier activation trigger 1 1 0: Setting prohibited 1 1: Operational amplifier 2: Operational amplifier activation trigger 0	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.

37.2.4 Operational Amplifier Control Register (AMPC)

Address(es): OPAMP.AMPC 4008 600Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	IREFE	—	—	—	—	AMPE[2]	—	—
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	AMPE[2]	OPAMP Operation Control	0: Operation amplifier 2 is stopped 1: Software trigger mode: Operation of operational amplifier 2 is enabled.*1 Activation trigger mode or activation and A/D trigger mode: Wait until AGT is enabled.	R/W
b6 to b3	—	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 37.2 shows the operational amplifier activation triggers associated with events.

Table 37.2 Operational amplifier activation triggers associated with 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

37.2.5 Operational Amplifier Monitor Register (AMPMON)

Address(es): OPAMP.AMPMON 4008 600Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	AMPMON[2]	—	—
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	R
b2	AMPMON[2]	Operational Amplifier 2 Status	0: Operational amplifier 2 is stopped 1: Operational amplifier 2 is operating.	R
b7 to b3	—	Reserved	These bits are read as 0	R

Note: This register is used to asynchronously reflect whether each operational amplifier is operating or 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 or stop the operational amplifier can be estimated, such as for checking normal operation. In this case, read this register after 1 CPU/peripheral clock cycle when the associated 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.

37.3 Operation

37.3.1 State Transitions

Figure 37.2 shows state transitions when the operational amplifier and reference current circuit are activated or stopped using the operational amplifier control circuit.

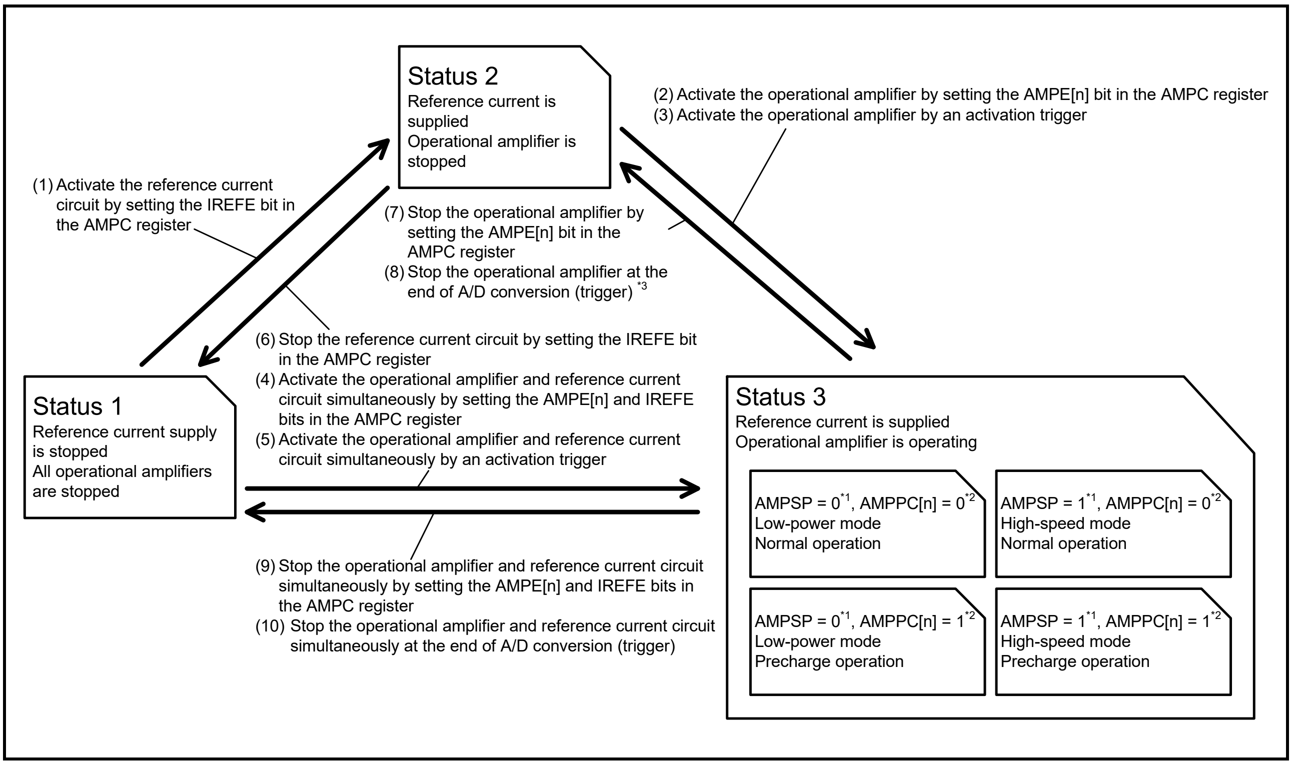

Status 2
Reference current is supplied
Operational amplifier is stopped
(2) Activate the operational amplifier by setting the AMPE[n] bit in the AMPC register
(3) Activate the operational amplifier by an activation trigger
(1) Activate the reference current circuit by setting the IREFE bit in the AMPC register
(7) Stop the operational amplifier by setting the AMPE[n] bit in the AMPC register
(8) Stop the operational amplifier at the end of A/D conversion (trigger) *3
(6) Stop the reference current circuit by setting the IREFE bit in the AMPC register
(4) Activate the operational amplifier and reference current circuit simultaneously by setting the AMPE[n] and IREFE bits in the AMPC register
(5) Activate the operational amplifier and reference current circuit simultaneously by an activation trigger
Status 1
Reference current supply is stopped
All operational amplifiers are stopped
Status 3
Reference current is supplied
Operational amplifier is operating
AMPSP = 0*1, AMPPC[n] = 0*2
Low-power mode
Normal operation
AMPSP = 1*1, AMPPC[n] = 0*2
High-speed mode
Normal operation
AMPSP = 0*1, AMPPC[n] = 1*2
Low-power mode
Precharge operation
AMPSP = 1*1, AMPPC[n] = 1*2
High-speed mode
Precharge operation
(9) Stop the operational amplifier and reference current circuit simultaneously by setting the AMPE[n] and IREFE bits in the AMPC register
(10) Stop the operational amplifier and reference current circuit simultaneously at the end of A/D conversion (trigger)

Figure 37.2 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[2] 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 required to preset operation of the reference current circuit to be enabled (operate the operational amplifier by status 2).

A stabilization wait time is required 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 48, 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 or stop only the operational amplifier that is preset to be used by setting the AMPTRM register.

37.3.2 Operational Amplifier Control Operation

Figure 37.3 to Figure 37.6 show operational amplifier control operation.

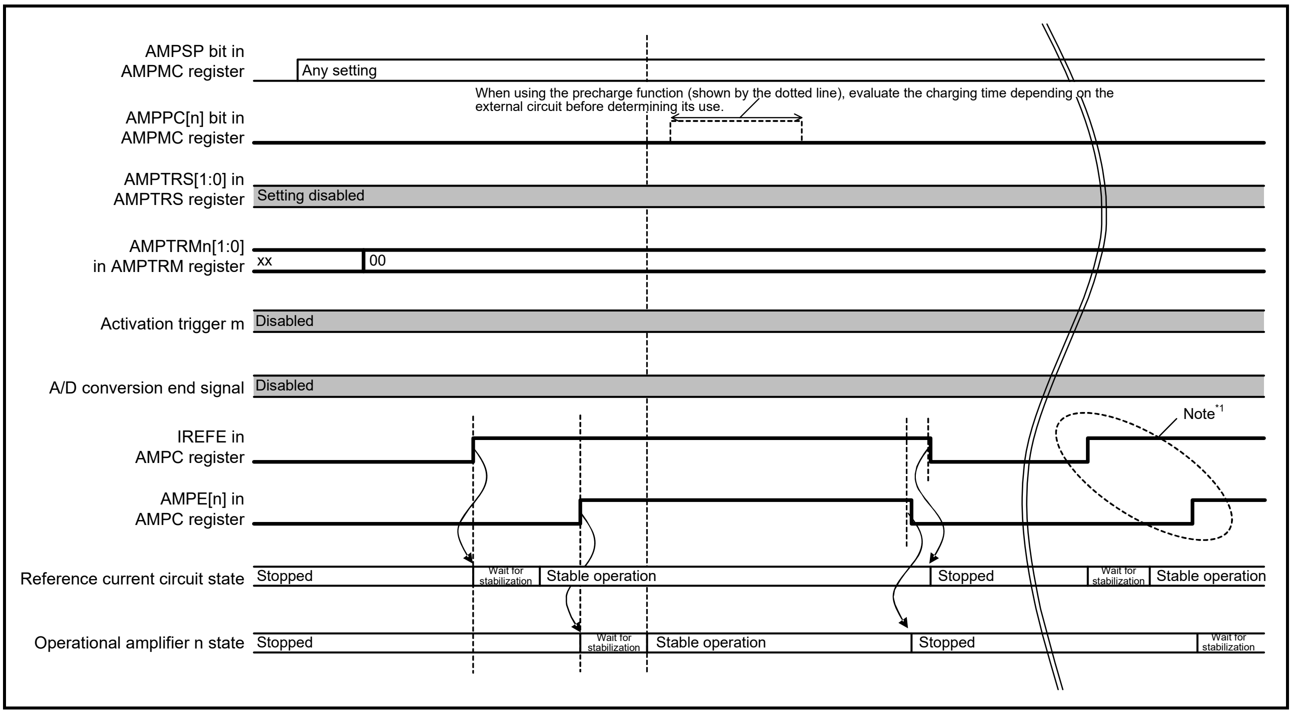

AMPSP bit in AMPMC register
Any setting
When using the precharge function (shown by the dotted line), evaluate the charging time depending on the external circuit before determining its use.
AMPPC[n] bit in AMPMC register
AMPTRS[1:0] in AMPTRS register
Setting disabled
AMPTRMn[1:0] in AMPTRM register
xx
00
Activation trigger m
Disabled
A/D conversion end signal
Disabled
IREFE in AMPC register
Note*1
AMPE[n] in AMPC register
Reference current circuit state
Stopped
Wait for stabilization
Stable operation
Stopped
Wait for stabilization
Stable operation
Operational amplifier n state
Stopped
Wait for stabilization
Stable operation
Stopped
Wait for stabilization

Figure 37.3 Operational amplifier control operation in software trigger mode used for control when the reference current circuit and operational amplifier are activated/stopped by software trigger mode

Note: n: Unit number (n = 2).
m: An activation trigger used to control operational amplifier unit n selected by the AMPTRS register.

Note 1. When operating or stopping the operational amplifier continuously, set the IREFE and AMPE[n] bits again as in the first setting after the operational amplifier is stopped.

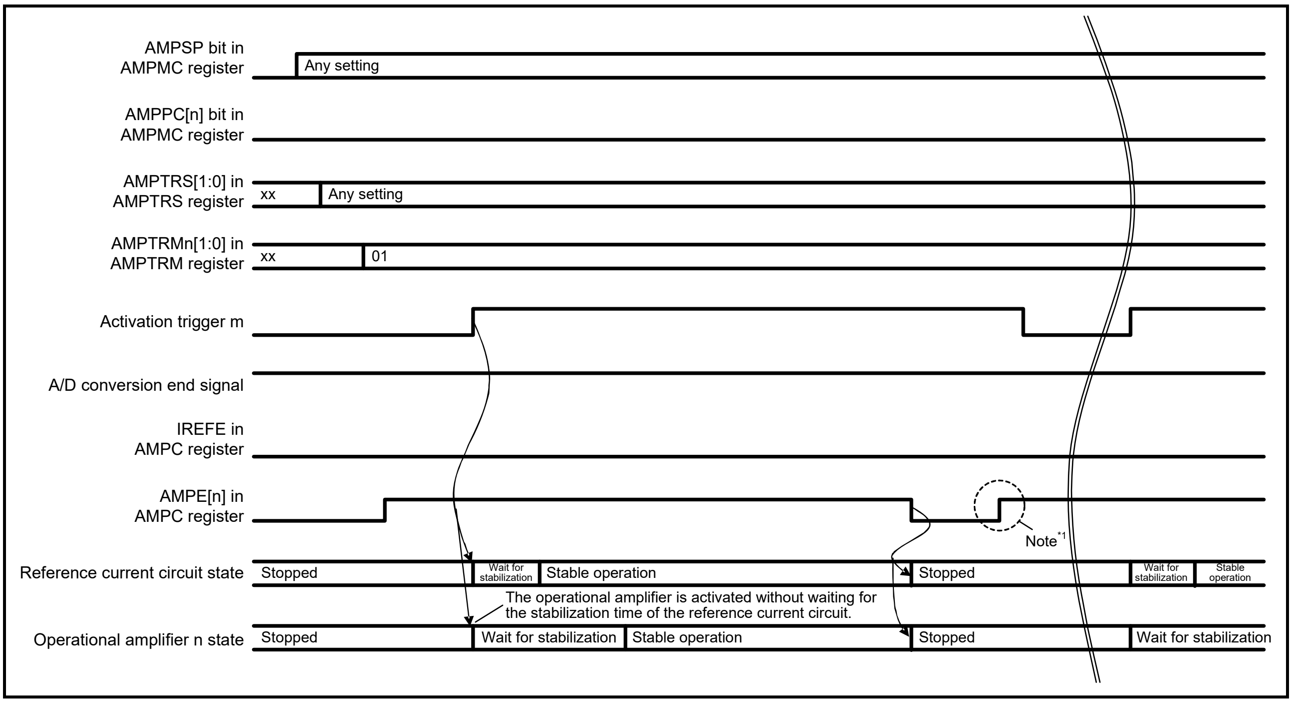

AMPSP bit in AMPMC register
Any setting
AMPPC[n] bit in AMPMC register
AMPTRS[1:0] in AMPTRS register
xx
Any setting
AMPTRMn[1:0] in AMPTRM register
xx
01
Activation trigger m
A/D conversion end signal
IREFE in AMPC register
AMPE[n] in AMPC register
Note*1
Reference current circuit state
Stopped
Wait for stabilization
Stable operation
Stopped
Wait for stabilization
Stable operation
The operational amplifier is activated without waiting for the stabilization time of the reference current circuit.
Operational amplifier n state
Stopped
Wait for stabilization
Stable operation
Stopped
Wait for stabilization

Figure 37.4 **Operational amplifier control operation when activation trigger mode is used for activation with the reference current circuit and operational amplifier activated by an activation trigger and stopped by setting the AMPC register**

Note: n: Unit number (n = 2).
m: An activation trigger used to control operational amplifier unit n selected in the AMPTRS register.
Set the AGT function.

Note 1. When operating or 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.

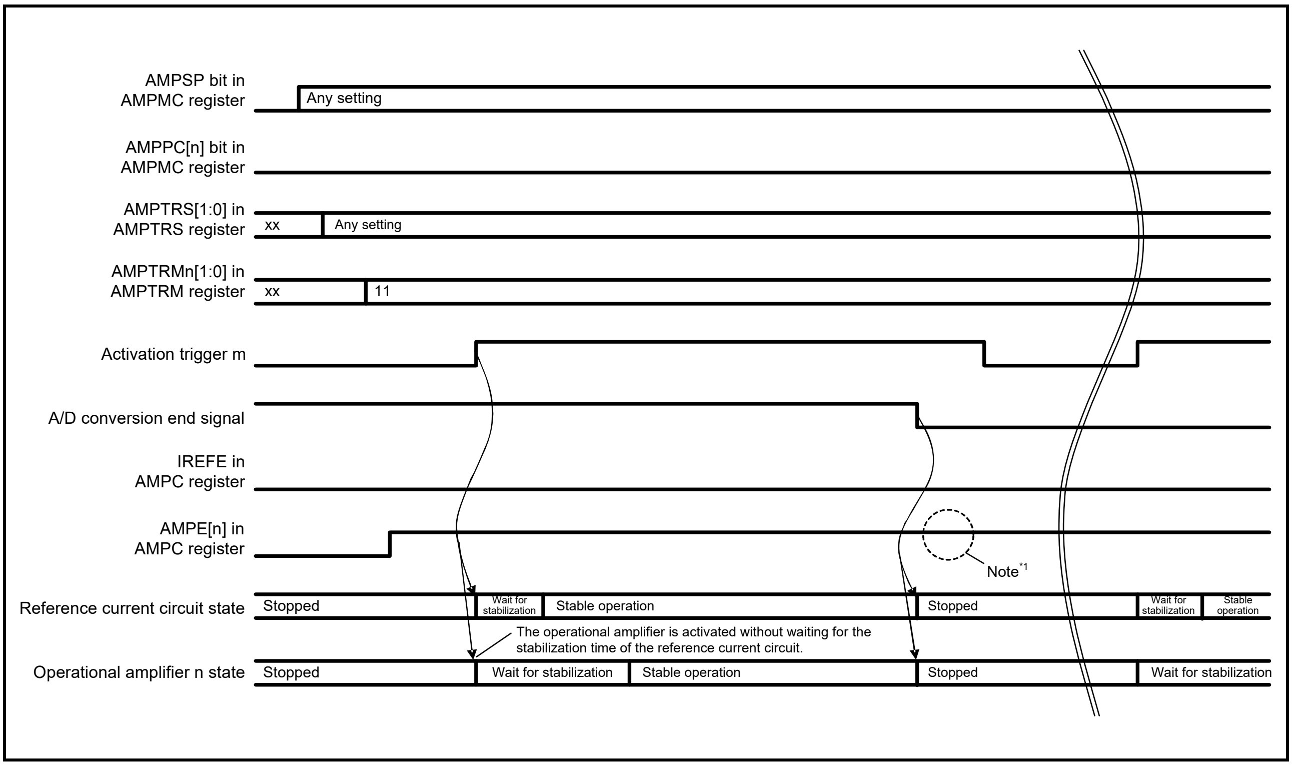

AMPSP bit in AMPMC register
Any setting
AMPPC[n] bit in AMPMC register
AMPTRS[1:0] in AMPTRS register
xx
Any setting
AMPTRMn[1:0] in AMPTRM register
xx
11
Activation trigger m
A/D conversion end signal
IREFE in AMPC register
AMPE[n] in AMPC register
Note*1
Reference current circuit state
Stopped
Wait for stabilization
Stable operation
Stopped
Wait for stabilization
Stable operation
The operational amplifier is activated without waiting for the stabilization time of the reference current circuit.
Operational amplifier n state
Stopped
Wait for stabilization
Stable operation
Stopped
Wait for stabilization

Figure 37.5 **Operational amplifier control operation in activation and A/D trigger mode (1) with the reference current circuit and operational amplifier activated by an activation trigger and stopped by an A/D conversion end (trigger)**

Note: n: Unit number (n = 2).
m: An activation trigger used to control operational amplifier unit n selected by the AMPTRS register.
Set the AGT function.

Note 1. When operating or stopping the operational amplifier continuously, it is not required to set the registers again because the operational amplifier waits for an activation trigger after it is stopped.

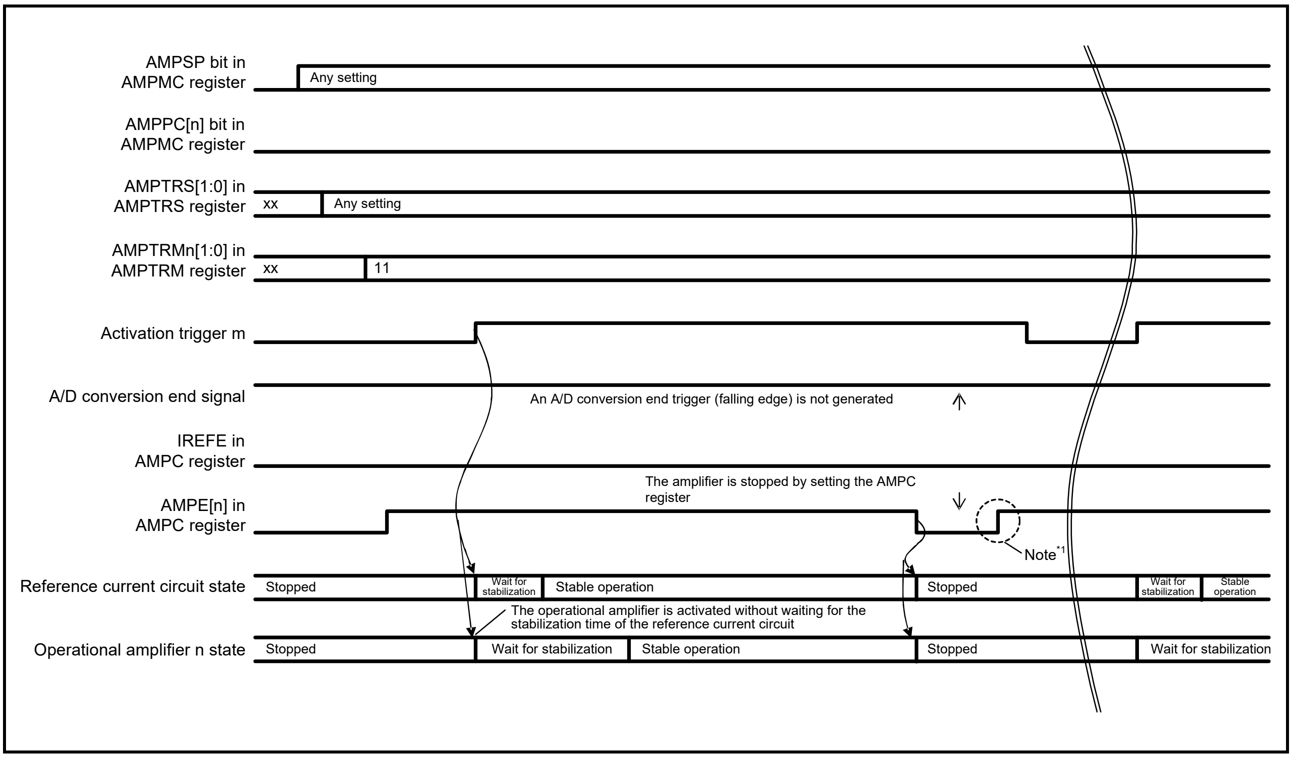

AMPSP bit in AMPMC register
Any setting
AMPPC[n] bit in AMPMC register
AMPTRS[1:0] in AMPTRS register
xx
Any setting
AMPTRMn[1:0] in AMPTRM register
xx
11
Activation trigger m
A/D conversion end signal
An A/D conversion end trigger (falling edge) is not generated
IREFE in AMPC register
The amplifier is stopped by setting the AMPC register
AMPE[n] in AMPC register
Note*1
Reference current circuit state
Stopped
Wait for stabilization
Stable operation
Stopped
Wait for stabilization
Stable operation
The operational amplifier is activated without waiting for the stabilization time of the reference current circuit
Operational amplifier n state
Stopped
Wait for stabilization
Stable operation
Stopped
Wait for stabilization

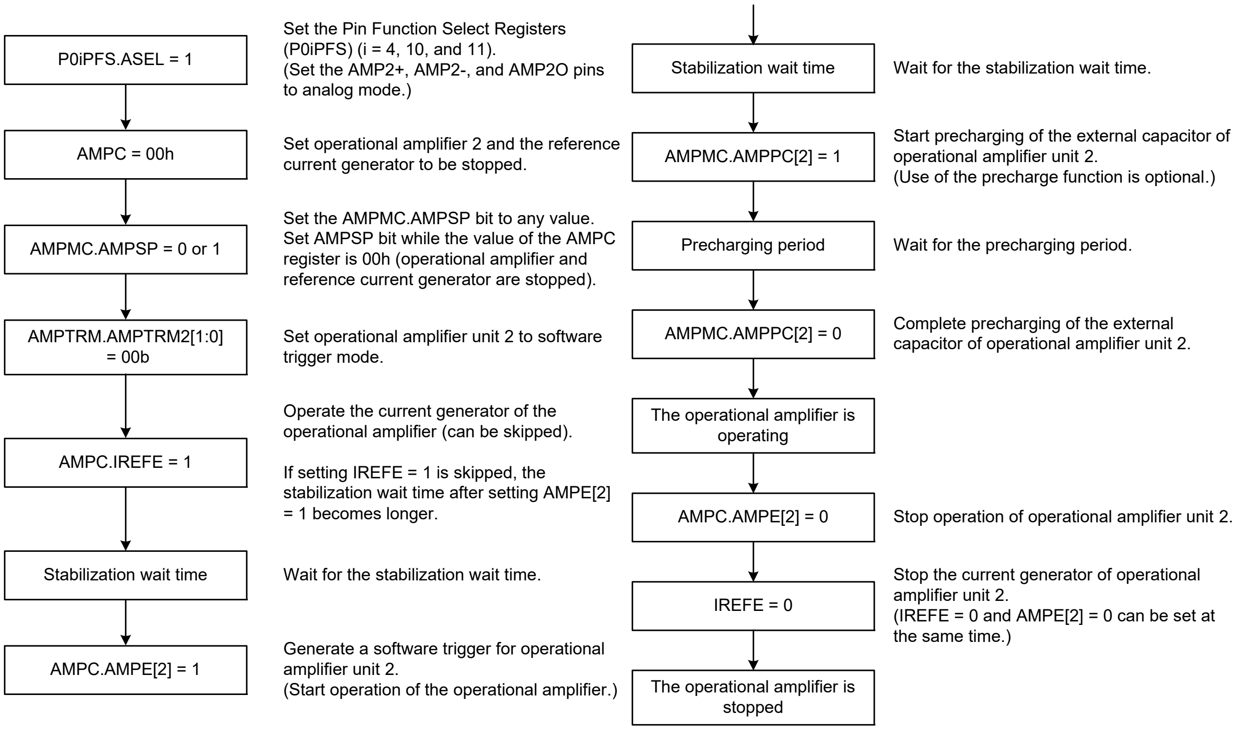
Figure 37.6 Operational amplifier control operation in activation and A/D trigger mode (2) with the reference current circuit and operational amplifier stopped by setting the AMPC register to be activated by an activation trigger and stopped by an A/D conversion end (trigger)

Note: n: Unit number (n = 2).
m: An activation trigger used to control operational amplifier unit n selected by the AMPTRS register.
Set the AGT function. See section 37.4, Software Trigger Mode for the procedure to activate the operational amplifier with an activation trigger.

Note 1. When operating or 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.

37.4 Software Trigger Mode

This section describes the procedure to activate and stop the operational amplifier using a software trigger. Figure 37.7 shows an example of each register setting.


P0iPFS.ASEL = 1
AMPC = 00h
AMPMC.AMPSP = 0 or 1
AMPTRM.AMPTRM2[1:0] = 00b
AMPC.IREFE = 1
Stabilization wait time
AMPC.AMPE[2] = 1
Stabilization wait time
AMPMC.AMPPC[2] = 1
Precharging period
AMPMC.AMPPC[2] = 0
The operational amplifier is operating
AMPC.AMPE[2] = 0
IREFE = 0
The operational amplifier is stopped

Set the Pin Function Select Registers (P0iPFS) (i = 4, 10, and 11). (Set the AMP2+, AMP2-, and AMP2O pins to analog mode.)

Set operational amplifier 2 and the reference current generator to be stopped.

Set the AMPMC.AMPSP bit to any value. Set AMPSP bit while the value of the AMPC register is 00h (operational amplifier and reference current generator are stopped).

Set operational amplifier unit 2 to software trigger mode.

Operate the current generator of the operational amplifier (can be skipped).

If setting IREFE = 1 is skipped, the stabilization wait time after setting AMPE[2] = 1 becomes longer.

Wait for the stabilization wait time.

Generate a software trigger for operational amplifier unit 2. (Start operation of the operational amplifier.)

Wait for the stabilization wait time.

Start precharging of the external capacitor of operational amplifier unit 2. (Use of the precharge function is optional.)

Wait for the precharging period.

Complete precharging of the external capacitor of operational amplifier unit 2.

Stop operation of operational amplifier unit 2.

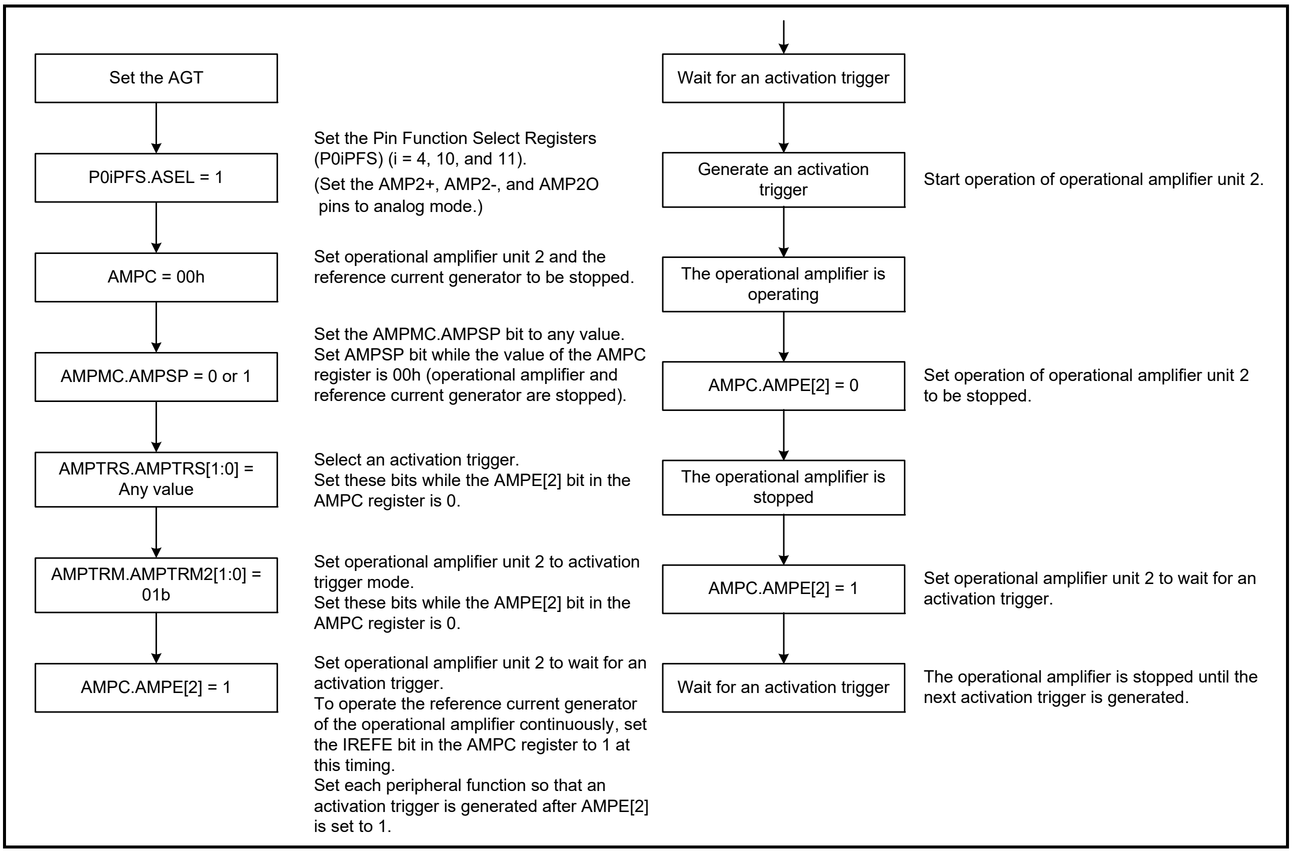
Stop the current generator of operational amplifier unit 2. (IREFE = 0 and AMPE[2] = 0 can be set at the same time.)

Note: For details on the stabilization wait time, see section 48, Electrical Characteristics

Figure 37.7 Procedure to start and stop OPAMP in software trigger mode

37.5 Activation Trigger Mode

This section describes the procedure to activate the operational amplifier using an activation trigger and to stop the amplifier with software. Figure 37.8 shows an example of each register setting.


Set the AGT
P0iPFS.ASEL = 1
AMPC = 00h
AMPMC.AMPSP = 0 or 1
AMPTRS.AMPTRS[1:0] = Any value
AMPTRM.AMPTRM2[1:0] = 01b
AMPC.AMPE[2] = 1
Wait for an activation trigger
Generate an activation trigger
The operational amplifier is operating
AMPC.AMPE[2] = 0
The operational amplifier is stopped
AMPC.AMPE[2] = 1
Wait for an activation trigger

Set the Pin Function Select Registers (P0iPFS) (i = 4, 10, and 11).
(Set the AMP2+, AMP2-, and AMP2O pins to analog mode.)

Set operational amplifier unit 2 and the reference current generator to be stopped.

Set the AMPMC.AMPSP bit to any value.
Set AMPSP bit while the value of the AMPC register is 00h (operational amplifier and reference current generator are stopped).

Select an activation trigger.
Set these bits while the AMPE[2] bit in the AMPC register is 0.

Set operational amplifier unit 2 to activation trigger mode.
Set these bits while the AMPE[2] bit in the AMPC register is 0.

Set operational amplifier unit 2 to wait for an activation trigger.
To operate the reference current generator of the operational amplifier continuously, set the IREFE bit in the AMPC register to 1 at this timing.
Set each peripheral function so that an activation trigger is generated after AMPE[2] is set to 1.

Start operation of operational amplifier unit 2.

Set operation of operational amplifier unit 2 to be stopped.

Set operational amplifier unit 2 to wait for an activation trigger.

The operational amplifier is stopped until the next activation trigger is generated.

Figure 37.8 Procedure to start and stop OPAMP in activation trigger mode

37.6 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. Figure 37.9 shows an example of each register setting.

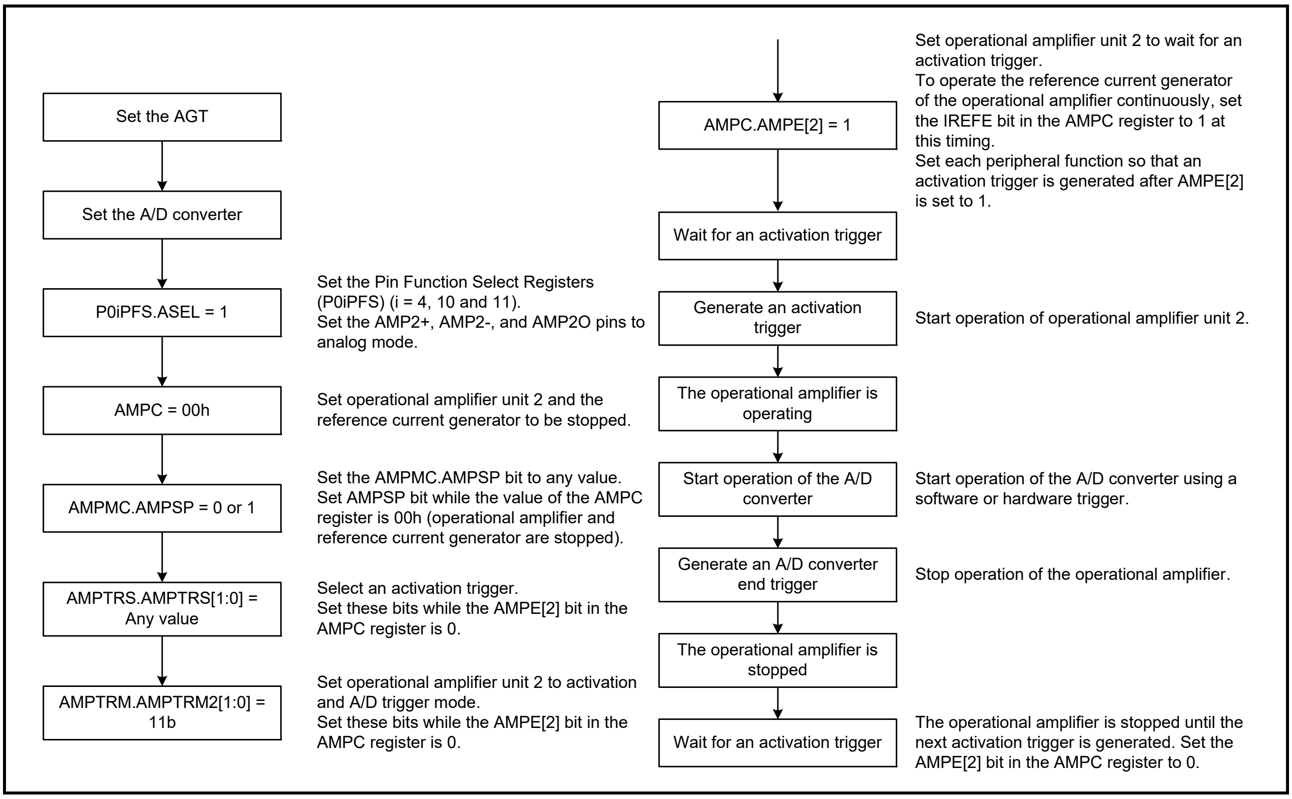

Set the AGT
Set the A/D converter
P0iPFS.ASEL = 1
Set the Pin Function Select Registers (P0iPFS) (i = 4, 10 and 11).
Set the AMP2+, AMP2-, and AMP2O pins to analog mode.
AMPC = 00h
Set operational amplifier unit 2 and the reference current generator to be stopped.
AMPMC.AMPSP = 0 or 1
Set the AMPMC.AMPSP bit to any value.
Set AMPSP bit while the value of the AMPC register is 00h (operational amplifier and reference current generator are stopped).
AMPTRS.AMPTRS[1:0] = Any value
Select an activation trigger.
Set these bits while the AMPE[2] bit in the AMPC register is 0.
AMPTRM.AMPTRM2[1:0] = 11b
Set operational amplifier unit 2 to activation and A/D trigger mode.
Set these bits while the AMPE[2] bit in the AMPC register is 0.
AMPC.AMPE[2] = 1
Set operational amplifier unit 2 to wait for an activation trigger.
To operate the reference current generator of the operational amplifier continuously, set the IREFE bit in the AMPC register to 1 at this timing.
Set each peripheral function so that an activation trigger is generated after AMPE[2] is set to 1.
Wait for an activation trigger
Generate an activation trigger
Start operation of operational amplifier unit 2.
The operational amplifier is operating
Start operation of the A/D converter
Start operation of the A/D converter using a software or hardware trigger.
Generate an A/D converter end trigger
Stop operation of the operational amplifier.
The operational amplifier is stopped
Wait for an activation trigger
The operational amplifier is stopped until the next activation trigger is generated. Set the AMPE[2] bit in the AMPC register to 0.

Figure 37.9 Procedure to activate the operational amplifier using an activation trigger and to stop the operational amplifier with an A/D conversion end trigger

37.7 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 the A/D conversion. The reference current circuit can be stopped at the end of the 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.

38. Low Power Analog Comparator (ACMPLP)

38.1 Overview

The Low-Power Analog Comparator (ACMPLP) compares a reference input voltage with an analog input voltage. Comparator channels ACMPLP0 and ACMPLP1 are independent of each other.

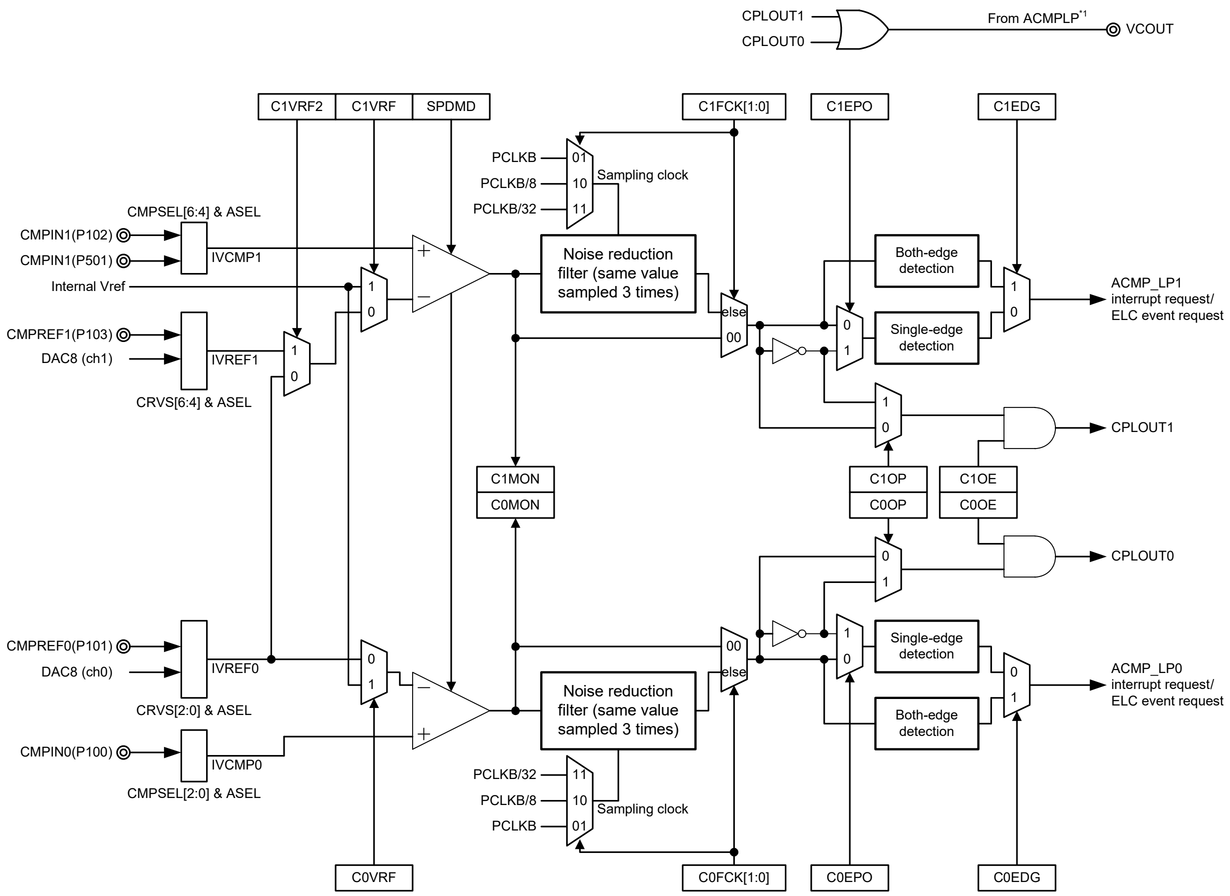
The comparison result of the reference input voltage and analog input voltage can be read by software. The comparison result can also be output externally. The reference input voltage can be selected from an input to the CMPREFi (i = 0, 1) pin, an output from the internal 8-bit D/A converter, and the 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 38.1 lists the ACMPLP specifications, Figure 38.1 shows a block diagram of the ACMPLP when the window function is disabled, and Figure 38.2 shows a block diagram of the ACMPLP when the window function is enabled. Table 38.2 lists the I/O pins of the ACMPLP.

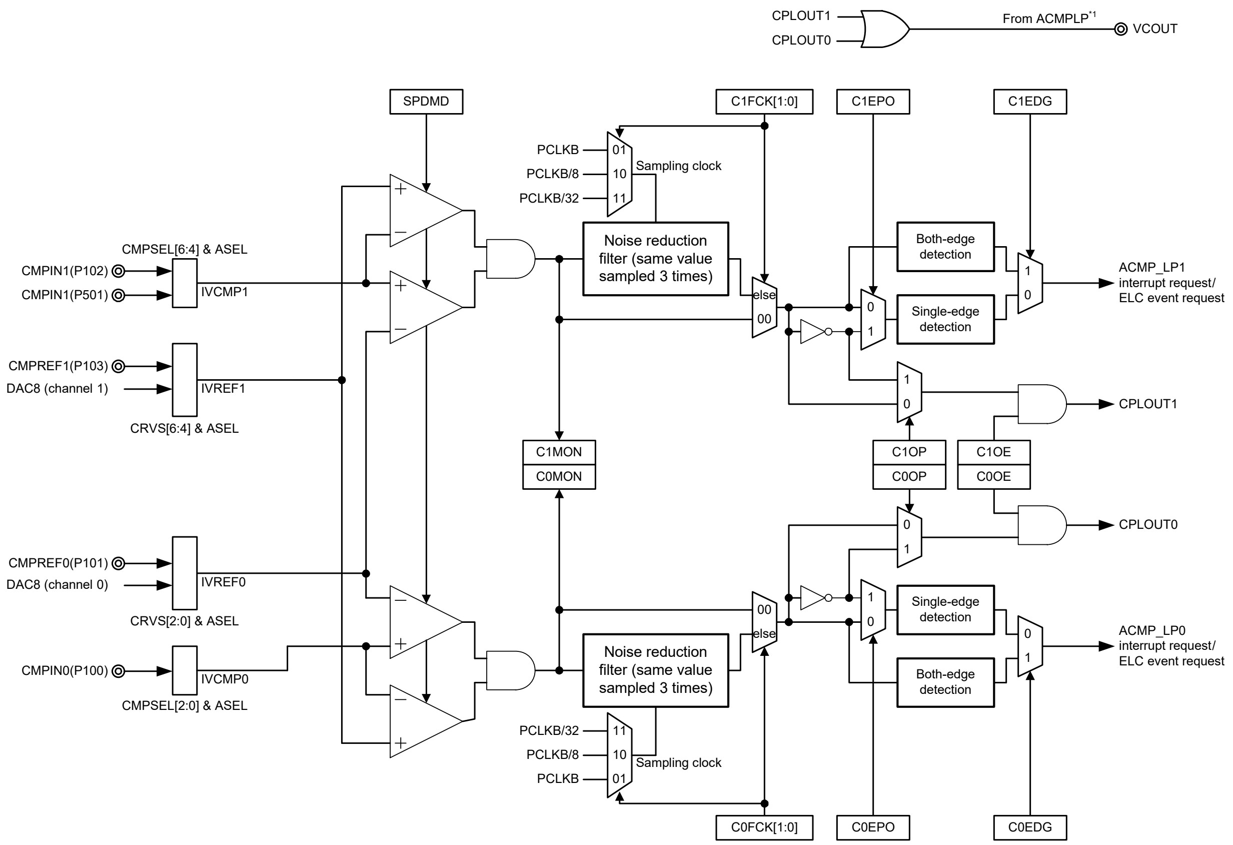
Table 38.1 ACMPLP specifications

Parameter	Specifications
Number of channels	2 channels: ACMPLP0 and ACMPLP1
Analog input voltage	Input from CMPINi (i = 0, 1) pin
Reference voltage	• Standard mode One of the following can be selected: - Internal reference voltage (Vref) - Input from CMPREFi (i = 0, 1) pin - Output from internal 8-bit D/A converter. • Window mode One of the following can be selected: - Input from CMPREFi (i = 0, 1) pin (CMPREF0: low reference, CMPREF1: high reference) - Output from internal 8-bit D/A converter.
Comparator output	• Comparison result • Generation of ELC event output • Monitor output from register.
Interrupt request signal	• Interrupt request generated on valid edge detection from comparison result • Rising edge, falling edge, or both edges can be selected.
Selectable functions	• Noise filter function - One of three sampling frequencies can be selected - Not using the filter function can be selected. • Window function - Window function is used or not used can be selected. • Low-Power Analog Comparator response speed - High-speed mode or low-speed mode can be selected.


CPLOUT1
CPLOUT0
From ACMPLP*1
VCOUT
C1VRF2
C1VRF
SPDMD
C1FCK[1:0]
C1EPO
C1EDG
PCLKB
PCLKB/8
PCLKB/32
01
10
11
Sampling clock
CMPSEL[6:4] & ASEL
CMPIN1(P102)
CMPIN1(P501)
IVCMP1
Internal Vref
1
0
+
−
Noise reduction filter (same value sampled 3 times)
else
00
Both-edge detection
Single-edge detection
ACMP_LP1 interrupt request/ ELC event request
CMPREF1(P103)
DAC8 (ch1)
IVREF1
CRVS[6:4] & ASEL
CPLOUT1
C1MON
C0MON
C1OP
C0OP
C1OE
C0OE
CPLOUT0
CMPREF0(P101)
DAC8 (ch0)
IVREF0
CRVS[2:0] & ASEL
ACMP_LP0 interrupt request/ ELC event request
CMPIN0(P100)
IVCMP0
CMPSEL[2:0] & ASEL
Noise reduction filter (same value sampled 3 times)
PCLKB/32
PCLKB/8
PCLKB
Sampling clock
C0VRF
C0FCK[1:0]
C0EPO
C0EDG

Note 1. ACMPLP results are output to the VCOUT pin.

Figure 38.1 ACMPLP block diagram when window function is disabled


CPLOUT1
CPLOUT0
From ACMPLP*1
VCOUT
SPDMD
C1FCK[1:0]
C1EPO
C1EDG
PCLKB
PCLKB/8
PCLKB/32
01
10
11
Sampling clock
Noise reduction filter (same value sampled 3 times)
else
00
Both-edge detection
Single-edge detection
ACMP_LP1 interrupt request/ ELC event request
CMPSEL[6:4] & ASEL
CMPIN1(P102)
CMPIN1(P501)
IVCMP1
CMPREF1(P103)
DAC8 (channel 1)
IVREF1
CRVS[6:4] & ASEL
CPLOUT1
C1MON
C0MON
C1OP
C0OP
C1OE
C0OE
CPLOUT0
CMPREF0(P101)
DAC8 (channel 0)
IVREF0
CRVS[2:0] & ASEL
CMPIN0(P100)
IVCMP0
CMPSEL[2:0] & ASEL
ACMP_LP0 interrupt request/ ELC event request
C0FCK[1:0]
C0EPO
C0EDG

Note 1. ACMPLP results are output to the VCOUT pin.

Figure 38.2 ACMPLP block diagram when window function is enabled

Table 38.2 Comparator pin configuration

Comparator	Reference voltage input pin		Analog voltage input pin		Output pin
	Standard mode	Window function mode	Standard mode	Window function mode	
ACMPLP0	• IVREF0 (CMPREF0 (P101)/DAC8 (channel 0)) • Internal Vref (Selectable)	Low reference voltage: • IVREF0 (CMPREF0(P101)/ DAC8 (channel 0)) High reference voltage: • IVREF1 (CMPREF1 (P103)/DAC8 (channel 1))	• IVCMP0 (CMPIN0 (P100)		VCOUT*1
ACMPLP1	• IVREF0 (CMPREF0(P101)/ DAC8 (channel 0)) • IVREF1 (CMPREF1 (P103)/DAC8 (channel 1)) • Internal Vref (Selectable)		• IVCMP1 (CMPIN1 (P102)/CMPIN1 (P501)		

Note 1. ACMPLP0 and ACMPLP1 compare outputs are output to the VCOUT pin.

38.2 Register Descriptions

38.2.1 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,*6	0: Disable window function for ACMPLP0 1: Enable window function for ACMPLP0.	R/W
b2	C0VRF	ACMPLP0 Reference Voltage Selection*6	0: IVREF0 1: Internal reference voltage (Vref).*4	R/W
b3	C0MON	ACMPLP0 Monitor Flag*3	When the window function is disabled: 0: IVCMP0 < ACMPLP0 reference voltage 1: IVCMP0 > ACMPLP0 reference voltage. When the window function is enabled: 0: IVCMP0 < IVREF0 or IVCMP0 > IVREF1 1: IVREF0 < IVCMP0 < IVREF1.	R
b4	C1ENB	ACMPLP1 Operation Enable	0: Disable ACMPLP1 operation 1: Enable ACMPLP1 operation.	R/W
b5	C1WDE	ACMPLP1 Window Function Mode Enable*1,*2,*5	0: Disable ACMPLP1 window function mode 1: Enable ACMPLP1 window function mode.	R/W
b6	C1VRF	ACMPLP1 Reference Voltage Selection*5	0: IVREF0 or IVREF1 1: Internal reference voltage (Vref).*4	R/W
b7	C1MON	ACMPLP1 Monitor Flag*3	When the window function is disabled: 0: IVCMP1 < ACMPLP1 reference voltage 1: IVCMP1 > ACMPLP1 reference voltage. When the window function is enabled: 0: IVCMP1 < IVREF0 or IVCMP1 > IVREF1 1: IVREF0 < IVCMP1 < IVREF1.	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.

Note 4. The setting is valid only when in standard mode. When in window function mode, IVREF0 or IVREF1 is selected regardless of the setting of this bit.

Note 5. To change C1WDE and C1VRF, the CRV[6:4] and CRV[2:0] bits must be 000b.

Note 6. To change C0WDE and C0VRF, the CRV[2:0] bits must be 000b.

38.2.2 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 on the rising edge 1: Interrupt and ELC event request on the 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 on the rising edge 1: Interrupt and ELC event request on the 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 can be generated. Change these bits only after setting event link to deselect. Also, be sure to clear the associated interrupt request flag.

38.2.3 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/W
b1	C0OE	ACMPLP0 VCOUT Pin Output Enable*1	0: Disable ACMPLP0 VCOUT pin output 1: Enable ACMPLP0 VCOUT pin output.	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/W
b5	C1OE	ACMPLP1 VCOUT Pin Output Enable*1	0: Disable ACMPLP1 VCOUT pin output 1: Enable ACMPLP1 VCOUT pin output.	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 output to the VCOUT pin.
Note 2. Set the CiENB bit (i = 0, 1) in the COMPMDR register to 0 before rewriting the SPDMD bit.

38.2.4 Comparator Input Select Register (COMPSEL0)

Address(es): ACMPLP.COMPSEL0 4008 5E04h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CMPSEL[6:4]			—	CMPSEL[2:0]		
Value after reset:	0	0	0	1	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CMPSEL[2:0]	ACMPLP0 Input (IVCMP0) Selection*1	b2 b0 0 0 0: No input 0 0 1: CMPIN0 (P100) 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	CMPSEL[6:4]	ACMPLP1 Input (IVCMP1) Selection*2	b6 b4 0 0 0: No input 0 0 1: CMPIN1 (P102) 1 0 0: CMPIN1 (P501) Other settings are prohibited.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Writing a value other than 000b is prohibited while CMPSEL[2:0] is not 000b.
Note 2. Writing a value other than 000b is prohibited while CMPSEL[6:4] is not 000b.

38.2.5 Comparator Reference Voltage Select Register (COMPSEL1)

Address(es): ACMPLP.COMPSEL1 4008 5E05h

	b7	b6	b5	b4	b3	b2	b1	b0
	C1VRF2	CRVS[6:4]			—	CRVS[2:0]		
Value after reset:	1	0	0	1	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CRVS[2:0]	ACMPLP0 Reference Voltage (IVREF0) Selection*1	b2 b0 0 0 0: No input 0 0 1: CMPREF0 (P101) 0 1 0: DAC8 (channel 0) output Other settings are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b04	CRVS[6:4]	ACMPLP1 Reference Voltage (IVREF1) Selection*2	b6 b4 0 0 0: No input 0 0 1: CMPREF1 (P103) 0 1 0: DAC8 (channel 1) output Other settings are prohibited.	R/W
b7	C1VRF2	ACMPLP1 Reference Voltage Selection 2*3	0: IVREF0 selected 1: IVREF1 selected.	R/W

Note 1. Writing a value other than 000b is prohibited while CRVS[2:0] is not 000b.
Note 2. Writing a value other than 000b is prohibited while CRVS[6:4] is not 000b.
Note 3. To change C1VRF2, bits CRVS[6:4] and CRVS[2:0] must be 000b.

38.3 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 the comparator operation. Table 38.3 shows the procedure for setting the ACMPLP registers.

Table 38.3 Procedure for setting the ACMPLP associated registers (i = 0, 1)

Step	Register	Bit	Setting	
1	MSTPCRD	MSTPD29	0: Input clock supply.	
2	Corresponding Port mn Pin Function Select Register (PmnPFS)	ASEL	Select the analog input.	
	COMPSEL0	CMPSEL[2:0], CMPSEL[6:4]		
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*5	Select the reference voltage.	Window comparator operation (reference = IVREF0 and IVREF1*3).
	COMPSEL1	CRVS[2:0], CRVS[6:4], C1VRF2		
	COMPMDR	CiENB	1: Operation enabled.	
5	Waiting for the comparator stabilization time T_{cmp} (minimum 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/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[7:0]	When using an interrupt: select the interrupt status flag, ICU event link select.*3	
9	ELSRn	ELS[7:0]	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 comparator setting, an unnecessary interrupt might occur until operation becomes stable, so initialize the interrupt flag.
Note 4. After the comparator setting, an unnecessary interrupt might occur until operation becomes stable, so initialize the event link select.
Note 5. To change to internal reference voltage (Vref), follow the procedure in section 38.2.1, ACMPLP Mode Setting Register (COMPMDR).

Figure 38.3 shows an operating example of the ACMPLPi (i = 0, 1) when window function is disabled. The reference input voltage (IVREFi) or internal reference voltage (Vref) and the analog input voltage (IVCMPi) are compared as follows:

- 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 the ICU. For details on the interrupt, see section 38.5, ACMPLP Interrupts. ACMPLPi also outputs an event signal to the ELC to activate other modules. For details on the ELC, see section 38.6, ELC Event Output. Do not change the values of the registers during the comparison.

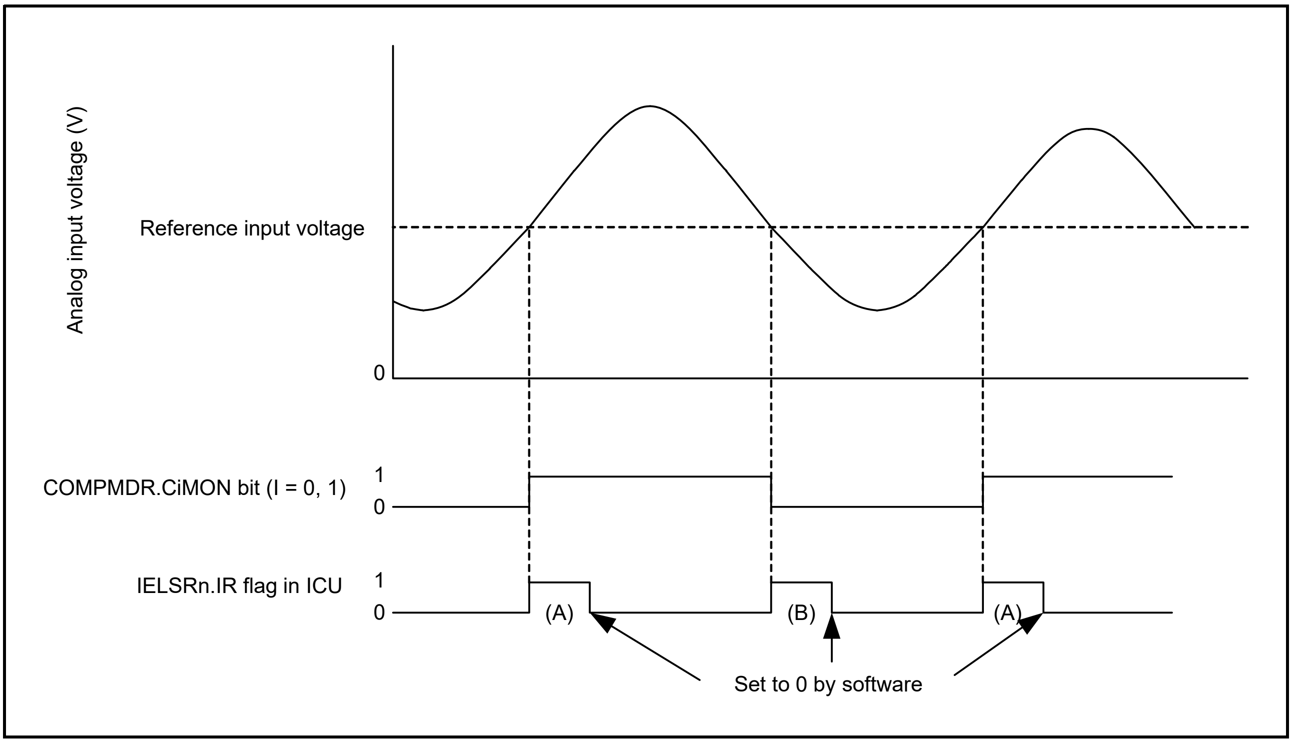

Analog input voltage (V)
Reference input voltage
0
COMPMDR.CiMON bit (I = 0, 1)
1
0
IELSRn.IR flag in ICU
1
0
(A)
(B)
(A)
Set to 0 by software

Figure 38.3 Operating example of ACMPLPi (i = 0, 1) when window function is disabled

Figure 38.3 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 38.4 shows an operation example of ACMPLPi (i = 0, 1) when the window function is enabled.

The reference voltage (IVREF0/IVREF1) and the analog input voltage are compared. The CiMON bit:

- Is set to 1 when IVREF0 < the analog input voltage < IVREF1
- Is set to 0 when the analog input voltage < IVREF0 or IVREF1 < the analog input voltage.

ACMPLPi outputs an interrupt to the ICU. For details on the interrupt, see section 38.5, ACMPLP Interrupts. ACMPLPi also outputs an event signal to the ELC to activate other modules. For details on the ELC, see section 38.6, ELC Event Output. Do not change the values of the registers during the comparison.

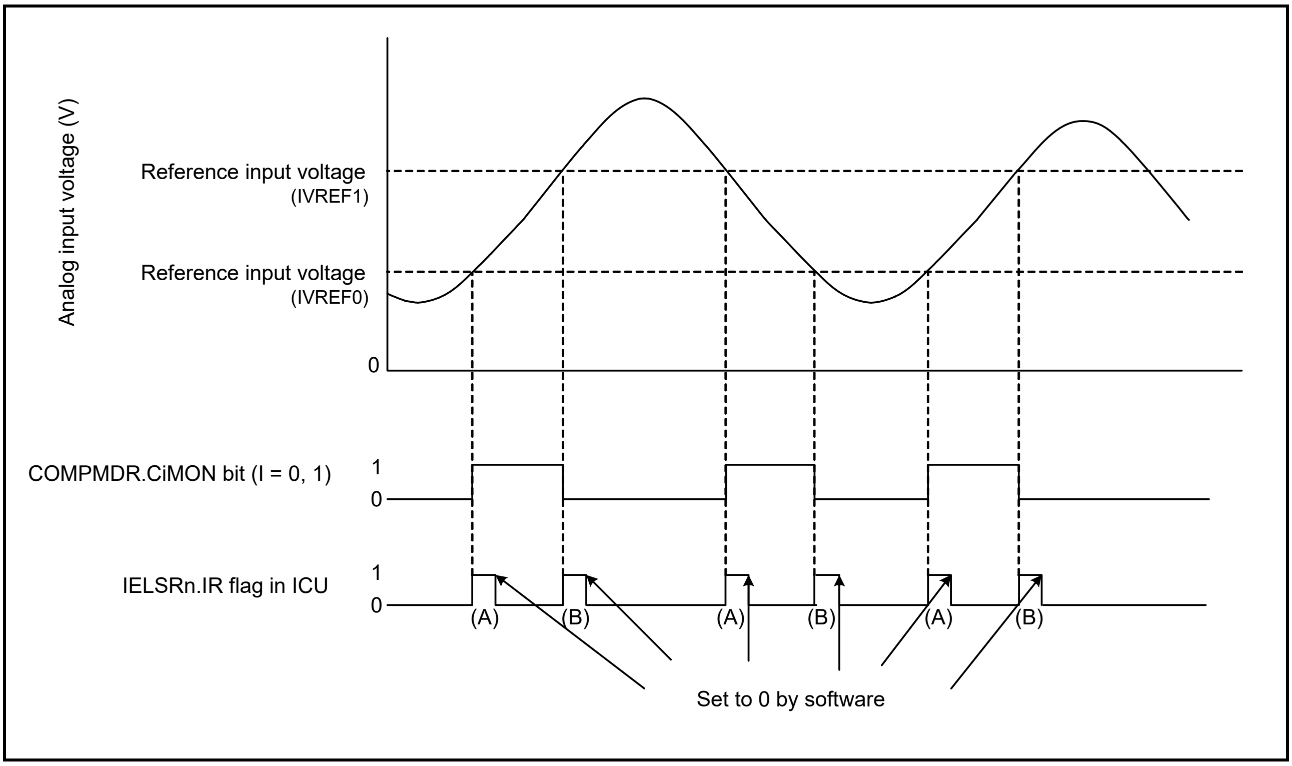

Analog input voltage (V)
Reference input voltage
(IVREF1)
Reference input voltage
(IVREF0)
0
COMPMDR.CiMON bit (I = 0, 1)
1
0
IELSRn.IR flag in ICU
1
0
(A)
(B)
(A)
(B)
(A)
(B)
Set to 0 by software

Figure 38.4 Operating example of ACMPLPi (i = 0, 1) when window function is enabled

Figure 38.4 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.

38.4 Noise Filter

Figure 38.5 shows the configuration of the ACMPLPi noise filter, and Figure 38.6 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 in Software Standby mode, set the COMPFIR.CiFCK[1:0] bits to 00b (bypass).

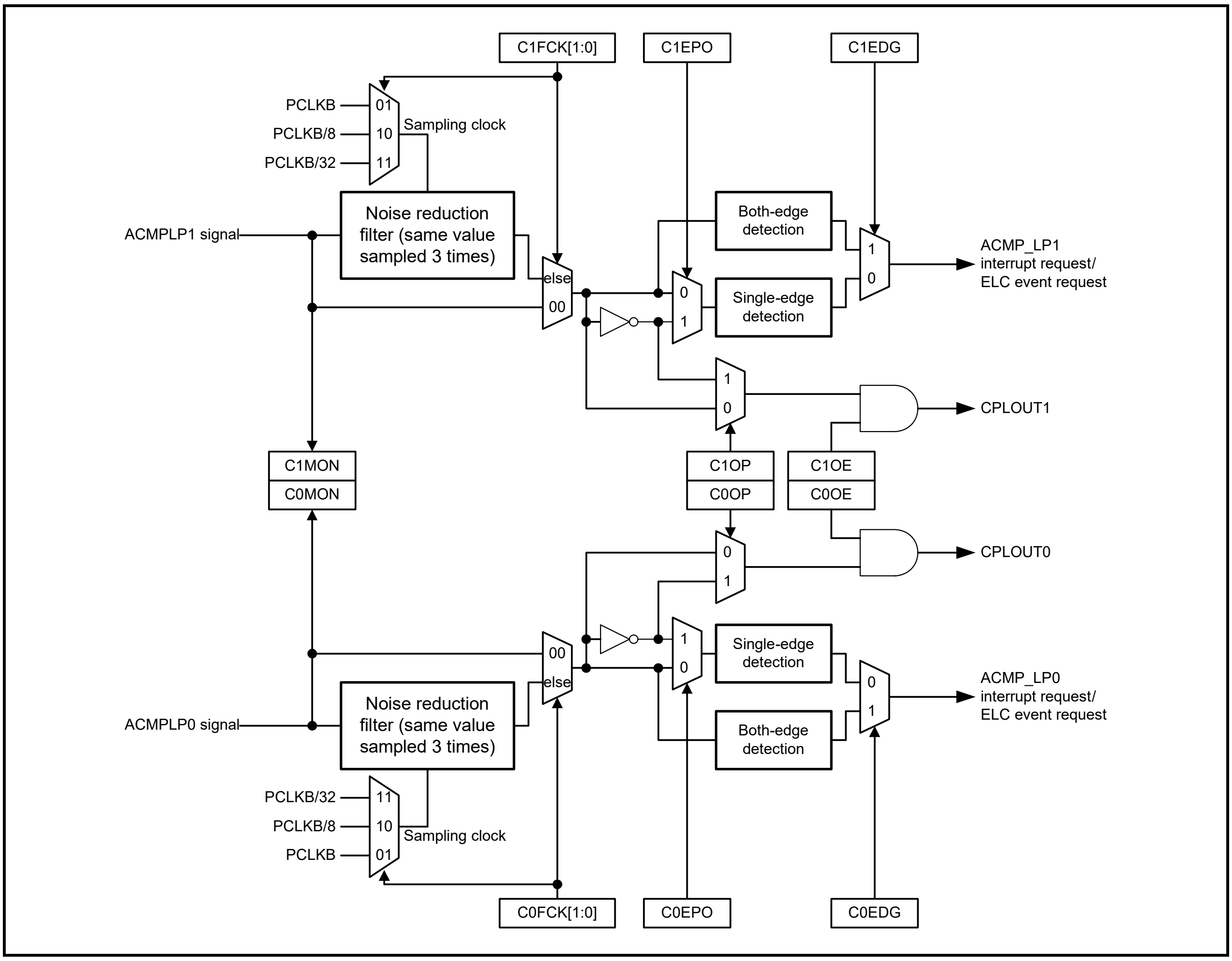

C1FCK[1:0]
C1EPO
C1EDG
PCLKB
PCLKB/8
PCLKB/32
01
10
11
Sampling clock
Noise reduction filter (same value sampled 3 times)
ACMPLP1 signal
else
00
Both-edge detection
Single-edge detection
ACMP_LP1 interrupt request/ ELC event request
CPLOUT1
C1MON
C0MON
C1OP
C0OP
C1OE
C0OE
CPLOUT0
ACMP_LP0 interrupt request/ ELC event request
ACMPLP0 signal
C0FCK[1:0]
C0EPO
C0EDG

Figure 38.5 Noise filter and edge detection configuration

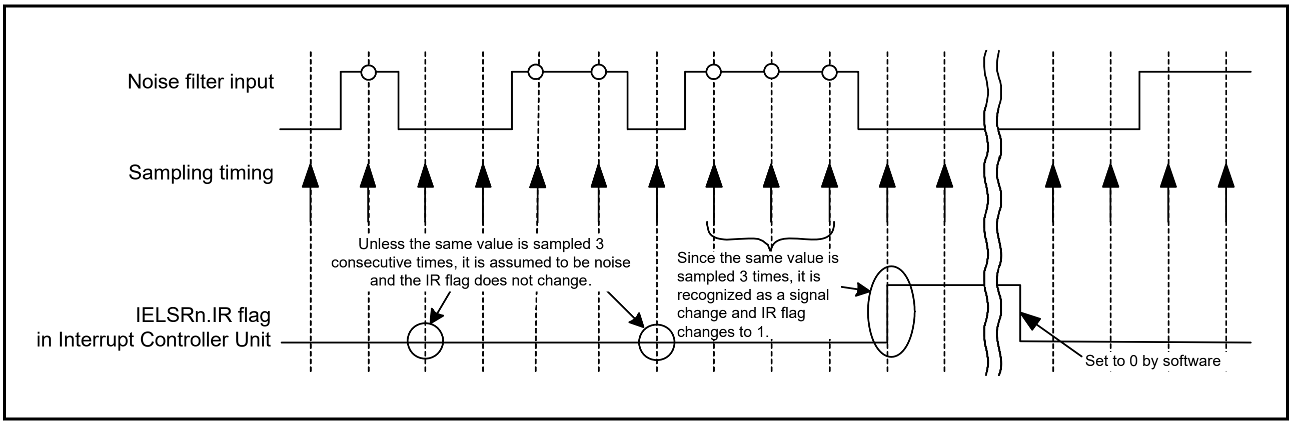

Noise filter input
Sampling timing
IELSRn.IR flag
in Interrupt Controller Unit
Unless the same value is sampled 3 consecutive times, it is assumed to be noise and the IR flag does not change.
Since the same value is sampled 3 times, it is recognized as a signal change and IR flag changes to 1.
Set to 0 by software

Figure 38.6 Noise filter and interrupt operation example

38.5 ACMPLP Interrupts

The ACMPLP generates interrupt requests from the ACMPLP0 and ACMPLP1 sources. To use the ACMPLPi (i = 0 and 1) interrupt, select it in the IELSRn register in the ICU. You can 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 the 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 (no sampling). The ACMPLP1 interrupt request cannot be used to release Software Standby mode or Snooze mode.

38.6 ELC Event Output

The ELC uses the ACMPLP interrupt request signal as an ELC event signal, enabling link operation for the preset module. To use the ELC events of the ACMPLP, select them in the ELSRn register in the ELC. When using ELC event request, set the COMPFIR.CiFCK[1:0] bits to 01b, 10b, or 11b.

38.7 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 in the CiEPO bit.

38.8 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.

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 registers.

For the register settings and the associated comparator output, see section 38.2.3, ACMPLP Output Control Register (COMPOCR).

38.9 Usage Notes

38.9.1 Settings for the Module-Stop State

The Module Stop Control Register can enable or disable the ACMPLP operation. The ACMPLP is initially stopped after reset. c. For details, see section 11, Low Power Modes.

38.9.2 Relationship with A/D converter

Constraints apply on the simultaneous use of ACMPLP analog input and A/D converter analog input. For details, see section 34.8.13, Relationship between the ADC14, OPAMP, and ACMPLP.

39. 8-Bit D/A Converter (DAC8)

39.1 Overview

Table 39.1 lists the specifications of the 8-bit D/A converter, and Figure 39.1 shows a block diagram.

Table 39.1 8-bit D/A converter specifications

Parameter	Specifications
Resolution	8 bits
Output channels	2 channels
Module-stop function	The module-stop state can be set to reduce power consumption.

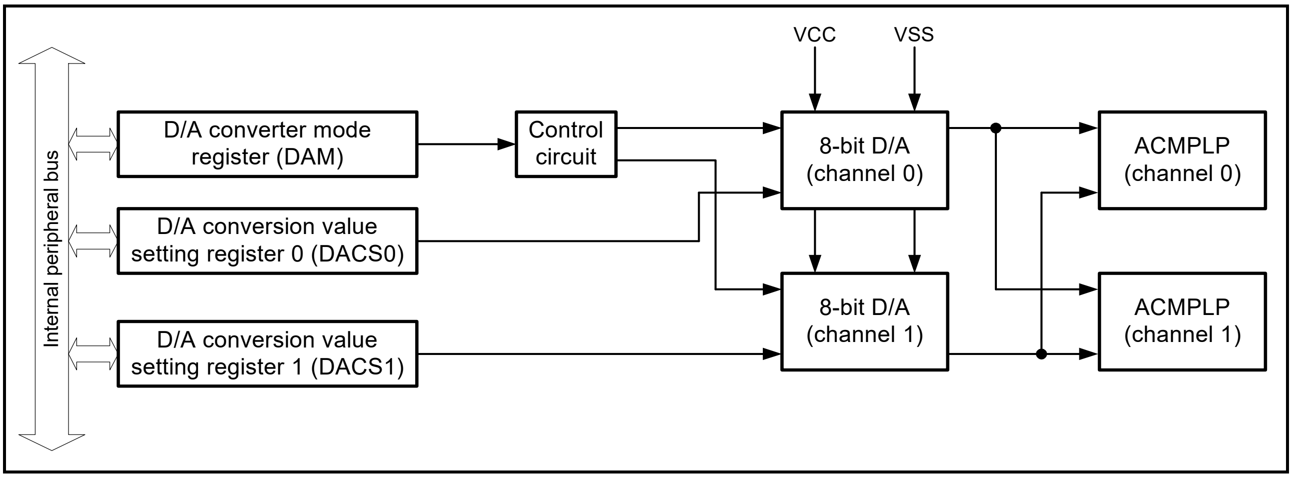

Internal peripheral bus
D/A converter mode register (DAM)
D/A conversion value setting register 0 (DACS0)
D/A conversion value setting register 1 (DACS1)
Control circuit
VCC
VSS
8-bit D/A (channel 0)
8-bit D/A (channel 1)
ACMPLP (channel 0)
ACMPLP (channel 1)

Figure 39.1 8-bit D/A converter block diagram

39.2 Register Descriptions

39.2.1 D/A Conversion Value Setting Register n (DACSn) (n = 0, 1)

Address(es): DAC8.DACS0 4009 E000h, DAC8.DACS1 4009 E001h

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

The DACSn register is an 8-bit read/write register that stores data for D/A conversion. When D/A conversion is enabled, the value in the DACSn register is converted and output to an ACMPLP.

When 8-bit D/A converter output is selected as the reference input for the ACMPLP in the COMPSEL1 register, and ACMPLP operation is enabled (COMPMDR.CnENB = 1), changing the DACS[7:0] bits for the channel in use is prohibited.

39.2.2 D/A Converter Mode Register (DAM)

Address(es): DAC8.DAM 4009 E003h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DACE1	DACE0	—	—	—	—
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	DACE0	D/A Operation Enable 0	0: D/A conversion disabled for channel 0 1: D/A conversion enabled for channel 0.	R/W
b5	DACE1	D/A Operation Enable 1	0: D/A conversion disabled for channel 1 1: D/A conversion enabled for channel 1.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DACEn bit (D/A Operation Enable n) (n = 0, 1)

The DACEn bit enables or disables D/A conversion.

When an 8-bit D/A converter output is selected as the reference input for the ACMPLP in the COMPSEL1 register, and ACMPLP operation is enabled (COMPMDR.CnENB = 1), changing the DACEn bits for the channel in use is prohibited.

39.3 Operation

The 8-bit D/A converter includes D/A conversion circuits for two channels, each of which can operate independently. When the DAM.DACEn bit (n = 0, 1) is set to 1, the 8-bit D/A converter is enabled and the conversion result is output to the ACMPLP.

The following sequence describes an operation example when performing D/A conversion for channel 0:

1. Set the data for D/A conversion to the DACS0 register.
2. Set the DAM.DACE0 bit to 1 to start D/A conversion. The conversion result is output to ACMPLP. The conversion result is continuously being output until the DACS0 is rewritten or the DAM.DACE0 bit is set to 0 (D/A conversion disabled).
   The output value (reference) is calculated with the following formula:

$$\frac{\text{DACS0 register}}{256} \times \text{VCC}$$

3. Set the COMPSEL1 register, and select the 8-bit D/A converter as the reference voltage.
4. Set the COMPMDR.CiENB bit to 1.
5. Wait for the comparator stabilization time T_{cmp} (min. 100 µs). For details, see section 38, Low Power Analog Comparator (ACMPLP).

39.4 Usage Notes

39.4.1 Module-Stop State

The Module Stop Control Register can enable or disable operation of the 8-bit D/A converter. The 8-bit D/A converter is stopped after a reset. The registers become accessible when the module-stop state is canceled. For details, see section 11, Low Power Modes.

39.4.2 Operation of the 8-bit D/A Converter in Module-Stop State

When the MCU enters the module-stop state with D/A conversion enabled, D/A outputs are saved. The power supply current is the same as the one during D/A conversion. If the power supply current must be reduced in the module-stop state, disable D/A conversion by setting the DAM.DACEn bits to 0.

39.4.3 8-bit D/A Converter in Software Standby Mode Operation

When the MCU enters Software Standby mode with D/A conversion enabled, D/A outputs are saved. The power supply current is the same as the one during D/A conversion. If the power supply current must be reduced in Software Standby mode, disable D/A conversion by setting the DAM.DACEn bits to 0.

39.4.4 When Not Using the D/A Converter

When not using the 8-bit D/A converter, set the DAM.DACEn bit to 0 (output disabled), and the DACSn register to 00h, so that current does not flow and the current consumption can be reduced.

40. Capacitive Touch Sensing Unit (CTSU)

40.1 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 40.1 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.

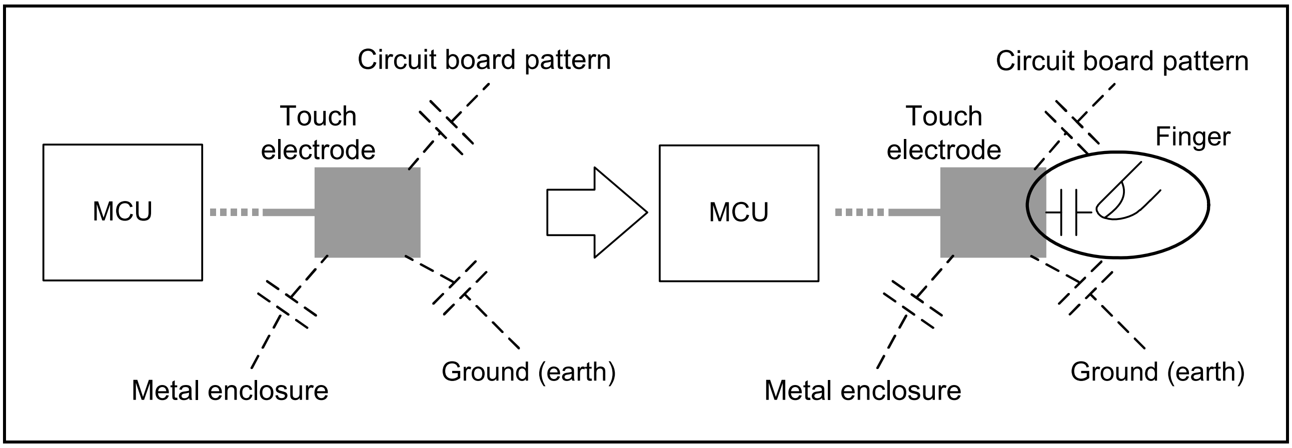

Circuit board pattern
Touch electrode
MCU
Metal enclosure
Ground (earth)
Circuit board pattern
Touch electrode
Finger
MCU
Metal enclosure
Ground (earth)

Figure 40.1 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.

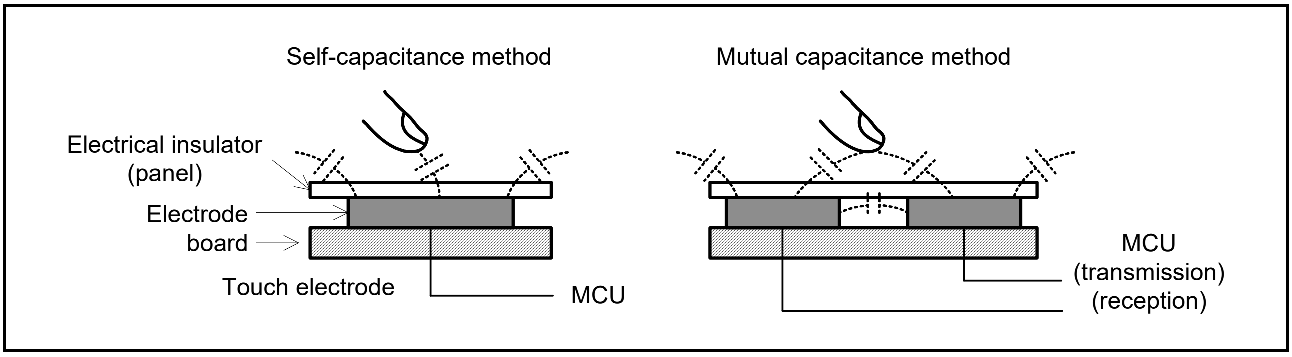

Self-capacitance method
Mutual capacitance method
Electrical insulator (panel)
Electrode
board
Touch electrode
MCU
MCU
(transmission)
(reception)

Figure 40.2 Self-capacitance and mutual-capacitance methods

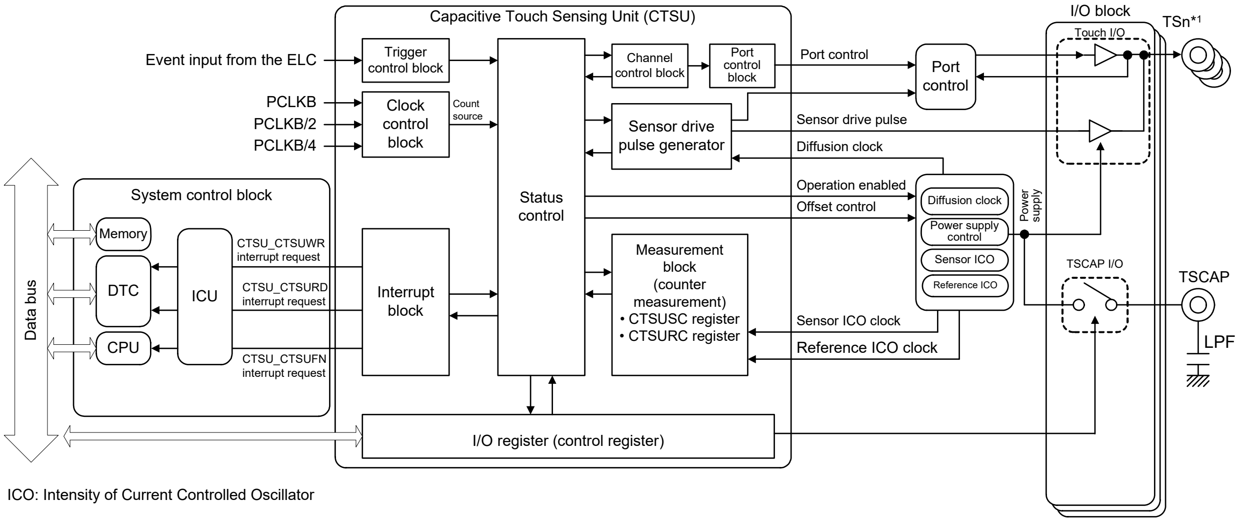
Electrostatic capacitance is measured by counting a clock signal whose frequency changes according to the amount of charged or discharged current, for a specified period. For details on the measurement principles of the CTSU, see section 40.3.1, Principles of Measurement Operation. Table 40.1 lists the CTSU specifications and Figure 40.3 shows a block diagram.

Table 40.1 CTSU specifications

Parameter		Description
Operating clocks		PCLKB, PCLKB/2, or PCLKB/4
Pins	Electrostatic capacitance measurement	11 channels (TS00, TS01, TS03, TS10, TS12, TS13, TS18, TS28, TS30, TS31, TS34)
	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 40.3 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.


Capacitive Touch Sensing Unit (CTSU)
Event input from the ELC
Trigger control block
PCLKB
PCLKB/2
PCLKB/4
Clock control block
Count source
Status control
Channel control block
Port control block
Port control
Port control
Sensor drive pulse generator
Sensor drive pulse
Diffusion clock
Operation enabled
Offset control
Diffusion clock
Power supply control
Sensor ICO
Reference ICO
Power supply
Measurement block (counter measurement)
• CTSUSC register
• CTSURC register
Sensor ICO clock
Reference ICO clock
I/O block
Touch I/O
TSn*1
TSCAP I/O
TSCAP
LPF
Data bus
System control block
Memory
DTC
CPU
ICU
CTSU_CTSUWR interrupt request
CTSU_CTSURD interrupt request
CTSU_CTSUFN interrupt request
Interrupt block
I/O register (control register)

ICO: Intensity of Current Controlled Oscillator

Note 1. n = 00, 01, 03, 10, 12, 13, 18, 28, 30, 31, 34

Figure 40.3 CTSU block diagram

Table 40.2 CTSU pin configuration

Pin name	I/O	Function
TS00, TS01, TS03, TS10, TS12, TS13, TS18, TS28, TS30, TS31, TS34	Input	Electrostatic capacitive measurement pins (touch pins)
TSCAP	-	LPF connection pin

40.2 Register Descriptions

40.2.1 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.	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 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 software writes 1 to the CTSUSTRT bit (software trigger), and stops when 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 40.3 lists the CTSU states.

Table 40.3 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 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 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 40.4 shows the CTSU power supply state control.

Table 40.4 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 initializes the internal control registers.

Do not write 1 to the CTSUINIT bit when the CTSUSTRT bit is 1.

40.2.2 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

Bit	Symbol	Bit name	Description	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
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 40.3.2, Measurement Modes.

40.2.3 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 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 40.2.21, 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.

40.2.4 CTSU Sensor Stabilization Wait Control Register (CTSUSST)

Address(es): CTSU.CTSUSST 4008 1003h

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

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUSST[7:0]	CTSU Sensor Stabilization Wait Control	Set 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.

40.2.5 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 0 0 0: TS00 0 0 0 0 0 1: TS01 0 0 0 0 1 1: TS03 0 0 1 0 1 0: TS10 0 0 1 1 0 0: TS12 0 0 1 1 0 1: TS13 0 1 0 0 1 0: TS18 0 1 1 1 0 0: TS28 0 1 1 1 1 0: TS30 0 1 1 1 1 1: TS31 1 0 0 0 1 0: TS34 Other than those specified, the starting measurement operation when CTSUCR0.CTSUSTRT = 1 is prohibited after these bits are set.	R/W*1
			In other measurement modes, these bits indicate the channel that is currently being measured: b5 b0 0 0 0 0 0 0: TS00 0 0 0 0 0 1: TS01 0 0 0 0 1 1: TS03 0 0 1 0 1 0: TS10 0 0 1 1 0 0: TS12 0 0 1 1 0 1: TS13 0 1 0 0 1 0: TS18 0 1 1 1 0 0: TS28 0 1 1 1 1 0: TS30 0 1 1 1 1 1: TS31 1 0 0 0 1 0: TS34 1 1 1 1 1 1: 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, 001010b, 001100b, 001101b, 010010b, 011100b, 011110b, 011111b, and 100010b). In other modes, these indicate the receive channel that is being measured and writing to these bits has no effect.

40.2.6 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 0 0 0: TS00 0 0 0 0 0 1: TS01 0 0 0 0 1 1: TS03 0 0 1 0 1 0: TS10 0 0 1 1 0 0: TS12 0 0 1 1 0 1: TS13 0 1 0 0 1 0: TS18 0 1 1 1 0 0: TS28 0 1 1 1 1 0: TS30 0 1 1 1 1 1: TS31 1 0 0 0 1 0: TS34 1 1 1 1 1 1: Measurement is stopped.	R
b7, b6	—	Reserved	These bits are read as 0.	R

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.

40.2.7 CTSU Channel Enable Control Register 0 (CTSUCHAC0)

Address(es): CTSU.CTSUCHAC0 4008 1006h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHAC0[7:0]*1							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHAC0[7:0] *1	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 pins.	R/W

Note 1. The MCU does not support TS02 and TS04 to TS07 pins. Therefore, CTSUCHAC0[2] and CTSUCHAC0[4:7] are read as 0. The write value should be 0.

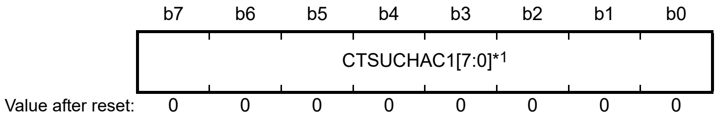
Only set the CTSUCHAC0 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC0[7:0]*1 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[3] with TS03.

40.2.8 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]*1	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 TS10, TS12, and TS13 pins.	R/W

Note 1. The MCU does not support TS08, TS09, TS11, TS14 and TS15 pins. Therefore, CTSUCHAC1[0], CTSUCHAC1[1], CTSUCHAC1[3], and CTSUCHAC[7:6] are read as 0. The write value should be 0.

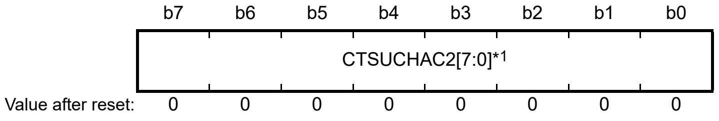
Only set the CTSUCHAC1 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC1[7:0]*1 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[2] is associated with TS10 and CTSUCHAC1[5] with TS13.

40.2.9 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. This bit specifies the TS18 pin.	R/W

Note 1. The MCU does not support TS16, TS17 and TS19 to TS23 pins. Therefore, CTSUCHAC2[7:3] and CTSUCHAC2[1:0] are read as 0. The write value should be 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[2] is associated with TS18.

40.2.10 CTSU Channel Enable Control Register 3 (CTSUCHAC3)

Address(es): CTSU.CTSUCHAC3 4008 1009h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHAC3[7:0]*1							
Value after reset:	0	0	0	0	0	0	0	0

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 TS28, TS30, and TS31 pins.	R/W

Note 1. The MCU does not support TS24 to TS27 and TS29 pins. Therefore, CTSUCHAC3[3:0] and CTSUCHAC3[5] are 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[4] is associated with TS28 and CTSUCHAC3[7] with TS31.

40.2.11 CTSU Channel Enable Control Register 4 (CTSUCHAC4)

Address(es): CTSU.CTSUCHAC4 4008 100Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CTSUCHAC4[3:0]*1			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	CTSUCHAC4[3:0] *1	CTSU Channel Enable Control 4	These bits select whether the associated TS pin is measured: 0: Do not measure 1: Measure. This bit specifies the TS34 pin.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The MCU does not support TS32, TS33, and TS35 pins. Therefore, CTSUCHAC4[3] and CTSUCHAC4[1:0] are read as 0. The write value should be 0.

Only set the CTSUCHAC4 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC4[3:0]*1 bits (CTSU Channel Enable Control 4)

The CTSUCHAC4[3:0] bits select the receive and transmit pins whose electrostatic capacitance is to be measured. CTSUCHAC4[2] is associated with TS34.

40.2.12 CTSU Channel Transmit/Receive Control Register 0 (CTSUCHTRC0)

Address(es): CTSU.CTSUCHTRC0 4008 100Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHTRC0[7:0]*1							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHTRC0[7:0] *1	CTSU Channel Transmit/Receive Control 0	0: Reception 1: Transmission. These bits specify the TS00, TS01, TS03 pins.	R/W

Note 1. The MCU does not support TS02 and TS04 to TS07 pins. Therefore, CTSUCHTRC0[2] and CTSUCHTRC0[4:7] are read as 0. The write value should be 0.

Only set the CTSUCHTRC0 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC0[7:0]*1 bits (CTSU Channel Transmit/Receive Control 0)

In full-scan mode, the CTSUCHTRC0[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. CTSUCHTRC0[0] is associated with TS00 and CTSUCHTRC0[3] with TS03.

40.2.13 CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1)

Address(es): CTSU.CTSUCHTRC1 4008 100Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHTRC1[7:0]*1							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHTRC1[7:0]*1	CTSU Channel Transmit/Receive Control 1	0: Reception 1: Transmission. These bits specify the TS10, TS12, and TS13 pins.	R/W

Note 1. The MCU does not support TS08, TS09, TS11, TS14, and TS15 pins. Therefore, CTSUCHTRC1[0], CTSUCHTRC1[1], CTSUCHTRC1[3], and CTSUCHTRC[7:6] are read as 0. The write value should be 0.

Only set the CTSUCHTRC1 when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC1[7:0]*1 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 of these bits is ignored in self-capacitance single scan and multi-scan modes. CTSUCHTRC1[2] is associated with TS10 and CTSUCHTRC1[5] with TS13.

40.2.14 CTSU Channel Transmit/Receive Control Register 2 (CTSUCHTRC2)

Address(es): CTSU.CTSUCHTRC2 4008 100Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHTRC2[7:0]*1							
Value after reset:	0	0	0	0	0	0	0	0

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 specifies the TS18 pin.	R/W

Note 1. The MCU does not support TS16, TS17, and TS19 to TS23 pins. Therefore, CTSUCHTRC2[7:3] and CTSUCHTRC2[1:0] are 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[2] is associated with TS18.

40.2.15 CTSU Channel Transmit/Receive Control Register 3 (CTSUCHTRC3)

Address(es): CTSU.CTSUCHTRC3 4008 100Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHTRC3[7:0]*1							
Value after reset:	0	0	0	0	0	0	0	0

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 TS28, TS30, and TS31 pins.	R/W

Note 1. The MCU does not support TS24 to TS27, and TS29 pins. Therefore, CTSUCHTRC3[3:0] and CTSUCHTRC3[5] are 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[4] is associated with TS28 and CTSUCHTRC3[7] with TS31.

40.2.16 CTSU Channel Transmit/Receive Control Register 4 (CTSUCHTRC4)

Address(es): CTSU.CTSUCHTRC4 4008 100Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CTSUCHTRC4[3:0]*1			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	CTSUCHTRC4[3:0] *1	CTSU Channel Transmit/ Receive Control 4	0: Reception 1: Transmission. This bit specifies the TS34 pin.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The MCU does not support TS32, TS33, and TS35 pins. Therefore, CTSUCHTRC4[3] and CTSUCHTRC4[1:0] are read as 0. The write value should be 0.

Only set the CTSUCHTRC4 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC4[3:0]*1 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[2] is associated with TS34.

40.2.17 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 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.

40.2.18 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
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 40.3.2.2, 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 indicates when the sensor counter, CTSUSC, overflows. On overflow, the counter value reads as FFFFh. Measurement processing continues for the specified period.

No interrupt is generated when an overflow occurs. To determine the channel on which the overflow occurred, read the measurement result of each channel after measurement completes, signaled by a measurement end interrupt.

This flag is cleared when 0 is written after 1 is read by 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, CTSURC, 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, signaled by a measurement end interrupt.

This flag is cleared when 0 is written after 1 is read by 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.

40.2.19 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 40.5.

Table 40.5 Relationship between base clock frequencies and CTSUSSDIV[3:0] bit settings (1 of 2)

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

Table 40.5 Relationship between base clock frequencies and CTSUSSDIV[3:0] bit settings (2 of 2)

Base clock frequency fb (MHz)	CTSUSSDIV[3:0] bit setting
0.29 ≤ fb < 0.31	1101b
0.27 ≤ fb < 0.29	1110b
fb < 0.27	1111b

40.2.20 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 is generated.

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.

40.2.21 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
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, then to the CTSUSO0 register, and then to the CTSUSO1 register. The write to the CTSUSO1 register causes a transition to Status 3 (see Table 40.6 and Table 40.7). 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 40.3.2.1, 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 these 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.

40.2.22 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 clock. Read these bits after a CTSU_CTSURD interrupt is generated. 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.

40.2.23 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 (counter value/measurement time) for the input current amount 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.

40.2.24 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.

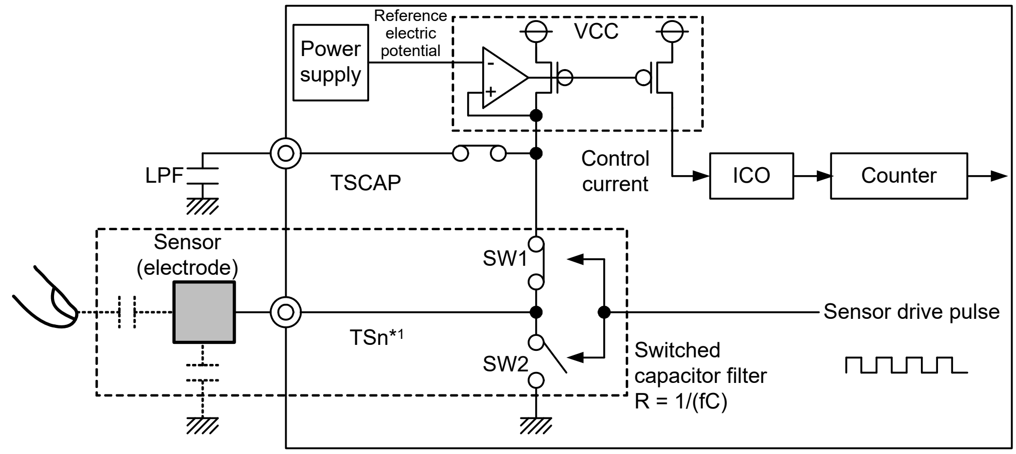
If the TSCAP voltage becomes abnormal, the sensor ICO counter value is undefined, but touch measurement completes normally, therefore it is difficult to detect an abnormality by reading the sensor ICO counter value. If the CTSU reference ICO current adjustment bits (CTSURICOA[7:0]) in the CTSUSO1 register 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.

40.3 Operation

40.3.1 Principles of Measurement Operation

Figure 40.4 shows the measurement circuit.

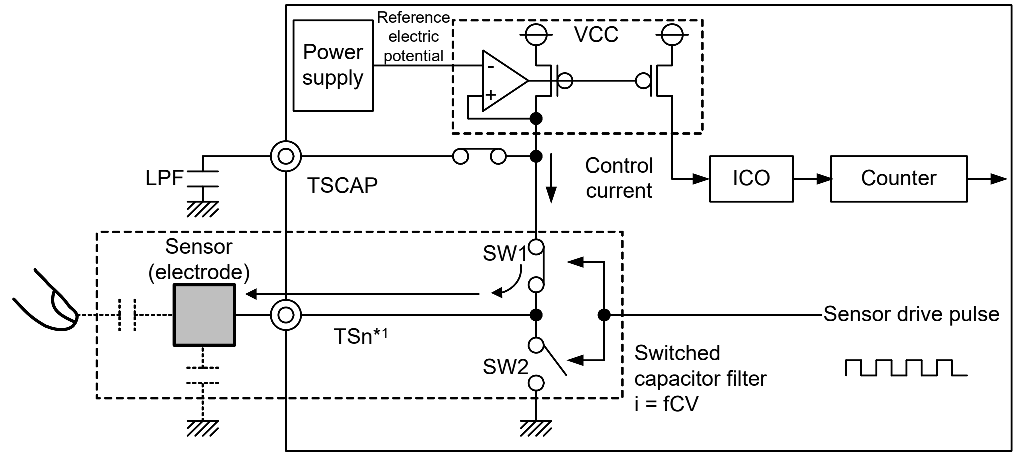

Power supply
Reference electric potential
VCC
LPF
TSCAP
Control current
ICO
Counter
Sensor (electrode)
SW1
TSn*1
SW2
Sensor drive pulse
Switched capacitor filter
R = 1/(fC)

Note 1. n = 00, 01, 03, 10, 12, 13, 18, 28, 30, 31, 34

Figure 40.4 Measurement circuit

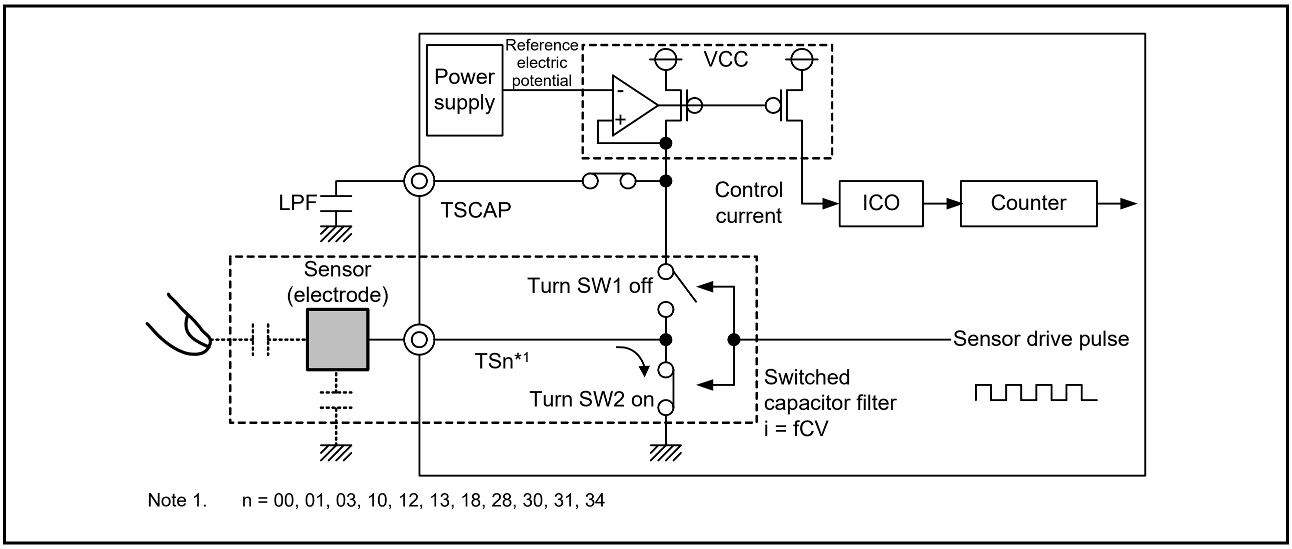
Figure 40.5 to Figure 40.7 explain the electrostatic capacitance measurement operation principles of the CTSU current frequency conversion method. The operation is as follows:

1. The electrostatic capacitance of the electrode is charged by turning SW1 on and SW2 off (Figure 40.5).
2. The charged capacitance is discharged by turning SW1 off and SW2 on (See Figure 40.6).
3. Current flows to the switched capacitor filter by repeatedly charging and discharging the electrodes as in step 1. and step 2.. At this point, if a finger is in close proximity, the capacitance and the flowing current change. A clock is generated by supplying the control current, a 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 40.7).


Power supply
Reference electric potential
VCC
LPF
TSCAP
Control current
ICO
Counter
Sensor (electrode)
SW1
TSn*1
SW2
Sensor drive pulse
Switched capacitor filter
i = fCV

Note 1. n = 00, 01, 03, 10, 12, 13, 18, 28, 30, 31, 34

Figure 40.5 Charging operation


Power supply
Reference electric potential
VCC
LPF
TSCAP
Control current
ICO
Counter
Sensor (electrode)
Turn SW1 off
TSn*1
Turn SW2 on
Sensor drive pulse
Switched capacitor filter
i = fCV

Note 1. n = 00, 01, 03, 10, 12, 13, 18, 28, 30, 31, 34

Figure 40.6 Discharging operation

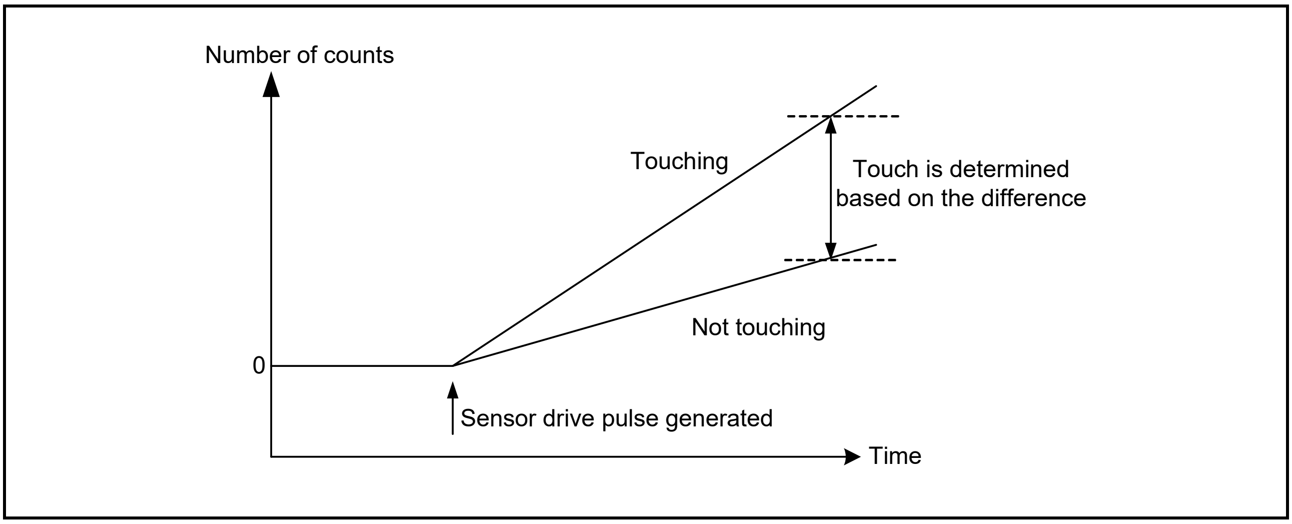


Figure 40.7 Change in measured value when finger is touching and not touching

40.3.2 Measurement Modes

The CTSU supports self-capacitance and mutual-capacitance methods. Figure 40.8 shows these methods.

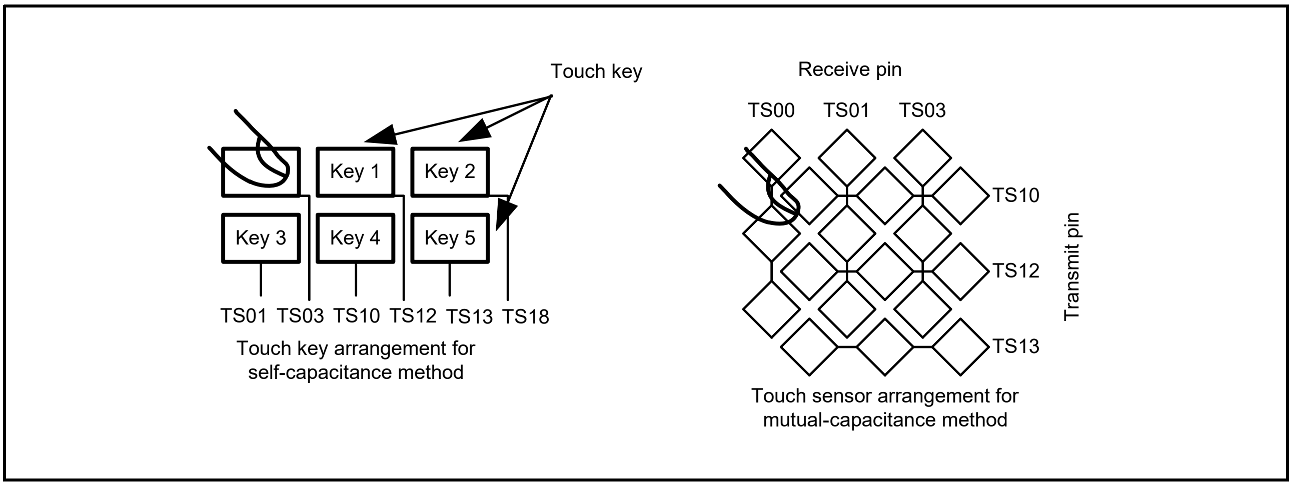
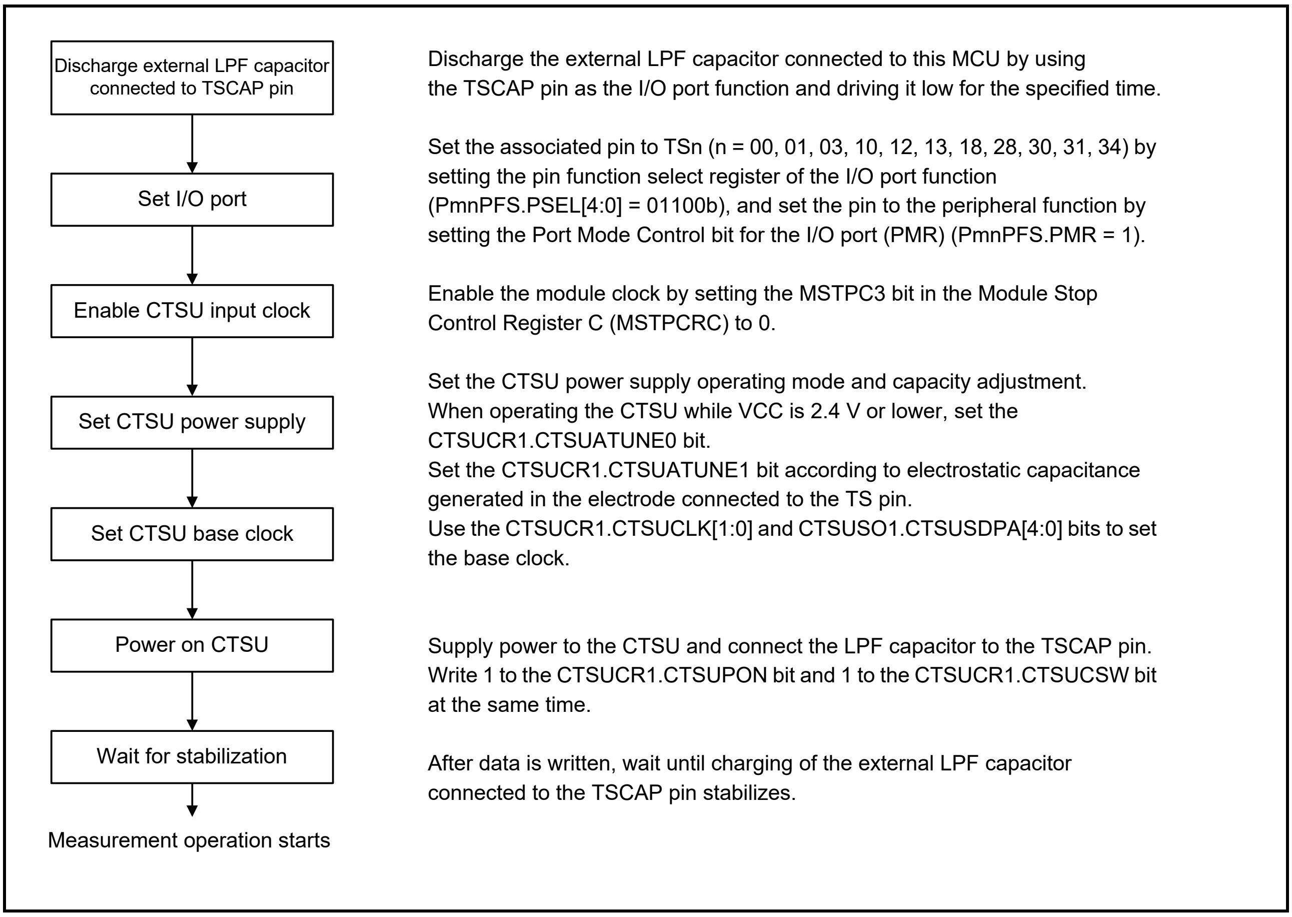

Touch key
Key 1
Key 2
Key 3
Key 4
Key 5
TS01 TS03 TS10 TS12 TS13 TS18
Touch key arrangement for self-capacitance method
Receive pin
TS00 TS01 TS03
TS10
TS12
TS13
Transmit pin
Touch sensor arrangement for mutual-capacitance method

Figure 40.8 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, single scan and multi-scan can be used as measurement modes. In the mutual-capacitance method, the capacitance between two opposing electrodes (transmit and receive pins) is measured.

40.3.2.1 Initial setting flow

Figure 40.9 shows the flow for the CTSU initial setting.


Discharge external LPF capacitor connected to TSCAP pin
Set I/O port
Enable CTSU input clock
Set CTSU power supply
Set CTSU base clock
Power on CTSU
Wait for stabilization
Measurement operation starts

Discharge the external LPF capacitor connected to this MCU by using the TSCAP pin as the I/O port function and driving it low for the specified time.

Set the associated pin to TSn (n = 00, 01, 03, 10, 12, 13, 18, 28, 30, 31, 34) by setting the pin function select register of the I/O port function (PmnPFS.PSEL[4:0] = 01100b), and set the pin to the peripheral function by setting the Port Mode Control bit for the I/O port (PMR) (PmnPFS.PMR = 1).

Enable the module clock by setting the MSTPC3 bit in the Module Stop Control Register C (MSTPCRC) to 0.

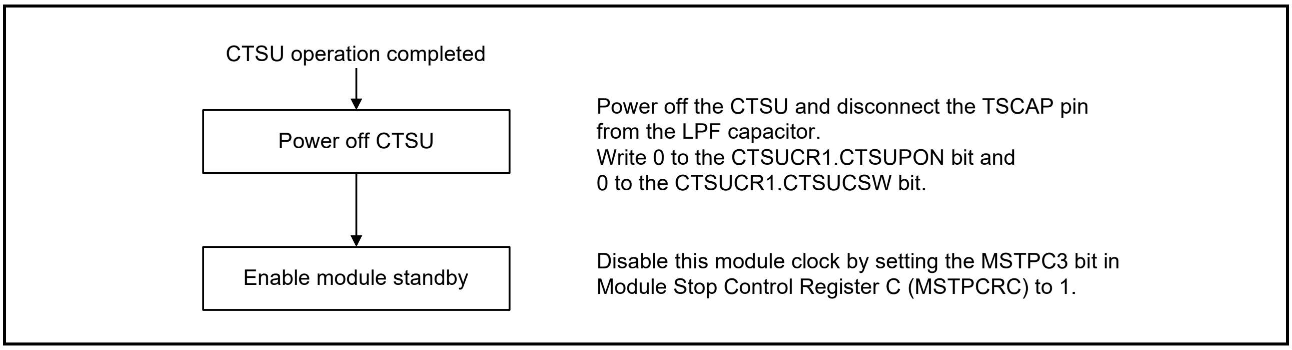
Set the CTSU power supply operating mode and capacity adjustment.
When operating the CTSU while VCC is 2.4 V or lower, set the CTSUCR1.CTSUATUNE0 bit.
Set the CTSUCR1.CTSUATUNE1 bit according to electrostatic capacitance generated in the electrode connected to the TS pin.
Use the CTSUCR1.CTSUCLK[1:0] and CTSUSO1.CTSUSDPA[4:0] bits to set the base clock.

Supply power to the CTSU and connect the LPF capacitor to the TSCAP pin. Write 1 to the CTSUCR1.CTSUPON bit and 1 to the CTSUCR1.CTSUCSW bit at the same time.

After data is written, wait until charging of the external LPF capacitor connected to the TSCAP pin stabilizes.

Figure 40.9 CTSU initial setting flow

Figure 40.10 shows the flow for stopping CTSU operation and invoking the standby state.


CTSU operation completed
Power off CTSU
Enable module standby

Power off the CTSU and disconnect the TSCAP pin from the LPF capacitor.
Write 0 to the CTSUCR1.CTSUPON bit and 0 to the CTSUCR1.CTSUCSW bit.

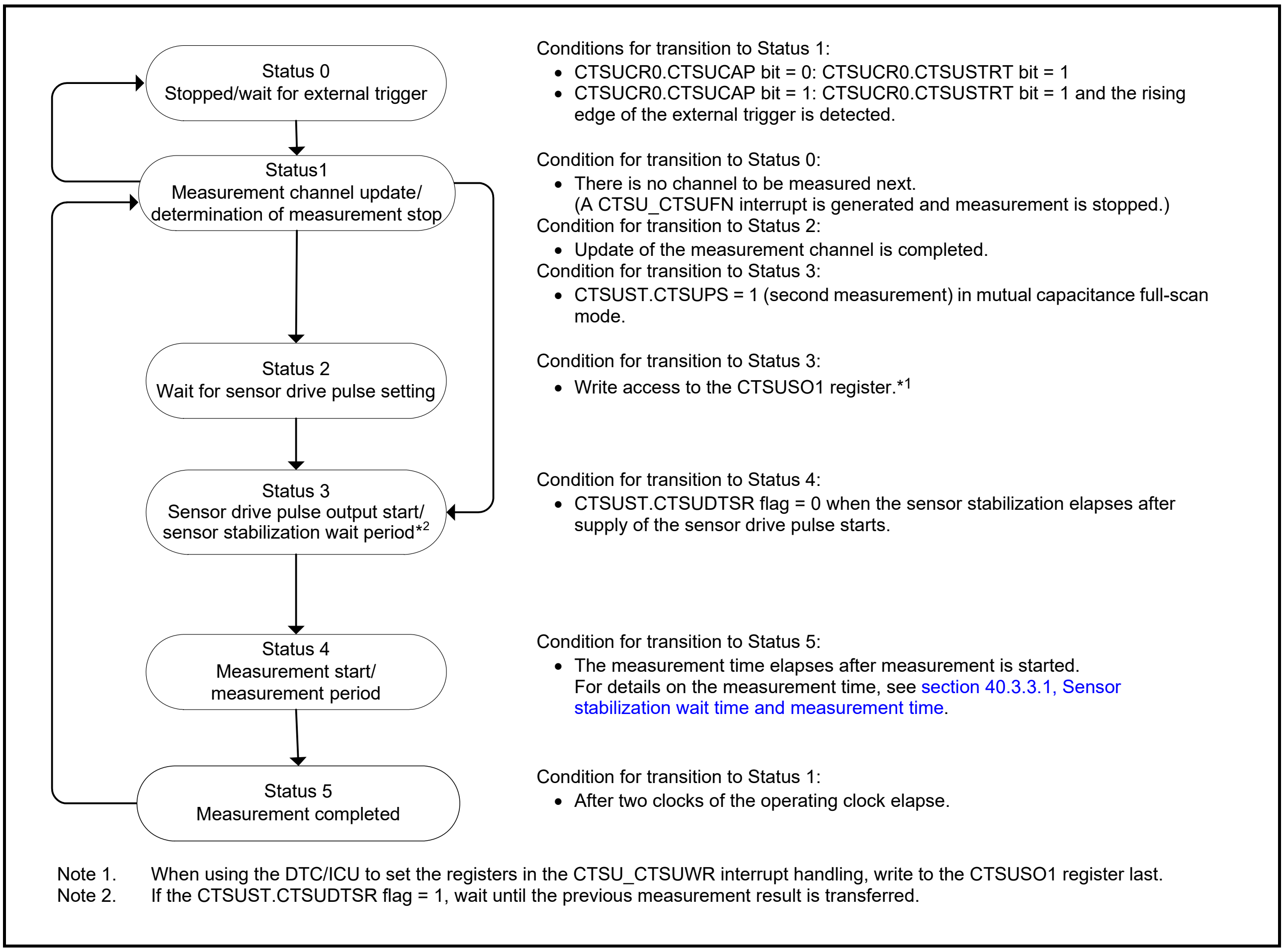
Disable this module clock by setting the MSTPC3 bit in Module Stop Control Register C (MSTPCRC) to 1.

Figure 40.10 CTSU stopping flow

To restart operation, follow the initial setting flow shown in Figure 40.9.

40.3.2.2 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 40.11 shows the status operation transitions.


Status 0
Stopped/wait for external trigger
Status1
Measurement channel update/
determination of measurement stop
Status 2
Wait for sensor drive pulse setting
Status 3
Sensor drive pulse output start/
sensor stabilization wait period*2
Status 4
Measurement start/
measurement period
Status 5
Measurement completed

Conditions for transition to Status 1:

- CTSUCR0.CTSUCAP bit = 0: CTSUCR0.CTSUSTRT bit = 1
- CTSUCR0.CTSUCAP bit = 1: CTSUCR0.CTSUSTRT bit = 1 and the rising edge of the external trigger is detected.

Condition for transition to Status 0:

- There is no channel to be measured next.
  (A CTSU_CTSUFN interrupt is generated and measurement is stopped.)

Condition for transition to Status 2:

- Update of the measurement channel is completed.

Condition for transition to Status 3:

- CTSUST.CTSUPS = 1 (second measurement) in mutual capacitance full-scan mode.

Condition for transition to Status 3:

- Write access to the CTSUSO1 register.*1

Condition for transition to Status 4:

- CTSUST.CTSUDTSR flag = 0 when the sensor stabilization elapses after supply of the sensor drive pulse starts.

Condition for transition to Status 5:

- The measurement time elapses after measurement is started.
  For details on the measurement time, see section 40.3.3.1, Sensor stabilization wait time and measurement time.

Condition for transition to Status 1:

- After two clocks of the operating clock elapse.

Note 1. When using the DTC/ICU to set the registers in the CTSU_CTSUWR interrupt handling, write to the CTSUSO1 register last.
Note 2. If the CTSUST.CTSUDTSR flag = 1, wait until the previous measurement result is transferred.

Figure 40.11 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.

If the channel to be measured is not set in the CTSUMCH0, CTSUCHAC0 to CTSUCHAC4, and CTSUCHTRC0 to CTSUCHTRC4 registers, a CTSU_CTSUFN interrupt is generated immediately after a transition to Status 1, then the status transitions to Status 0. The following are cases when 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.

40.3.2.3 Self-capacitance single scan mode operation

In self-capacitance single scan mode, electrostatic capacitance is measured on one channel. Figure 40.12 shows the software flow and an operation example, and Figure 40.13 shows the timing.

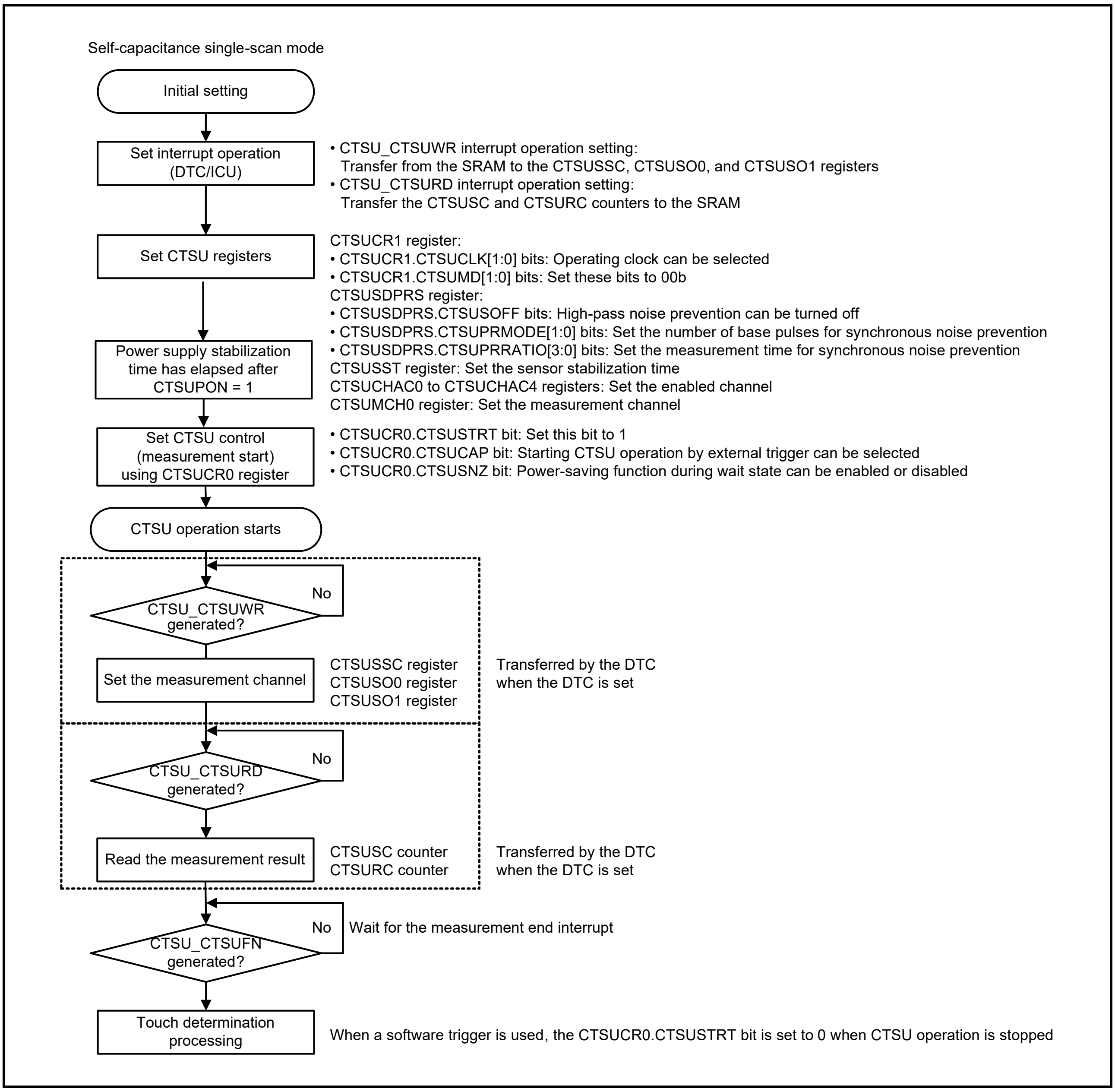

Self-capacitance single-scan mode
Initial setting
Set interrupt operation (DTC/ICU)
• CTSU_CTSUWR interrupt operation setting:
Transfer from the SRAM to the CTSUSSC, CTSUSO0, and CTSUSO1 registers
• CTSU_CTSURD interrupt operation setting:
Transfer the CTSUSC and CTSURC counters to the SRAM
Set CTSU registers
CTSUCR1 register:
• CTSUCR1.CTSUCLK[1:0] bits: Operating clock can be selected
• CTSUCR1.CTSUMD[1:0] bits: Set these bits to 00b
CTSUSDPRS register:
• CTSUSDPRS.CTSUSOFF bits: High-pass noise prevention can be turned off
• CTSUSDPRS.CTSUPRMODE[1:0] bits: Set the number of base pulses for synchronous noise prevention
• CTSUSDPRS.CTSUPRRATIO[3:0] bits: Set the measurement time for synchronous noise prevention
CTSUSST register: Set the sensor stabilization time
CTSUCHAC0 to CTSUCHAC4 registers: Set the enabled channel
CTSUMCH0 register: Set the measurement channel
Power supply stabilization time has elapsed after CTSUPON = 1
Set CTSU control (measurement start) using CTSUCR0 register
• CTSUCR0.CTSUSTRT bit: Set this bit to 1
• CTSUCR0.CTSUCAP bit: Starting CTSU operation by external trigger can be selected
• CTSUCR0.CTSUSNZ bit: Power-saving function during wait state can be enabled or disabled
CTSU operation starts
CTSU_CTSUWR generated?
No
Set the measurement channel
CTSUSSC register
CTSUSO0 register
CTSUSO1 register
Transferred by the DTC when the DTC is set
CTSU_CTSURD generated?
No
Read the measurement result
CTSUSC counter
CTSURC counter
Transferred by the DTC when the DTC is set
CTSU_CTSUFN generated?
No
Wait for the measurement end interrupt
Touch determination processing
When a software trigger is used, the CTSUCR0.CTSUSTRT bit is set to 0 when CTSU operation is stopped

Figure 40.12 Software flow and operation example for self-capacitance single scan mode

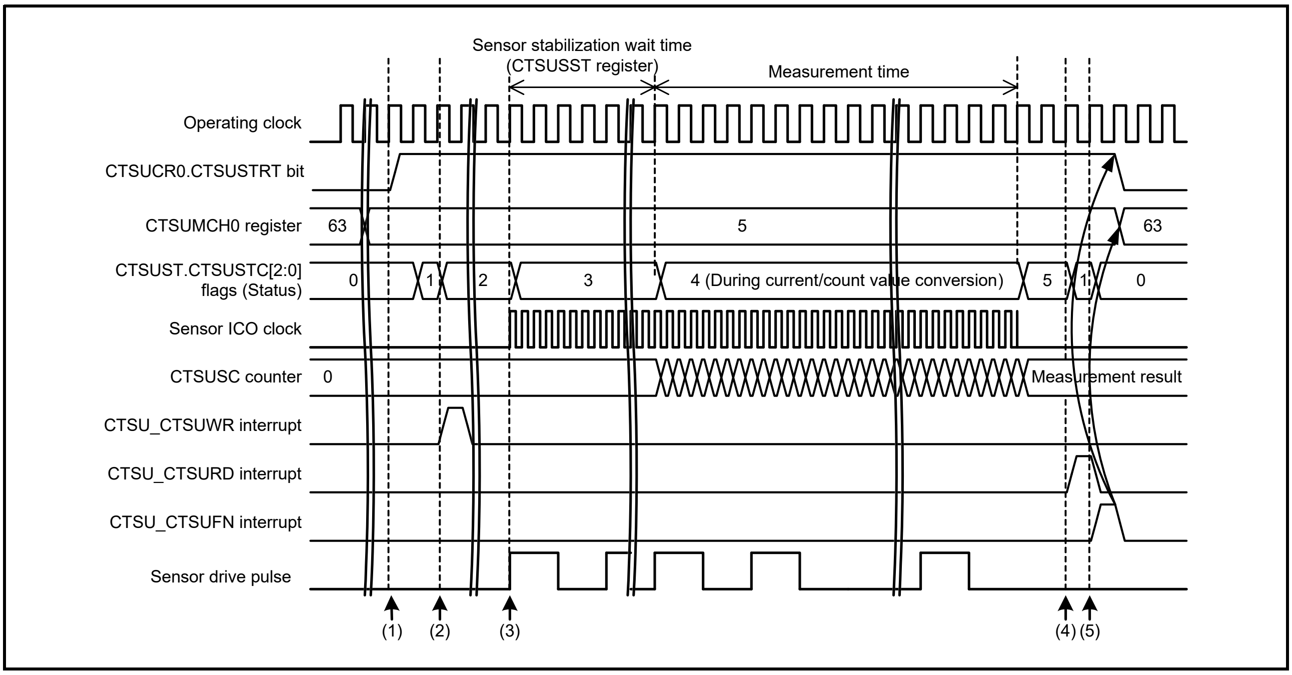

Sensor stabilization wait time
(CTSUSST register)
Measurement time
Operating clock
CTSUCR0.CTSUSTRT bit
CTSUMCH0 register
63
5
63
CTSUST.CTSUSTC[2:0] flags (Status)
0
1
2
3
4 (During current/count value conversion)
5
1
0
Sensor ICO clock
CTSUSC counter
0
Measurement result
CTSU_CTSUWR interrupt
CTSU_CTSURD interrupt
CTSU_CTSUFN interrupt
Sensor drive pulse
(1)
(2)
(3)
(4)
(5)

Figure 40.13 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 40.13:

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).

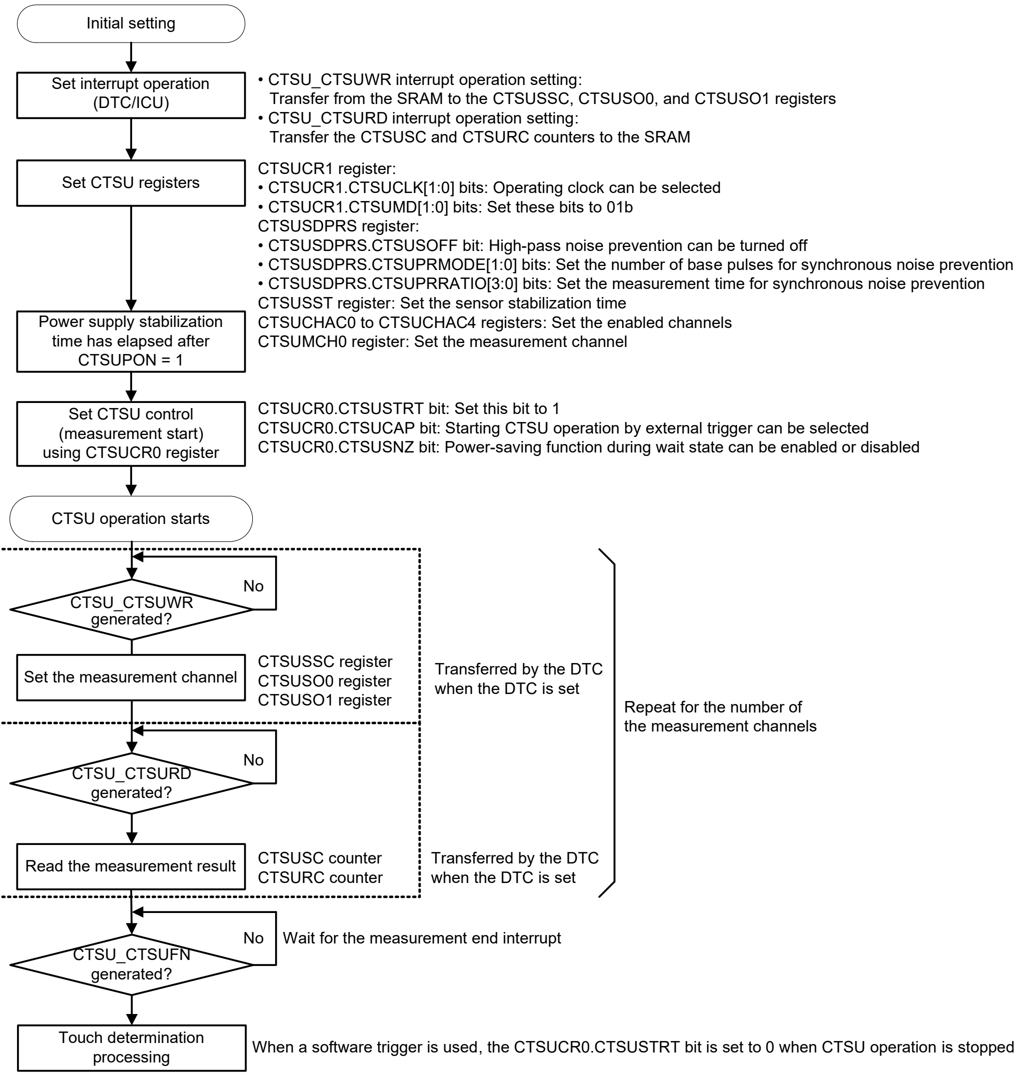
Table 40.6 lists the touch pin states in self-capacitance single scan mode.

Table 40.6 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

40.3.2.4 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 40.14 shows the software flow and an operation example, and Figure 40.15 shows the timing.


Initial setting
Set interrupt operation (DTC/ICU)
• CTSU_CTSUWR interrupt operation setting:
Transfer from the SRAM to the CTSUSSC, CTSUSO0, and CTSUSO1 registers
• CTSU_CTSURD interrupt operation setting:
Transfer the CTSUSC and CTSURC counters to the SRAM
Set CTSU registers
CTSUCR1 register:
• CTSUCR1.CTSUCLK[1:0] bits: Operating clock can be selected
• CTSUCR1.CTSUMD[1:0] bits: Set these bits to 01b
CTSUSDPRS register:
• CTSUSDPRS.CTSUSOFF bit: High-pass noise prevention can be turned off
• CTSUSDPRS.CTSUPRMODE[1:0] bits: Set the number of base pulses for synchronous noise prevention
• CTSUSDPRS.CTSUPRRATIO[3:0] bits: Set the measurement time for synchronous noise prevention
CTSUSST register: Set the sensor stabilization time
CTSUCHAC0 to CTSUCHAC4 registers: Set the enabled channels
CTSUMCH0 register: Set the measurement channel
Power supply stabilization time has elapsed after CTSUPON = 1
Set CTSU control (measurement start) using CTSUCR0 register
CTSUCR0.CTSUSTRT bit: Set this bit to 1
CTSUCR0.CTSUCAP bit: Starting CTSU operation by external trigger can be selected
CTSUCR0.CTSUSNZ bit: Power-saving function during wait state can be enabled or disabled
CTSU operation starts
CTSU_CTSUWR generated?
No
Set the measurement channel
CTSUSSC register
CTSUSO0 register
CTSUSO1 register
Transferred by the DTC when the DTC is set
Repeat for the number of the measurement channels
CTSU_CTSURD generated?
No
Read the measurement result
CTSUSC counter
CTSURC counter
Transferred by the DTC when the DTC is set
CTSU_CTSUFN generated?
No
Wait for the measurement end interrupt
Touch determination processing
When a software trigger is used, the CTSUCR0.CTSUSTRT bit is set to 0 when CTSU operation is stopped

Channel measurement sequence in self-capacitance multiscan mode

Setting:
- Select self-capacitance multi-scan mode (CTSUCR1.CTSUMD[1:0] bits = 01b)
- Set channels 0 and 3 to enabled channels (CTSUCHAC0.CTSUCHAC0[7:0] bits = 00001001b)

Receive Channels							
Channel 7	Channel 6	Channel 5	Channel 4	Channel 3	Channel 2	Channel 1	Channel 0
				(2)			(1)

Figure 40.14 Software flow and example operation for self-capacitance multi-scan mode

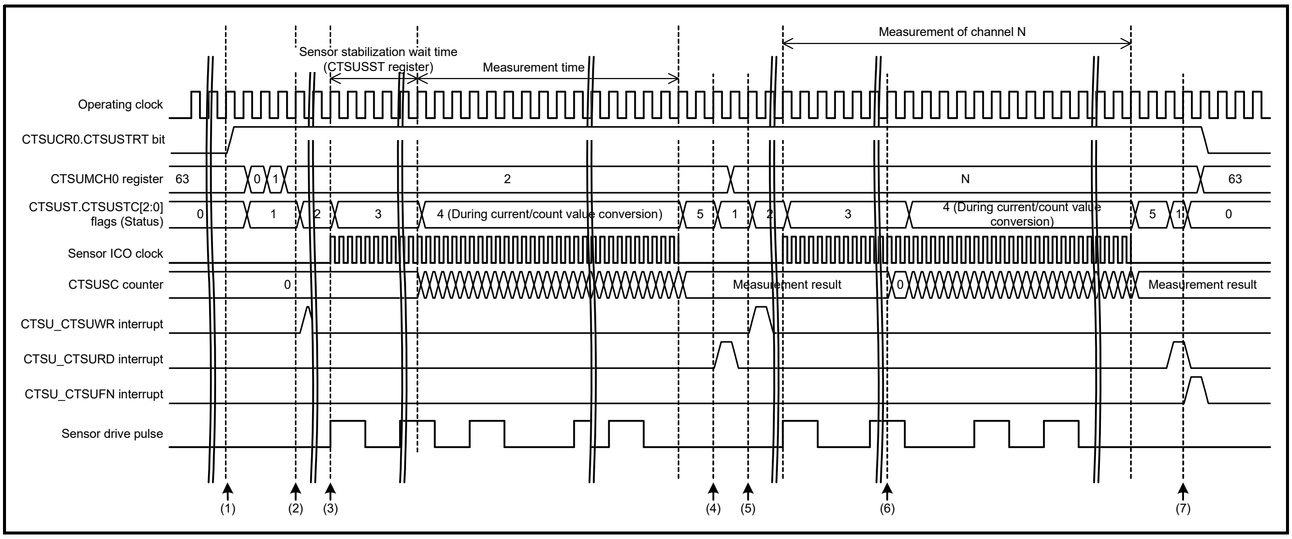

Measurement of channel N
Sensor stabilization wait time
(CTSUSST register)
Measurement time
Operating clock
CTSUCR0.CTSUSTRT bit
CTSUMCH0 register
63
0
1
2
N
63
CTSUST.CTSUSTC[2:0] flags (Status)
0
1
2
3
4 (During current/count value conversion)
5
1
2
3
4 (During current/count value conversion)
5
1
0
Sensor ICO clock
CTSUSC counter
0
Measurement result
0
Measurement result
CTSU_CTSUWR interrupt
CTSU_CTSURD interrupt
CTSU_CTSUFN interrupt
Sensor drive pulse
(1)
(2)
(3)
(4)
(5)
(6)
(7)

Figure 40.15 **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 40.15:

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 40.7 lists the touch pin states in self-capacitance multi-scan mode.

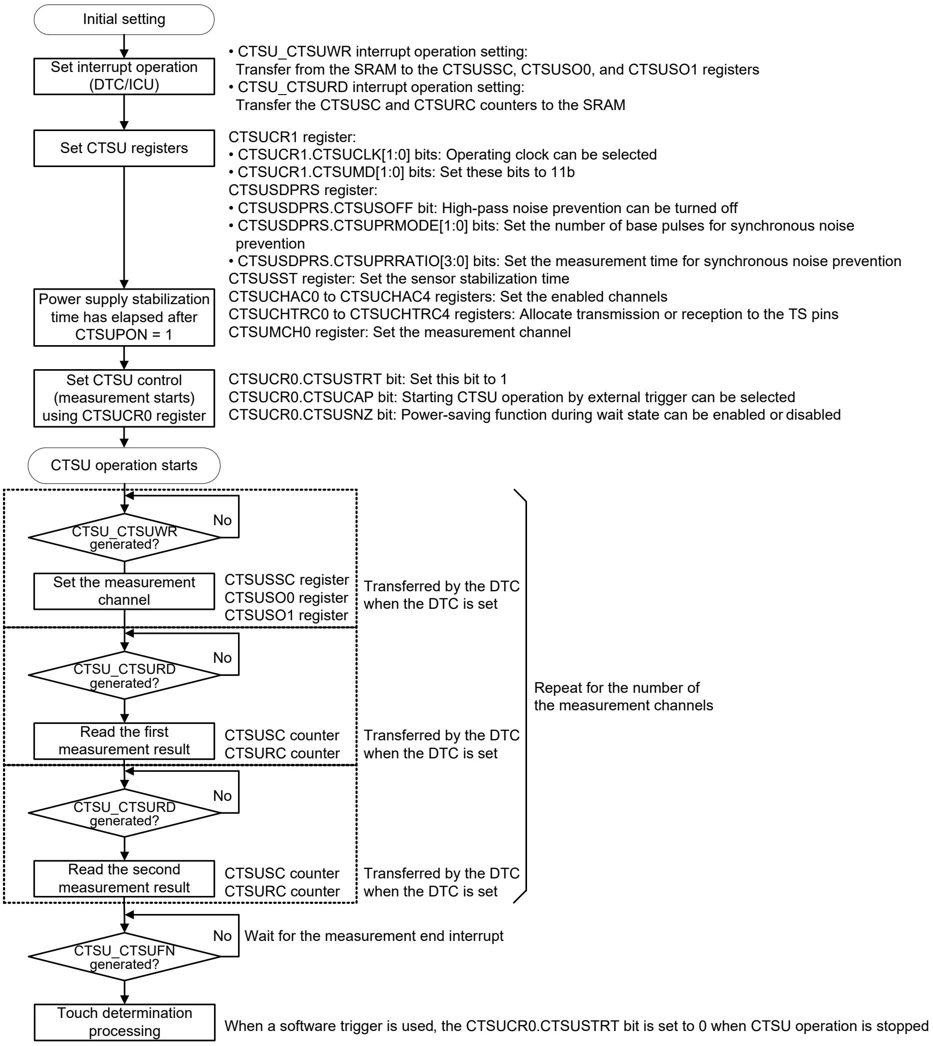
Table 40.7 **Touch pin states in self-capacitance multi-scan mode**

	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

40.3.2.5 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 40.16 shows the software flow and an operation example, and Figure 40.17 shows the timing.


Initial setting
Set interrupt operation (DTC/ICU)
Set CTSU registers
Power supply stabilization time has elapsed after CTSUPON = 1
Set CTSU control (measurement starts) using CTSUCR0 register
CTSU operation starts
CTSU_CTSUWR generated?
No
Set the measurement channel
CTSU_CTSURD generated?
No
Read the first measurement result
CTSU_CTSURD generated?
No
Read the second measurement result
CTSU_CTSUFN generated?
No
Touch determination processing

- CTSU_CTSUWR interrupt operation setting:
  Transfer from the SRAM to the CTSUSSC, CTSUSO0, and CTSUSO1 registers
- CTSU_CTSURD interrupt operation setting:
  Transfer the CTSUSC and CTSURC counters to the SRAM

CTSUCR1 register:
- CTSUCR1.CTSUCLK[1:0] bits: Operating clock can be selected
- CTSUCR1.CTSUMD[1:0] bits: Set these bits to 11b

CTSUSDPRS register:
- CTSUSDPRS.CTSUSOFF bit: High-pass noise prevention can be turned off
- CTSUSDPRS.CTSUPRMODE[1:0] bits: Set the number of base pulses for synchronous noise prevention
- CTSUSDPRS.CTSUPRRATIO[3:0] bits: Set the measurement time for synchronous noise prevention

CTSUSST register: Set the sensor stabilization time
CTSUCHAC0 to CTSUCHAC4 registers: Set the enabled channels
CTSUCHTRC0 to CTSUCHTRC4 registers: Allocate transmission or reception to the TS pins
CTSUMCH0 register: Set the measurement channel

CTSUCR0.CTSUSTRT bit: Set this bit to 1
CTSUCR0.CTSUCAP bit: Starting CTSU operation by external trigger can be selected
CTSUCR0.CTSUSNZ bit: Power-saving function during wait state can be enabled or disabled

CTSUSSC register
CTSUSO0 register
CTSUSO1 register
Transferred by the DTC when the DTC is set

CTSUSC counter
CTSURC counter
Transferred by the DTC when the DTC is set

CTSUSC counter
CTSURC counter
Transferred by the DTC when the DTC is set

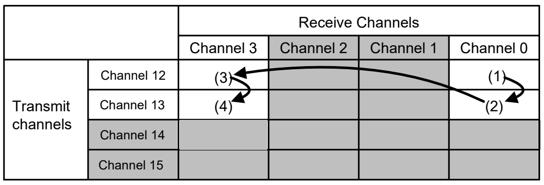
Repeat for the number of the measurement channels

Wait for the measurement end interrupt

When a software trigger is used, the CTSUCR0.CTSUSTRT bit is set to 0 when CTSU operation is stopped

Channel measurement sequence in mutual capacitance full scan mode

Setting:
- Select mutual-capacitance full-scan mode (CTSUCR1.CTSUMD[1:0] = 11b)
- Set channels 0, 3, 12, and 13 to enabled channels (CTSUCHAC0.CTSUCHAC0[7:0] bits = 00001001b, CTSUCHAC1.CTSUCHAC1[7:0] bits = 00110000b)
- Set channels 0 and 3 to receive channels and channels 12 and 13 to transmit channels (CTSUCHTRC0.CTSUCHTRC0[7:0] bits = 00000000b, CTSUCHTRC1.CTSUCHTRC1[7:0] bits = 00110000b)



		Receive Channels			
		Channel 3	Channel 2	Channel 1	Channel 0
Transmit channels	Channel 12	(3)			(1)
	Channel 13	(4)			(2)
	Channel 14				
	Channel 15				

Figure 40.16 Software flow and operation example for mutual capacitance full-scan mode

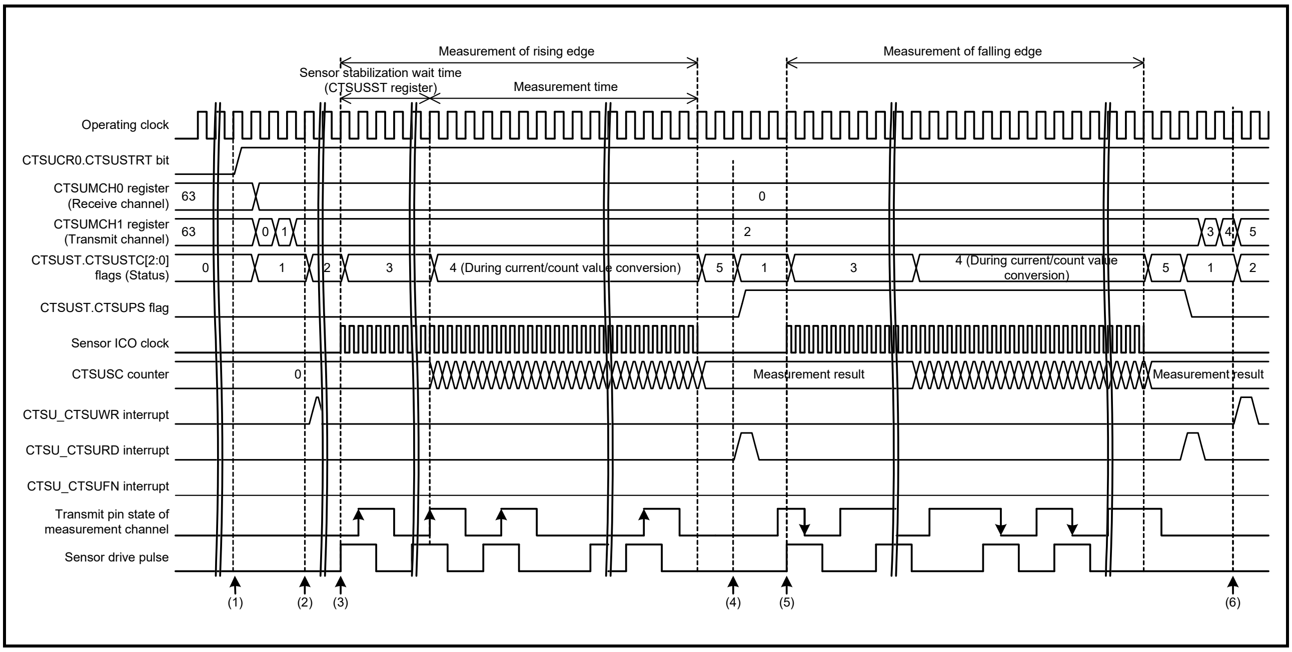

Measurement of rising edge
Sensor stabilization wait time (CTSUSST register)
Measurement time
Measurement of falling edge
Operating clock
CTSUCR0.CTSUSTRT bit
CTSUMCH0 register (Receive channel)
63
0
CTSUMCH1 register (Transmit channel)
63
0
1
2
3
4
5
CTSUST.CTSUSTC[2:0] flags (Status)
0
1
2
3
4 (During current/count value conversion)
5
1
3
4 (During current/count value conversion)
5
1
2
CTSUST.CTSUPS flag
Sensor ICO clock
CTSUSC counter
0
Measurement result
Measurement result
CTSU_CTSUWR interrupt
CTSU_CTSURD interrupt
CTSU_CTSUFN interrupt
Transmit pin state of measurement channel
Sensor drive pulse
(1)
(2)
(3)
(4)
(5)
(6)

Figure 40.17 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 40.17:

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 40.8 lists the touch pin states in mutual capacitance full-scan mode.

Table 40.8 Touch pin states in mutual capacitance full-scan mode (1 of 2)

	Touch pin for receive channels		Touch pin for transmit channels		
Status	Measured channel	Non-measured channel	Measured channel	Non-measured channel	Remarks
0	Low	Low	Low	Low	-
1	Low	Low	Low/High	Low	-
2	Low	Low	Low	Low	-

Table 40.8 Touch pin states in mutual capacitance full-scan mode (2 of 2)

	Touch pin for receive channels		Touch pin for transmit channels		
Status	Measured channel	Non-measured channel	Measured channel	Non-measured channel	Remarks
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	-

40.3.3 Parameters Common to Multiple Modes

40.3.3.1 Sensor stabilization wait time and measurement time

Figure 40.18 shows the timing of the sensor stabilization wait and measurement.

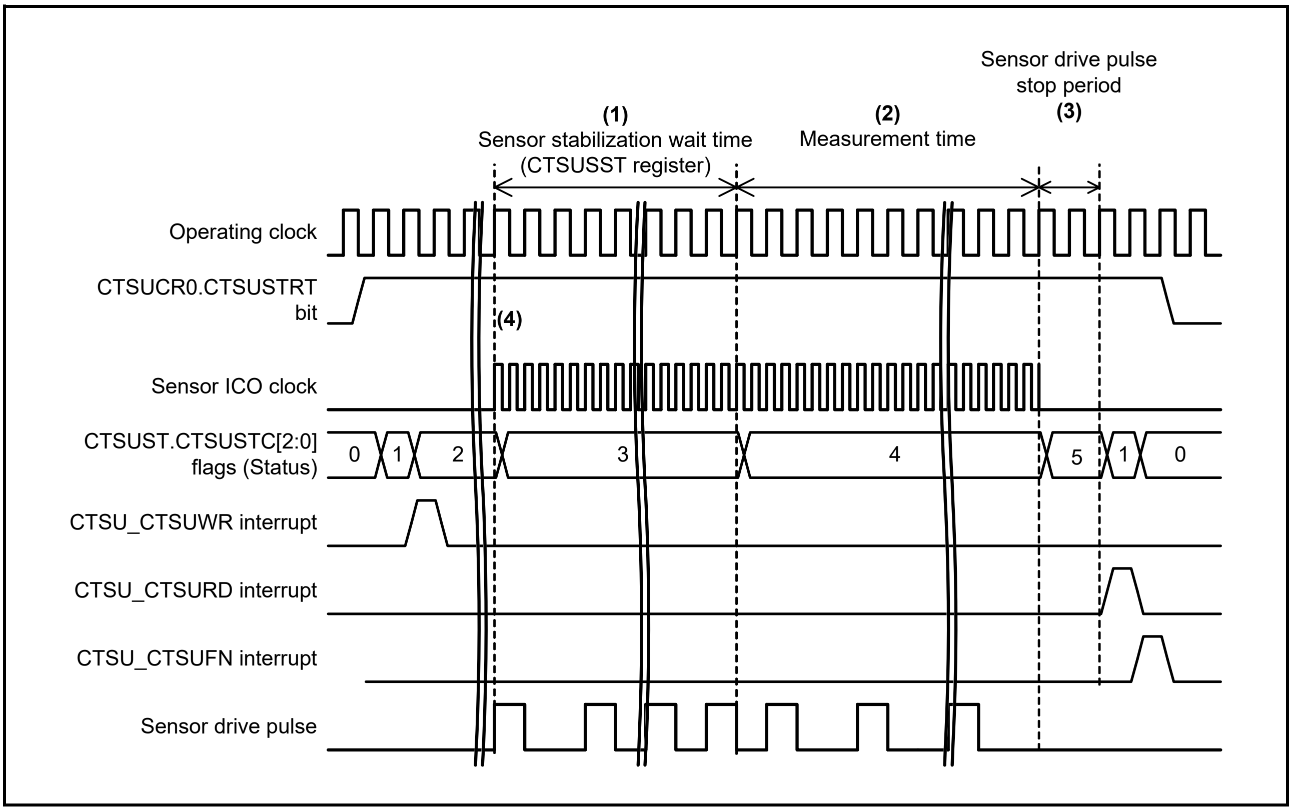

Sensor drive pulse stop period
(3)
(1)
Sensor stabilization wait time
(CTSUSST register)
(2)
Measurement time
Operating clock
CTSUCR0.CTSUSTRT bit
(4)
Sensor ICO clock
CTSUST.CTSUSTC[2:0] flags (Status)
0
1
2
3
4
5
1
0
CTSU_CTSUWR interrupt
CTSU_CTSURD interrupt
CTSU_CTSUFN interrupt
Sensor drive pulse

Figure 40.18 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 2 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 is completed, 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).

40.3.3.2 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 40.19). Because write access to the CTSUSO1 register controls the transition to the next status, be sure to set this register last.

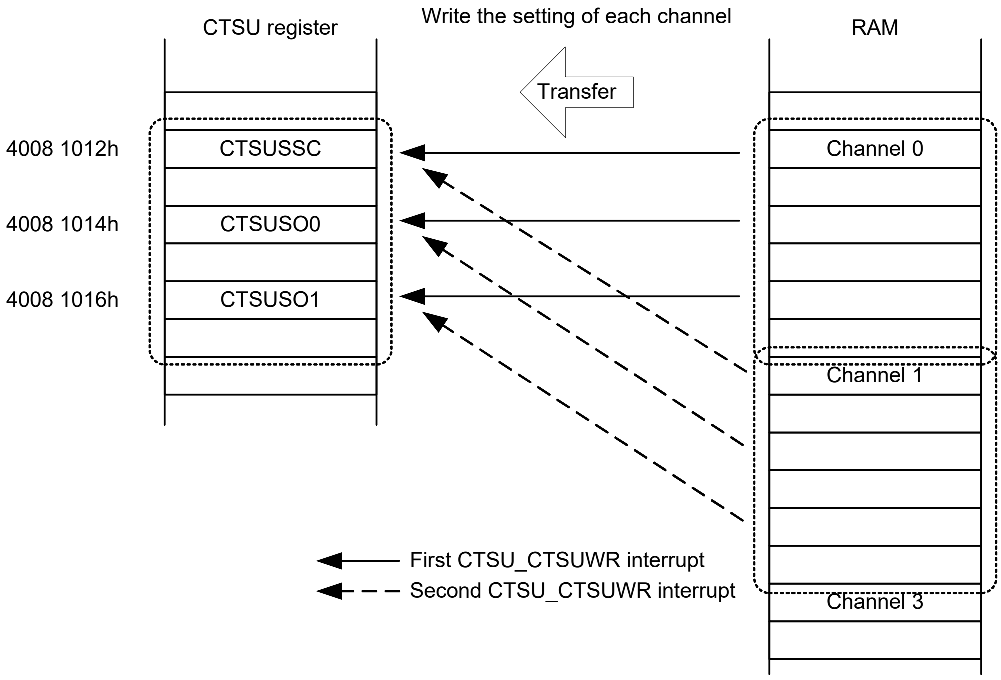

CTSU register
Write the setting of each channel
RAM
Transfer
4008 1012h
4008 1014h
4008 1016h
CTSUSSC
CTSUSO0
CTSUSO1
Channel 0
Channel 1
Channel 3
First CTSU_CTSUWR interrupt
Second CTSU_CTSUWR interrupt

Figure 40.19 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 40.20.

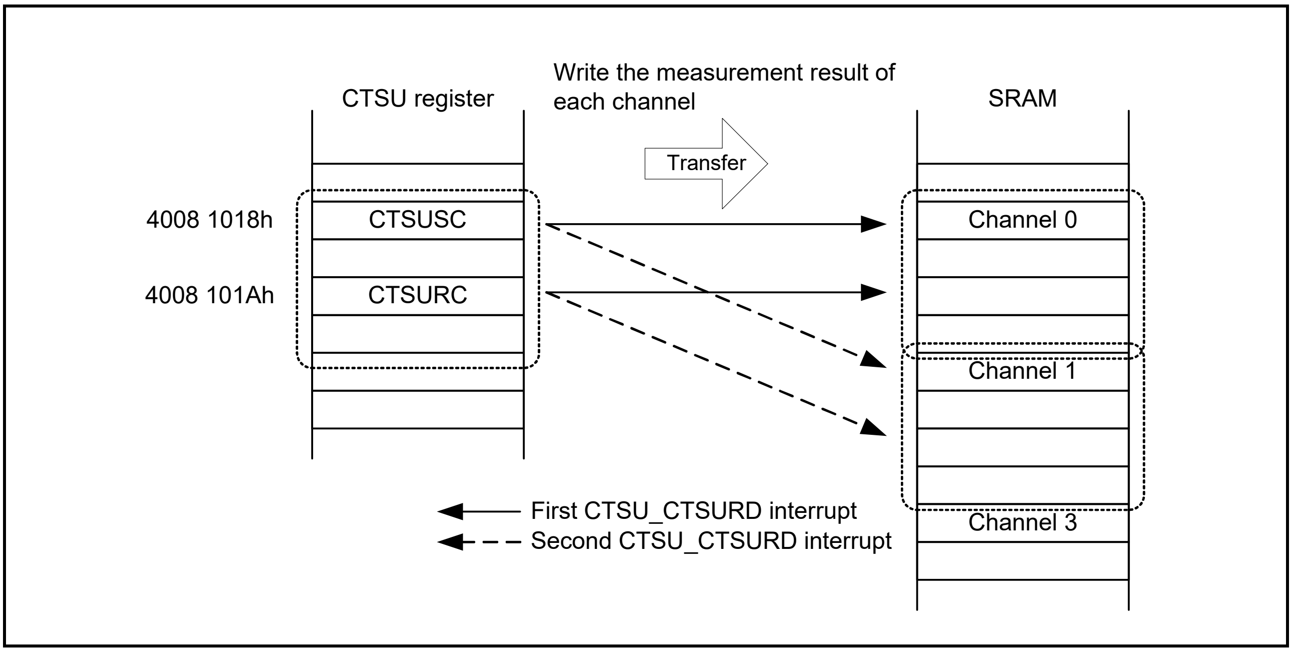

CTSU register
Write the measurement result of each channel
SRAM
Transfer
4008 1018h
CTSUSC
4008 101Ah
CTSURC
Channel 0
Channel 1
Channel 3
First CTSU_CTSURD interrupt
Second CTSU_CTSURD interrupt

Figure 40.20 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.

40.4 Usage Notes

40.4.1 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.

40.4.2 Constraints on 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.

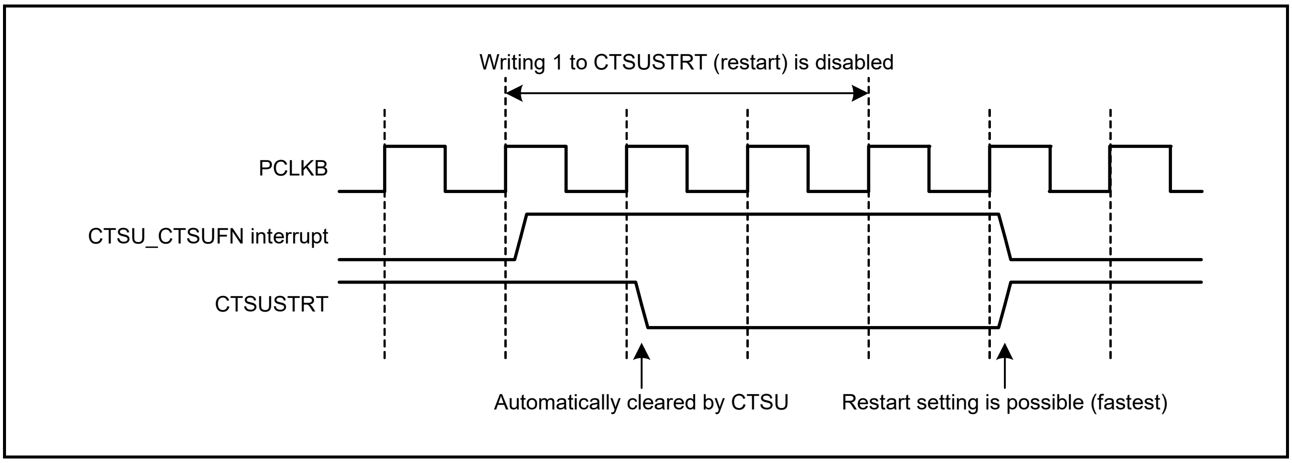

Writing 1 to CTSUSTRT (restart) is disabled
PCLKB
CTSU_CTSUFN interrupt
CTSUSTRT
Automatically cleared by CTSU
Restart setting is possible (fastest)

Figure 40.21 Notes on restarting measurement

40.4.3 Constraints on 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).

40.4.4 Constraints on Forced Stops

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 initialization of 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, also 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.

40.4.5 TSCAP Pin

The TSCAP pin requires an external decoupling capacitor to stabilize 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 a low level, before turning on the switch (CTSUCR1.CTSUCSW bit = 1) to establish a connection.

40.4.6 Constraints 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 40.9.

41. Data Operation Circuit (DOC)

41.1 Overview

The Data Operation Circuit (DOC) compares, adds, and subtracts 16-bit data. An interrupt can be generated when a selected condition applies.

Table 41.1 lists the DOC specifications and Figure 41.1 shows a block diagram.

Table 41.1 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.

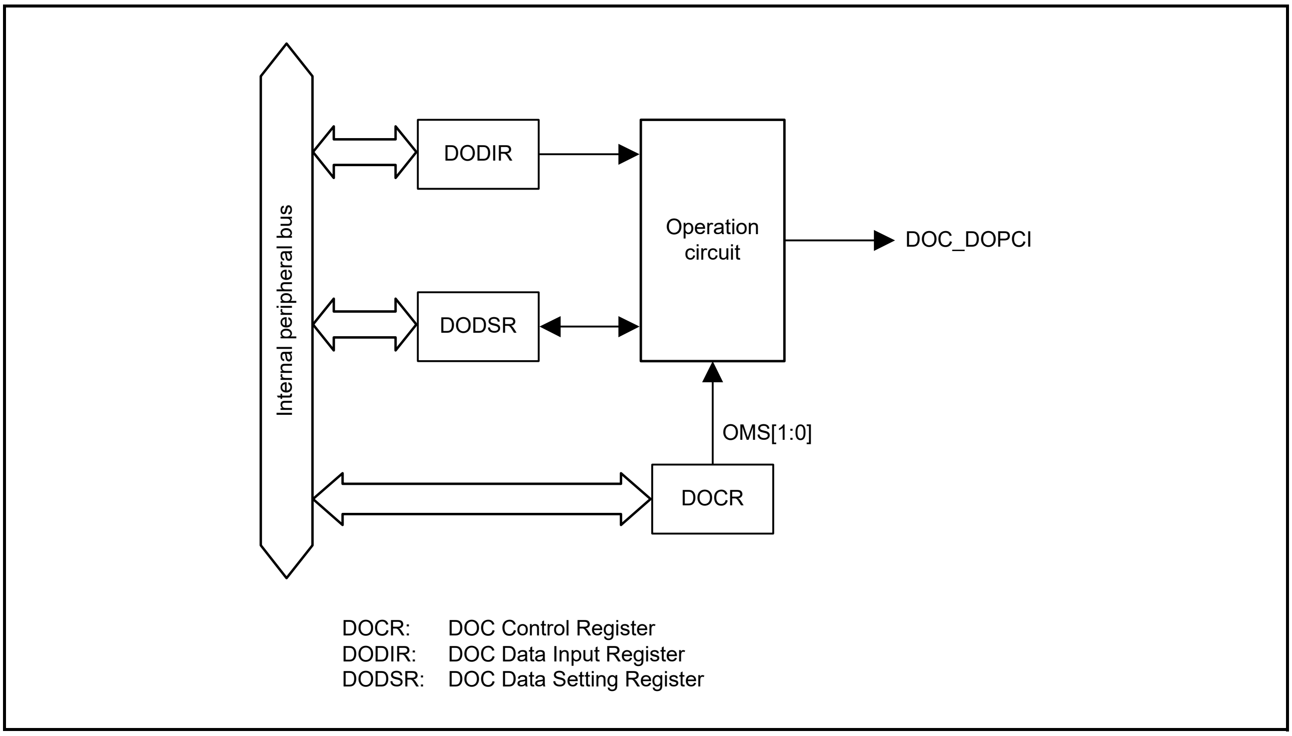

Internal peripheral bus
DODIR
DODSR
Operation circuit
DOC_DOPCI
OMS[1:0]
DOCR
DOCR: DOC Control Register
DODIR: DOC Data Input Register
DODSR: DOC Data Setting Register

Figure 41.1 DOC block diagram

41.2 Register Descriptions

41.2.1 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	These 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.

41.2.2 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

DODIR is a 16-bit read/write register that stores 16-bit data used in all operations.

41.2.3 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

DODSR 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.

41.3 Operation

41.3.1 Data Comparison Mode

Figure 41.2 shows an example DOC operation in data comparison mode. The following sequence is an example of operation when DCSEL 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 for comparison to the DODIR register.
4. Continue writing 16-bit data until all data to be compared is written to DODIR.
5. If a value written to DODIR does not match that in DODSR, the DOCR.DOPCF flag is set to 1.

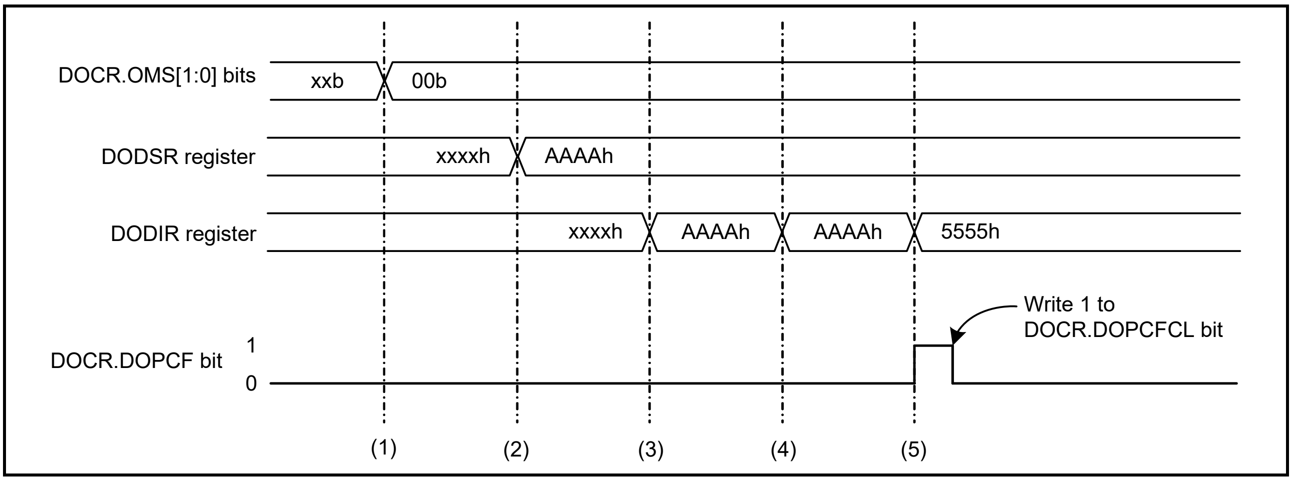


Figure 41.2 Example operation in data comparison mode

41.3.2 Data Addition Mode

Figure 41.3 shows an example for 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 the 16-bit data in the DODSR register as the initial value.
3. Write the 16-bit data to be added to the DODIR register. The result of the operation is stored in DODSR.
4. Continue writing 16-bit data until all data to be added is written to DODIR.
5. If the result of an operation is greater than FFFFh, the DOCR.DOPCF flag is set to 1.

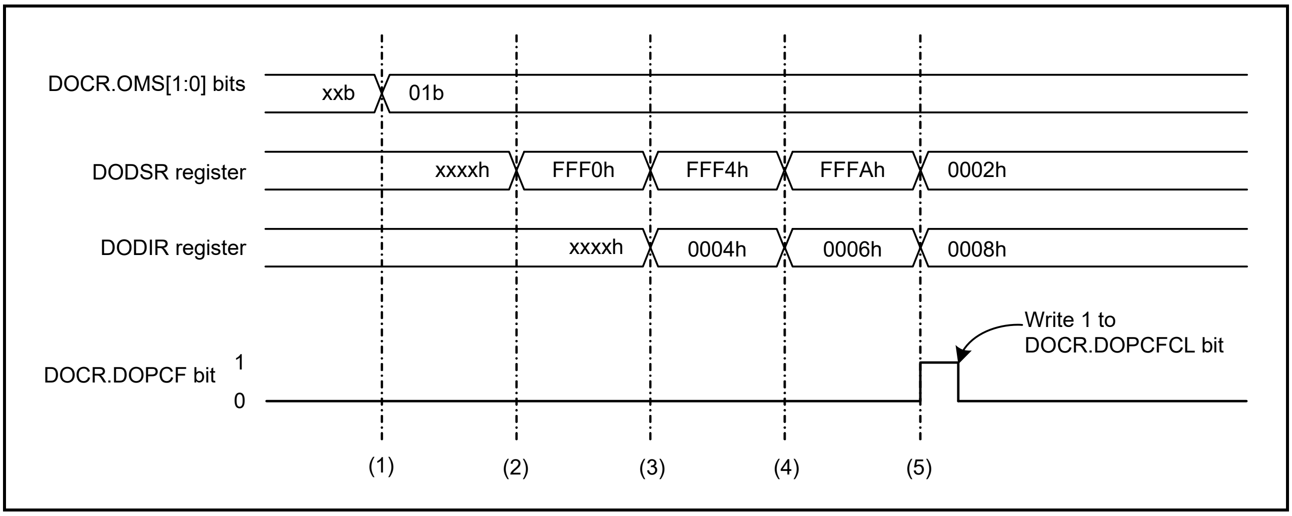

DOCR.OMS[1:0] bits
xxb
01b
DODSR register
xxxxh
FFF0h
FFF4h
FFFAh
0002h
DODIR register
xxxxh
0004h
0006h
0008h
Write 1 to
DOCR.DOPCFCL bit
DOCR.DOPCF bit
1
0
(1)
(2)
(3)
(4)
(5)

Figure 41.3 Example operation in data addition mode

41.3.3 Data Subtraction Mode

Figure 41.4 shows an example for 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.

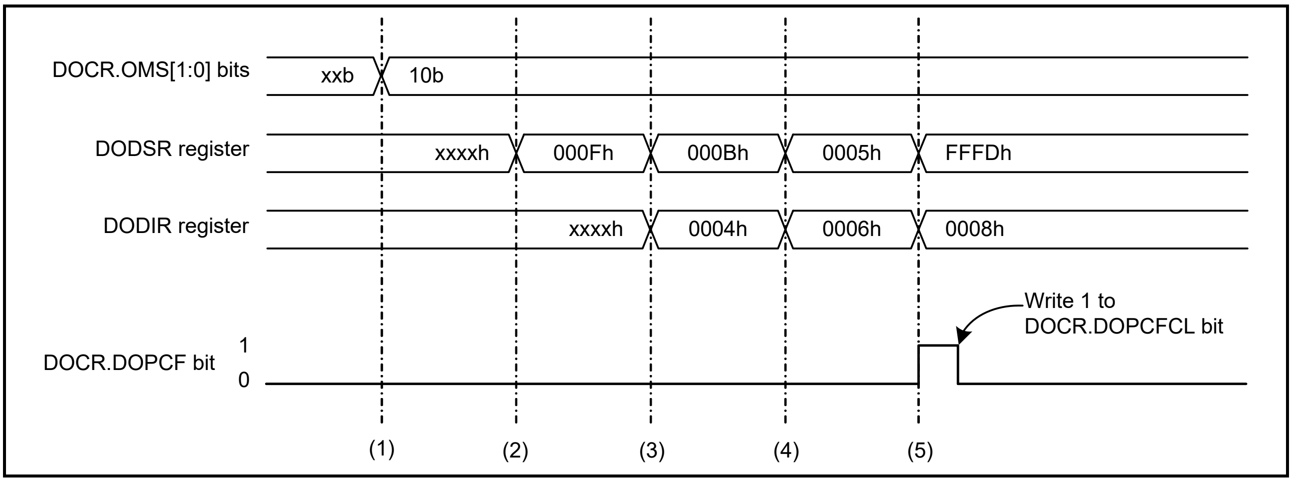

DOCR.OMS[1:0] bits
xxb
10b
DODSR register
xxxxh
000Fh
000Bh
0005h
FFFDh
DODIR register
xxxxh
0004h
0006h
0008h
Write 1 to
DOCR.DOPCFCL bit
DOCR.DOPCF bit
1
0
(1)
(2)
(3)
(4)
(5)

Figure 41.4 Example operation in data subtraction mode

41.4 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 can also be used as an interrupt request. When an event signal is generated, the Data Operation Circuit Flag (DOCR.DOPCF) is set to 1.

41.5 Usage Notes

41.5.1 Settings for the Module-Stop State

The Module Stop Control Register C (MSTPCRC) can enable or disable the 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.

42. SRAM

42.1 Overview

The MCU provides an on-chip high-speed SRAM module with either parity-bit checking or Error Correction Code (ECC). The first 16-KB area of SRAM0 is the ECC. Parity check is performed on the other areas.

Table 42.1 lists the SRAM specifications.

Table 42.1 SRAM specifications

Parameter	Without ECC	With ECC
SRAM capacity	SRAM0: 80 KB	SRAM0 (ECC area): 16 KB
SRAM address	SRAM0: 2000 4000h to 2001 7FFFh	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 42.3.7, Access Cycle.

42.2 Register Descriptions

42.2.1 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/W

The PARIOAD register controls the operation on detection of a parity error. The SRAM Protection Register (SRAMPRCR) protects this register against writes. Enable the SRAMPRCR bit in the SRAMPRCR register before writing to this register. Do not write to the PARIOAD register while accessing the SRAM.

OAD bit (Operation After Detection)

The OAD bit specifies either a reset or non-maskable interrupt when a parity error is detected. The OAD bit is used for SRAM0 (without ECC).

42.2.2 SRAM Protection Register (SRAMPRCR)

Address(es): SRAM.SRAMPRCR 4000 2004h

	b7	b6	b5	b4	b3	b2	b1	b0
	KW[6:0]							SRAMPRCR
Value after reset:	0	0	0	0	0	0	0	0

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.

42.2.3 ECC Operating Mode Control Register (ECCMODE)

Address(es): SRAM.ECCMODE 4000 20C0h

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

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/W

The ECCMODE register specifies the ECC operating mode. The ECC Protection Register (ECCPRCR) protects this register against writes. Before writing to this register, set the ECCPRCR bit in the ECCPRCR register to 1. 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.

42.2.4 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/W

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 a 2-bit error is detected while ECC operations are enabled and error checking is selected, the ECC2ERR bit is set to 1. The SRAM error signal is also asserted at this time. The 2-bit ECC error can be cleared by writing 0 to the ECC2ERR bit. 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.

42.2.5 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/W

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 writes. Before writing to this bit, set the ECCPRCR bit in the ECCPRCR register to 1 (write protection disabled).

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.

42.2.6 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/W

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 a 1-bit error is detected while ECC operations are enabled, error correction is selected, and updating of the 1-bit error information is enabled. The ECC1ERR bit is set to 1 and the SRAM error signal is asserted at this time. The 1-bit ECC error can be cleared by writing 0 to the ECC1ERR bit.

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.

42.2.7 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 writing 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 writing 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.

42.2.8 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 writing 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 to the ECCPRCR2 bit. When writing 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.

42.2.9 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/W

The ECC Protection Register 2 (ECCPRCR2) protects this register against writes. Before writing to this bit, set the ECCPRCR2 bit in the ECCPRCR2 register to 1 (write protection disabled). 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. When the ECC bypass function is used, the ECCMOD[1:0] bits in the ECCMODE register are set to 00b. The ECC must be accessed in 32 bits using the same address for 32-bit data. The ECC code is assigned to the lower 7 bits of the 32-bit data. When writing the ECC code, the upper 25 bits are ignored. When reading the ECC code, the upper 25 bits are undefined.

Note: For details on the ECC test, see section 42.3.4, ECC Decoder Testing.

42.2.10 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/W

The ECC Protection Register (ECCPRCR) protects this register against writes. Before writing to this bit, set the ECCPRCR bit in the ECCPRCR register to 1 (write protection disabled). 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 SRAM (ECC area).

42.3 Operation

42.3.1 Low Power Consumption Function

Power consumption can be reduced by setting the Module Stop Control Register A (MSTPCRA) to stop the supply of the clock signal to the SRAM. When both the MSTPA0 and the MSTPA6 bits in MSTPCRA are set to 1, supply of the clock signal to SRAM0 is stopped*1.

Note 1. The MSTPA0 bit and MSTPA6 bit in MSTPCRA must be set to the same value.

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.

42.3.2 ECC Function

You can enable or disable the ECC function by setting the ECCMODE register. By default, the ECC function is disabled and the ECC check type is SEC-DED (Single-Error Correction and Double-Error Detection).

When the ECC function is enabled, 7-bit check bits are appended to the 32-bit data for writes. For reads, 39-bit data (32-bit data and 7-bit 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 without error correction, and the ECC2ERR bit in the ECC2STS register is set to 1.

When the ECC function is enabled and the error checking is disabled, error correction is performed if a 1-bit error occurs but the 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 or ECC2ERR bit is not updated.

It is not possible to confirm the location where the error is detected. Therefore, after an error occurred, update all the data by writing 32-bit data to the SRAM.

When a read access is performed consecutively after a write access, the read access has priority. Therefore, during initialization, do not perform a read access successively after a write access.

42.3.3 ECC Error Generation

When the ECC function is enabled and error checking is applied to the SRAM (ECC area), an ECC error occurs when either the ECC2ERR bit in the ECC2STS register or the ECC1ERR bit in the ECC1STS register becomes 1 to indicate that a 2-bit error or a 1-bit error has occurred.

To mask ECC 1-bit errors, set the ECC1STSEN.E1STSEN bit to 0 to disable the ECC1ERR bit update. An ECC error is not generated when the ECC function is disabled or enabled without error checking.

An ECC error can generate 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.

42.3.4 ECC Decoder Testing

Figure 42.1 shows the ECC decoder testing.

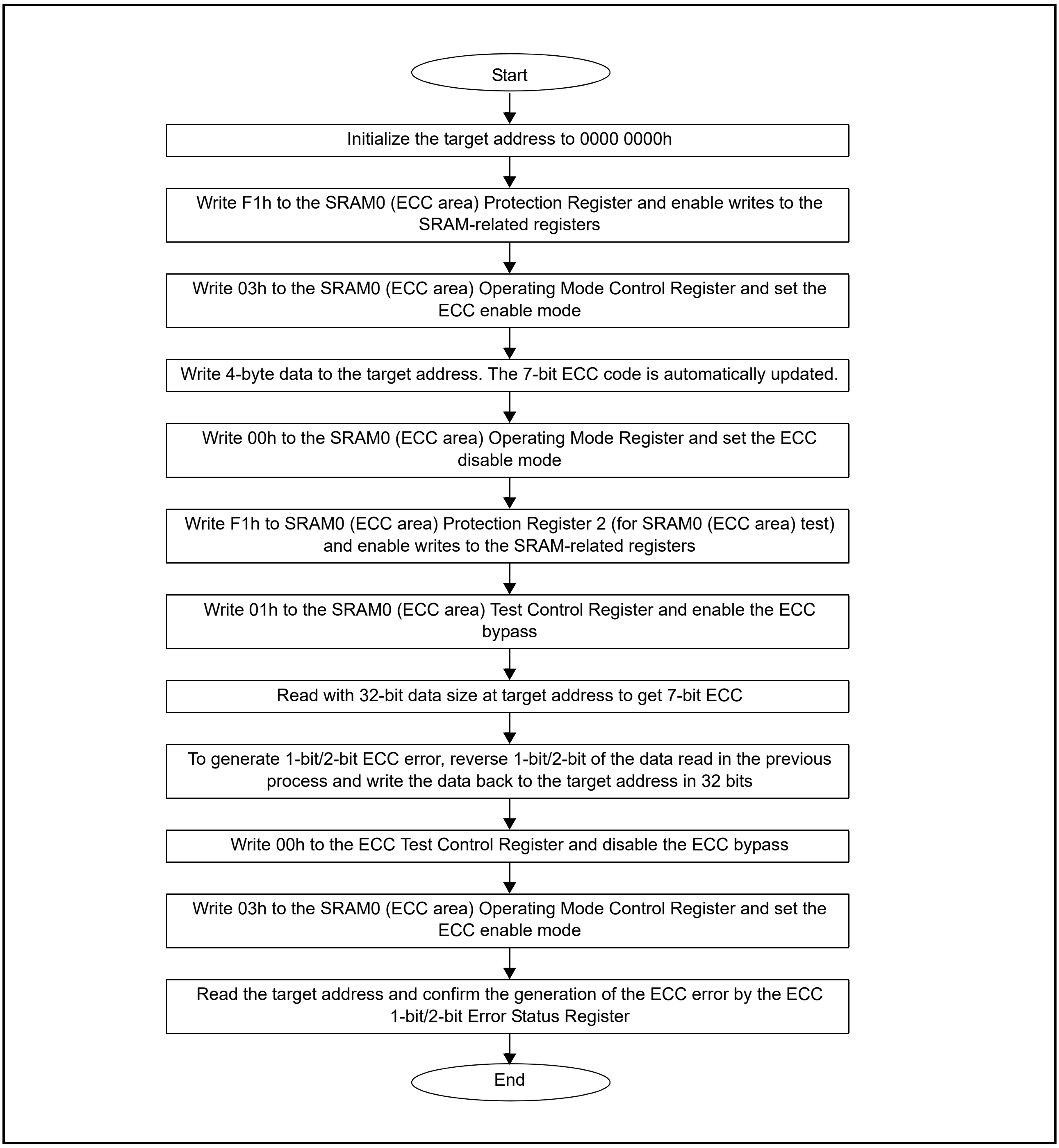

Start
Initialize the target address to 0000 0000h
Write F1h to the SRAM0 (ECC area) Protection Register and enable writes to the SRAM-related registers
Write 03h to the SRAM0 (ECC area) Operating Mode Control Register and set the ECC enable mode
Write 4-byte data to the target address. The 7-bit ECC code is automatically updated.
Write 00h to the SRAM0 (ECC area) Operating Mode Register and set the ECC disable mode
Write F1h to SRAM0 (ECC area) Protection Register 2 (for SRAM0 (ECC area) test) and enable writes to the SRAM-related registers
Write 01h to the SRAM0 (ECC area) Test Control Register and enable the ECC bypass
Read with 32-bit data size at target address to get 7-bit ECC
To generate 1-bit/2-bit ECC error, reverse 1-bit/2-bit of the data read in the previous process and write the data back to the target address in 32 bits
Write 00h to the ECC Test Control Register and disable the ECC bypass
Write 03h to the SRAM0 (ECC area) Operating Mode Control Register and set the ECC enable mode
Read the target address and confirm the generation of the ECC error by the ECC 1-bit/2-bit Error Status Register
End

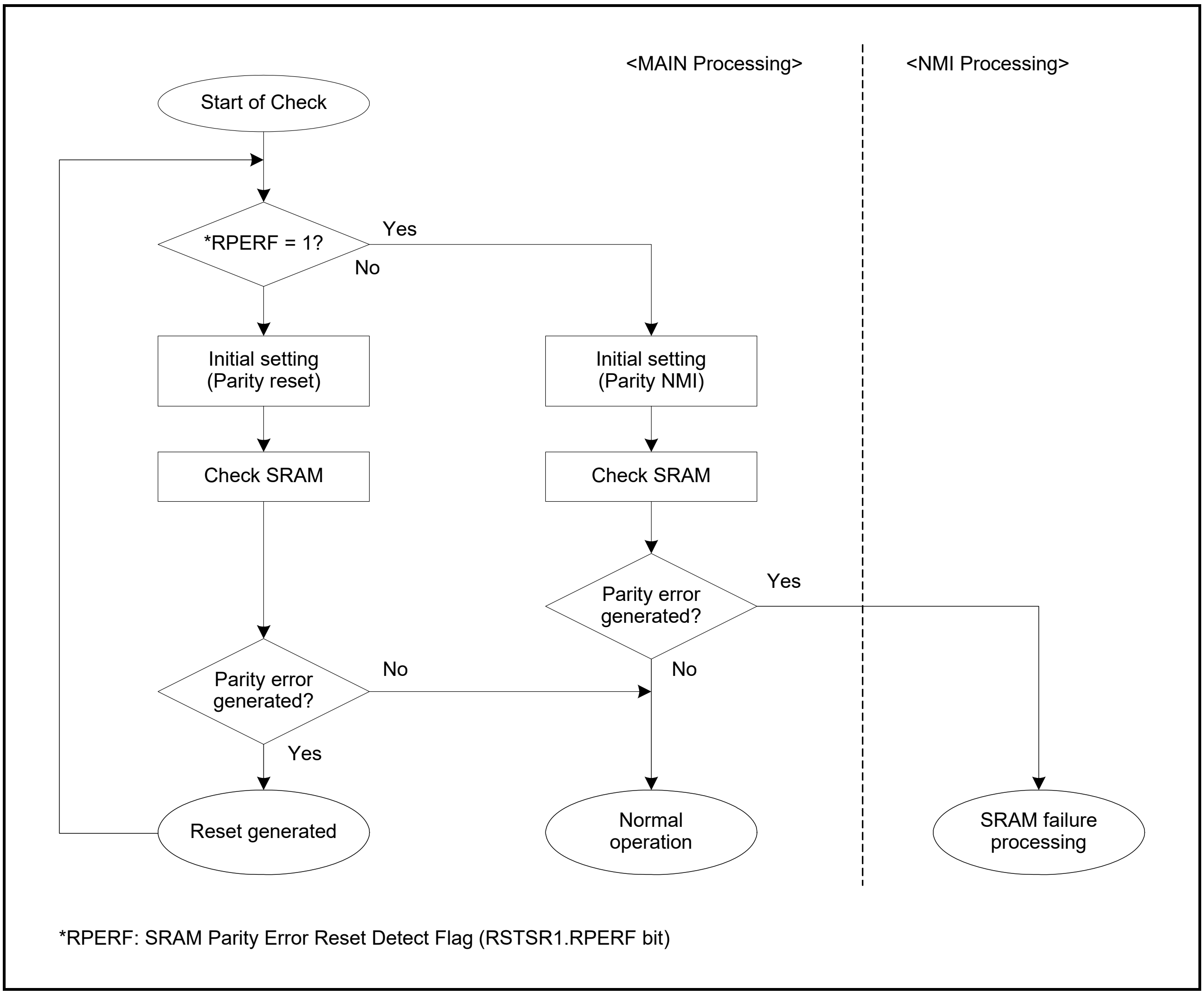
Figure 42.1 ECC decoder testing

42.3.5 Parity Calculation Function

The IEC60730 standard requires the checking of SRAM data. When data is written, a parity bit is added to every 8-bit data in the SRAM which has 32-bit data width, and when data is read, the parity is checked. When a parity error occurs, a parity-error notification is generated. This function can also be used to trigger a reset. The specification of SRAM0 without ECC 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 confirm whether the cause of the parity error is noise or corruption, follow the parity check flows shown in Figure 42.2 and Figure 42.3.


<MAIN Processing>
<NMI Processing>
Start of Check
*RPERF = 1?
Yes
No
Initial setting (Parity reset)
Initial setting (Parity NMI)
Check SRAM
Check SRAM
Parity error generated?
Yes
No
Parity error generated?
No
Yes
Reset generated
Normal operation
SRAM failure processing

*RPERF: SRAM Parity Error Reset Detect Flag (RSTSR1.RPERF bit)

Figure 42.2 Flow of SRAM parity check when SRAM parity reset is enabled

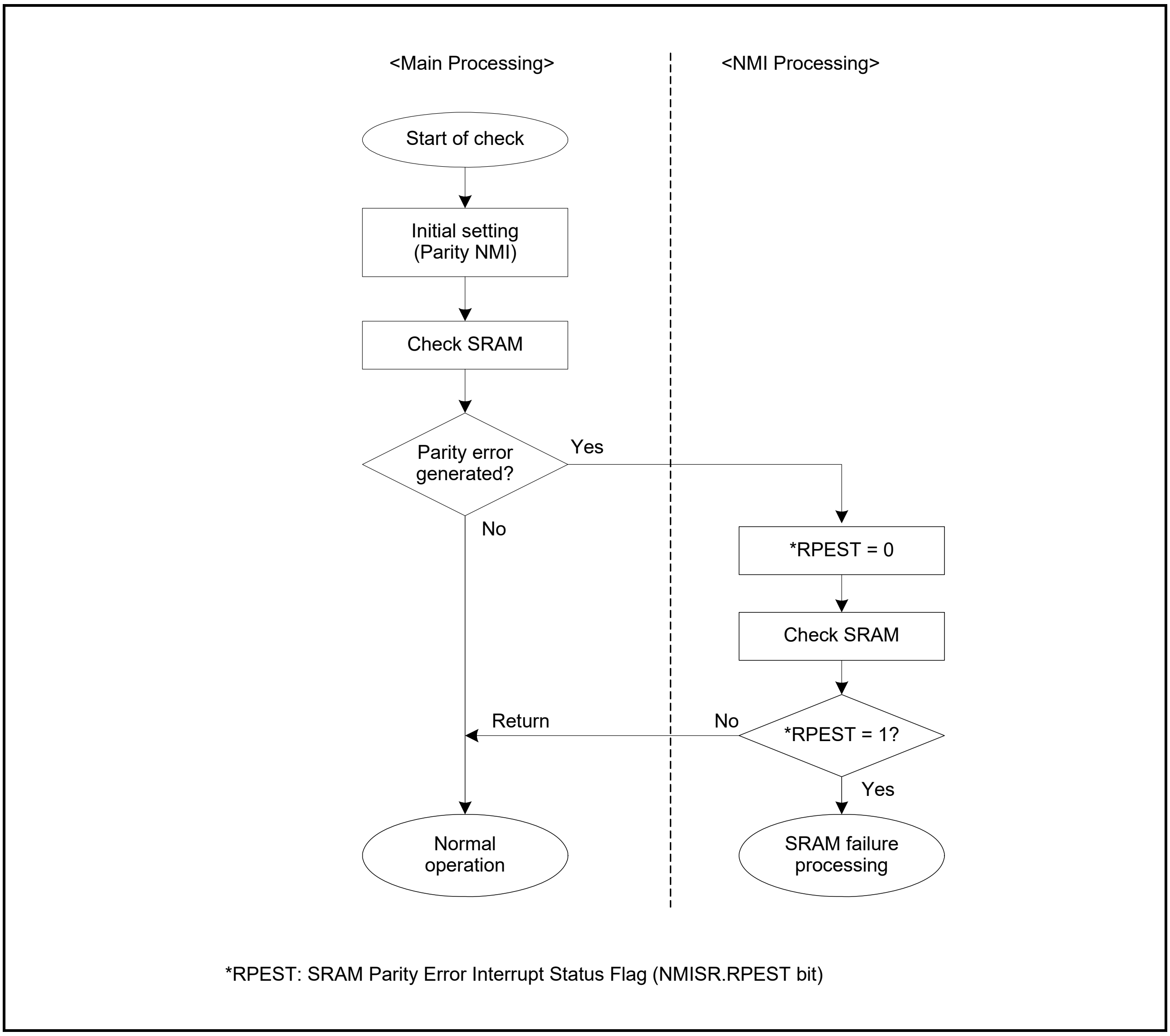

<Main Processing>
<NMI Processing>
Start of check
Initial setting (Parity NMI)
Check SRAM
Parity error generated?
Yes
No
*RPEST = 0
Check SRAM
*RPEST = 1?
No
Return
Yes
Normal operation
SRAM failure processing
*RPEST: SRAM Parity Error Interrupt Status Flag (NMISR.RPEST bit)

Figure 42.3 Flow of SRAM parity check when SRAM parity interrupt is enabled

42.3.6 SRAM Error Sources

An SRAM error source is either an ECC error or a parity error. ECC error or parity error can generate either a non-maskable interrupt or a reset, as selected with the OAD bit in the ECCOAD register for ECC error, or PARIOAD register for parity error.

Table 42.2 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)	Not possible	Not possible

42.3.7 Access Cycle

Table 42.3 SRAM0 (ECC area 2000 0000h to 2000 3FFFh)

	Read (cycle)		Write (cycle)	
Bit setting	**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 42.4 SRAM0 (Parity area 2000 4000h to 2001 7FFFh)

Read (cycle)		Write (cycle)	
Word Access	**Halfword/Byte access**	**Word access**	**Halfword/Byte access**
2		2	

42.4 Usage Notes

42.4.1 Instruction Fetch from SRAM area

When using SRAM0 to operate a program, initialize the SRAM area so that the CPU can correctly prefetch data. If the CPU prefetches data from an SRAM area that is not initialized, an ECC error or a parity error might occur. Initialize the additional 12-byte area from the end address of a program with a 4-byte boundary.

42.4.2 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 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).

43. Flash Memory

43.1 Overview

The MCU provides up to 512-KB code flash memory and 8-KB data flash memory. The Flash Control Block (FCB) controls the flash memory programming commands. This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

Table 43.1 lists the specifications of the code flash memory and data flash memory, and Figure 43.1 shows a block diagram of the related modules. Figure 43.2 shows the configuration of the code flash memory, and Figure 43.3 shows the configuration of the data flash memory.

Table 43.1 Specifications of code flash memory and data flash memory

Parameter	Code flash memory	Data flash memory
Memory capacity	512 KB of user area	8 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 FCLK cycles 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 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.	

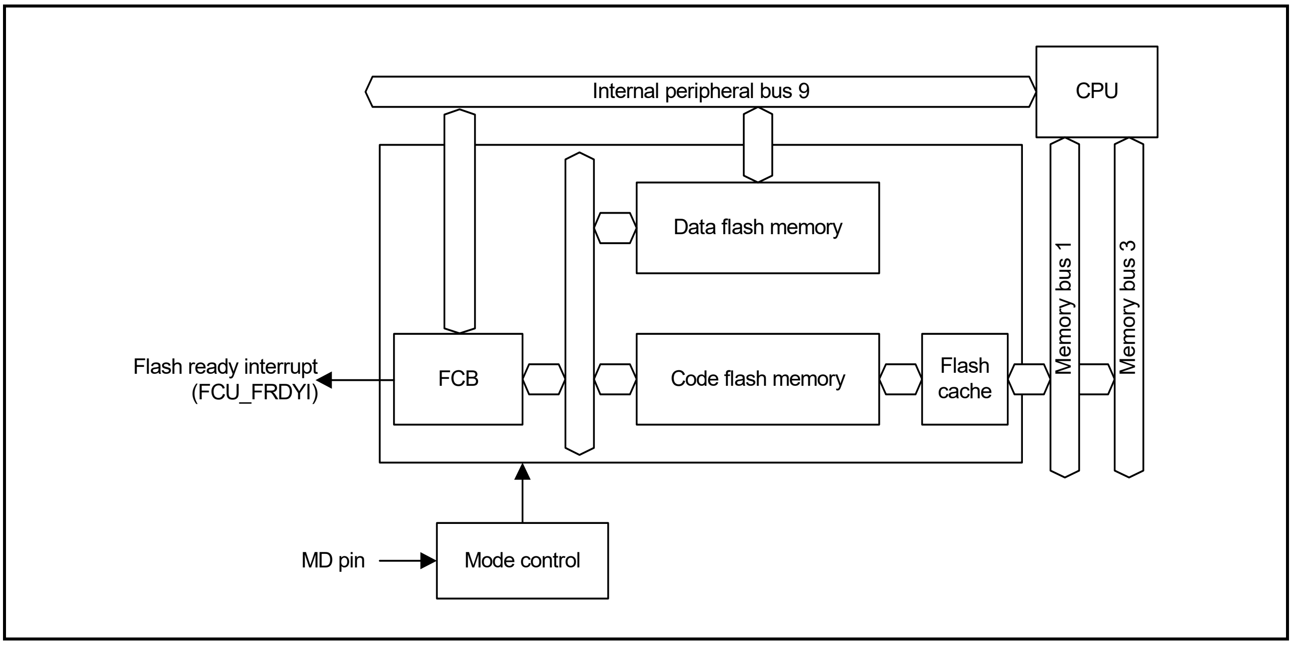

Internal peripheral bus 9
CPU
Data flash memory
Memory bus 1
Memory bus 3
Flash ready interrupt
(FCU_FRDYI)
FCB
Code flash memory
Flash cache
MD pin
Mode control

Figure 43.1 Flash memory-related modules block diagram

43.2 Memory Structure

Figure 43.2 shows the mapping of the code flash memory, and Table 43.2 shows the read and programming and erasure (P/E) addresses of the code flash memory. The user space of the code flash memory is divided into 2-KB blocks that serve as the units of erasure. The user area is available for storing the user program.

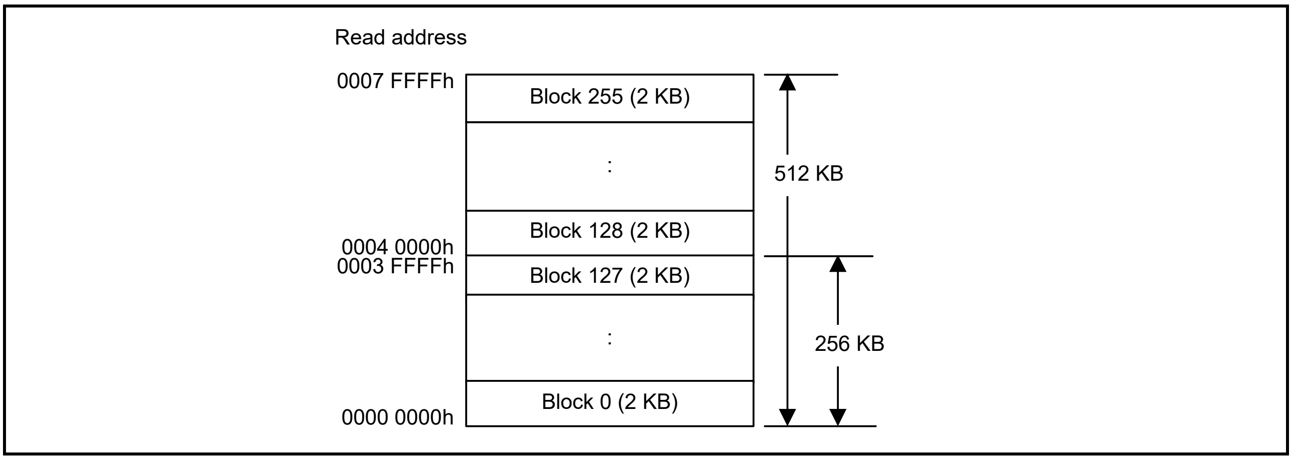

Read address
0007 FFFFh
Block 255 (2 KB)
:
512 KB
0004 0000h
0003 FFFFh
Block 128 (2 KB)
Block 127 (2 KB)
:
256 KB
0000 0000h
Block 0 (2 KB)

Figure 43.2 Mapping of the code flash memory

Table 43.2 Read and P/E addresses of the code flash memory

Size of code flash memory	Read address	P/E address	Number of blocks
512 KB	0000 0000h to 0007 FFFFh	0000 0000h to 0007 FFFFh	0 to 255

The data area of the data flash memory is divided into 1-KB blocks, with each being a unit for erasure. Figure 43.3 shows the mapping of the data flash memory, and Table 43.3 shows the read, programming and erasure addresses of the data flash memory.

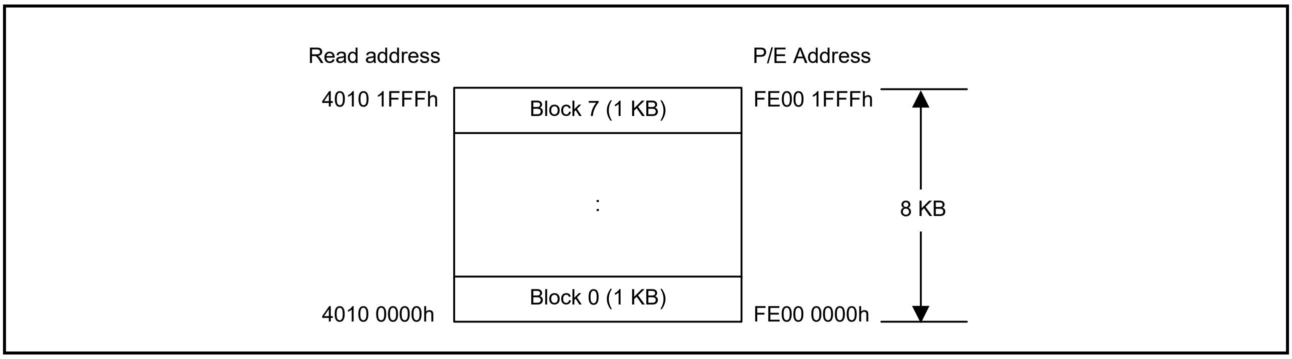


Figure 43.3 Mapping of the data flash memory

Table 43.3 Read and P/E addresses of the data flash memory

Size of data flash memory	Read address	P/E address	Number of blocks
8 KB	4010 0000h to 4010 1FFFh	FE00 0000h to FE00 1FFFh	0 to 7

43.3 Flash Cache

43.3.1 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 43.4 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	128 bytes	8 bytes	16 bytes
Associativity	2-way set associative	Fully associative	-
	• 64 bits/entry (64-bit aligned data) • 8 entries/ways.	• 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: MEMWAIT = 0: 0 wait MEMWAIT = 1: 1 or 2 waits	Cache hit: 0 wait Cache miss: According to SYSTEM.MEMWAIT register: MEMWAIT = 0: 0 wait MEMWAIT = 1: 1 or 2 waits	Cache hit: 0 wait Cache miss: According to SYSTEM.MEMWAIT register: MEMWAIT = 0: 0 wait MEMWAIT = 1: 1 or 2 waits

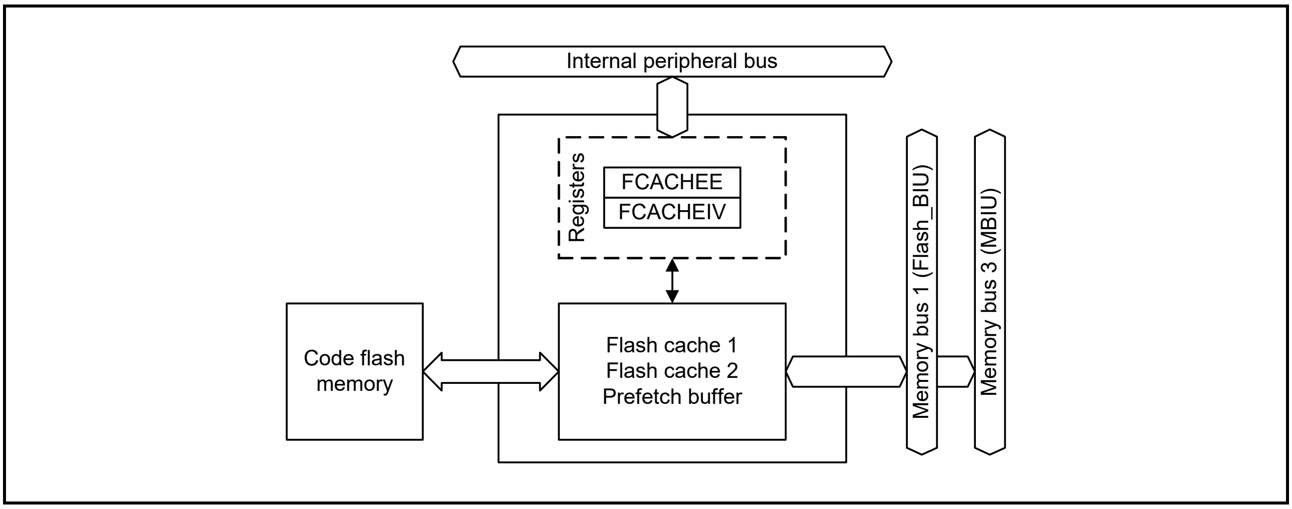

Internal peripheral bus
Registers
FCACHEE
FCACHEIV
Code flash memory
Flash cache 1
Flash cache 2
Prefetch buffer
Memory bus 1 (Flash_BIU)
Memory bus 3 (MBIU)

Figure 43.4 FCACHE block diagram

43.3.2 Register Descriptions

43.3.2.1 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. The write value should be 0.	R/W

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 15.5, Notes on using Flash Cache.

43.3.2.2 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 the write value is 1, FCACHE is invalidated. When the write value is 0, this setting is ignored.	R/W

Bit	Symbol	Bit name	Description	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

When 1 is written to the FCACHEIV.FCACHEIV bit, flash cache data in FCACHE1, FCACHE2, and FLPF is invalidated.

43.3.2.3 Data Flash Control Resister (DFLCTL)

Address(es): FLCN.DFLCTL 407E C090h

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

Bit	Symbol	Bit name	Description	R/W
b0	DFLEN	Data Flash Access Enable	0: Access to the data flash is disabled 1: Access to the data flash is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DFLCTL register is to enable or disable accessing (reading, programming, and erasing) of the data flash. After setting the DFLCTL.DFLEN bit, Data Flash STOP recovery time (t_{DSTOP}) is nccessary before reading the data flash or entering the data flash P/E mode.

The time setup takes differs for each operating mode.

<Setup time for each operating mode>

- HS (High-speed) mode: 5 µs
- LS (Low-speed) mode: 720 ns
- LP (Low-power) mode: 720 ns
- LV (Low-voltage) mode: 10 µs

43.4 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: Do not change operation mode (read mode, wait mode) when the flash cache is enabled.
Note 1. It is not necessary to disable the flash cache on the first setup after reset.

43.4.1 Notice to use Flash Cache

When using flash cache by access from the CPU, Arm® MPU should also be set to be cacheable.

See the *ARM®v7-M Architecture Reference Manual* and the *ARM® Cortex®-M4 Devices Generic User Guide*.

43.5 Operating Modes Associated with the Flash Memory

Figure 43.5 shows a diagram of the mode transitions associated with the flash memory. For information on setting up the modes, see section 3, Operating Modes.

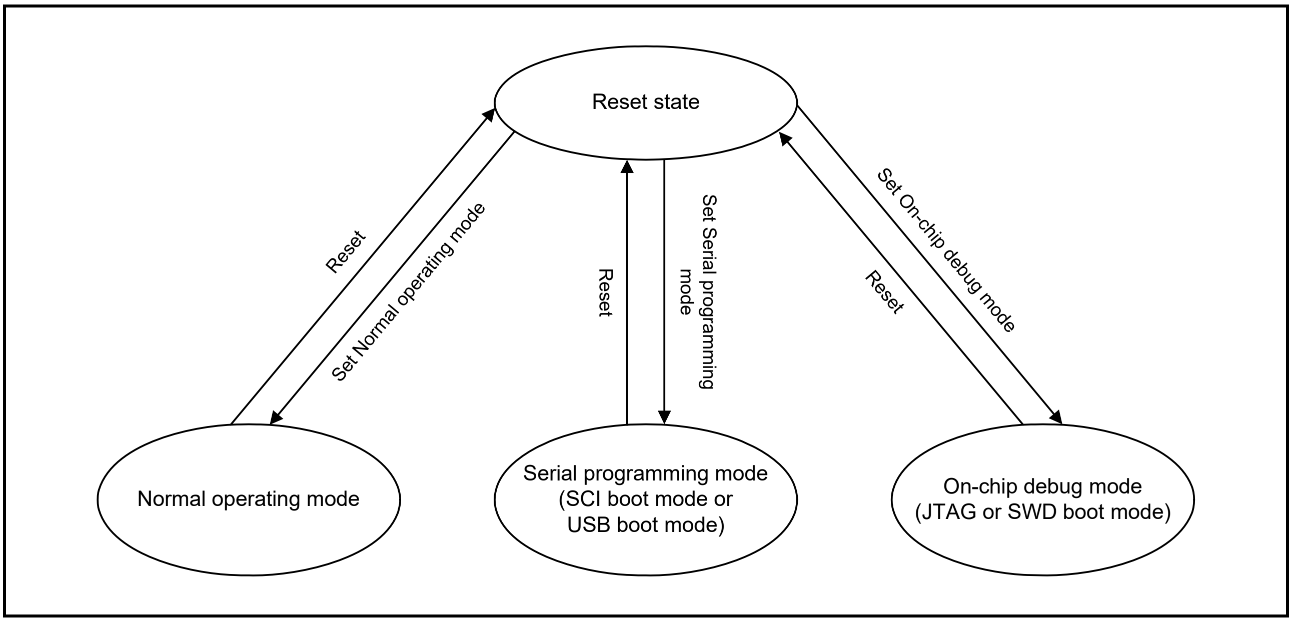

Reset state
Reset
Set Normal operating mode
Reset
Set Serial programming mode
Set On-chip debug mode
Reset
Normal operating mode
Serial programming mode (SCI boot mode or USB boot mode)
On-chip debug mode (JTAG or SWD boot mode)

Figure 43.5 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 mode. Table 43.5 shows the differences between the modes.

Table 43.5 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

43.5.1 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 43.6 shows how the ID code determines the authentication method.

Table 43.6 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 are 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. Mismatching 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 contents in the user flash (code and data) area, and the configuration area are erased. However, forced erasure is not performed 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. Mismatching 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 mismatching, and connection to the programmer or the on-chip debugger is prohibited.

43.6 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 43.7 lists the programming methods and the associated operating modes.

Table 43.7 Programming methods (1 of 2)

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

Table 43.7 Programming methods (2 of 2)

Programming method	Functional overview	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 connected 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 43.8 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 43.8 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 43.9 lists the security functions supported by the on-chip flash memory, and Table 43.10 lists the available operations and security settings.

Table 43.9 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 43.10 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 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

43.6.1 Configuration Area Bit Map

The bits used for ID authentication, startup area select, access window protection, and security configuration functions are mapped in Figure 43.6. The boot program must use these bits as hexadecimal data.

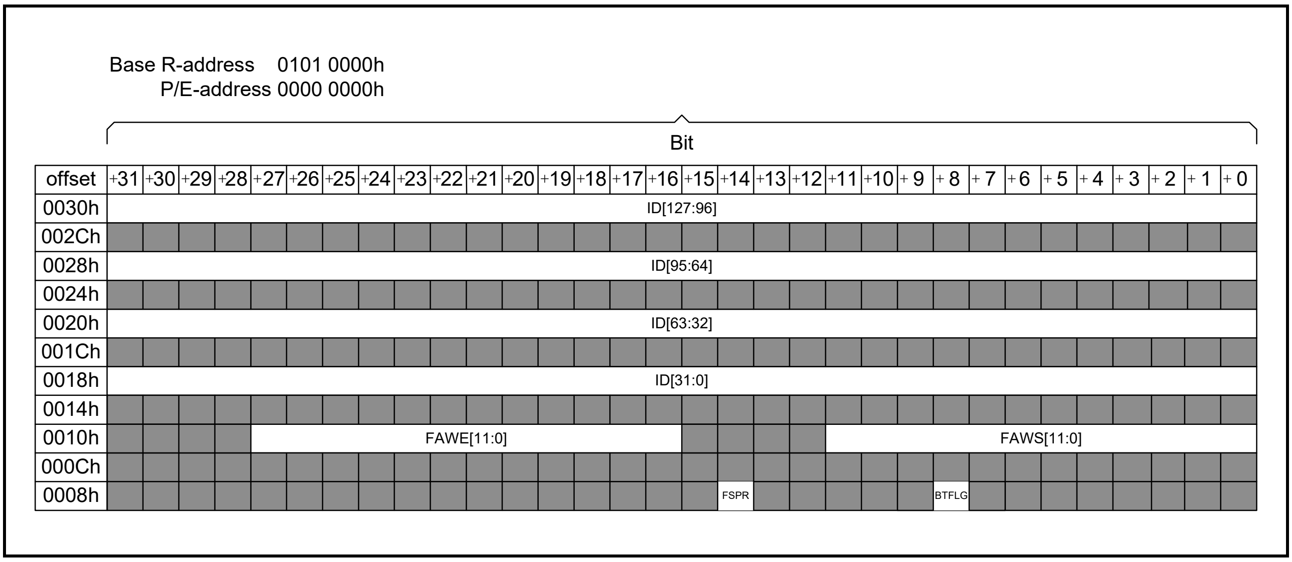
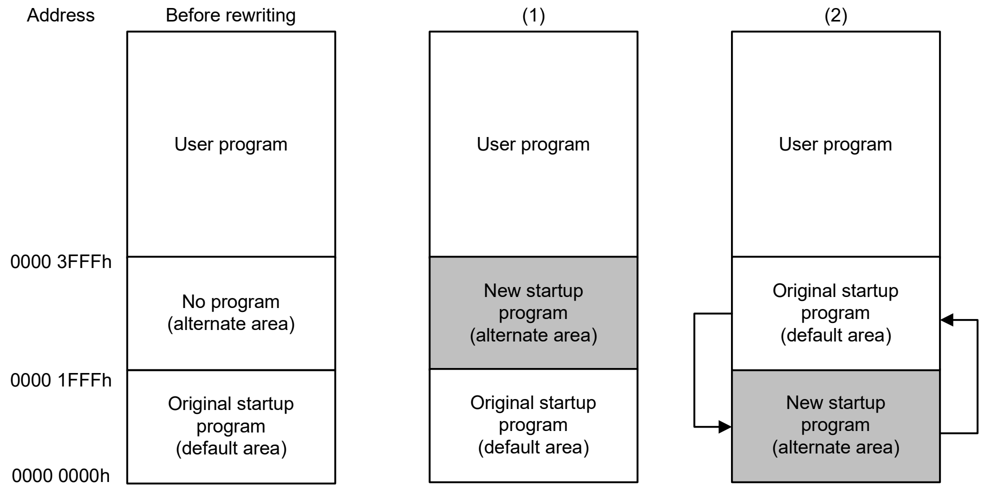

Base R-address 0101 0000h
P/E-address 0000 0000h
Bit
offset
+31 +30 +29 +28 +27 +26 +25 +24 +23 +22 +21 +20 +19 +18 +17 +16 +15 +14 +13 +12 +11 +10 +9 +8 +7 +6 +5 +4 +3 +2 +1 +0
0030h
ID[127:96]
002Ch
0028h
ID[95:64]
0024h
0020h
ID[63:32]
001Ch
0018h
ID[31:0]
0014h
0010h
FAWE[11:0]
FAWS[11:0]
000Ch
0008h
FSPR
BTFLG

Figure 43.6 Configuration area bit map

43.6.2 Startup Area Select

The startup area select function allows the boot program to be safely updated. The startup area is 8 KB of space located in the user area. The FCB controls the startup area address based on the Startup Area Select Flag (BTFLG) that is located in the AWSC register. The startup area can be locked by the FSPR bit.

Figure 43.7 shows an overview of the startup program protection.


Address
Before rewriting
(1)
(2)
User program
User program
User program
0000 3FFFh
No program
(alternate area)
New startup
program
(alternate area)
Original startup
program
(default area)
0000 1FFFh
Original startup
program
(default area)
Original startup
program
(default area)
New startup
program
(alternate area)
0000 0000h

(1) Program a new startup program in the alternate area. If the alternate area fails to be rewritten, the new startup program can be rewritten again after starting up, using the default area because the original startup program is in the default area.
(2) After the alternate area is successfully rewritten, the default area and the alternate area are switched using the self-programming library. After that, the program in the alternate area starts on a reset.

Figure 43.7 Overview of startup program protection

43.6.3 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 43.8 shows the flash area protection.

The access window is specified in both the FAWS[11:0] and FAWE[11:0] bits. Setting of the FAWE[11:0] and FAWS[11:0] 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[11:0] bits to one block lower than the block pointed to by the FAWE[11:0] bits.
- FAWE[11:0] < FAWS[11:0]: The P/E command cannot execute anywhere in the user area of the code flash memory.

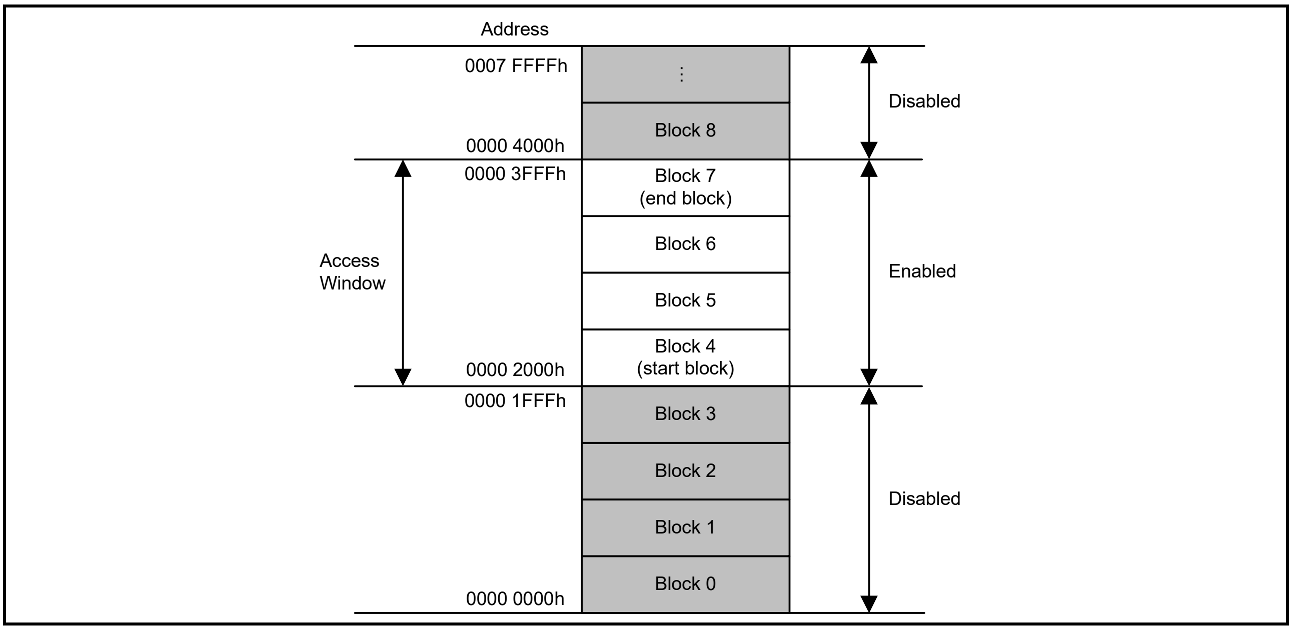

Address
0007 FFFFh
0000 4000h
0000 3FFFh
0000 2000h
0000 1FFFh
0000 0000h
⋮
Block 8
Block 7
(end block)
Block 6
Block 5
Block 4
(start block)
Block 3
Block 2
Block 1
Block 0
Access
Window
Disabled
Enabled
Disabled

Figure 43.8 Flash area protection overview

43.7 Programming Commands

The FCB controls the programming commands.

43.8 Suspend Operation

The forced stop command forces the blank check command or the 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 resetting the registers for command execution by copying the saved addresses.

43.9 Protection

The types of protection provided include:

- Software protection
- Error protection
- Boot program protection.

43.10 Serial Programming Mode

The serial programming modes include:

- Boot mode with SCI9
- USB boot mode with the USBFS.

Table 43.11 lists the I/O pins of the flash memory-related modules.

Table 43.11 I/O pins of flash memory-related modules (1 of 2)

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 SCI
P109/TXD9	Output		For host communication, to transmit data through SCI

Table 43.11 I/O pins of flash memory-related modules (2 of 2)

Pin name	I/O	Applicable modes	Function
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.

43.10.1 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 43.9 shows the system configuration for operations in boot mode.

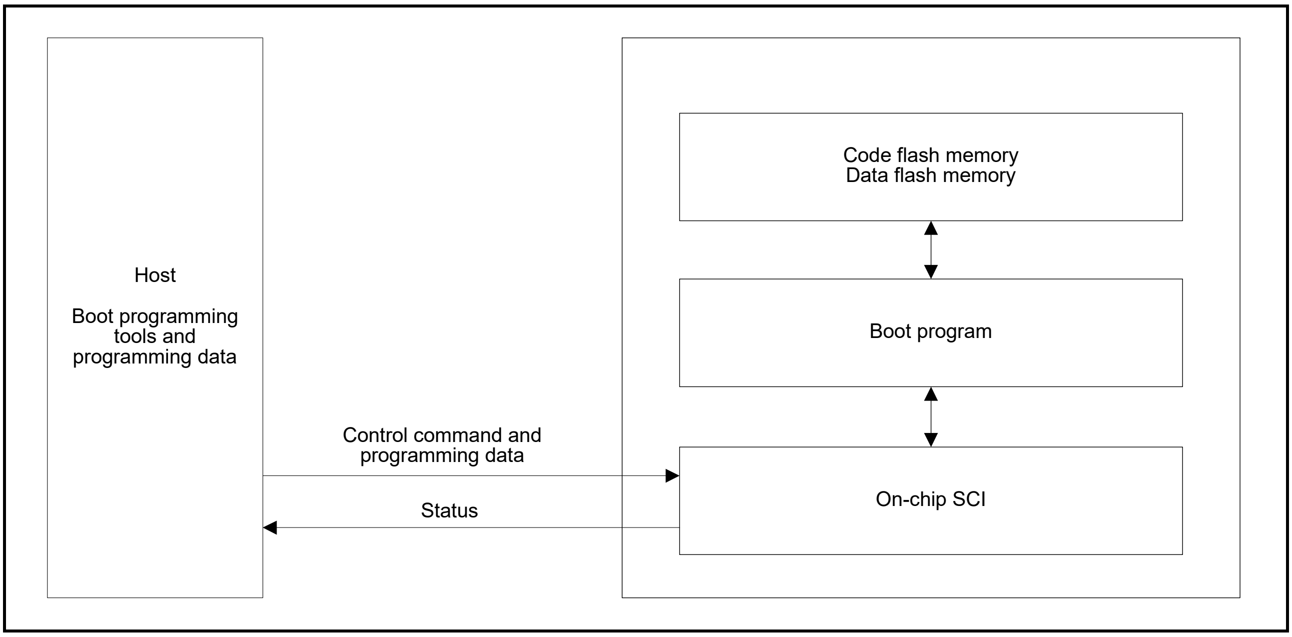

Host
Boot programming tools and programming data
Code flash memory
Data flash memory
Boot program
Control command and programming data
Status
On-chip SCI

Figure 43.9 System configuration in SCI boot mode

43.10.2 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 43.10 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.

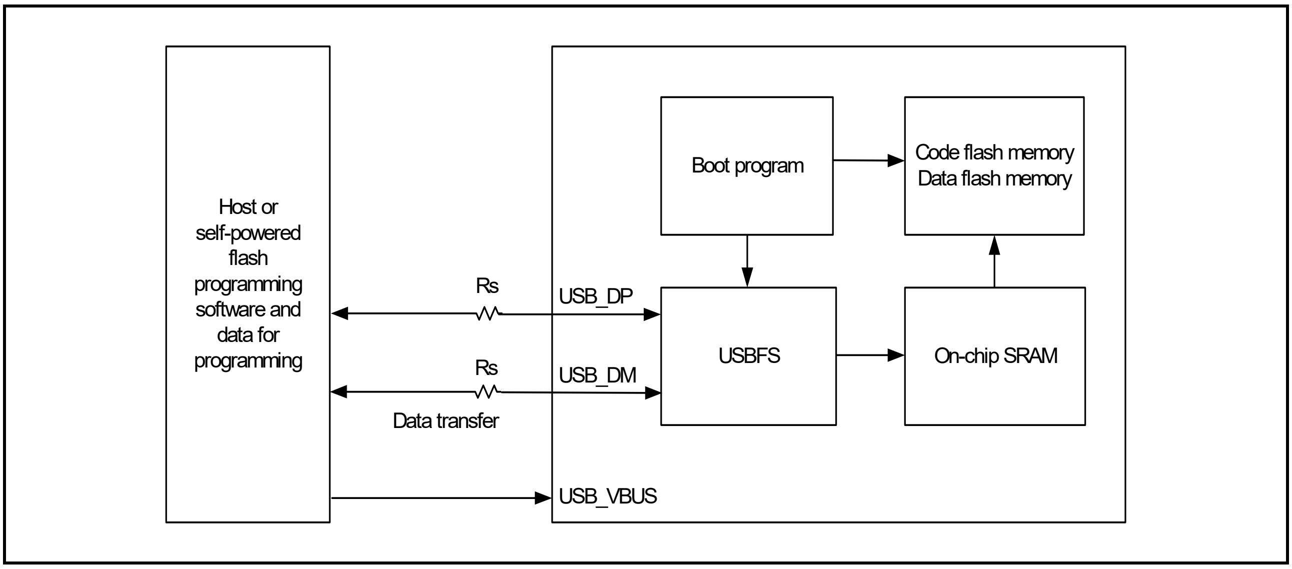

Host or
self-powered
flash
programming
software and
data for
programming
Rs
Rs
Data transfer
USB_DP
USB_DM
USB_VBUS
Boot program
Code flash memory
Data flash memory
USBFS
On-chip SRAM

Figure 43.10 System configuration in USB boot mode

43.11 Using a Serial Programmer

A dedicated flash memory programmer can be used to program the flash memory in serial programming mode.

43.11.1 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.

43.11.2 Programming Environment

Figure 43.11 shows the environment recommended by Renesas for programming the flash memory of the MCU with data.

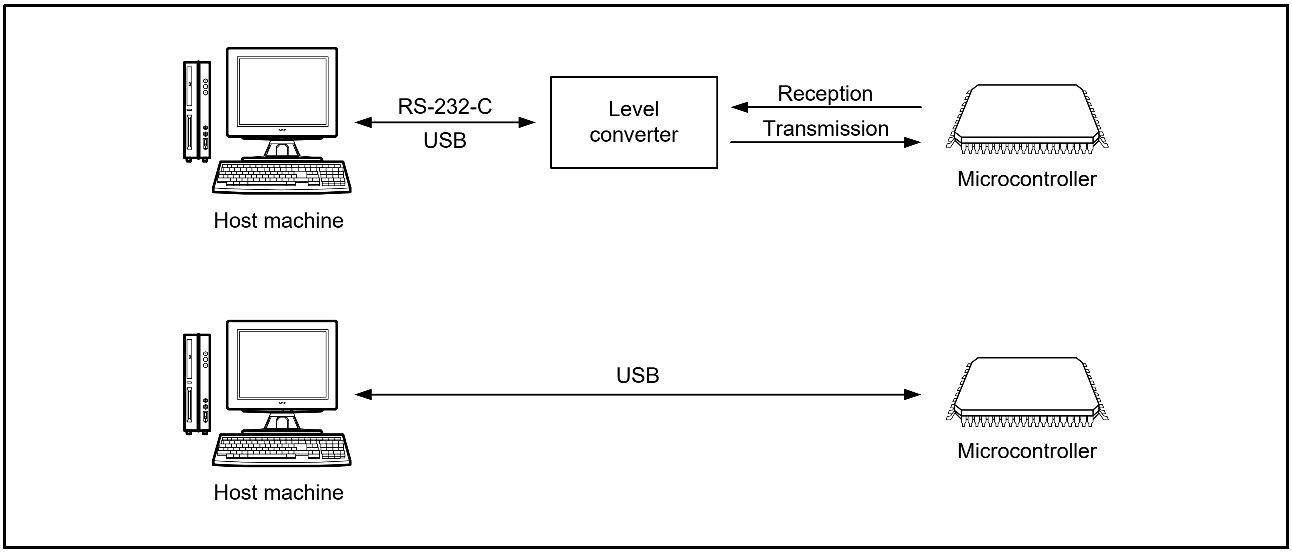

RS-232-C
USB
Level
converter
Reception
Transmission
Microcontroller
Host machine
USB
Microcontroller
Host machine

Figure 43.11 Environment for writing programs to the flash memory

43.12 Self-Programming

43.12.1 Overview

The MCU supports programming of the flash memory by the user program. The programming commands can be used with user programs for writing to the code and data flash memory. This enables updates to the user programs and overwriting of constant data fields.

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 43.12. This program can also be copied in advance to and executed from the internal SRAM. When executing from the internal SRAM, this program can also program the code flash memory area.

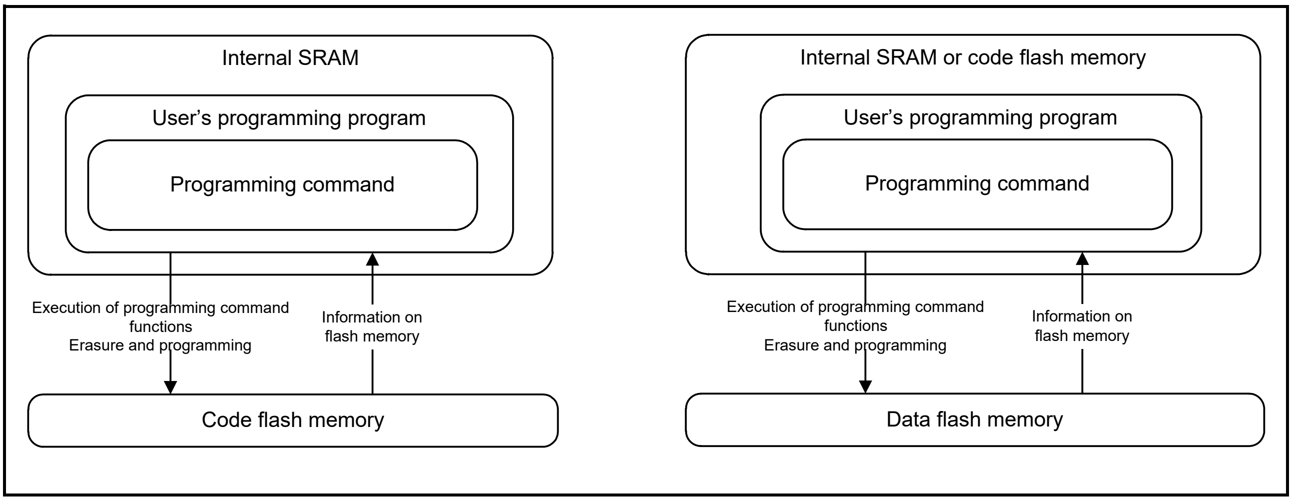

Internal SRAM
User's programming program
Programming command
Execution of programming command functions
Erasure and programming
Information on flash memory
Code flash memory
Internal SRAM or code flash memory
User's programming program
Programming command
Execution of programming command functions
Erasure and programming
Information on flash memory
Data flash memory

Figure 43.12 Schematic view of self-programming

43.12.2 Background Operation

Background operation can be used when a combination of the flash memory for writing and reading is as listed in Table 43.12.

Table 43.12 Conditions under which background operation is available

Product	Writable range	Readable range
All products	Data flash memory	Code flash memory

43.13 Reading the Flash Memory

43.13.1 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 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.

43.13.2 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 memory in the non-programmed state, all bits are read as 1s.

43.14 Usage Notes

43.14.1 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.

43.14.2 Suspension by Erase Suspend Commands

When suspending an erase operation with the erase suspend command, complete the operation with a resume command.

43.14.3 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.

43.14.4 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 48, 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.

43.14.5 Non-Maskable Interrupt Disabled during Programming and Erasure

When a non-maskable interrupt*1 occurs during a programming and erasure operation, the vectors are fetched from the code flash memory, and undefined data is read. Therefore, do not generate a non-maskable interrupt during a programming and erasure operation in the code flash memory. This constraint applies only to the code flash memory.

Note 1. A non-maskable interrupt is an NMI pin interrupt, oscillation stop detection interrupt, WDT underflow/refresh error interrupt, IWDT underflow/refresh error interrupt, voltage monitor 1 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.

43.14.6 Location of Interrupt Vectors during Programming and Erasure

When an interrupt occurs during a programming and erasure operation, the vector can be fetched from the code flash memory. 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.

43.14.7 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.

43.14.8 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 43.14.9, Actions Prohibited during Programming and Erasure, erase the area again.

43.14.9 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 programming or erasure operation to the code flash memory
- Do not change the DFLCTL.DFLEN bit value during a programming or erasure operation to the data flash memory.

44. Segment LCD Controller (SLCDC)

44.1 Overview

The MCU provides a controller for LCD display and display pins. Table 44.1 lists the SLCDC specifications.

Table 44.1 SLCDC specifications

Parameter	Description
Features	• Liquid crystal waveform (waveform A or B) selectable • LCD driver voltage generator can use external resistance division method • Automatic output of segment and shared signals based on automatic display data register read • LCD blinking and display selectable.
Number of pins	For details on the number of pins, see Table 44.2, SLCDC display function pins for 56-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 differ depending on the product. Table 44.2 shows the display function pins for products with different pin counts. Table 44.3 shows the maximum number of pixels for products with different pin counts. Figure 44.1 shows the SLDC block diagram.

Table 44.2 SLCDC display function pins for 56-pin products

Parameter	56 Pins															
LCD controller/ driver	Number of segment pins (SEG) : 9 Number of common pins (COM) : 4															
Multiplexed I/O port	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PORT1	-	-	-	-	-	SEG 53	SEG 52	-	COM 3	COM 2	COM 1	COM 0	VL4*1	-	VL2*1	VL1*1
PORT2	-	-	-	-	-	-	-	-	-	SEG 12	SEG 20	SEG 23	-	-	-	-
PORT3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PORT4	-	-	-	-	-	-	SEG9	-	SEG 11	-	-	-	-	SEG6	-	-
PORT5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SEG 49	-
PORT6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PORT7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PORT8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PORT9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Note 1. VL1, VL2, and VL4 are power supply pins for driving the LCD.

Table 44.3 Maximum number of pixels for 56-pin products (1 of 2)

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	9 (9 segment signals, 1 common signal)
		1/2	2	18 (9 segment signals, 2 common signals)
			3	27 (9 segment signals, 3 common signals)
		1/3	3	
			4	36 (9 segment signals, 4 common signals)

Table 44.3 Maximum number of pixels for 56-pin products (2 of 2)

Drive waveform for LCD driver	LCD driver voltage generator	Bias mode	Number of time slices	Maximum number of pixels
Waveform B	External resistance division	1/3	4	36 (9 segment signals, 4 common signals)

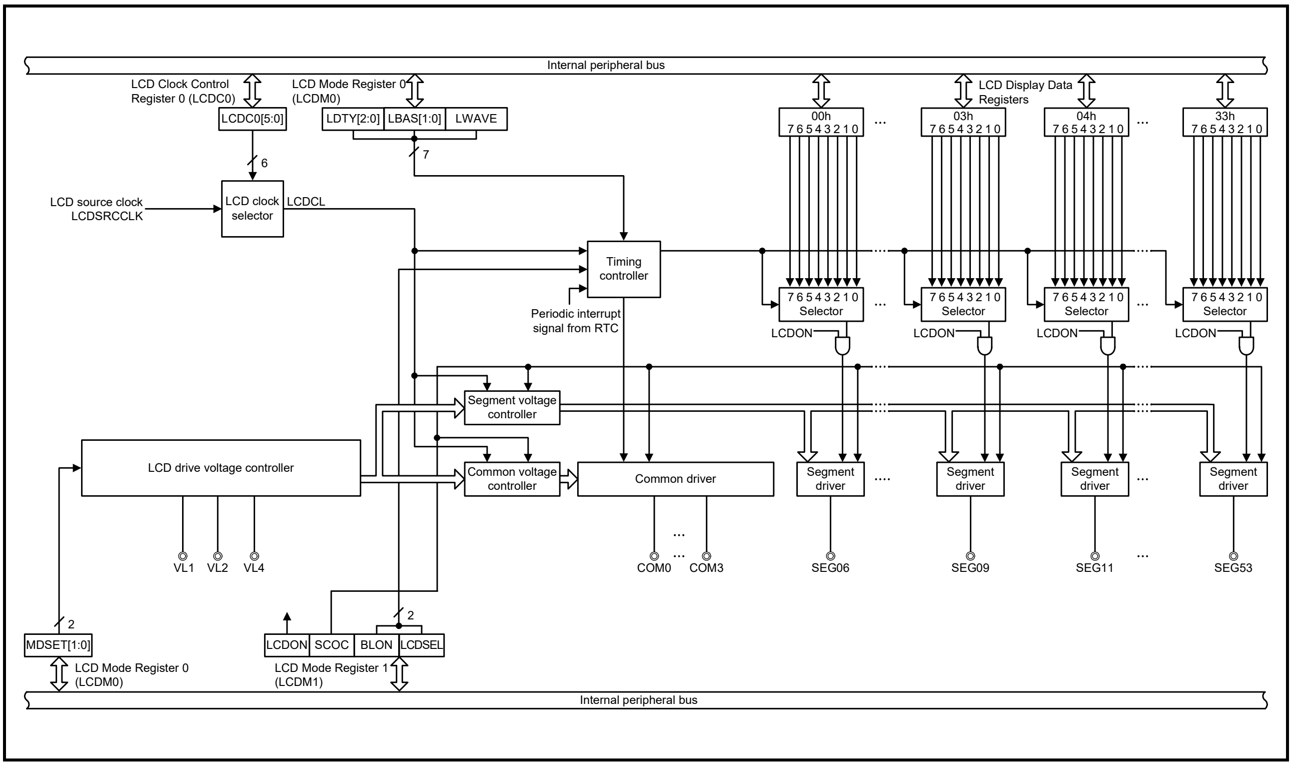

Internal peripheral bus
LCD Clock Control Register 0 (LCDC0)
LCDC0[5:0]
LCD Mode Register 0 (LCDM0)
LDTY[2:0]
LBAS[1:0]
LWAVE
6
7
LCD source clock LCDSRCCLK
LCD clock selector
LCDCL
Timing controller
Periodic interrupt signal from RTC
LCD Display Data Registers
00h
03h
04h
33h
7 6 5 4 3 2 1 0
Selector
LCDON
Segment voltage controller
Common voltage controller
Common driver
Segment driver
LCD drive voltage controller
VL1
VL2
VL4
COM0
COM3
SEG06
SEG09
SEG11
SEG53
2
MDSET[1:0]
LCD Mode Register 0 (LCDM0)
LCDON
SCOC
BLON
LCDSEL
LCD Mode Register 1 (LCDM1)
Internal peripheral bus

Figure 44.1 SLCDC block diagram

44.2 Register Descriptions

44.2.1 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: Setting prohibited 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 Other settings are prohibited.	R/W
b5	LWAVE	LCD Display Waveform Select	0: Waveform A 1: Waveform B.	R/W

Bit	Symbol	Bit name	Description	R/W
b7, b6	MDSET[1:0]	LCD Drive Voltage Generator Select	b7 b6 0 0: External resistance division method Other settings are 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 44.4 are supported. Combinations of settings not shown in Table 44.4 are prohibited.

Table 44.4 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
Waveform A	4	1/3	0	0	1	1	0	1		A
Waveform A	3	1/3	0	0	1	0	0	1		A
Waveform A	3	1/2	0	0	1	0	0	0		A
Waveform A	2	1/2	0	0	0	1	0	0		A
Waveform A	Static		0	0	0	0	0	0		A
Waveform B	4	1/3	1	0	1	1	0	1		A

A: Available, N/A: Not available

44.2.2 LCD Mode Register 1 (LCDM1)

Address(es): SLCDC.LCDM1 4008 2001h

	b7	b6	b5	b4	b3	b2	b1	b0
	LCDON	SCOC	—	BLON	LCDSEL	—	—	—
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	LCDSEL	Display Data Area Control	b4 b3 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
b4	BLON	Display Data Area Control		R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	SCOC	LCD Display Enable/Disable	b7 b6 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
b7	LCDON	LCD Display Enable/Disable		R/W

44.2.3 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: Set the frame frequency in a range from 32 Hz to 128 Hz.

Note: Do not set LCDC0 when the LCDM1.SCOC bit is 1.

44.3 LCD Display Data Registers

The LCD display data registers are mapped as shown in Table 44.5. The contents displayed on the LCD can be changed by changing the contents of the LCD display data registers.

Table 44.5 Relationship between LCD Display Data Register contents and segment/common outputs

Register name	Address	b7	b6	b5	b4	b3	b2	b1	b0	
		COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0	56-pin
SEG06	4008 2106h	SEG06 (B-pattern area)				SEG06 (A-pattern area)				A
SEG09	4008 2109h	SEG09 (B-pattern area)				SEG09 (A-pattern area)				A
SEG11	4008 210Bh	SEG11 (B-pattern area)				SEG11 (A-pattern area)				A
SEG12	4008 210Ch	SEG12 (B-pattern area)				SEG12 (A-pattern area)				A
SEG20	4008 2114h	SEG20 (B-pattern area)				SEG20 (A-pattern area)				A
SEG23	4008 2117h	SEG23 (B-pattern area)				SEG23 (A-pattern area)				A
SEG49	4008 2131h	SEG49 (B-pattern area)				SEG49 (A-pattern area)				A
SEG52	4008 2134h	SEG52 (B-pattern area)				SEG52 (A-pattern area)				A
SEG53	4008 2135h	SEG53 (B-pattern area)				SEG53 (A-pattern area)				A

A: Available, N/A: Not available

Note: All LCD display data registers have an initial value of 0h, and all bits that are read/write.

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 correspondence between A-pattern area data and COM signals is as follows:
bit [0] ↔ COM0, bit [1] ↔ COM1, bit [2] ↔ COM2, and bit [3] ↔ COM3.

The correspondence between B-pattern area data and COM signals is 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.

44.4 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, based on 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 an A-pattern and B-pattern area data (blinking display associated with the periodic interrupt timing of the Realtime Clock (RTC)).

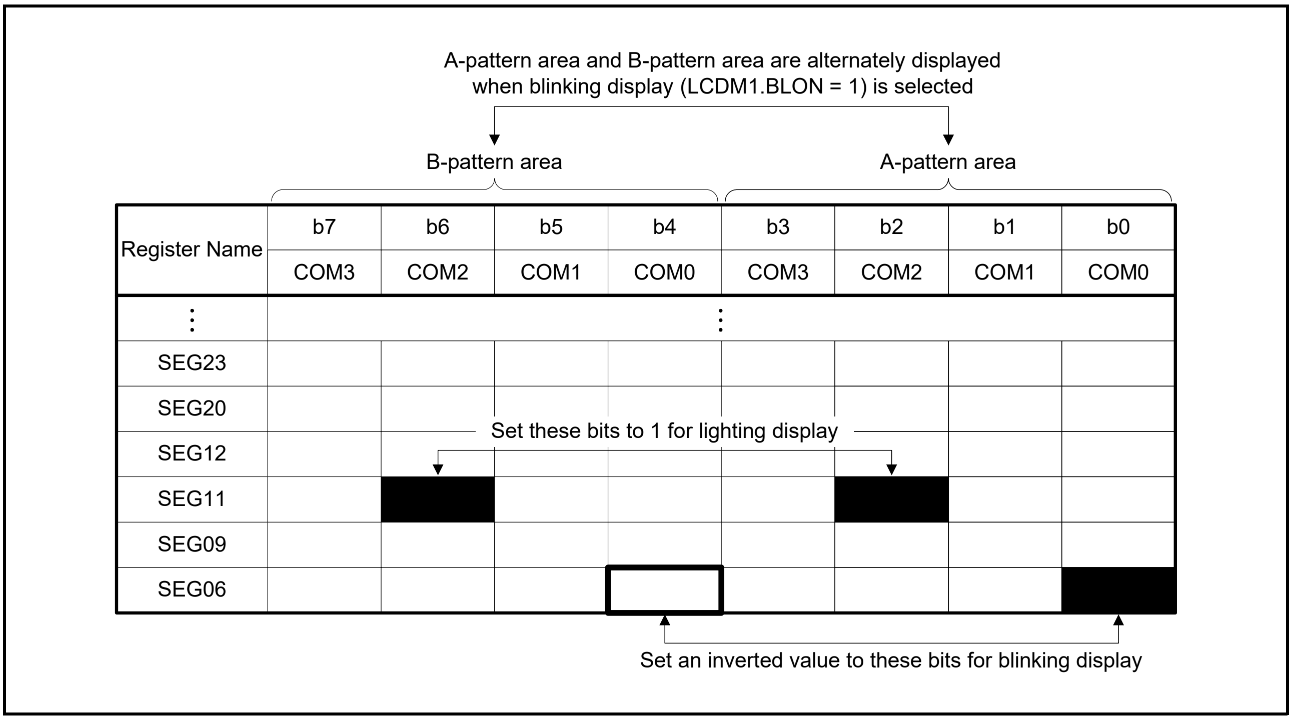

A-pattern area and B-pattern area are alternately displayed
when blinking display (LCDM1.BLON = 1) is selected
B-pattern area
A-pattern area
Register Name
b7
b6
b5
b4
b3
b2
b1
b0
COM3
COM2
COM1
COM0
COM3
COM2
COM1
COM0
SEG23
SEG20
SEG12
SEG11
SEG09
SEG06
Set these bits to 1 for lighting display
Set an inverted value to these bits for blinking display

Figure 44.2 Example for setting LCD display data registers when the pattern is changed

44.4.1 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 44.3, LCD Display Data Registers.

44.4.2 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 SEG06 register to 1, and bit [4] of SEG06 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 SEG09 register to 1, and set bit [6] of SEG09 register to 1 for lighting display. For details on the display area, see section 44.3, LCD Display Data Registers.

Figure 44.3 and Figure 44.4 show the timing operation of display switching.

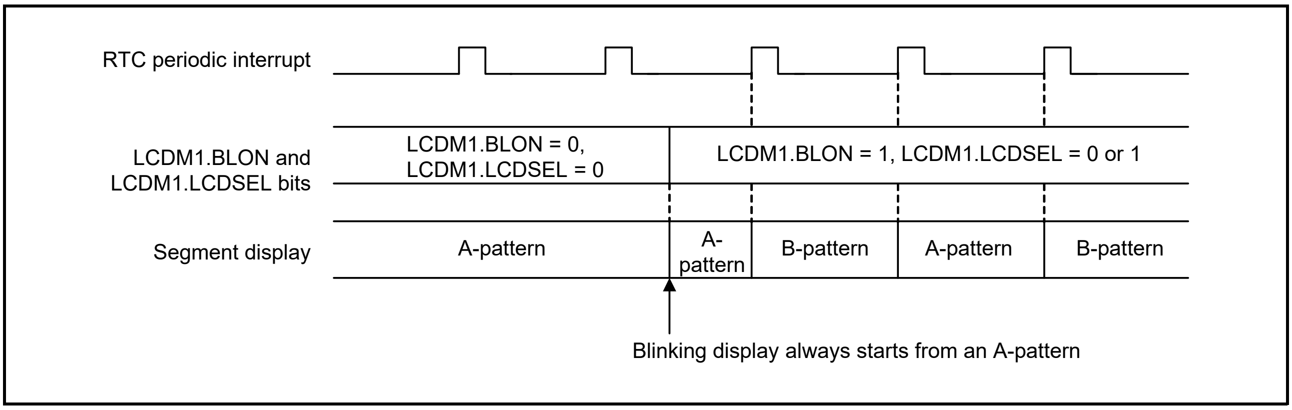

RTC periodic interrupt
LCDM1.BLON and LCDM1.LCDSEL bits
Segment display
LCDM1.BLON = 0, LCDM1.LCDSEL = 0
LCDM1.BLON = 1, LCDM1.LCDSEL = 0 or 1
A-pattern
A-pattern
B-pattern
A-pattern
B-pattern
Blinking display always starts from an A-pattern

Figure 44.3 Switching operation from A-pattern display to blinking display

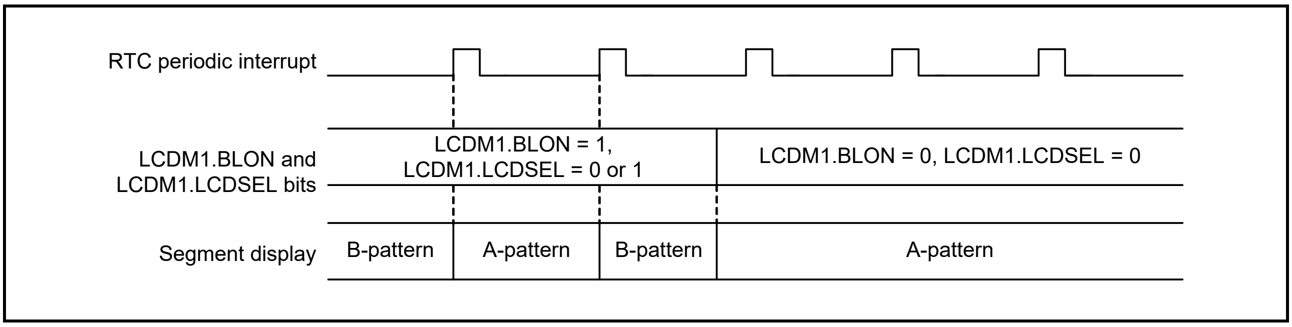

RTC periodic interrupt
LCDM1.BLON and LCDM1.LCDSEL bits
Segment display
LCDM1.BLON = 1, LCDM1.LCDSEL = 0 or 1
LCDM1.BLON = 0, LCDM1.LCDSEL = 0
B-pattern
A-pattern
B-pattern
A-pattern

Figure 44.4 Switching operation from blinking display to A-pattern display

44.5 Setting LCD Controller/Driver

To operate the LCD controller/driver, follow procedures in this section. Otherwise, the LCD operation is not guaranteed.

(1) External resistance division method during normal liquid crystal waveform display

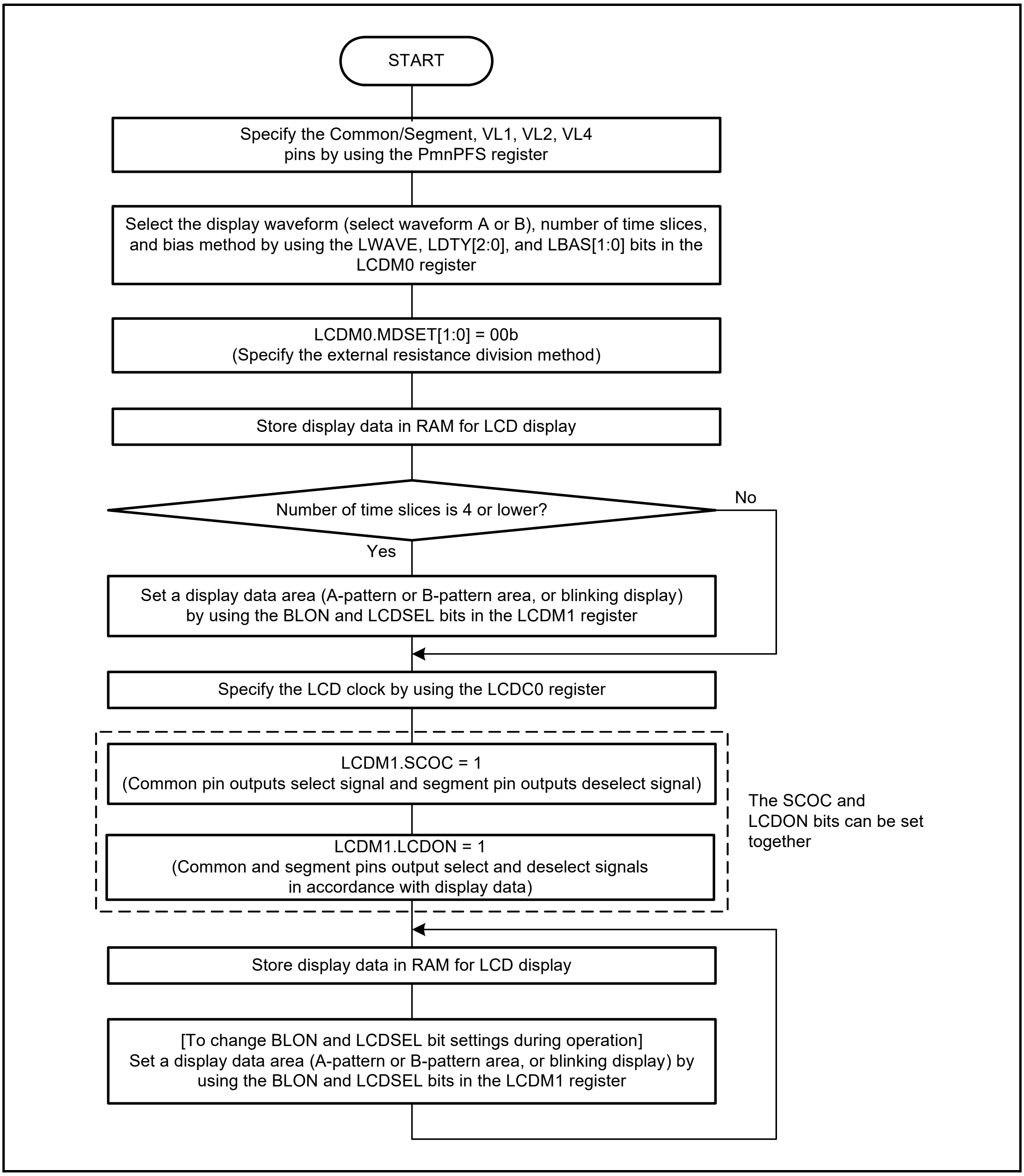

START
Specify the Common/Segment, VL1, VL2, VL4 pins by using the PmnPFS register
Select the display waveform (select waveform A or B), number of time slices, and bias method by using the LWAVE, LDTY[2:0], and LBAS[1:0] bits in the LCDM0 register
LCDM0.MDSET[1:0] = 00b (Specify the external resistance division method)
Store display data in RAM for LCD display
Number of time slices is 4 or lower?
No
Yes
Set a display data area (A-pattern or B-pattern area, or blinking display) by using the BLON and LCDSEL bits in the LCDM1 register
Specify the LCD clock by using the LCDC0 register
LCDM1.SCOC = 1 (Common pin outputs select signal and segment pin outputs deselect signal)
LCDM1.LCDON = 1 (Common and segment pins output select and deselect signals in accordance with display data)
The SCOC and LCDON bits can be set together
Store display data in RAM for LCD display
[To change BLON and LCDSEL bit settings during operation] Set a display data area (A-pattern or B-pattern area, or blinking display) by using the BLON and LCDSEL bits in the LCDM1 register

Figure 44.5 Setting procedure for external resistance division method during normal liquid crystal waveform display

44.6 Operation Stop Procedure

To stop the operation of the LCD, follow the steps shown in Figure 44.6.

The LCD stops operating when the LCDM1.LDCON and LCDM1.SCOC bits are set to 0.

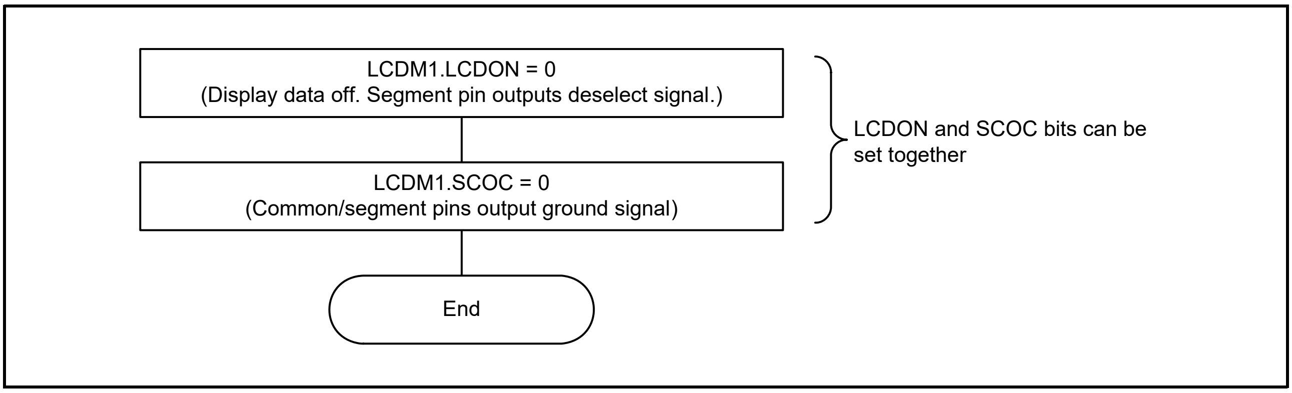

LCDM1.LCDON = 0
(Display data off. Segment pin outputs deselect signal.)
LCDM1.SCOC = 0
(Common/segment pins output ground signal)
End
LCDON and SCOC bits can be set together

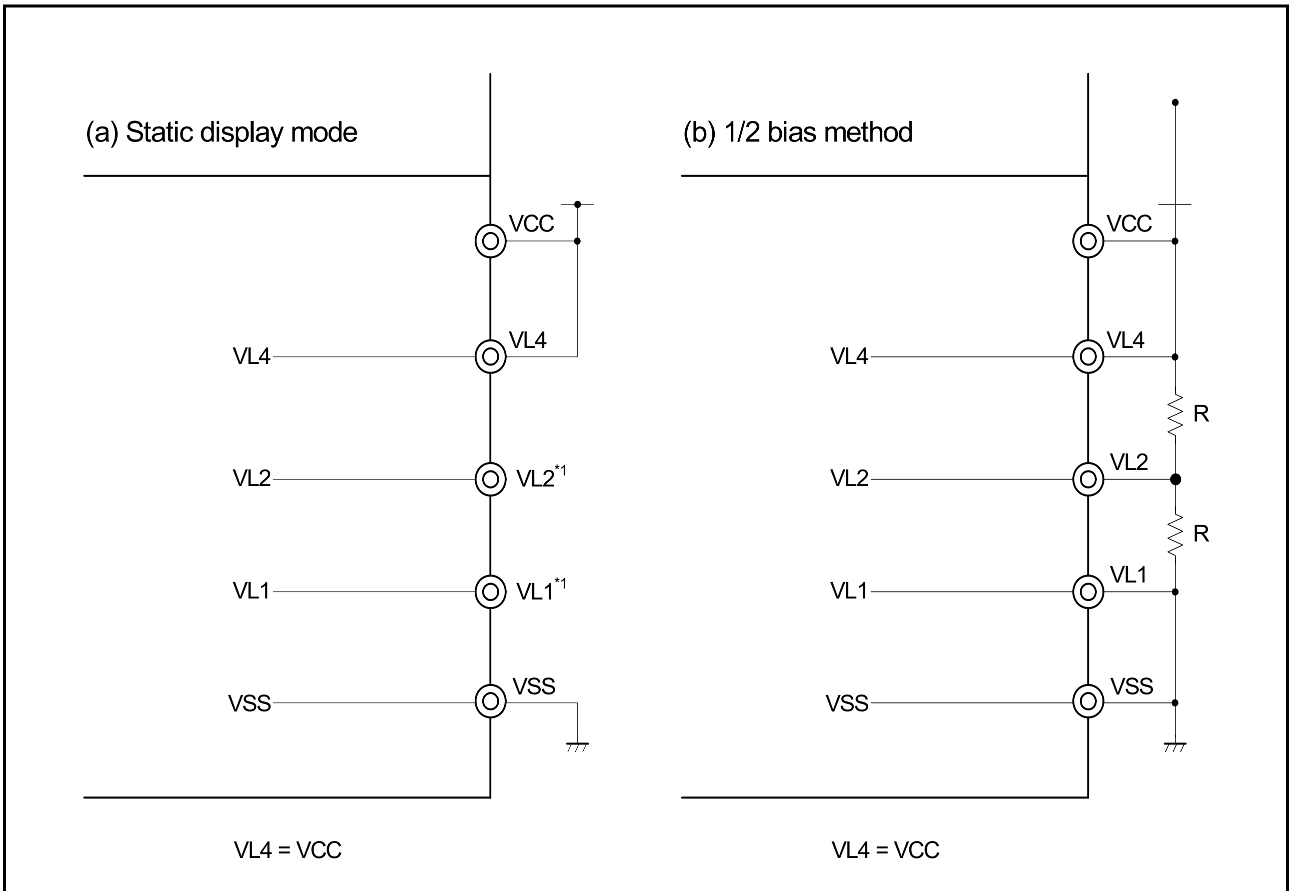
Figure 44.6 Operation stop procedure during normal liquid crystal waveform (A or B) display

44.7 Supplying LCD Drive Voltages VL1, VL2, and VL4

The power supply voltages for the LCD driver can be produced through external resistance division.

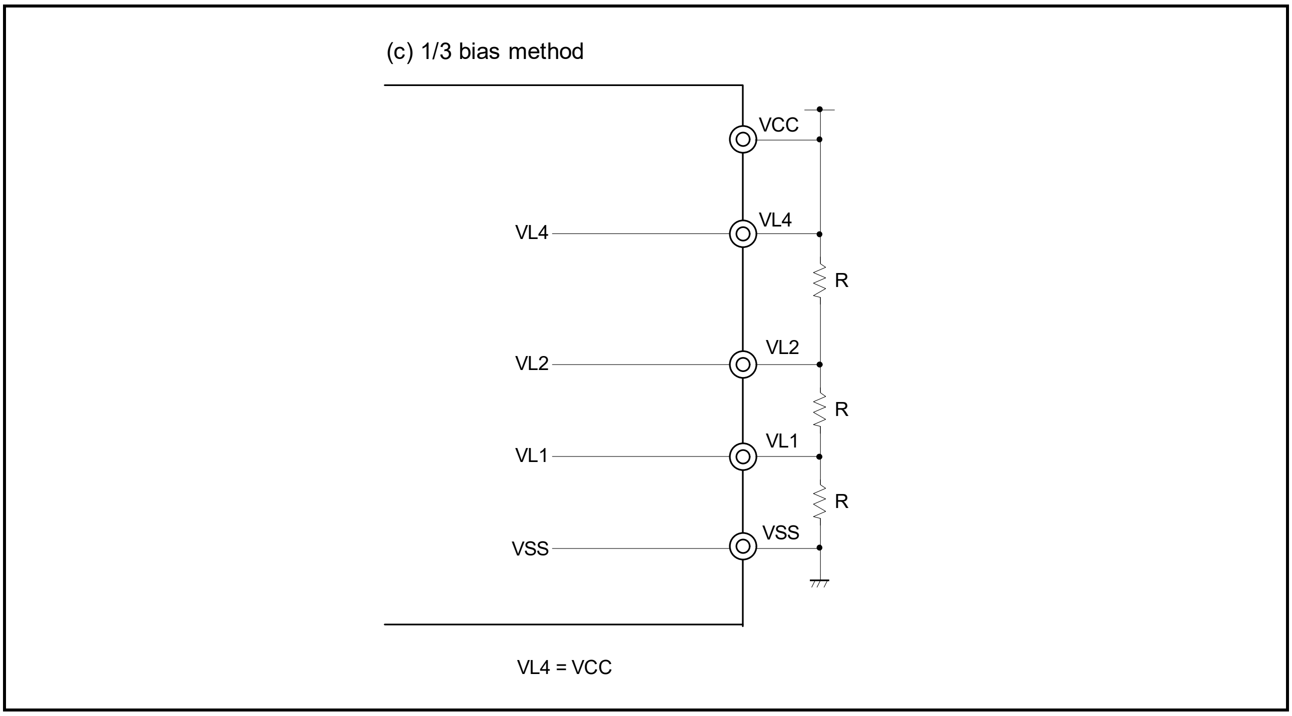
44.7.1 External Resistance Division Method

Figure 44.7 and Figure 44.8 show examples of the LCD drive power supply connection, associated with each bias method.


(a) Static display mode
VCC
VL4
VL4
VL2
VL2*1
VL1
VL1*1
VSS
VSS
VL4 = VCC
(b) 1/2 bias method
VCC
VL4
VL4
R
VL2
VL2
R
VL1
VL1
VSS
VSS
VL4 = VCC

Note 1. Connect VL1 and VL2 to GND or leave open.

Figure 44.7 Examples of LCD drive power connections using the external resistance division method (1 of 2)


(c) 1/3 bias method

VL4 = VCC

Figure 44.8 Examples of LCD drive power connections using external resistance division method (2 of 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.

44.8 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 44.6. 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.

Table 44.6 COM signal

Number of time slices	COM0	COM1	COM2	COM3
Static display mode	→	→	→	→
Two-time-slice mode	→		Open	Open
Three-time-slice mode	→			Open
Four-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 44.3, LCD Display Data Registers).

Bit [0] to bit [3] of each byte in A-pattern area are read in synchronization with COM0 to COM3, and bit [4] to bit [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 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 44.7 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 44.7 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

Figure 44.9 and Figure 44.10 show the common signal waveforms. Figure 44.11 to Figure 44.16 show the voltages and phases of the common and segment signals.

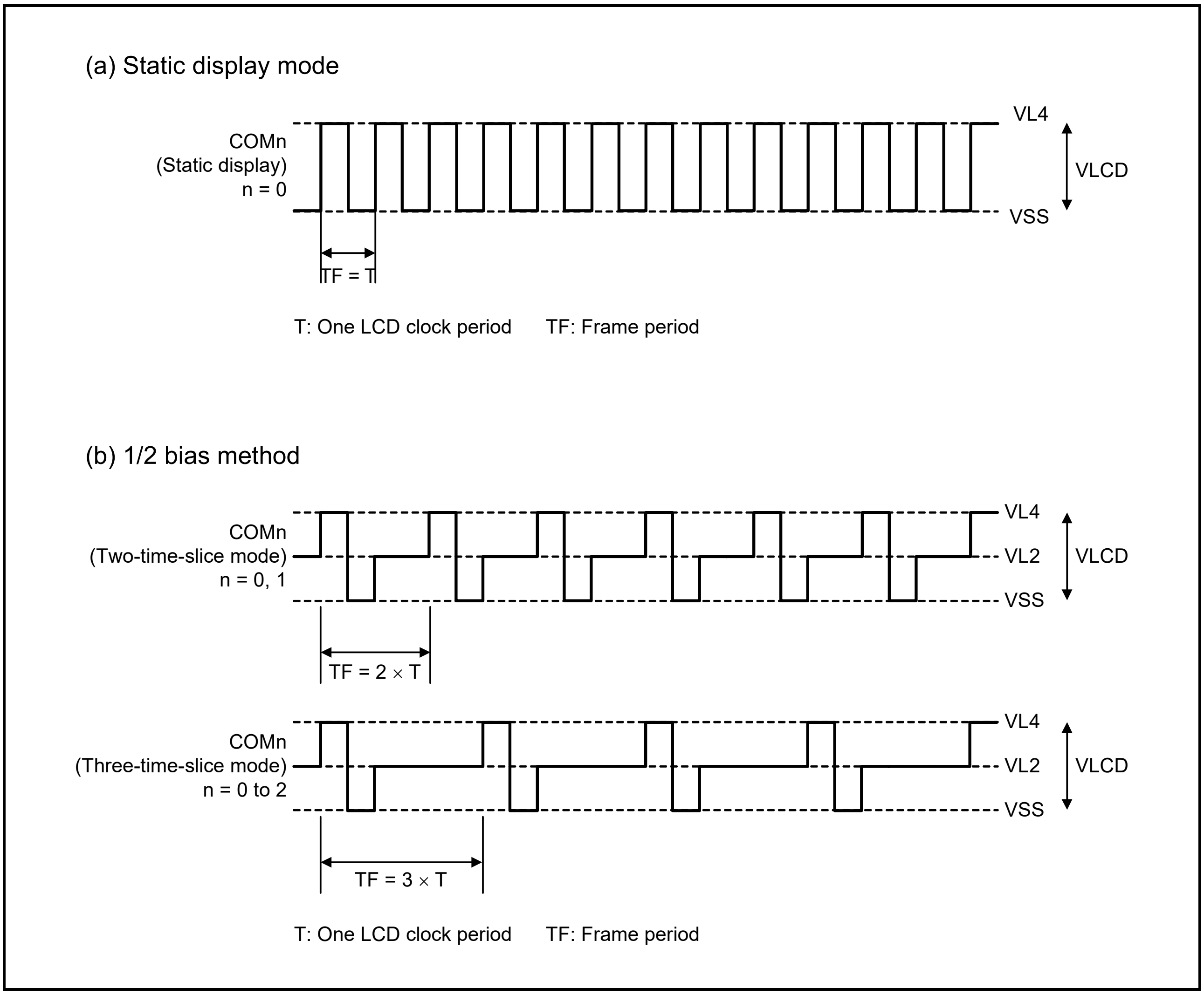
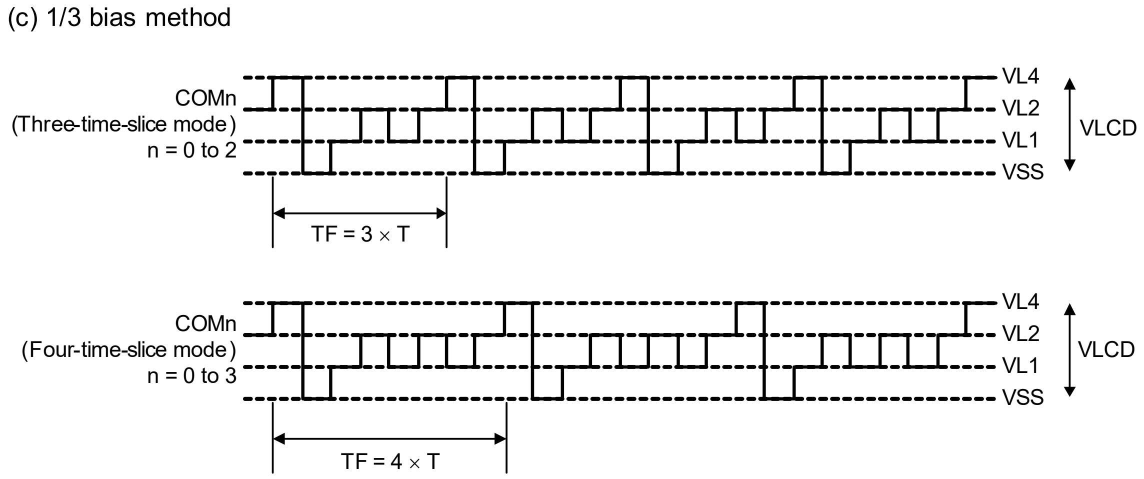

(a) Static display mode
COMn
(Static display)
n = 0
VL4
VLCD
VSS
TF = T
T: One LCD clock period
TF: Frame period
(b) 1/2 bias method
COMn
(Two-time-slice mode)
n = 0, 1
VL4
VL2
VLCD
VSS
TF = 2 × T
COMn
(Three-time-slice mode)
n = 0 to 2
VL4
VL2
VLCD
VSS
TF = 3 × T
T: One LCD clock period
TF: Frame period

Figure 44.9 Common signal waveforms (1 of 2)

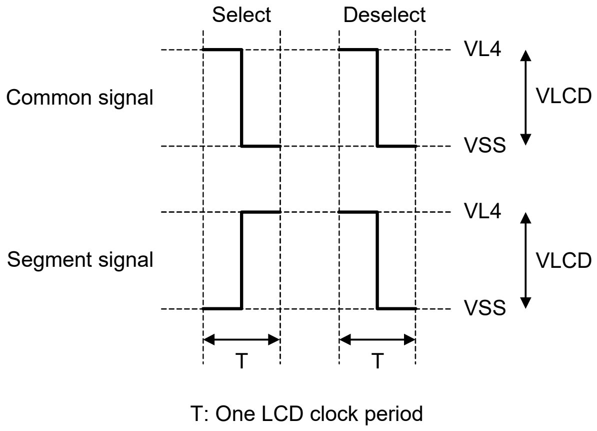

(c) 1/3 bias method
COMn
(Three-time-slice mode)
n = 0 to 2
VL4
VL2
VL1
VSS
VLCD
TF = 3 × T
COMn
(Four-time-slice mode)
n = 0 to 3
VL4
VL2
VL1
VSS
VLCD
TF = 4 × T

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

Figure 44.10 Common signal waveforms (2 of 2)

(a) Static display mode (waveform A)


Select
Deselect
Common signal
VL4
VSS
VLCD
Segment signal
VL4
VSS
VLCD
T
T
T: One LCD clock period

(b) 1/2 bias method (waveform A)

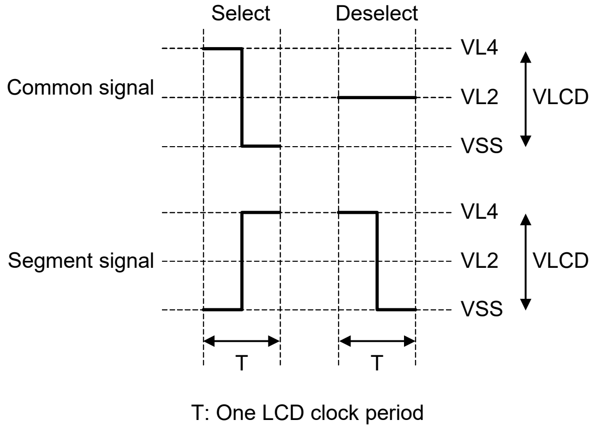
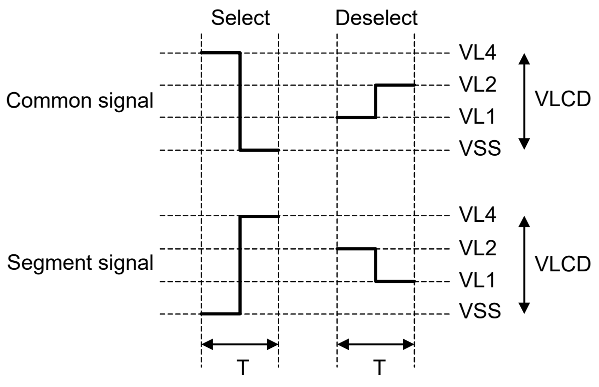

Select
Deselect
Common signal
VL4
VL2
VSS
VLCD
Segment signal
VL4
VL2
VSS
VLCD
T
T
T: One LCD clock period

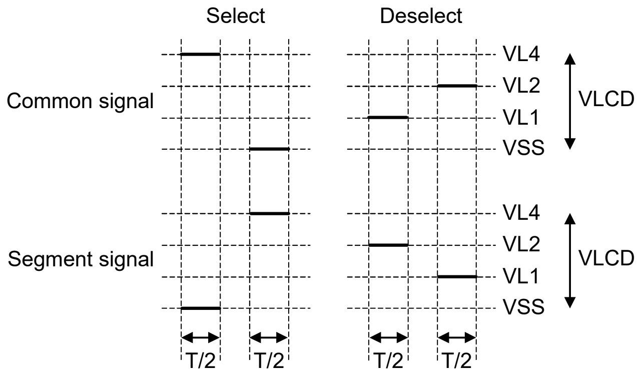
Figure 44.11 Voltages and phases of common and segment signals (1 of 3)

(c) 1/3 bias method (waveform A)


Select
Deselect
Common signal
VL4
VL2
VL1
VSS
VLCD
Segment signal
VL4
VL2
VL1
VSS
VLCD
T
T

T: One LCD clock period

(d) 1/3 bias method (waveform B)


Select
Deselect
Common signal
VL4
VL2
VL1
VSS
VLCD
Segment signal
VL4
VL2
VL1
VSS
VLCD
T/2
T/2
T/2
T/2

T: One LCD clock period

Figure 44.12 Voltages and phases of common and segment signals (2 of 3)

44.9 Display Modes

44.9.1 Four-Time-Slice Display Example

Figure 44.14 shows how a 12-digit LCD panel with the display pattern shown in Figure 44.13 is connected to the segment signals and the common signals (COM0 to COM3). This example displays “56.78” 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 seventh digit. To display “6.” in the LCD panel, the select or deselect voltage must be applied to the SEG20 and SEG23 pins at the select timing of the common signals COM0 to COM3. See Figure 44.13 for the relationship between the segment signals and LCD segments.

Table 44.8 Example of select (1) and deselect (0) data (COM0 to COM3) (1 of 2)

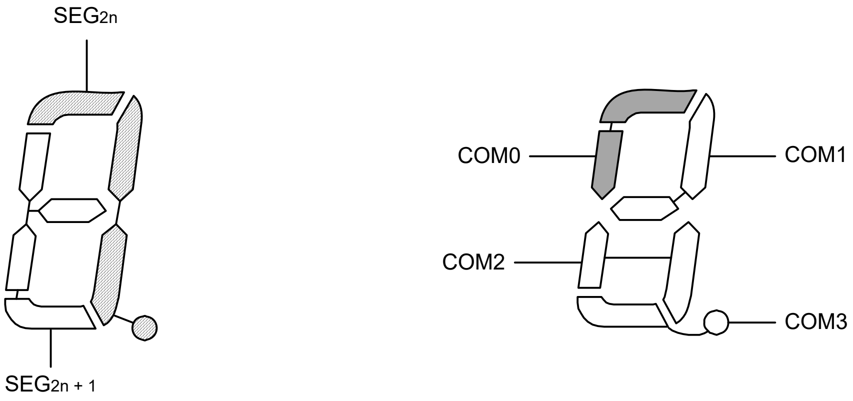
	Segment	
Common	**SEG20**	**SEG23**
COM0	Select	Select

Table 44.8 Example of select (1) and deselect (0) data (COM0 to COM3) (2 of 2)

	Segment	
Common	**SEG20**	**SEG23**
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to Table 44.8, the display data register location that corresponds to SEG20 must contain "1101b".

Figure 44.15 shows examples of LCD drive waveforms between the SEG20 signal and each common signal. When the select voltage is applied to SEG20 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 44.13 Four-time-slice LCD display pattern and electrode connections

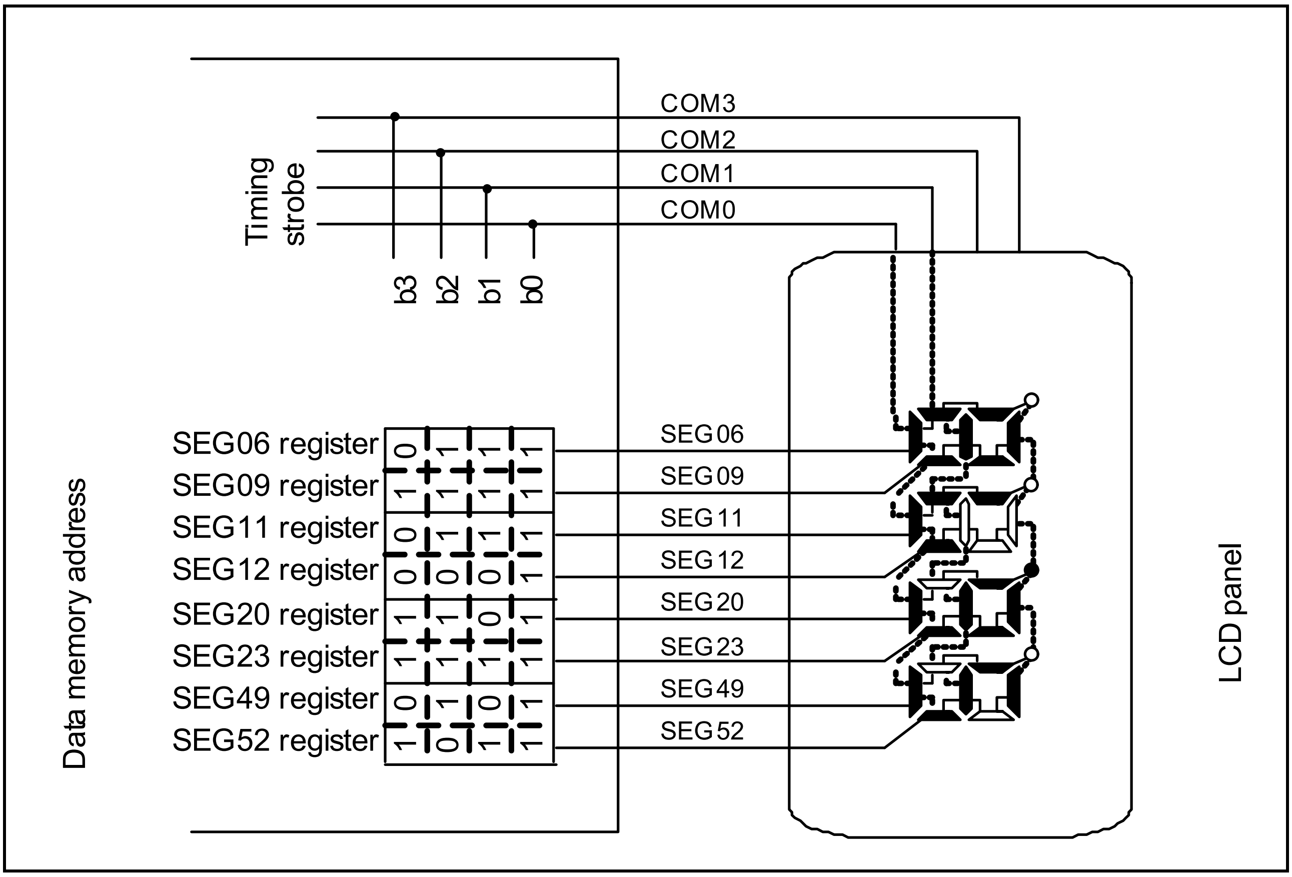

Timing strobe
COM3
COM2
COM1
COM0
b3
b2
b1
b0
Data memory address
SEG06 register 0 1 1 1
SEG09 register 1 1 1 1
SEG11 register 0 1 1 1
SEG12 register 0 0 0 1
SEG20 register 1 1 0 1
SEG23 register 1 1 1 1
SEG49 register 0 1 0 1
SEG52 register 1 0 1 1
SEG06
SEG09
SEG11
SEG12
SEG20
SEG23
SEG49
SEG52
LCD panel

Figure 44.14 Example for connecting four-time-slice LCD panel

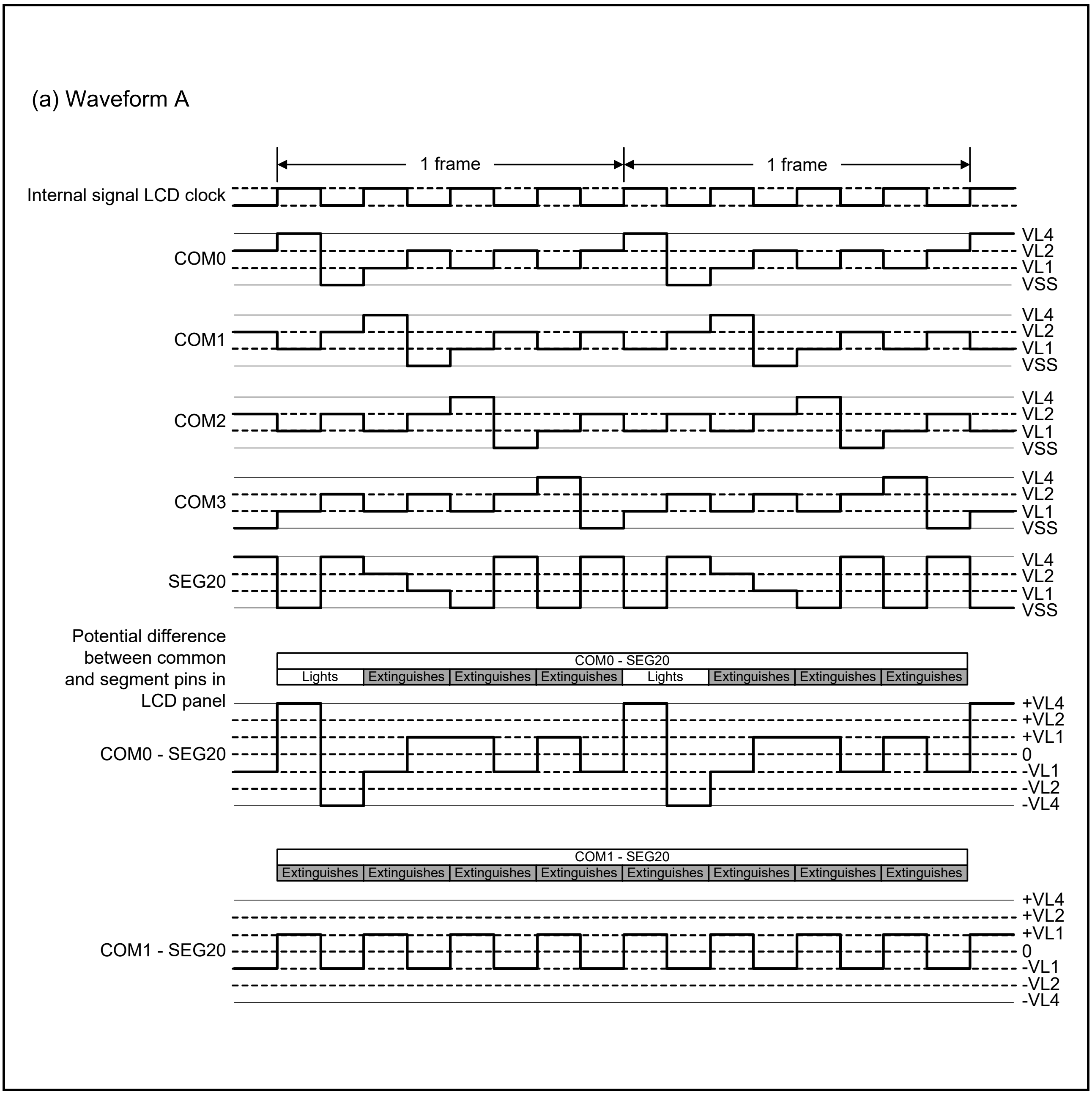

(a) Waveform A
1 frame
1 frame
Internal signal LCD clock
COM0
COM1
COM2
COM3
SEG20
VL4
VL2
VL1
VSS
Potential difference between common and segment pins in LCD panel
COM0 - SEG20
Lights
Extinguishes
Extinguishes
Extinguishes
Lights
Extinguishes
Extinguishes
Extinguishes
+VL4
+VL2
+VL1
0
-VL1
-VL2
-VL4
COM1 - SEG20
Extinguishes
Extinguishes
Extinguishes
Extinguishes
Extinguishes
Extinguishes
Extinguishes
Extinguishes

Figure 44.15 Four-time-slice LCD drive waveform examples between SEG20 and each common signal using 1/3 bias method (1 of 2)

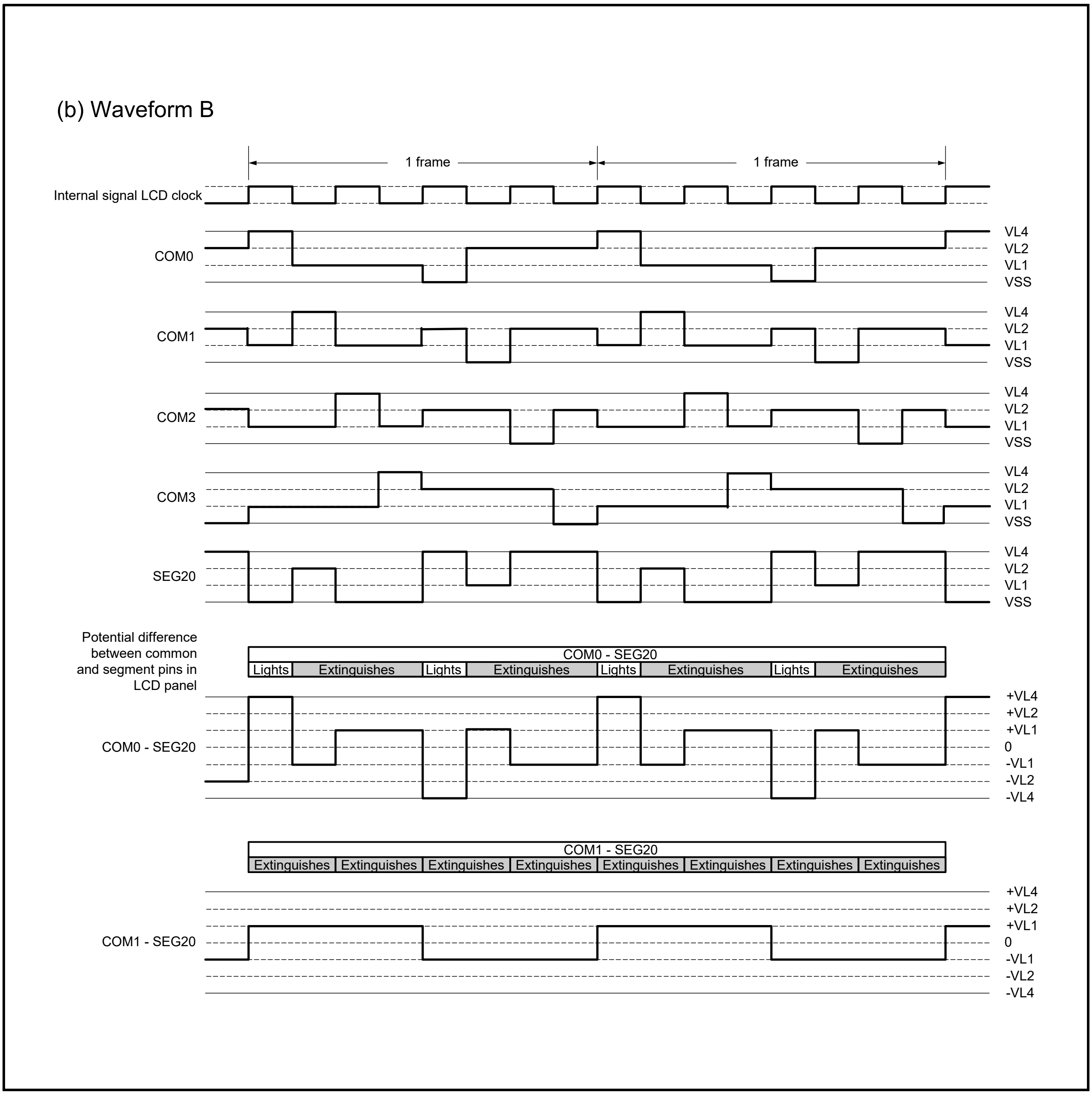

(b) Waveform B
1 frame
1 frame
Internal signal LCD clock
COM0
COM1
COM2
COM3
SEG20
VL4
VL2
VL1
VSS
Potential difference between common and segment pins in LCD panel
COM0 - SEG20
Lights
Extinguishes
Lights
Extinguishes
Lights
Extinguishes
Lights
Extinguishes
+VL4
+VL2
+VL1
0
-VL1
-VL2
-VL4
COM1 - SEG20
Extinguishes
Extinguishes
Extinguishes
Extinguishes
Extinguishes
Extinguishes
Extinguishes
Extinguishes

Figure 44.16 Four-time-slice LCD drive waveform examples between SEG12 and each common signal using 1/3 bias method (2 of 2)

45. Secure Cryptographic Engine (SCE5)

45.1 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.

Table 45.1 shows the SCE5 specifications and Figure 45.1 shows the SCE5 block diagram.

Table 45.1 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 - 61 PCLKA cycles for 256-bit key. 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

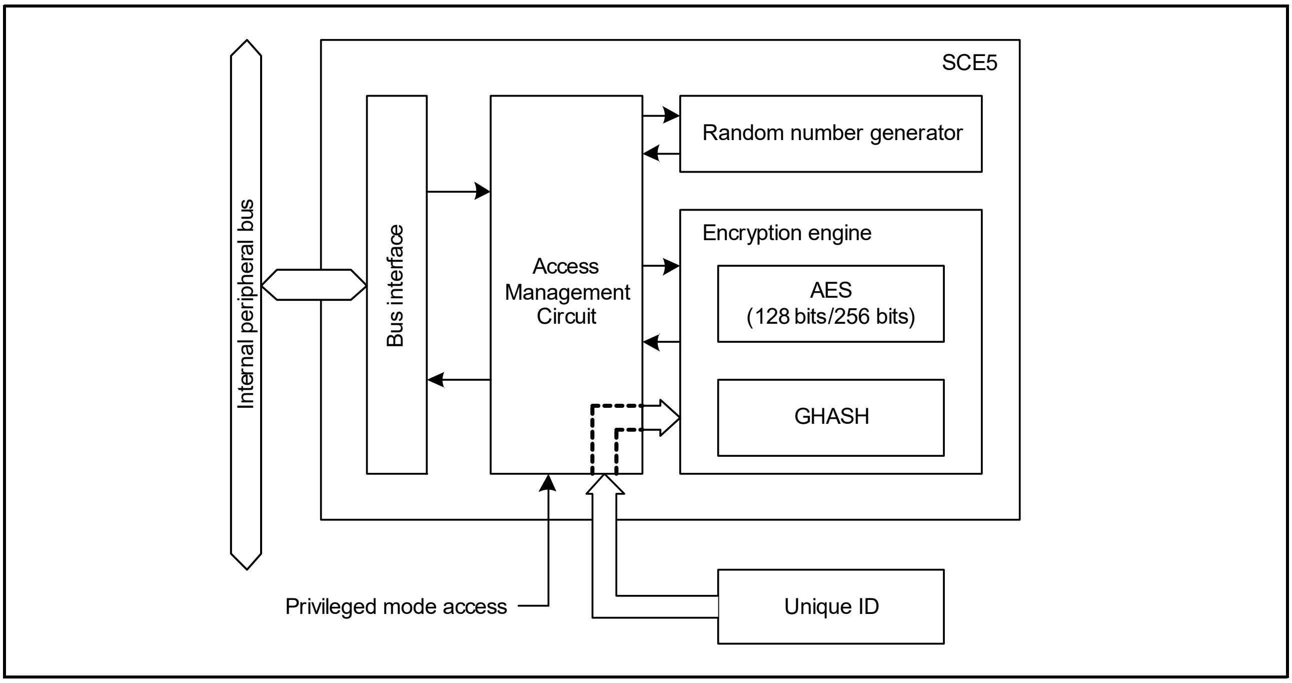

SCE5
Internal peripheral bus
Bus interface
Access Management Circuit
Random number generator
Encryption engine
AES (128 bits/256 bits)
GHASH
Privileged mode access
Unique ID

Figure 45.1 SCE5 block diagram

45.2 Operation

45.2.1 Encryption Engine

The encryption engine performs the following operation in hardware, as shown in Figure 45.2:

- Plaintext to ciphertext encryption
- Ciphertext to plaintext decryption.

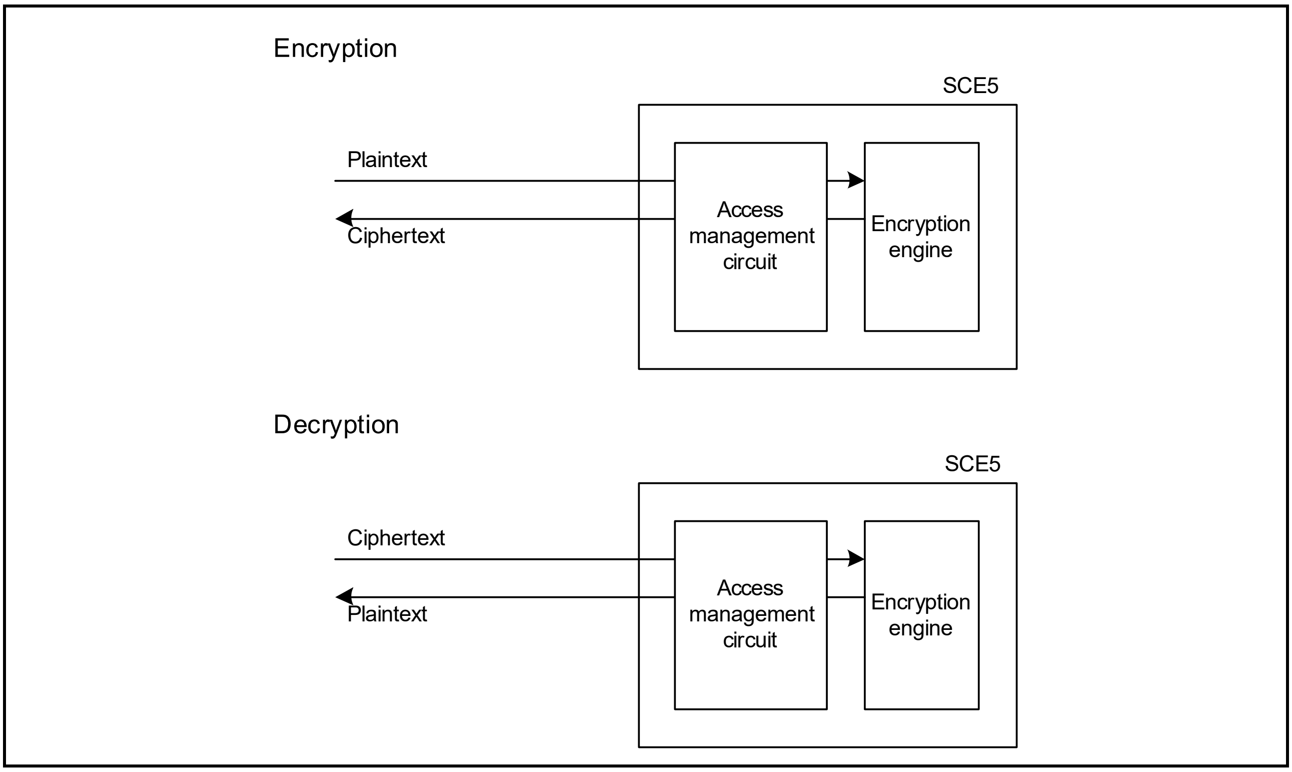

Encryption
SCE5
Plaintext
Ciphertext
Access management circuit
Encryption engine
Decryption
SCE5
Ciphertext
Plaintext
Access management circuit
Encryption engine

Figure 45.2 Encryption and decryption processes by encryption engine

45.2.2 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 45.3 shows the encryption engine timing.

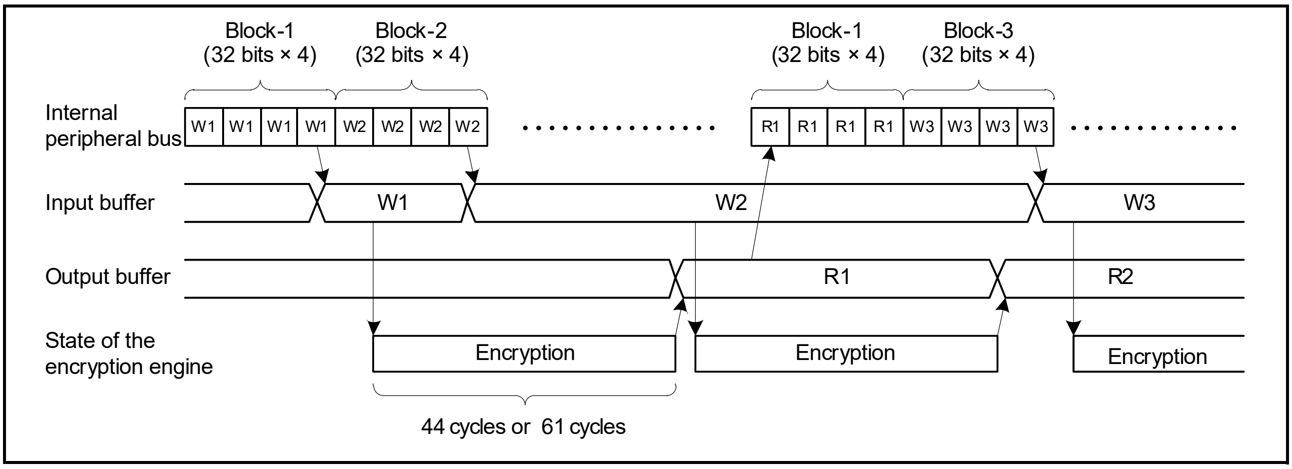

Block-1
(32 bits × 4)
Block-2
(32 bits × 4)
Block-1
(32 bits × 4)
Block-3
(32 bits × 4)
Internal peripheral bus
W1 W1 W1 W1 W2 W2 W2 W2
R1 R1 R1 R1 W3 W3 W3 W3
Input buffer
W1
W2
W3
Output buffer
R1
R2
State of the encryption engine
Encryption
Encryption
Encryption
44 cycles or 61 cycles

Figure 45.3 Encryption and decryption timing (AES)

45.3 Usage Notes

45.3.1 Software Standby Mode

If the MCU enters Software Standby mode while the encryption engine is processing, proper processing cannot be resumed after exiting Software Standby mode. Therefore, it is necessary to enter Software Standby mode while the encryption engine is not running.

45.3.2 Settings for the Module-Stop Function

The Module Stop Control Register C (MSTPCRC) can enable or disable SCE5 operation. The SCE5 module is initially stopped after reset. Releasing the module-stop state enables access to the registers.

46. Bluetooth Low Energy (BLE)

This MCU has a Bluetooth Low Energy (BLE), which consists of an RF transceiver compliant with Bluetooth 5.0 Low Energy (single mode), a link layer, and an RF transceiver power-supply.

The BLE is controlled by a Bluetooth middlewear available from Renesas Electronics Corporation.

46.1 Overview

Table 46.1 lists the Specifications of the BLE. Figure 46.1 is a BLE Block Diagram.

Table 46.1 Specifications of the BLE

Item	Description		
Features	• RF transceiver compliant with the Bluetooth 5.0 Low Energy specification and a link layer • An on-chip matching circuit allows reducing the number of external parts • On-chip BLE-dedicated high-precision low-speed oscillator (32.768 kHz) • Transmission power is selectable as 0 dBm or +4 dBm. • The minimum receiving sensitivity is selectable from among –105 dBm (125 kbps), –100 dBm (500 kbps), –95 dBm (1 Mbps), or –92 dBm (2 Mbps).		
Bluetooth 5.0 functions	Classification	Function	Remark
	Device Address	Public or random address	The address can be set as a desired address.
	Advertising	Extended or periodic	
		Multiple advertising	Maximum number of sets: 4
		Advertising or Scan Response Data	Maximum data length=1650 bytes
	Scanning	Passive, active, or periodic	Number of units for concurrent synchronization with periodic advertising=2
		Whitelist or periodic advertiser list	Number of units registered in the whitelist: 4 Number of units registered in the periodic advertiser list: 4
	Master or slave	Data transmission or reception	Maximum payload length=251 bytes MoreData function is supported. Master/slave multi-role function is supported.
	Other	Bit rates	125 kbps, 500 kbps, 1 Mbps, 2 Mbps Bit-rate combinations for transmission and reception can be set as desired.
		Frequency hopping	Channel Selection Algorithm #2
		Encryption circuit for Bluetooth	On-chip Bluetooth-dedicated AES-CCM (128 bits) circuit
Other functions	RF transceiver power-supply (DC-to-DC converter, and linear regulator)		

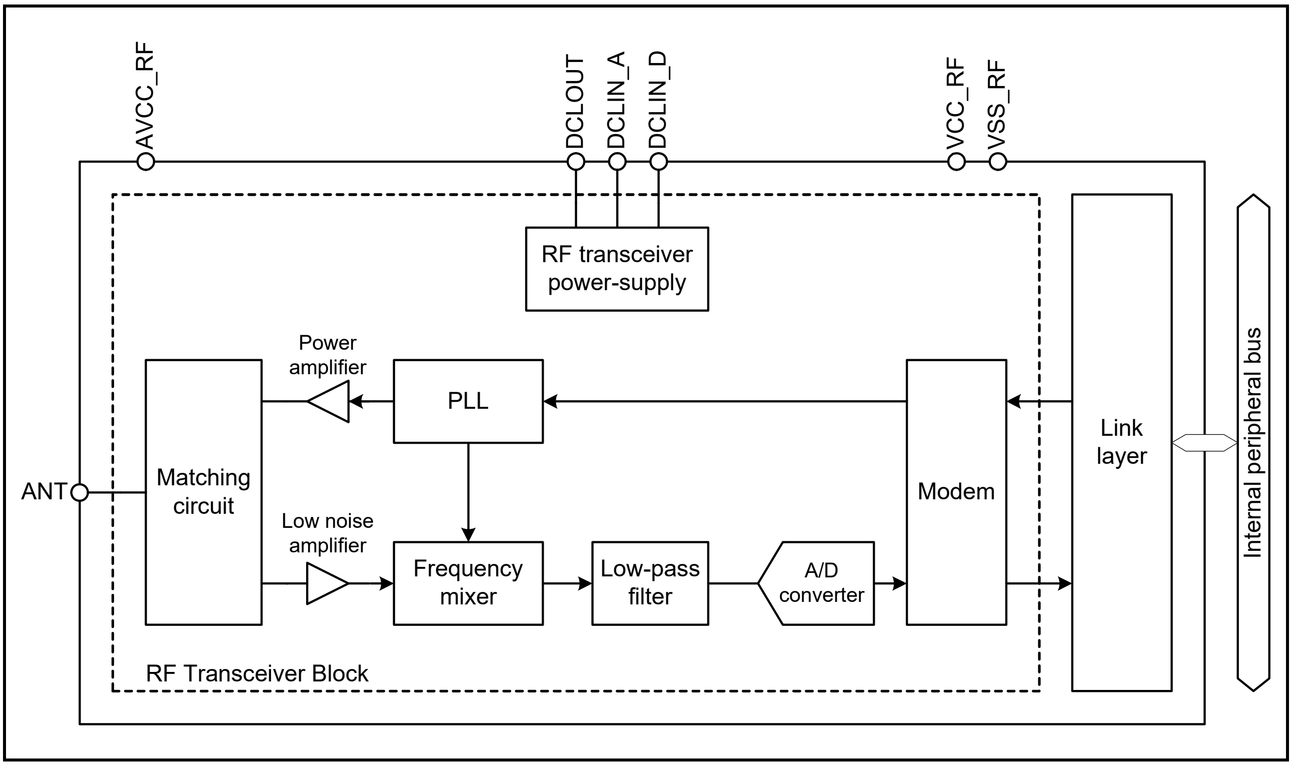

AVCC_RF
DCLOUT
DCLIN_A
DCLIN_D
VCC_RF
VSS_RF
RF transceiver power-supply
Power amplifier
PLL
Modem
Link layer
Internal peripheral bus
ANT
Matching circuit
Low noise amplifier
Frequency mixer
Low-pass filter
A/D converter
RF Transceiver Block

Figure 46.1 **BLE Block Diagram**

Table 46.2 lists the BLE I/O Pins.

Table 46.2 **BLE I/O Pins**

Pin name	I/O	Description
ANT	I/O	RF single I/O pin for RF transceiver Set the impedance of the signal line to 50 Ω.
DCLOUT	Output	Output pin for the RF transceiver power-supply
DCLIN_A, DCLIN_D	Input	If the DC-to-DC converter is to be selected as the power supply for the RF transceiver, connect an inductor and capacitor between the DCLOUT pin and the converter. If the linear regulator is to be selected as the power supply for the RF transceiver, connect a capacitor between the DCLOUT pin and the converter.
VCC_RF	Input	RF transceiver power-supply pin
AVCC_RF	Input	RF transceiver power-supply pin
VSS_RF	Input	RF transceiver ground pin

Table 46.3 lists the Bluetooth Low Energy functions that this MCU supports. The use of these functions requires the Bluetooth middleware.

Table 46.3 **List of Supported Bluetooth Low Energy Functions (1 of 2)**

Function	Bluetooth® core spec
Low Energy Controller (PHY and LL)	v4.0
Low Energy Host (L2CAP and Security Manager)	
Attribute Protocol and Generic Attribute Profile	

Table 46.3 List of Supported Bluetooth Low Energy Functions (2 of 2)

Function	Bluetooth® core spec
Appearance Data Type	v4.1
Low Duty Cycle Directed Advertising	
32-bit UUID Support in LE	
LE L2CAP Connection Oriented Channel Support	
LE Link Layer Topology	
LE Ping	
LE Data Packet Length Extension	v4.2
LE Secure Connections	
Link Layer Privacy	
Link Layer Extended Filter Policies	
LE 2M PHY	v5.0
LE Coded PHY	
High Duty Cycle Non-Connectable Advertising	
LE Advertising Extensions	
LE Channel Selection Algorithm #2	

46.2 Operation

46.2.1 State Transitions

Figure 46.2 is the State Transition Diagram of the BLE.

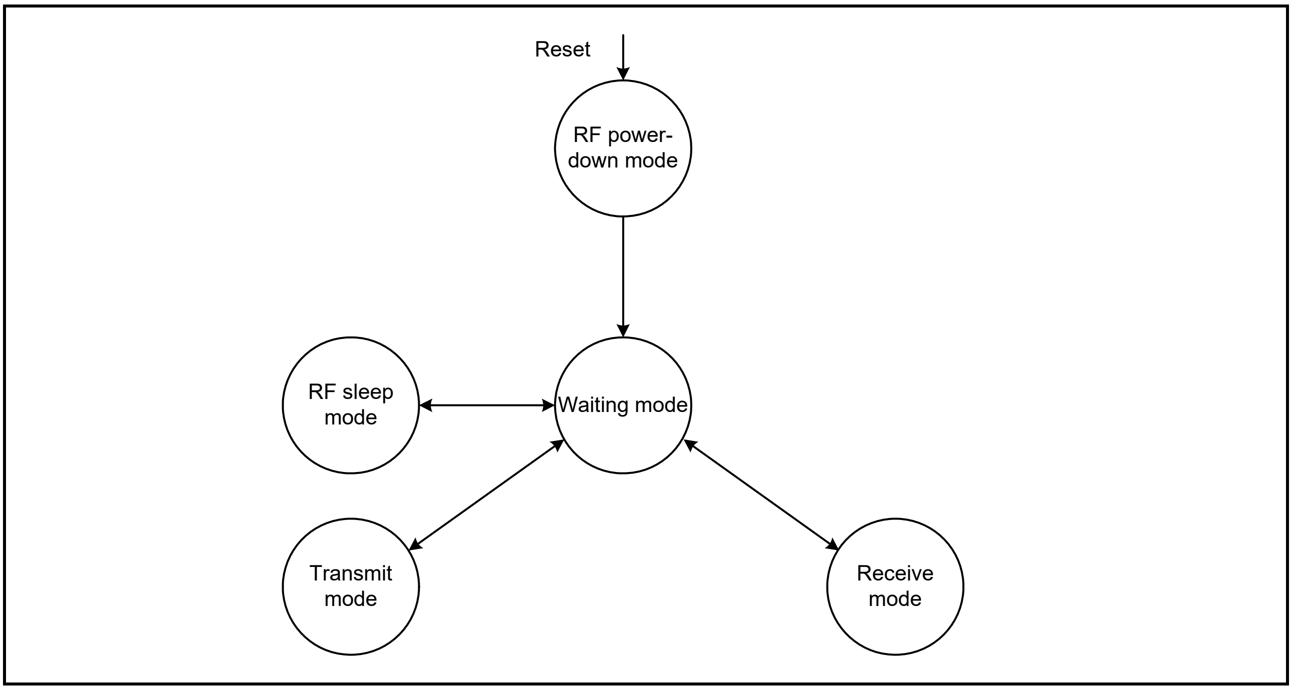


Figure 46.2 State Transition Diagram of the BLE

RF Power-Down Mode

The RF power-down mode is the initial mode of the BLE after release from the reset state.

Power is supplied to the MCU, but not to the RF transceiver.

Waiting Mode

In waiting mode, the BLE waits until the transmission or reception of data or a transition to sleep mode is requested.

RF Sleep Mode

The RF sleep mode is a low-power operating mode and the power supply, except to part of the link-layer circuit, is stopped.

Transmit Mode

The transmit mode is for the transmission of data.

The mode shifts to waiting mode after the data have been transmitted.

Receive Mode

The receive mode is for the reception of data.

The mode shifts to waiting mode after the data have been received.

46.3 Interrupts

Table 46.4 shows the interrupt sources.

The Bluetooth middleware executes processing in response to this interrupts.

Table 46.4 BLE Interrupt Sources

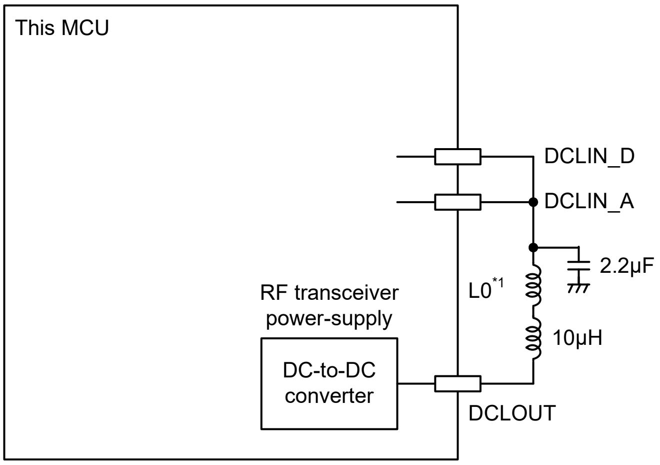
Name	DTC Activation	DMAC Activation
BLEIRQ	Not possible	Not possible

46.4 Usage Notes

46.4.1 RF Transceiver Power-Supply

The RF transceiver power-supply is selectable as either from the DC-to-DC converter or linear regulator.

Figure 46.3 and Figure 46.4 show an example of the external connection circuit when the DC-to-DC converter or the linear regulator is selected, respectively.


This MCU
DCLIN_D
DCLIN_A
2.2μF
RF transceiver power-supply
L0*1
10μH
DC-to-DC converter
DCLOUT

Note 1. Including wiring for the inductor L0 is recommended to reduce the noise of the DC-to-DC converter when designing the basic board.
Details are explained in the guidelines for designing Bluetooth boards (R01AN4886).
Obtain the latest version of this document from the Renesas website.

Figure 46.3 Example of the BLE Externally Connected Circuit When the DC-to-DC Converter is Selected

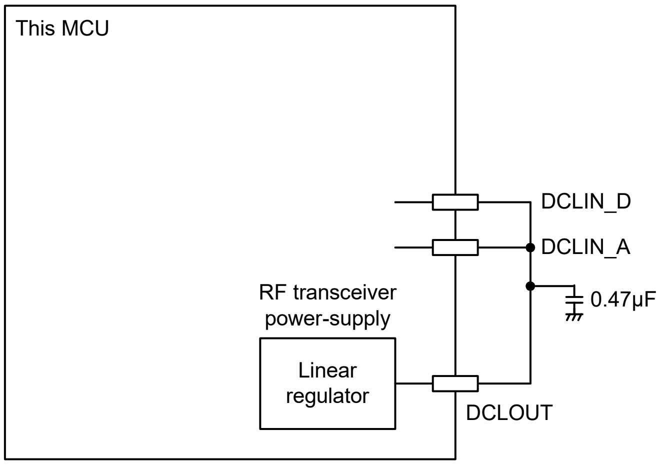

This MCU
DCLIN_D
DCLIN_A
0.47μF
RF transceiver power-supply
Linear regulator
DCLOUT

Figure 46.4 Example of the BLE Externally Connected Circuit When the Linear Regulator is Selected

Do not connect other pins or external circuits to the I/O pins (DCLOUT, DCLIN_D, or DCLIN_A) of the RF transceiver power-supply whether the DC to DC converter or linear regulator is selected.

46.4.2 Wireless Standards

International standards and domestic regulations stipulate the use of wireless receivers and transmitters.

Use the device in compliance with the standards or regulations of the country in which the device is being used.

Main standards applicable to the 2.4 GHz band:

- Japan: ARIB STD-T66

- U.S.A.: FCC CFR Title 47 parts 15.247 and 15.249
- Europe: EN 300 440 and EN 300 328

46.4.3 Notes on Board Design

The applicability of the notes on board design differs with the wireless standard to be employed.

Notes on board design are explained in more detail in the guidelines for designing Bluetooth boards (R01AN4886). Obtain the latest version of this document from the Renesas website.

47. Internal Voltage Regulator

47.1 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.

47.2 Operation

Table 47.1 lists the LDO mode pin settings, and Figure 47.1 shows the LDO mode settings. In LDO mode, the internal voltage is generated from VCC.

Table 47.1 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.

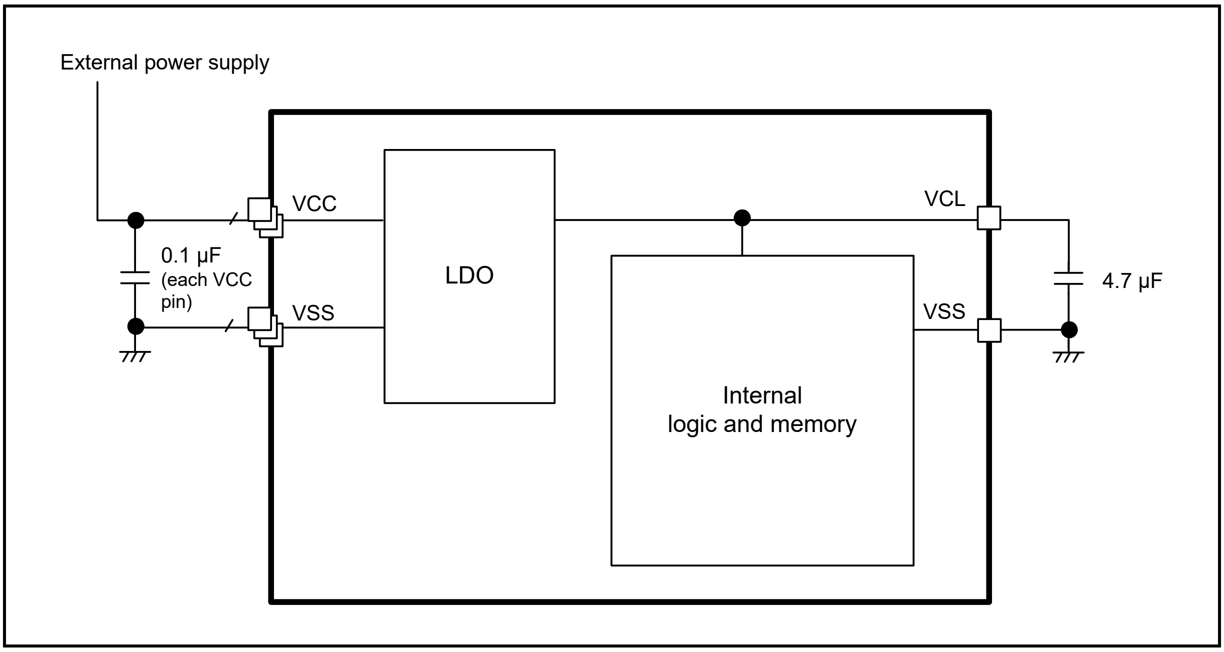

External power supply
VCC
0.1 µF
(each VCC
pin)
VSS
LDO
VCL
VSS
4.7 µF
Internal
logic and memory

Figure 47.1 LDO mode settings

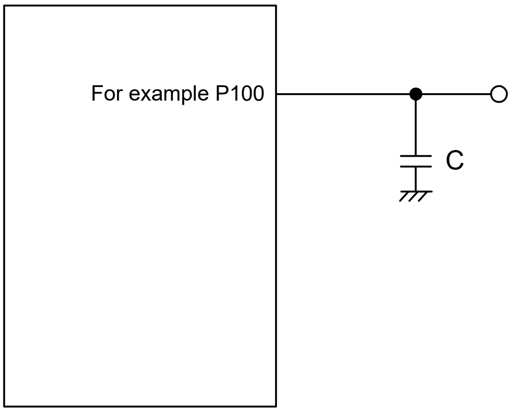
48. Electrical Characteristics

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 = VCC_RF = AVCC_RF = 1.8 to 3.6V, VREFH0 = 1.8 to AVCC0, VBATT = 1.8 to 3.6V, VSS = AVSS0 = VREFL0 = VSS_RF = 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 48.1 shows the timing conditions.



V_{OH} = VCC × 0.7, V_{OL} = VCC × 0.3
V_{IH} = VCC × 0.7, V_{IL} = VCC × 0.3
Load capacitance C = 30 pF

Figure 48.1 Input or output timing measurement conditions

The measurement conditions of timing specifications in each peripheral 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.

48.1 Absolute Maximum Ratings

Table 48.1 Absolute maximum ratings

Parameter		Symbol	Value	Unit
Power supply voltage		VCC	–0.5 to +4.0	V
Input voltage	5V-tolerant ports*1	V_{in}	–0.3 to +6.5	V
	P004, P010, P011, P014, P015	V_{in}	–0.3 to AVCC0 + 0.3	V
	ANT	V_{in}	–1.0 to +1.4	V
	XTAL1_RF, XTAL2_RF	V_{in}	–0.3 to +1.4	V
	DCLIN_A, DCLIN_D	V_{in}	–0.3 to +2.2	V
	Others	V_{in}	–0.3 to VCC + 0.3	V
Reference power supply voltage		VREFH0	–0.3 to +4.0	V
VBATT power supply voltage		VBATT	–0.5 to +4.0	V
Analog power supply voltage		AVCC0	–0.5 to +4.0	V
		VCC_RF	–0.3 to +4.0	V
		AVCC_RF	–0.3 to +4.0	V
USB power supply voltage		VCC_USB	–0.5 to +4.0	V
		VCC_USB_LDO	–0.5 to +4.0	V
Analog input voltage	When AN004 to AN006, AN009, AN010 are used	V_{AN}	–0.3 to AVCC0 + 0.3	V
	When AN017, AN019, AN020 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 +4.0	V
	VL4 voltage	V_{L4}	–0.3 to +4.0	V
Operating temperature*2		T_{opr}	–40 to +85	°C
Storage temperature		T_{stg}	–55 to +125	°C

Note 1. Ports P205, P206, P402, P407 are 5V-tolerant.
Note 2. See section 48.2.1, Tj/Ta Definition.

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 VCC_RF and VSS_RF pins, between the AVDD_RF and VSS_RF pins, between the VCC_USB and VSS_USB pins, between the VREFH0 and VREFL0 pins. Place capacitors with values of about 2.2 µF in the case of the VCC_RF pin and about 0.1 µF otherwise as close as possible to every power supply pin, and use the shortest and thickest possible traces for the connections. 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 48.2 Recommended operating conditions

Parameter	Symbol	Value	Min	Typ	Max	Unit
Power supply voltages	VCC[*1, *2]	When USBFS is not used	1.8	-	3.6	V
		When USBFS is used USB Regulator Disable	VCC_USB	-	3.6	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	-	VCC	-	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	1.8	-	3.6	V
Analog power supply voltages	AVCC0[*1, *2]		1.8	-	3.6	V
	AVSS0		-	0	-	V
	VREFH0	When used as ADC14 Reference	1.8	-	AVCC0	V
	VREFL0		-	0	-	V
BLE power supply voltages	VCC_RF[*3]		1.8	-	3.6	V
	AVCC_RF[*3]		1.8	-	3.6	
	VSS_RF		-	0	-	

Note: Bluetooth power supply voltage
VCC_RF [*3] 1.8 - 3.6 V
Note: AVCC_RF [*3] 1.8 - 3.6 V
Note: VCC_RF - 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.
Note 3. Use VCC = VCC_RF = AVCC_RF

48.2 DC Characteristics

48.2.1 Tj/Ta Definition

Table 48.3 DC characteristics

Conditions: Products with operating temperature (T_a) –40 to +85°C

Parameter	Symbol	Typ	Max	Unit	Test conditions
Permissible junction temperature	Tj	-	105*[1]	°C	High-speed mode Middle-speed mode Low-voltage mode Low-speed mode Subosc-speed mode

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. For details, see section 1.3, Part Numbering. If the part number shows the operation temperature at 85°C, then the maximum value of Tj is 105°C.

48.2.2 I/O V_{IH}, V_{IL}

Table 48.4 I/O V_{IH}, V_{IL} (1)

Conditions: VCC = AVCC0 = VCC_USB = VCC_USB_LDO = 2.7 to 3.6V, VBATT = 1.8 to 3.6 V, VSS = AVSS0 = 0 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Schmitt trigger input voltage	IIC*[1]	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)	5V-tolerant ports*[2]	V_{IH}	VCC × 0.8	-	5.8		
		V_{IL}	-	-	VCC × 0.2		
	P914, P915	V_{IH}	VCC_USB × 0.8	-	VCC_USB + 0.3		
		V_{IL}	-	-	VCC_USB × 0.2		
	P004, P010	V_{IH}	AVCC0 × 0.8	-	-		
		V_{IL}	-	-	AVCC0 × 0.2		
	EXTAL Input ports pins except for P004, P010, P914, P915	V_{IH}	VCC × 0.8	-	-		
		V_{IL}	-	-	VCC × 0.2		
When V_{BATT} power supply is selected	P402	V_{IH}	V_{BATT} × 0.8	-	V_{BATT} + 0.3		
		V_{IL}	-	-	V_{BATT} × 0.2		
		ΔV_T	V_{BATT} × 0.05	-	-		

Note 1. P205, P206, P407 (total 3 pins).

Note 2. P205, P206, P402, P407 (total 4 pins).

Table 48.5 I/O V_{IH}, V_{IL} (2)

Conditions: VCC = AVCC0 = VCC_USB = VCC_USB_LDO = 1.8 to 2.7 V, VBATT = 1.8 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		
	P914, P915	V_{IH}	VCC_USB × 0.8	-	VCC_USB + 0.3		
		V_{IL}	-	-	VCC_USB × 0.2		
	P004, P010	V_{IH}	AVCC0 × 0.8	-	-		
		V_{IL}	-	-	AVCC0 × 0.2		
	EXTAL Input ports pins except for P004, P010	V_{IH}	VCC × 0.8	-	-		
		V_{IL}	-	-	VCC × 0.2		
When V_{BATT} power supply is selected	P402, 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, P402, P407 (total 4 pins).

48.2.3 I/O I_{OH}, I_{OL}

Table 48.6 I/O I_{OH}, I_{OL}

Conditions: VCC = AVCC0 = VCC_USB = VCC_USB_LDO = 1.8 to 3.6 V

Parameter			Symbol	Min	Typ	Max	Unit
Permissible output current (average value per pin)	Ports P212, P213	-	I_{OH}	-	-	–4.0	mA
			I_{OL}	-	-	4.0	mA
	Port 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 3.6 V	I_{OH}	-	-	–20.0	mA
			I_{OL}	-	-	20.0	mA
	Ports P100 to P111, P201, P204, P300, P501 (total 16 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
	Ports P914, P915	-	I_{OH}	-	-	–4.0	mA
			I_{OL}	-	-	4.0	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 P212, P213	-	I_{OH}	-	-	–4.0	mA
			I_{OL}	-	-	4.0	mA
	Port 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 3.6 V	I_{OH}	-	-	–20.0	mA
			I_{OL}	-	-	20.0	mA
	Ports P100 to P111, P201, P204, P300, P501 (total 16 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
	Ports P914, P915	-	I_{OH}	-	-	–4.0	mA
			I_{OL}	-	-	4.0	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 P004, P010		$\Sigma I_{OH\ (max)}$	-	-	–30	mA
			$\Sigma I_{OL\ (max)}$	-	-	30	mA
	Ports P914, P915		$\Sigma I_{OH\ (max)}$	-	-	–4.0	mA
			$\Sigma I_{OL\ (min)}$	-	-	4.0	mA
	Total of all output pin*5		$\Sigma I_{OH\ (max)}$	-	-	–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.
Note 4. This is the value when middle driving ability for IIC Fast-mode is selected with the Port Drive Capability bit in PmnPFS register.
Note 5. For details on the permissible output current used with CTSU, see section 48.11, CTSU Characteristics.

48.2.4 I/O V_{OH}, V_{OL}, and Other Characteristics

Table 48.7 I/O V_{OH}, V_{OL} (1)

Conditions: VCC = AVCC0 = VCC_USB = VCC_USB_LDO = 2.7 to 3.6 V

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage	IIC*1		V_{OL}	-	-	0.4	V	I_{OL} = 3.0 mA
			V_{OL}*2,*5	-	-	0.6		I_{OL} = 6.0 mA
	Ports 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 P004, P010	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
	Ports P914, P915		V_{OH}	VCC_USB – 0.5	-	-		I_{OH} = –1.0 mA
			V_{OL}	-	-	0.5		I_{OL} = 1.0 mA
	Other output pins *4, *6	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. P100, P101, P204, P205, P206, P407 (total 6 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.
Note 6. This excludes the CLKOUT_RF pin.

Table 48.8 I/O V_{OH}, V_{OL} (2)

Conditions: VCC = AVCC0 = VCC_USB = VCC_USB_LDO = 1.8 to 2.7 V

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage	Ports P004, P010	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
	Ports P914, P915		V_{OH}	VCC_USB – 0.3	-	-		I_{OH} = –0.5 mA
			V_{OL}	-	-	0.3		I_{OL} = 0.5 mA
	Other output pins *1, *3	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.
Note 3. This excludes the CLKOUT_RF pin.

Table 48.9 I/O V_{OH}, V_{OL} (3)

Conditions: 3.0V ≤ VCC = AVCC0 = VCC_USB = VCC_USB_LDO = VCC_RF = AVCC_RF ≤ 3.6 V

Parameter		Symbol	Min	Max	Unit	Test conditions
Output low	CLKOUT_RF	V_{OL}	-	0.3	V	I_{OL} = 0.5 mA
Output high	CLKOUT_RF	V_{OH}	VCC_RF – 0.3	-	V	I_{OH} = –0.5 mA

Table 48.10 I/O other characteristics

Conditions: VCC = AVCC0 = 1.8 to 3.6 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, P914, P915)	R_U	10	20	50	kΩ	V_{in} = 0 V
Input capacitance	P914, P915, P100 to P103, P111, P200	C_{in}	-	-	30	pF	V_{in} = 0 V f = 1 MHz T_a = 25°C
	Other input pins		-	-	15		

48.2.5 I/O Pin Output Characteristics of Low Drive Capacity

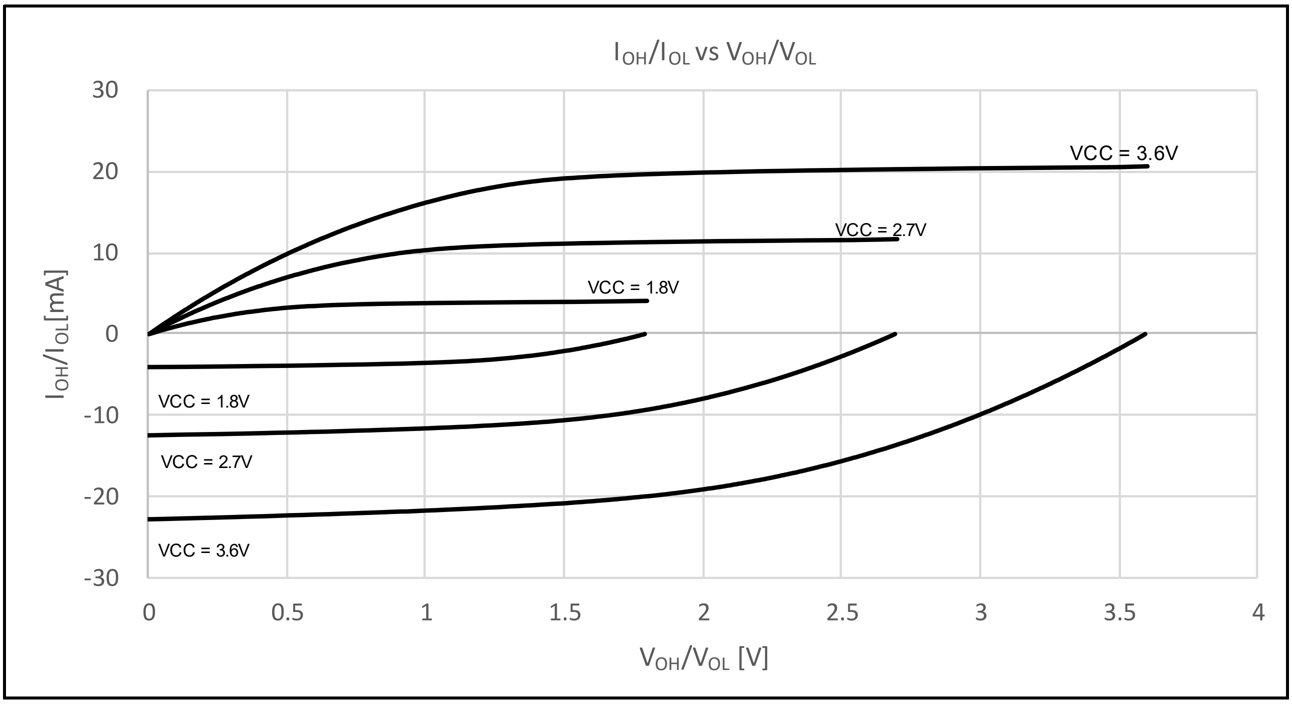


Figure 48.2 V_{OH}/V_{OL} and I_{OH}/I_{OL} voltage characteristics at Ta = 25°C when low drive output is selected (reference data)

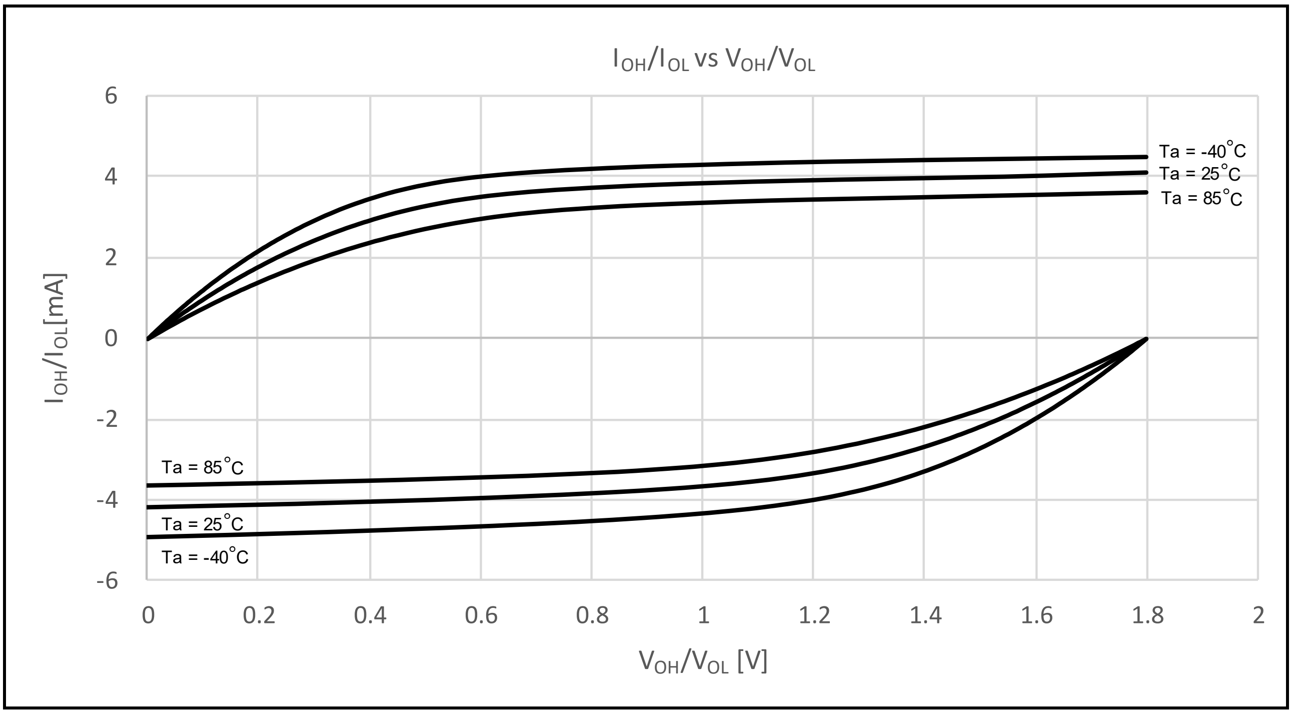


Figure 48.3 V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 1.8 V when low drive output is selected (reference data)

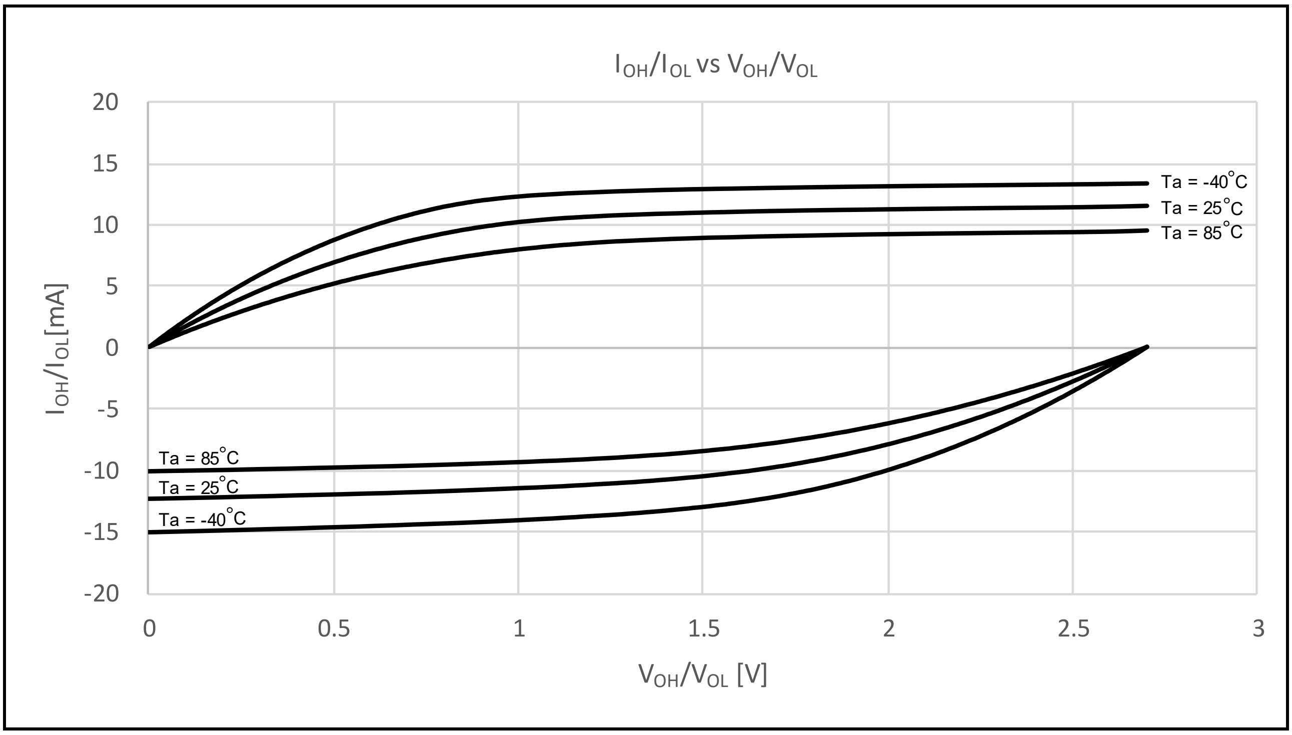


Figure 48.4 V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 2.7 V when low drive output is selected (reference data)

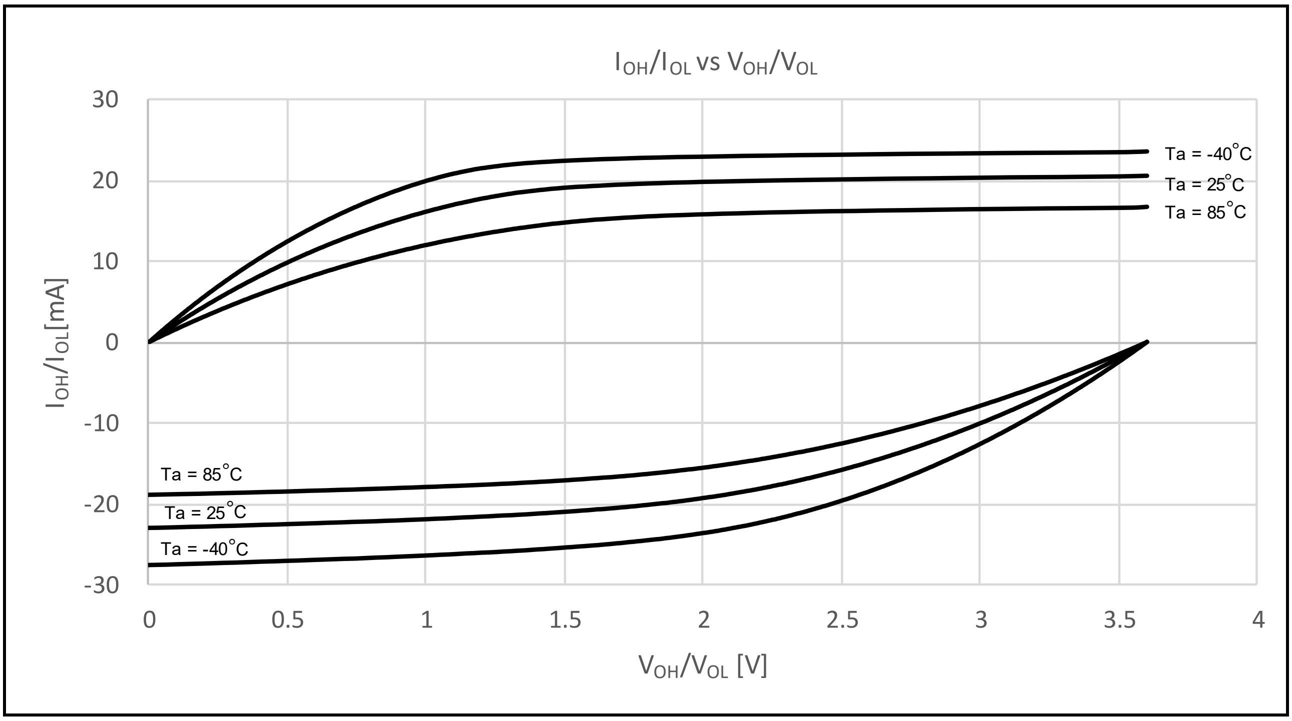


Figure 48.5 V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 3.6 V when low drive output is selected (reference data)

48.2.6 I/O Pin Output Characteristics of Middle Drive Capacity

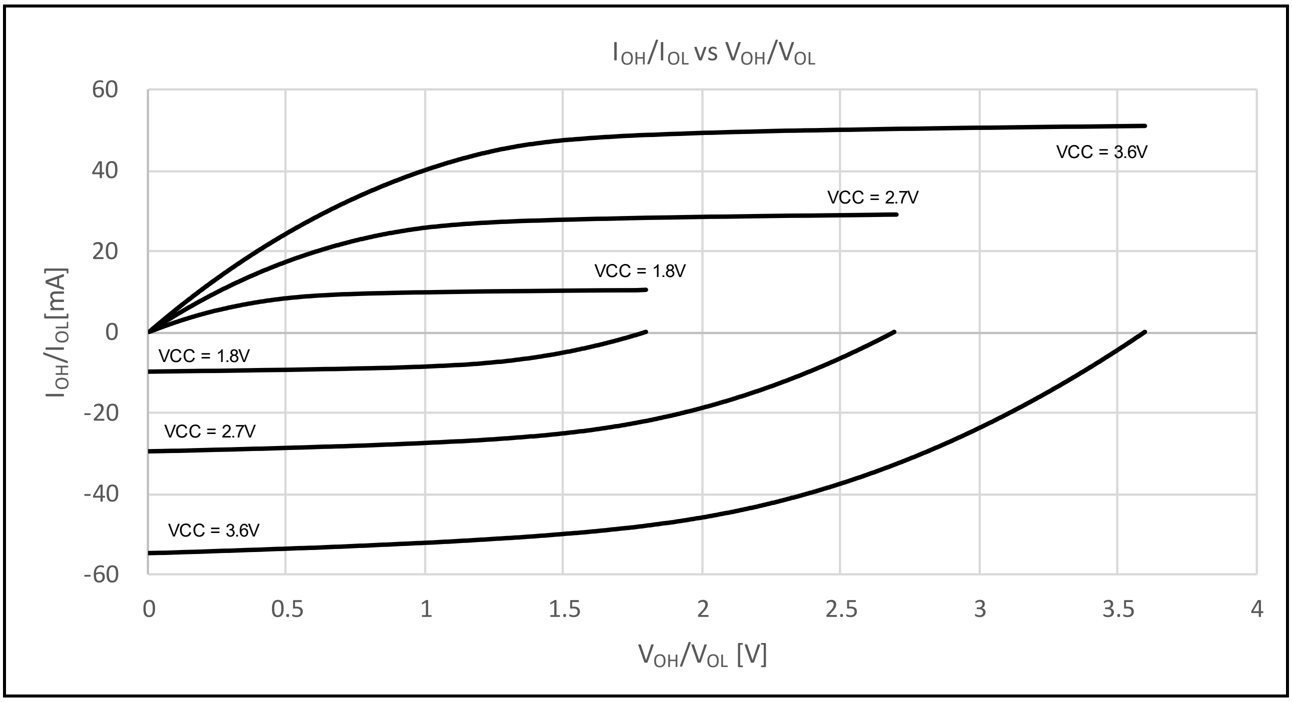


Figure 48.6 V_{OH}/V_{OL} and I_{OH}/I_{OL} voltage characteristics at Ta = 25°C when middle drive output is selected (reference data)

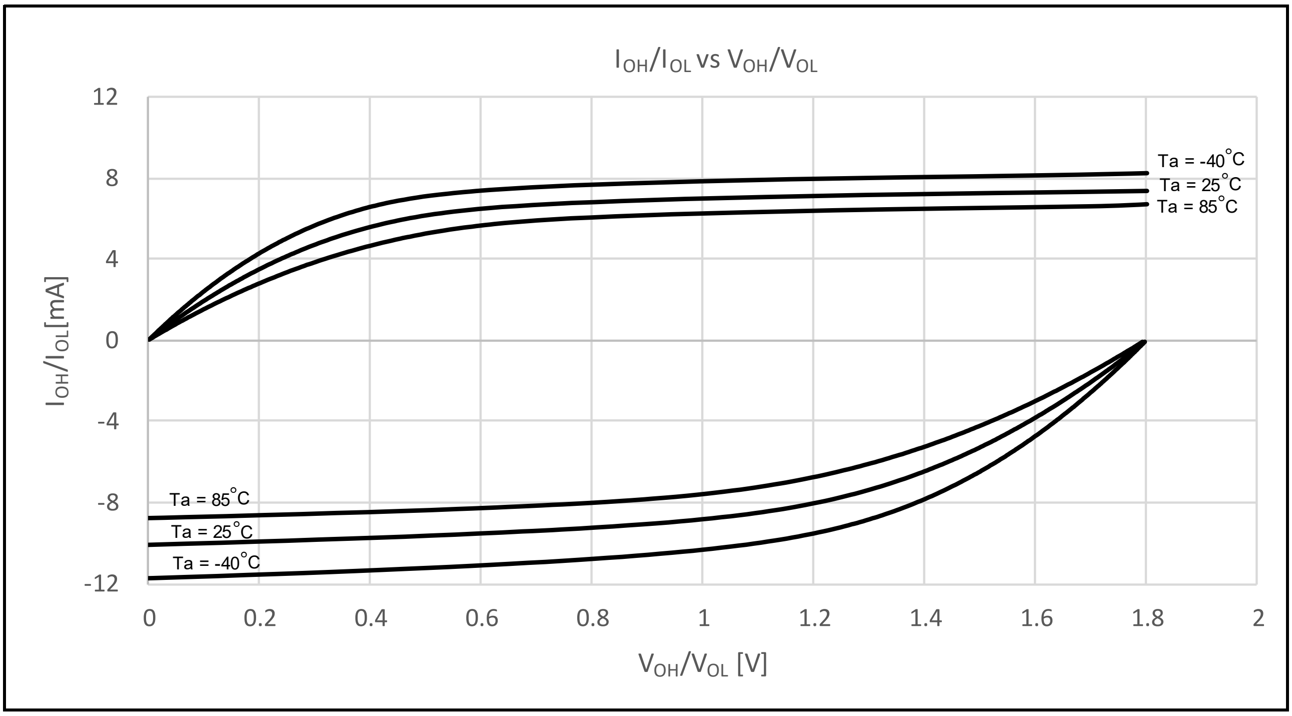


Figure 48.7 V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 1.8 V when middle drive output is selected (reference data)

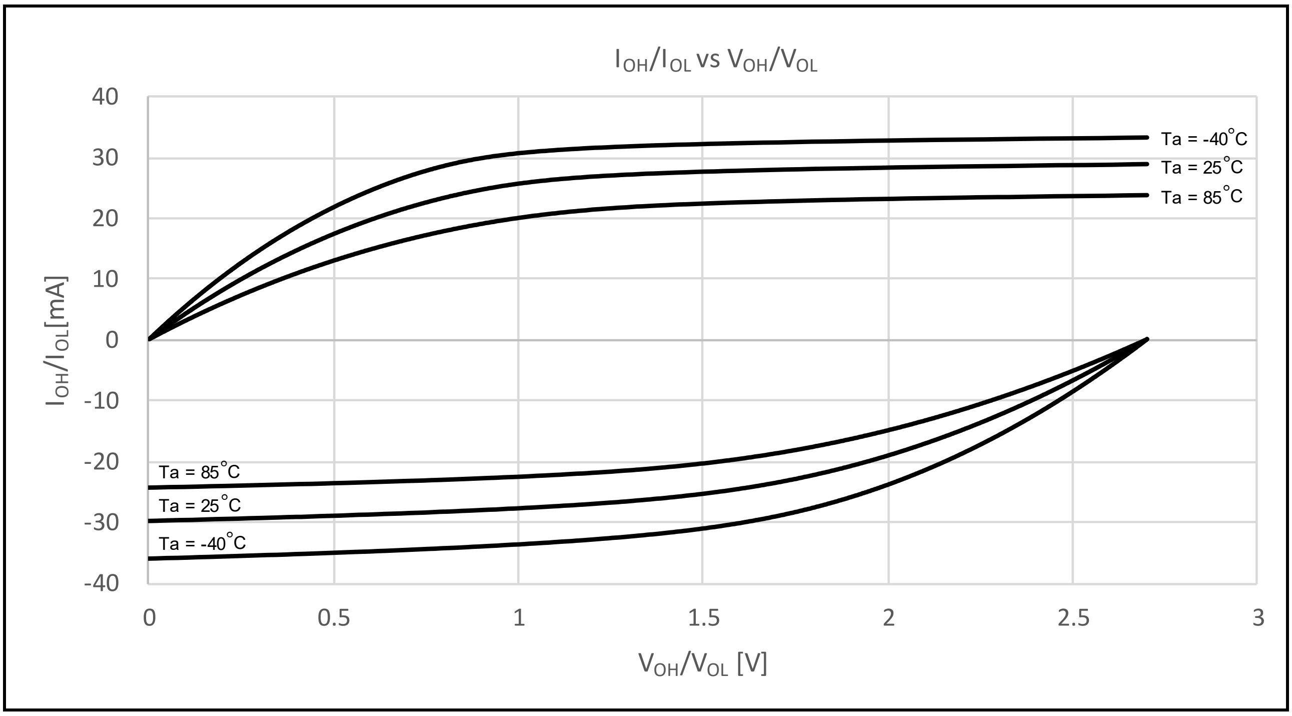


Figure 48.8 V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 2.7 V when middle drive output is selected (reference data)

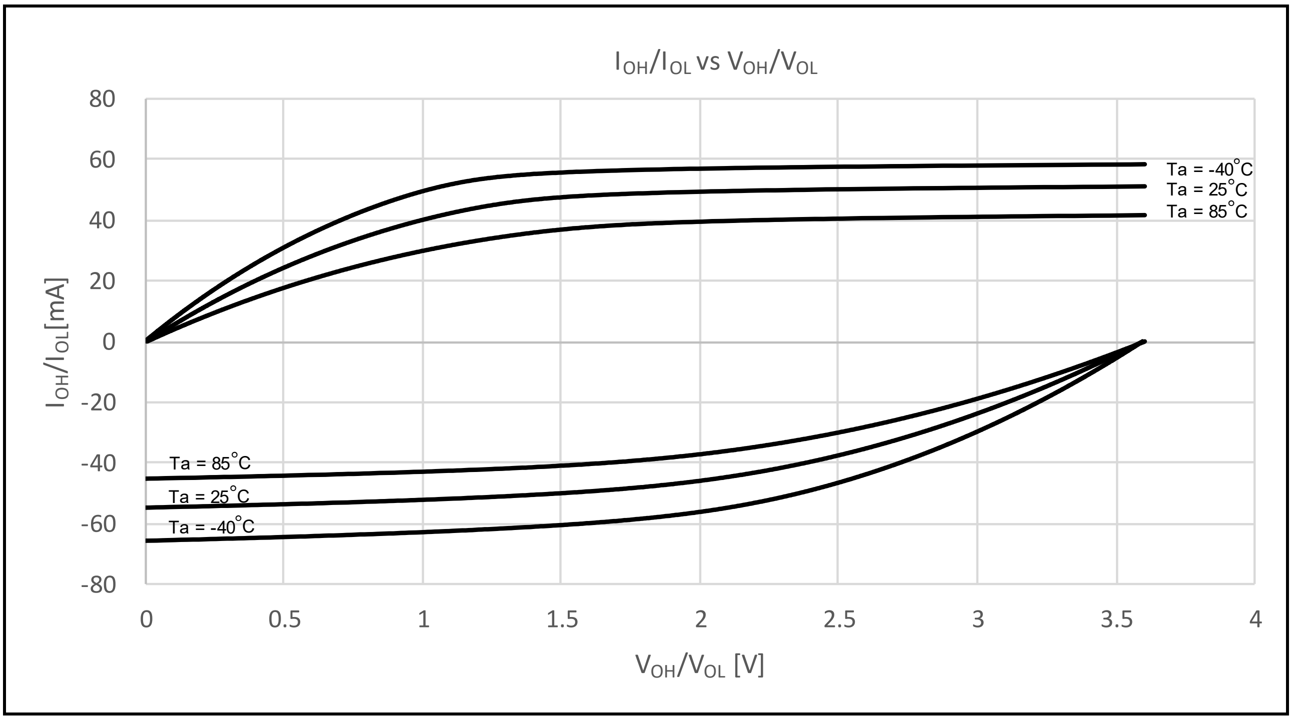


Figure 48.9 V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 3.6 V when middle drive output is selected (reference data)

48.2.7 P409 I/O Pin Output Characteristics of Middle Drive Capacity

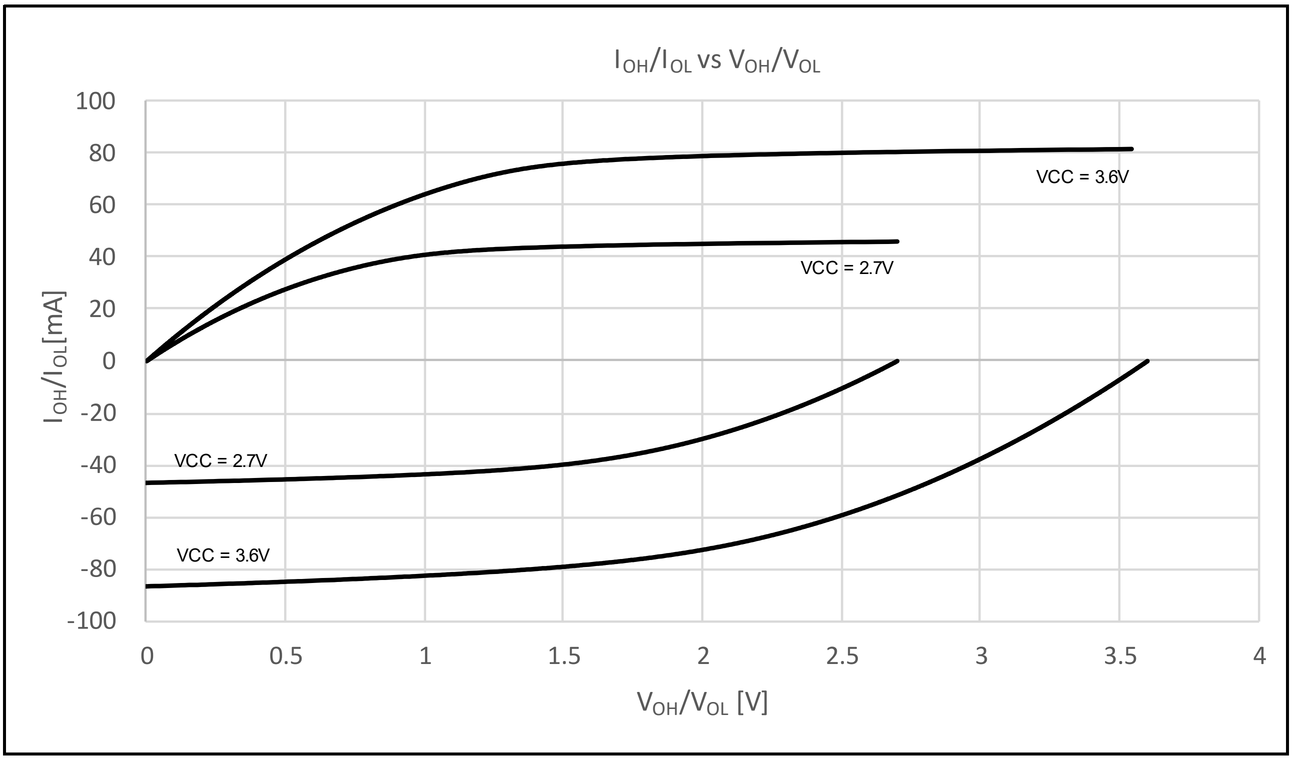


Figure 48.10 V_{OH}/V_{OL} and I_{OH}/I_{OL} voltage characteristics at Ta = 25°C when middle drive output is selected (reference data)

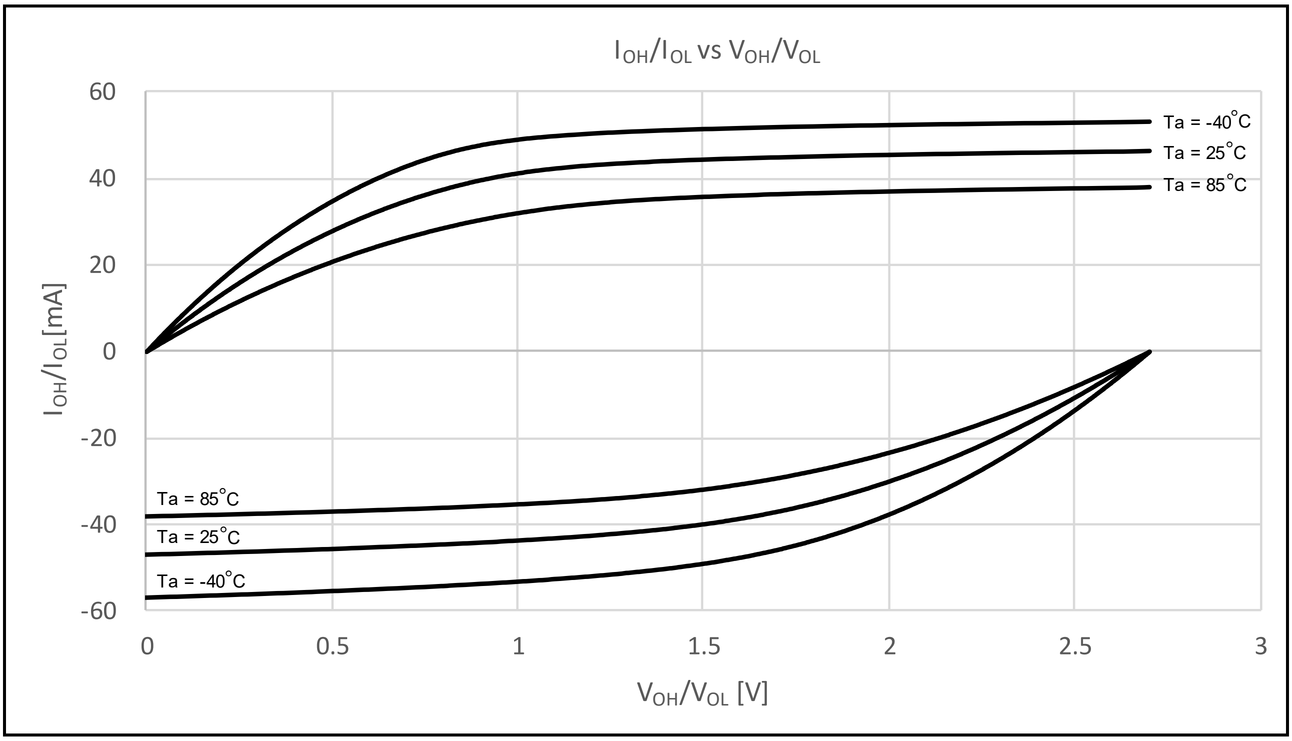


Figure 48.11 V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 2.7 V when middle drive output is selected (reference data)

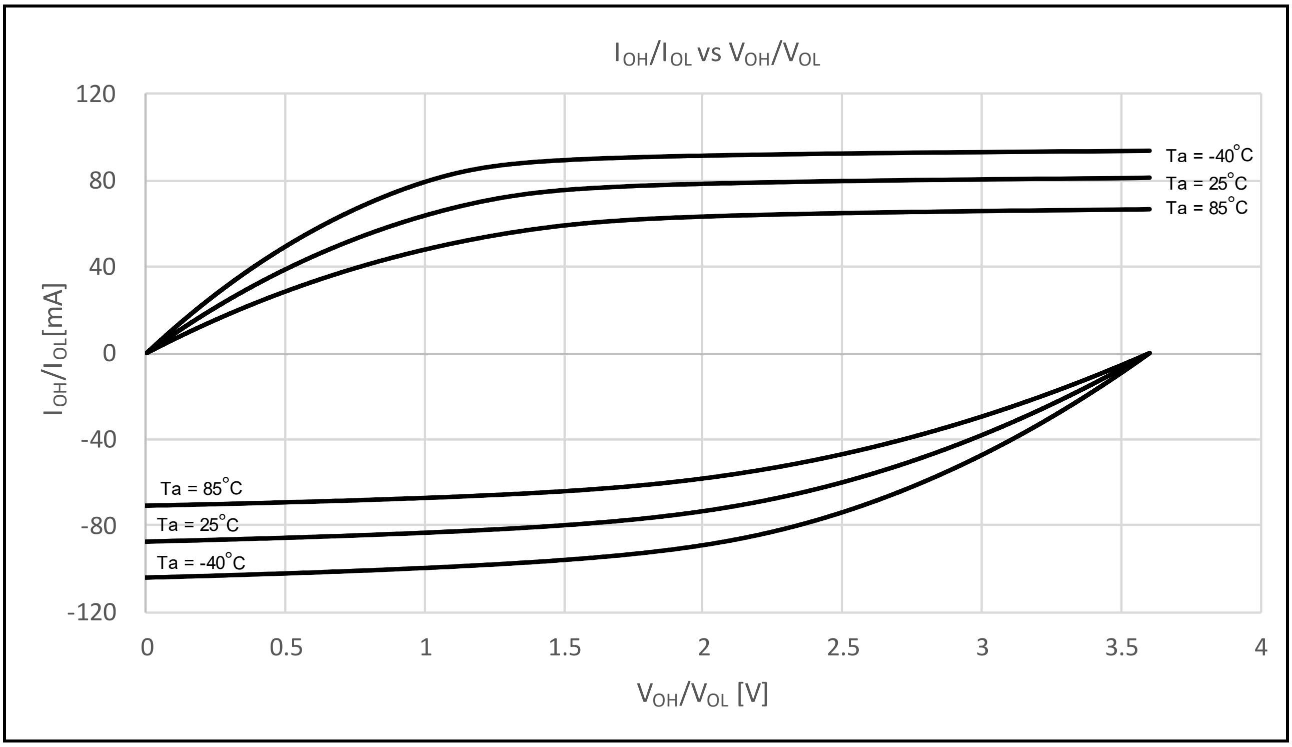


Figure 48.12 V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 3.6 V when middle drive output is selected (reference data)

48.2.8 IIC I/O Pin Output Characteristics

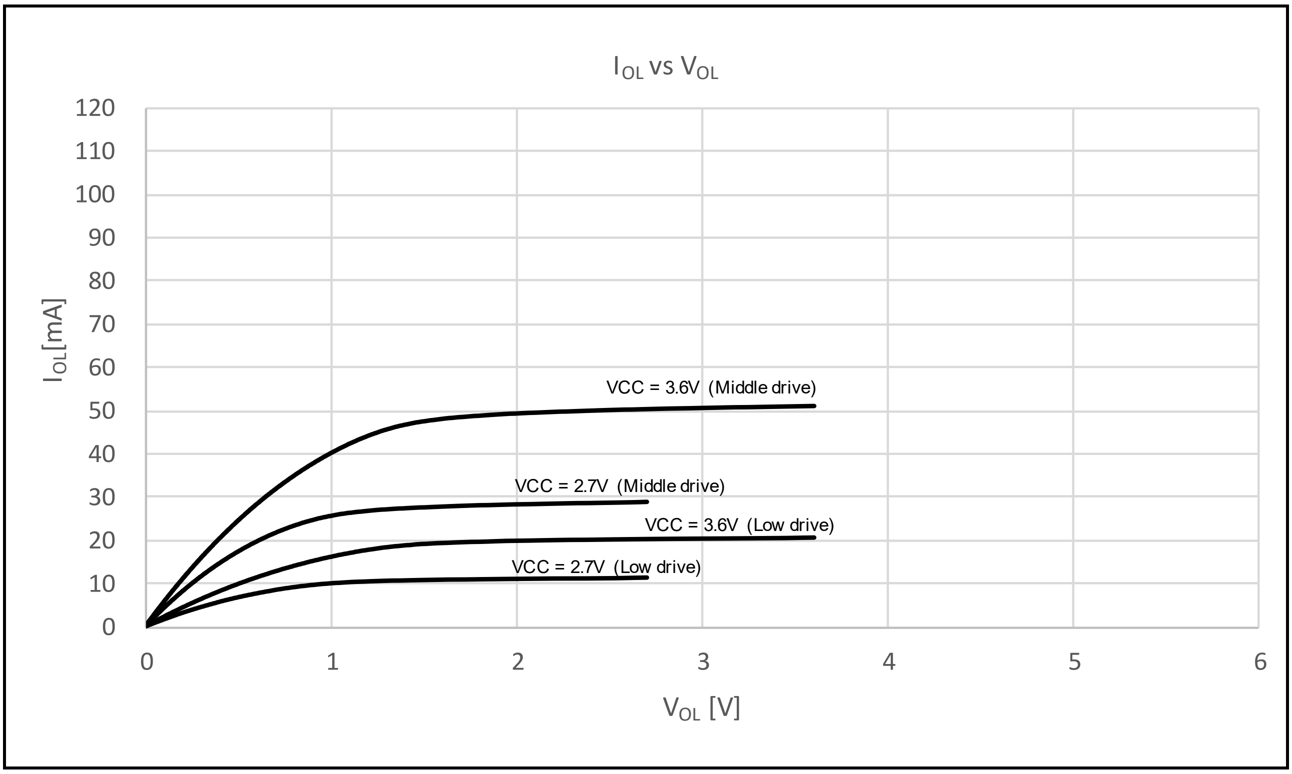


Figure 48.13 V_{OH}/V_{OL} and I_{OH}/I_{OL} voltage characteristics at Ta = 25°C

48.2.9 Operating and Standby Current

Table 48.11 Operating and standby current (1) (1 of 2)

Conditions: VCC = AVCC0 = 1.8 to 3.6 V

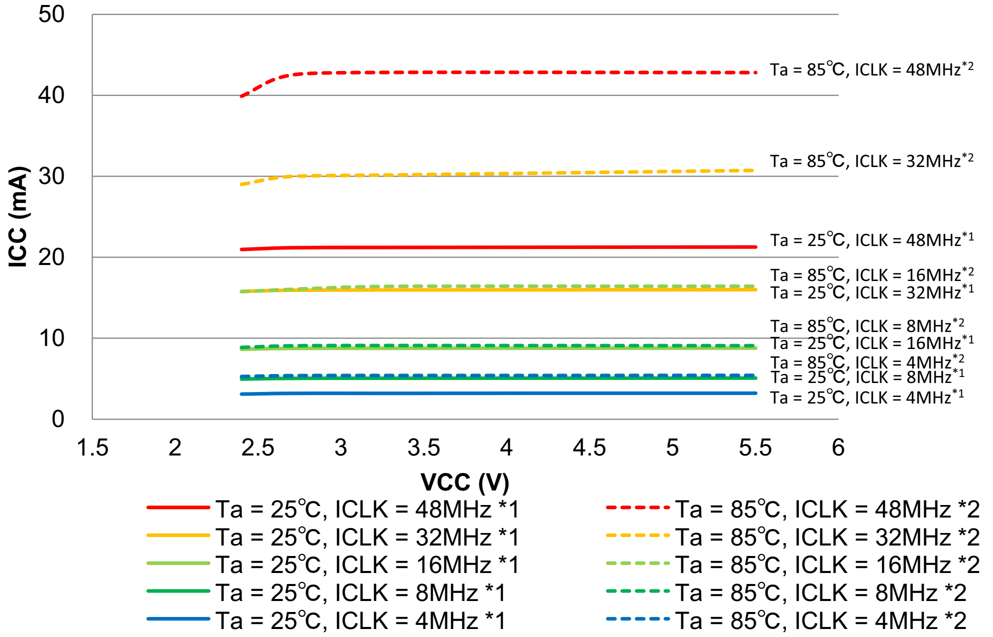
Parameter					Symbol	Typ*10	Max	Unit	Test conditions
Supply current*1	High-speed mode*2	Normal mode	All peripheral clock disabled, while (1) code executing from flash*5	ICLK = 48 MHz	I_{CC}	8.4	-	mA	*7
				ICLK = 32 MHz		5.9	-		
				ICLK = 16 MHz		3.5	-		
				ICLK = 8 MHz		2.3	-		
			All peripheral clock disabled, CoreMark code executing from flash*5	ICLK = 48 MHz		17.9	-		
				ICLK = 32 MHz		12.4	-		
				ICLK = 16 MHz		7.0	-		
				ICLK = 8 MHz		4.3	-		
			All peripheral clock enabled, while (1) code executing from flash*5	ICLK = 48 MHz		21.2	-		*9
				ICLK = 32 MHz		16.0	-		*8
				ICLK = 16 MHz		8.8	-		
				ICLK = 8 MHz		5.1	-		
			All peripheral clock enabled, code executing from SRAM*5	ICLK = 48 MHz		-	56.0		*9
		Sleep mode	All peripheral clock disabled*5	ICLK = 48 MHz		3.7	-		*7
				ICLK = 32 MHz		2.7	-		
				ICLK = 16 MHz		2.0	-		
				ICLK = 8 MHz		1.5	-		
			All peripheral clock enabled*5	ICLK = 48 MHz		16.4	-		*9
				ICLK = 32 MHz		12.7	-		*8
				ICLK = 16 MHz		7.2	-		
				ICLK = 8 MHz		4.3	-		
		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}	2.5	-	mA	*7
				ICLK = 8 MHz		2.1	-		
				ICLK = 1 MHz		1.0	-		
			All peripheral clock disabled, CoreMark code executing from flash*5	ICLK = 12 MHz		5.2	-		
				ICLK = 8 MHz		4.0	-		
				ICLK = 1 MHz		1.3	-		
			All peripheral clock enabled, while (1) code executing from flash*5	ICLK = 12 MHz		6.5	-		*8
				ICLK = 8 MHz		4.8	-		
				ICLK = 1 MHz		1.6	-		
			All peripheral clock enabled, code executing from SRAM*5	ICLK = 12 MHz		-	23.0		
		Sleep mode	All peripheral clock disabled*5	ICLK = 12 MHz		1.4	-		*7
				ICLK = 8 MHz		1.3	-		
				ICLK = 1 MHz		0.9	-		
			All peripheral clock enabled*5	ICLK = 12 MHz		5.3	-		*8
				ICLK = 8 MHz		4.0	-		
				ICLK = 1 MHz		1.5	-		
		Increase during BGO operation*6				2.5	-		-

Table 48.11 Operating and standby current (1) (2 of 2)

Conditions: VCC = AVCC0 = 1.8 to 3.6 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}	0.4	-	mA	*7
			All peripheral clock disabled, CoreMark code executing from flash*5	ICLK = 1 MHz		0.6	-		
			All peripheral clock enabled, while (1) code executing from flash*5	ICLK = 1 MHz		1.1	-		*8
			All peripheral clock enabled, code executing from SRAM*5	ICLK = 1 MHz		-	2.5		
		Sleep mode	All peripheral clock disabled*5	ICLK = 1 MHz		0.3	-		*7
			All peripheral clock enabled*5	ICLK = 1 MHz		1.0	-		*8
	Low-voltage mode*3	Normal mode	All peripheral clock disabled, while (1) code executing from flash*5	ICLK = 4 MHz	I_{CC}	1.8	-	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		3.3	-		*8
			All peripheral clock enabled, code executing from SRAM*5	ICLK = 4 MHz		-	9.0		
		Sleep mode	All peripheral clock disabled*5	ICLK = 4 MHz		1.4	-		*7
			All peripheral clock enabled*5	ICLK = 4 MHz		2.9	-		*8
	Subosc-speed mode*4	Normal mode	All peripheral clock disabled, while (1) code executing from flash*5	ICLK = 32.768 kHz	I_{CC}	9.3	-	μA	*8
			All peripheral clock enabled, while (1) code executing from flash*5	ICLK = 32.768 kHz		17.2	-		
			All peripheral clock enabled, code executing from SRAM*5	ICLK = 32.768 kHz		-	106.0		
		Sleep mode	All peripheral clock disabled*5	ICLK = 32.768 kHz		6.0	-		
			All peripheral clock enabled*5	ICLK = 32.768 kHz		14.0	-		

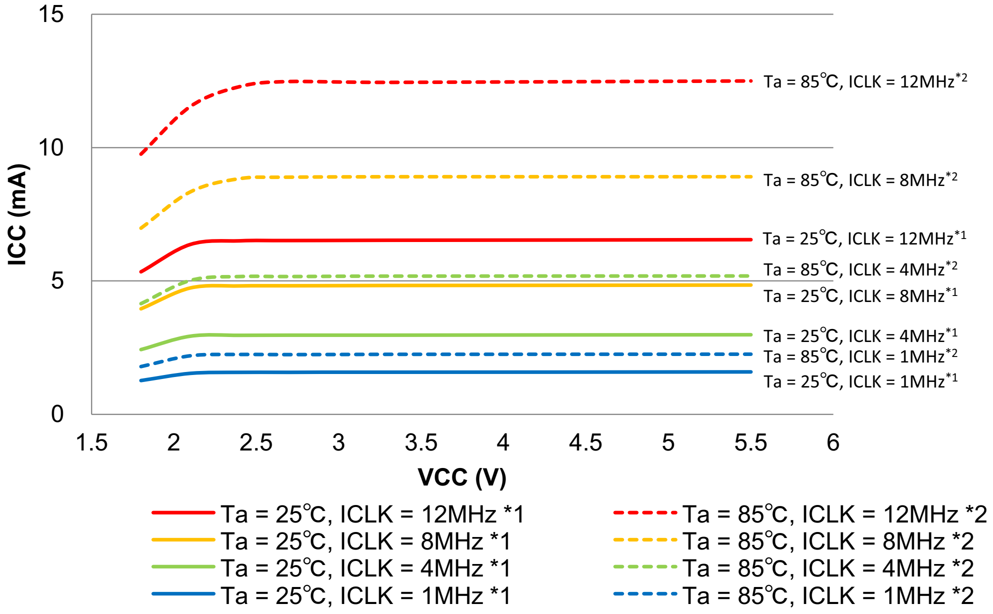
Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.
Note 2. The clock source is HOCO.
Note 3. The clock source is MOCO.
Note 4. The clock source is the sub-clock oscillator.
Note 5. This does not 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, PCLKA, PCLKB, PCLKC and PCLKD are set to divided by 64.
Note 8. FCLK, PCLKA, PCLKB, PCLKC and PCLKD are the same frequency as that of ICLK.
Note 9. FCLK 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.

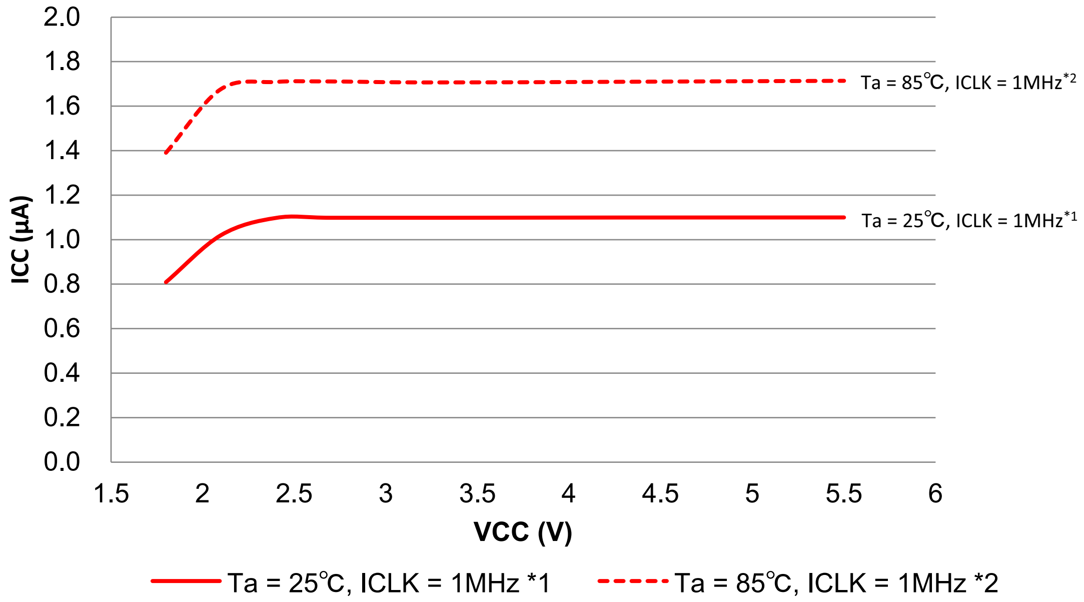
Figure 48.14 Voltage dependency in high-speed mode (reference data)



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.

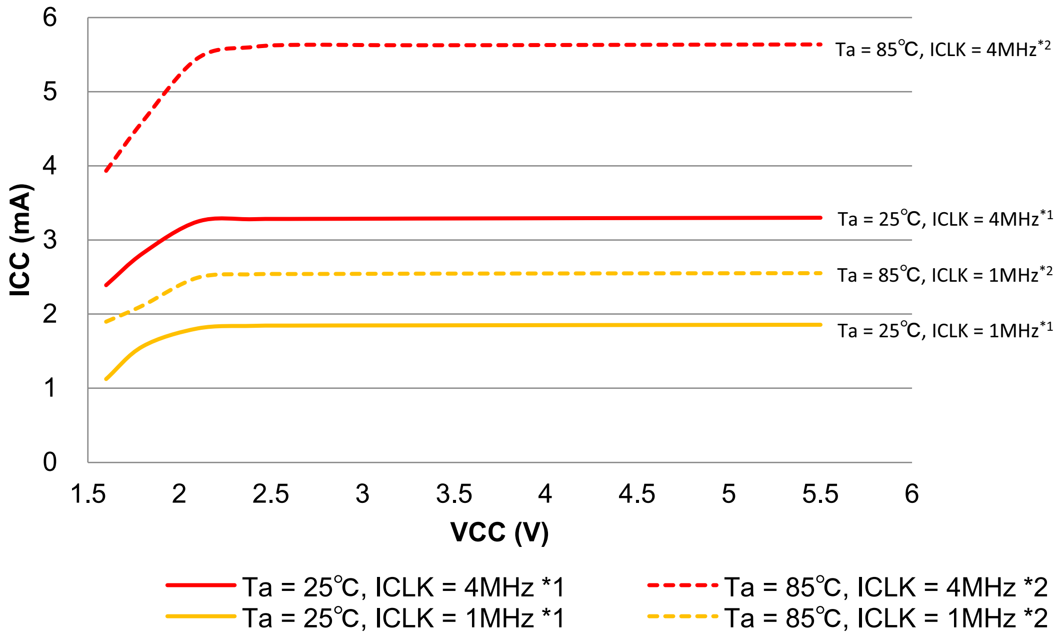
Figure 48.15 Voltage dependency in middle-speed mode (reference data)



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.

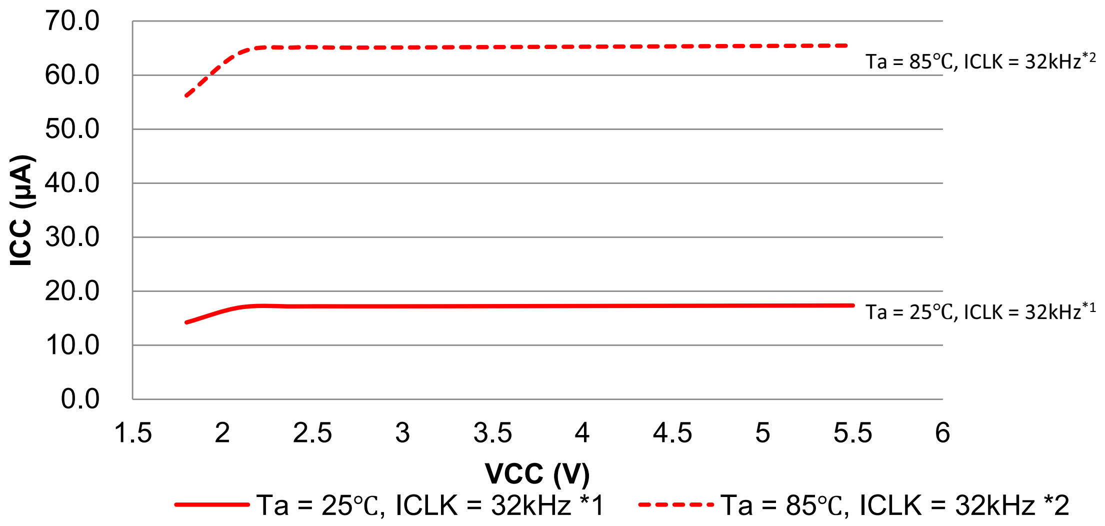
Figure 48.16 Voltage dependency in low-speed mode (reference data)



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 48.17 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 48.18 Voltage dependency in subosc-speed mode (reference data)

Table 48.12 Operating and standby current (2)

Conditions: VCC = AVCC0 = 1.8 to 3.6 V

Parameter			Symbol	Typ*4	Max	Unit	Test conditions
Supply current*1	Software Standby mode*2	T_a = 25°C	I_{CC}	0.9	5.0	μA	PSMCR.PSMC[1:0] = 01b (48-KB SRAM on)
		T_a = 55°C		1.5	8.1		
		T_a = 85°C		3.6	22.1		
		T_a = 25°C		1.0	5.6		PSMCR.PSMC[1:0] = 00b (All SRAM on)
		T_a = 55°C		1.6	8.4		
		T_a = 85°C		4.3	26.7		
	Increment for RTC operation with low-speed on-chip oscillator*3			0.5	-		-
	Increment for RTC operation with sub-clock oscillator*3			0.4	-		SOMCR.SODRV[1:0] are 11b (Low power mode 3)
				1.2	-		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.

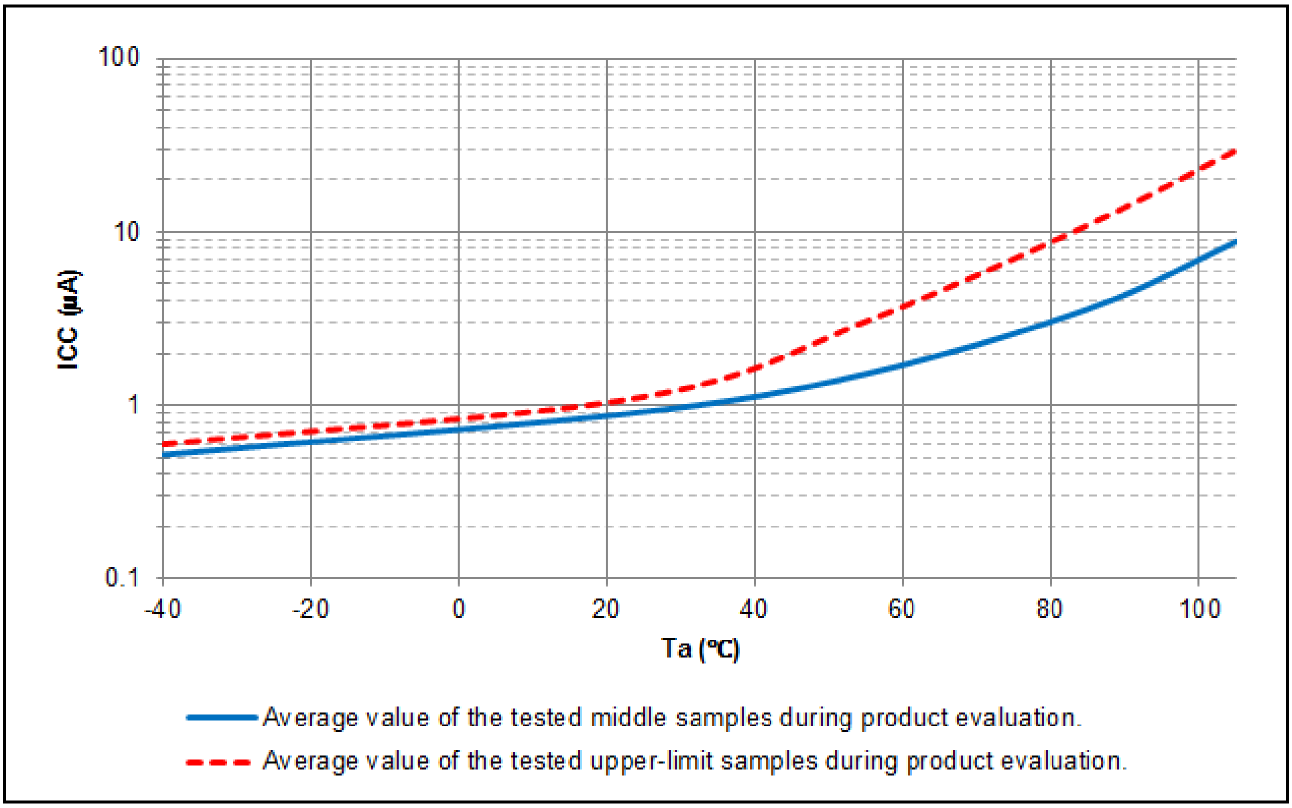


Figure 48.19 Temperature dependency in Software Standby mode 48-KB SRAM on (reference data)

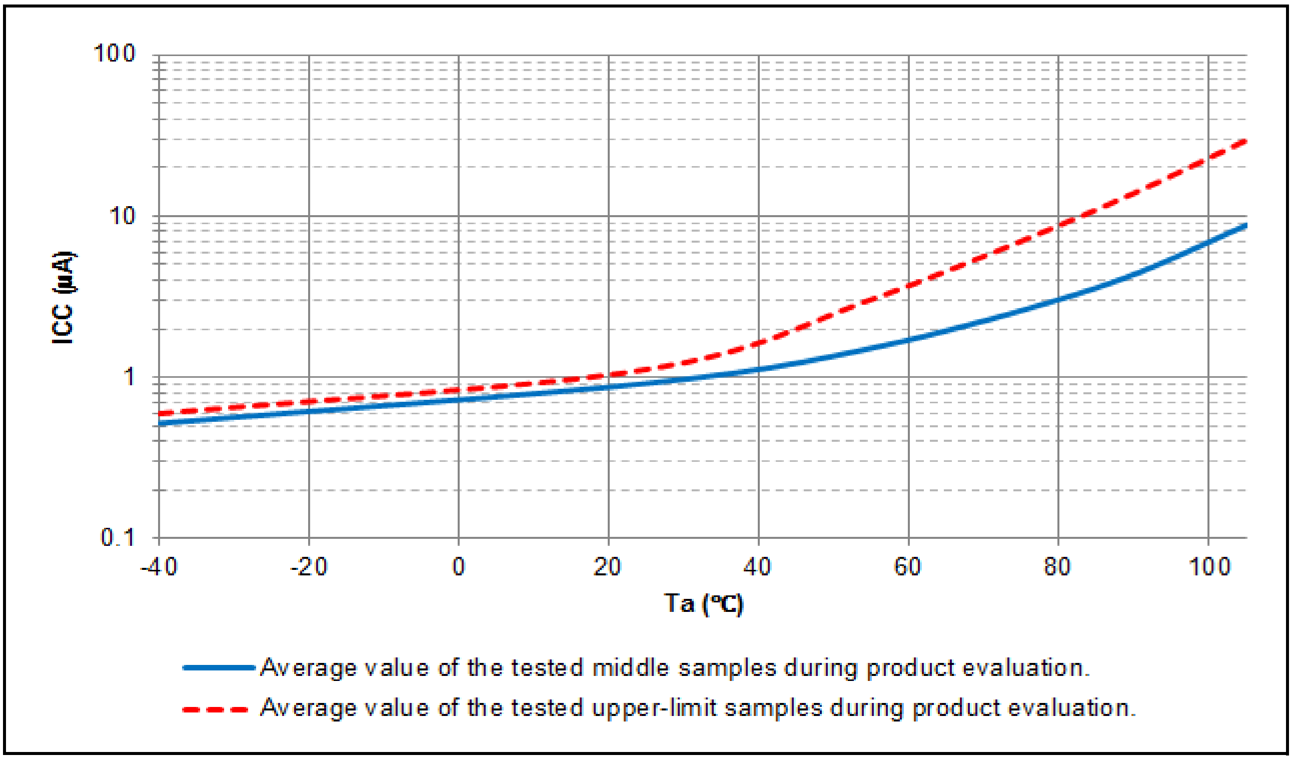


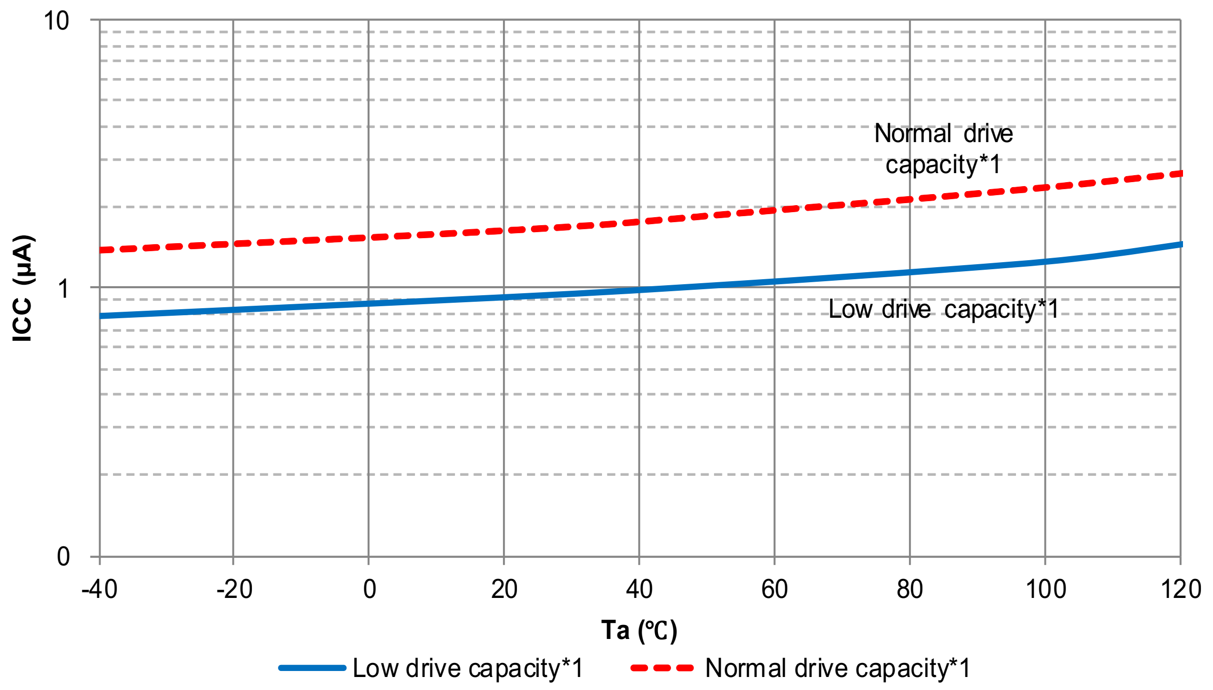
Figure 48.20 Temperature dependency in Software Standby mode all SRAM on (reference data)

Table 48.13 Operating and standby current (3)

Conditions: VCC = AVCC0 = 0V, VBATT = 1.8 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}	0.8	-	μA	VBATT = 2.0 V SOMCR.SORDRV[1:0] = 11b (Low power mode 3)
		T_a = 55°C		0.9	-		
		T_a = 85°C		1.1	-		
		T_a = 25°C		0.9	-		VBATT = 3.3 V SOMCR.SORDRV[1:0] = 11b (Low power mode 3)
		T_a = 55°C		1.0	-		
		T_a = 85°C		1.2	-		
		T_a = 25°C		1.6	-		VBATT = 2.0 V SOMCR.SORDRV[1:0] = 00b (Normal mode)
		T_a = 55°C		1.8	-		
		T_a = 85°C		2.1	-		
		T_a = 25°C		1.7	-		VBATT = 3.3 V SOMCR.SORDRV[1:0] = 00b (Normal mode)
		T_a = 55°C		1.9	-		
		T_a = 85°C		2.2	-		

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 48.21 Temperature dependency of RTC operation with VCC off (reference data)

Table 48.14 Operating and standby current (4)

Conditions: VCC = AVCC0 = 1.8 to 3.6 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	-
	Comparator Low-speed mode using DAC8			-	820	-	μA	-
Operational Amplifier operating current	Low power mode	1 unit operating	I_{AMP}	-	2.5	4.0	μA	-
	High speed mode	1 unit operating		-	140	220	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	-
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: • Device 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: • Device 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. The reference power supply current is included in the power supply current value for D/A conversion.

Note 2. Current consumed only by the USBFS.

Note 3. Includes the 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. Current flowing only to the LCD controller. Not including the current that flows through the LCD panel.

Note 6. When the MCU is in Software Standby mode or the MSTPCRD.MSTPD16 (ADC140 Module Stop bit) is in the module-stop state.

Table 48.15 Operating and standby current (5)

Conditions: VCC = VCC_RF = AVCC_RF = 3.3 V, VSS = VSS_RF = 0 V, Ta = +25°C

Parameter		Symbol	Min	Typ		Max	Unit	Test conditions
				Transmit output power				
				0 dBm	4 dBm			
BLE operating current (When DC-DC converter is selected)	Transmit mode, 2 Mbps	Idd_tx	-	4.5	8.7	-	mA	-
	Transmit mode, 1 Mbps		-			-	mA	-
	Transmit mode, 500 kbps		-			-	mA	-
	Transmit mode, 125 kbps		-			-	mA	-
	Receive mode, 2 Mbps Prf = -67 dBm	Idd_rx	-	3.3	3.5	-	mA	-
	Receive mode, 1 Mbps Prf = -67 dBm		-			-	mA	-
	Receive mode, 500 kbps Prf = -72 dBm		-			-	mA	-
	Receive mode, 125 kbps Prf = -79 dBm		-			-	mA	-
	Idle mode	Idd_idle	-	0.5		-	mA	-
	Deep sleep mode	Idd_slp	-	1.5		-	μA	-
	Power down mode	Idd_down	-	0.1		-	μA	-
BLE operating current (When linear regulator is selected)	Transmit mode, 2 Mbps	Idd_tx	-	10.2	18.1	-	mA	-
	Transmit mode, 1 Mbps		-			-	mA	-
	Transmit mode, 500 kbps		-			-	mA	-
	Transmit mode, 125 kbps		-			-	mA	-
	Receive mode, 2M bps Prf = -67 dBm	Idd_rx	-	6.9		-	mA	-
	Receive mode, 1 Mbps Prf = -67 dBm		-	6.9		-	mA	-
	Receive mode, 500 kbps Prf = -72 dBm		-	6.9		-	mA	-
	Receive mode, 125 kbps Prf = -79 dBm		-	7.1		-	mA	-
	Idd_idle	Idd_idle	-	0.7		-	mA	-
	Idd_slp	Idd_slp	-	1.5		-	μA	-
	Idd_down	Idd_down	-	0.1		-	μA	-

48.2.10 VCC Rise and Fall Gradient and Ripple Frequency

Table 48.16 Rise and fall gradient characteristics

Conditions: VCC = AVCC0 = 0 to 3.6 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Power-on VCC rising gradient	Voltage monitor 0 reset disabled at startup (normal 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 48.17 Rising and falling gradient and ripple frequency characteristics

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V

The ripple voltage must meet the allowable ripple frequency $f_{r(VCC)}$ within the range between the VCC upper limit (3.6 V) and lower limit (1.8 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 48.22 $V_{r\ (VCC)} \leq VCC \times 0.2$
		-	-	1	MHz	Figure 48.22 $V_{r\ (VCC)} \leq VCC \times 0.08$
		-	-	10	MHz	Figure 48.22 $V_{r\ (VCC)} \leq VCC \times 0.06$
Allowable voltage change rising and falling gradient	dt/dVCC	1.0	-	-	ms/V	When VCC change exceeds VCC ±10%

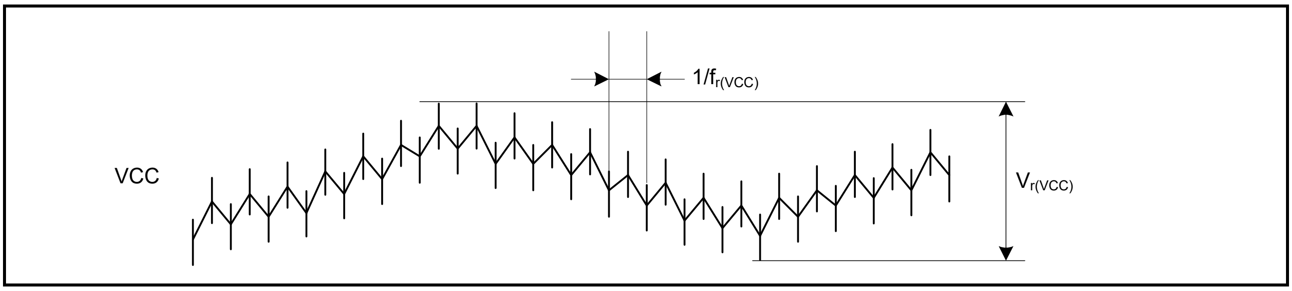


Figure 48.22 Ripple waveform

48.3 AC Characteristics

48.3.1 Frequency

Table 48.18 Operation frequency value in high-speed operating mode

Conditions: VCC = AVCC0 = 2.4 to 3.6 V

Parameter			Symbol	Min	Typ	Max*5	Unit
Operation frequency	System clock (ICLK)*4	2.7 to 3.6 V	f	0.032768	-	48	MHz
		2.4 to 2.7 V		0.032768	-	16	
	FlashIF clock (FCLK)*1, *2, *4	2.7 to 3.6 V		0.032768	-	32	
		2.4 to 2.7 V		0.032768	-	16	
	Peripheral module clock (PCLKA)*4	2.7 to 3.6 V		-	-	48	
		2.4 to 2.7 V		-	-	16	
	Peripheral module clock (PCLKB)*4	2.7 to 3.6 V		-	-	32	
		2.4 to 2.7 V		-	-	16	
	Peripheral module clock (PCLKC)*3, *4	2.7 to 3.6 V		-	-	64	
		2.4 to 2.7 V		-	-	16	
	Peripheral module clock (PCLKD)*4	2.7 to 3.6 V		-	-	64	
		2.4 to 2.7 V		-	-	16	

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, and FCLK.

Note 5. The maximum value of operation frequency does not include the internal oscillator errors. The operation can be guaranteed with the errors of the internal oscillator. For details on the range for guaranteed operation, see Table 48.23, Clock timing.

Table 48.19 Operation frequency value in Middle-speed mode

Conditions: VCC = AVCC0 = 1.8 to 3.6 V

Parameter			Symbol	Min	Typ	Max*5	Unit
Operation frequency	System clock (ICLK)*4	2.7 to 3.6 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 3.6 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 3.6 V		-	-	12	
		2.4 to 2.7 V		-	-	12	
		1.8 to 2.4 V		-	-	8	
	Peripheral module clock (PCLKB)*4	2.7 to 3.6 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 3.6 V		-	-	12	
		2.4 to 2.7 V		-	-	12	
		1.8 to 2.4 V		-	-	8	
	Peripheral module clock (PCLKD)*4	2.7 to 3.6 V		-	-	12	
		2.4 to 2.7 V		-	-	12	
		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, and FCLK.
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 48.23, Clock timing.

Table 48.20 Operation frequency value in Low-speed mode

Conditions: VCC = AVCC0 = 1.8 to 3.6 V

Parameter			Symbol	Min	Typ	Max*4	Unit
Operation frequency	System clock (ICLK)*3	1.8 to 3.6 V	f	0.032768	-	1	MHz
	FlashIF clock (FCLK)*1, *3	1.8 to 3.6 V		0.032768	-	1	
	Peripheral module clock (PCLKA)*3	1.8 to 3.6 V		-	-	1	
	Peripheral module clock (PCLKB)*3	1.8 to 3.6 V		-	-	1	
	Peripheral module clock (PCLKC)*2, *3	1.8 to 3.6 V		-	-	1	
	Peripheral module clock (PCLKD)*3	1.8 to 3.6 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, and FCLK.
Note 4. The maximum value of operation frequency does not include the internal oscillator errors. The operation can be guaranteed with the errors of the internal oscillator. For details on the range for guaranteed operation, see Table 48.23, Clock timing.

Table 48.21 Operation frequency value in low-voltage mode

Conditions: VCC = AVCC0 = 1.8 to 3.6 V

Parameter			Symbol	Min	Typ	Max*5	Unit
Operation frequency	System clock (ICLK)*4	1.8 to 3.6 V	f	0.032768	-	4	MHz
	FlashIF clock (FCLK)*1, *2, *4	1.8 to 3.6 V		0.032768	-	4	
	Peripheral module clock (PCLKA)*4	1.8 to 3.6 V		-	-	4	
	Peripheral module clock (PCLKB)*4	1.8 to 3.6 V		-	-	4	
	Peripheral module clock (PCLKC)*3, *4	1.8 to 3.6 V		-	-	4	
	Peripheral module clock (PCLKD)*4	1.8 to 3.6 V		-	-	4	

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, and FCLK.
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 48.23, Clock timing.

Table 48.22 Operation frequency value in Subosc-speed mode

Conditions: VCC = AVCC0 = 1.8 to 3.6 V

Parameter			Symbol	Min	Typ	Max	Unit
Operation frequency	System clock (ICLK)*3	1.8 to 3.6 V	f	27.8528	32.768	37.6832	kHz
	FlashIF clock (FCLK)*1, *3	1.8 to 3.6 V		27.8528	32.768	37.6832	
	Peripheral module clock (PCLKA)*3	1.8 to 3.6 V		-	-	37.6832	
	Peripheral module clock (PCLKB)*3	1.8 to 3.6 V		-	-	37.6832	
	Peripheral module clock (PCLKC)*2, *3	1.8 to 3.6 V		-	-	37.6832	
	Peripheral module clock (PCLKD)*3	1.8 to 3.6 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.

48.3.2 Clock Timing

Table 48.23 Clock timing (1 of 2)

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
EXTAL external clock input cycle time	t_{Xcyc}	50	-	-	ns	Figure 48.23
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 ≤ 3.6
		-	-	8		1.8 ≤ VCC < 2.4
Main clock oscillator oscillation frequency	f_{MAIN}	1	-	20	MHz	2.4 ≤ VCC ≤ 3.6
		1	-	8		1.8 ≤ VCC < 2.4
Main clock oscillation stabilization wait time (crystal)*9	$t_{MAINOSCWT}$	-	-	-*9	ms	
LOCO clock oscillation frequency	f_{LOCO}	27.8528	32.768	37.6832	kHz	-
LOCO clock oscillation stabilization time	t_{LOCO}	-	-	100	μs	Figure 48.24
IWDT-dedicated clock oscillation frequency	f_{ILOCO}	12.75	15	17.25	kHz	-
Bluetooth-dedicated clock oscillation frequency	f_{BLECK}	-	32	-	MHz	
Bluetooth-dedicated low-speed on-chip oscillator oscillation frequency	$f_{BLELOCO}$	-	32.768	-	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 ≤ 3.6
		23.76	24	24.24		Ta = –20 to 85°C 1.8 ≤ VCC ≤ 3.6
	f_{HOCO32}	31.52	32	32.48		Ta = –40 to -20°C 1.8 ≤ VCC ≤ 3.6
		31.68	32	32.32		Ta = –20 to 85°C 1.8 ≤ VCC ≤ 3.6
	f_{HOCO48}*4	47.28	48	48.72		Ta = –40 to –20°C 1.8 ≤ VCC ≤ 3.6
		47.52	48	48.48		Ta = –20 to 85°C 1.8 ≤ VCC ≤ 3.6
	f_{HOCO64}*5	63.04	64	64.96		Ta = –40 to –20°C 2.4 ≤ VCC ≤ 3.6
		63.36	64	64.64		Ta = –20 to 85°C 2.4 ≤ VCC ≤ 3.6

Table 48.23 Clock timing (2 of 2)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
HOCO clock oscillation stabilization time*6, *7	Except low-voltage mode	t_{HOCO24} t_{HOCO32}	-	-	37.1	μs	Figure 48.25
		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 48.27
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}	-	-	-*3	s	Figure 48.28

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 3.6 V.

Note 3. After changing the setting of the SOSCCR.SOSTP bit so that the sub-clock oscillator operates, only start using the sub-clock oscillator after the sub-clock oscillation stabilization wait time elapses, that is greater than or equal to the value recommended by the oscillator manufacturer.

Note 4. The 48-MHz HOCO can be used within a VCC range of 1.8 V to 3.6 V.

Note 5. The 64-MHz HOCO can be used within a VCC range of 2.4 V to 3.6 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.

Note 9. When setting up the main clock, ask the oscillator manufacturer for an oscillation evaluation and use the results as the recommended oscillation stabilization time. Set the MOSCWTCR register to a value equal to or greater than the recommended stabilization time. After changing the setting of the MOSCCR.MOSTP bit so that the main clock oscillator operates, read the OSCSF.MOSCSF flag to confirm that it is 1, then start using the main clock.

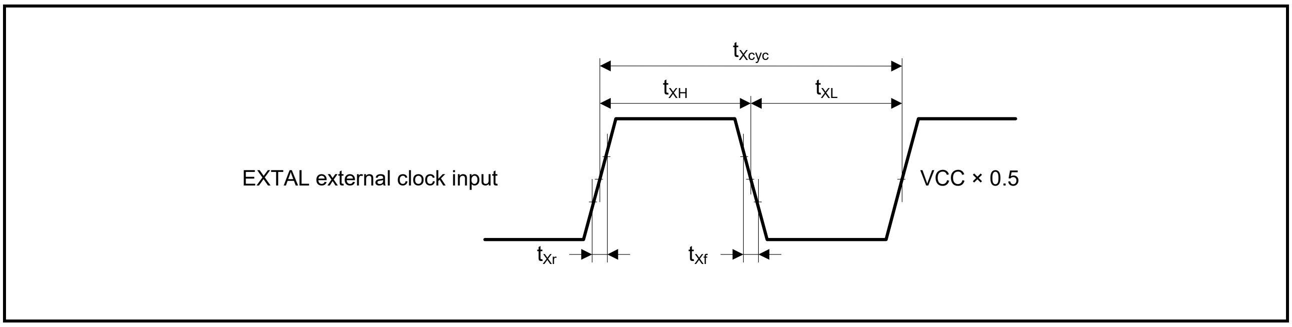


Figure 48.23 EXTAL external clock input timing

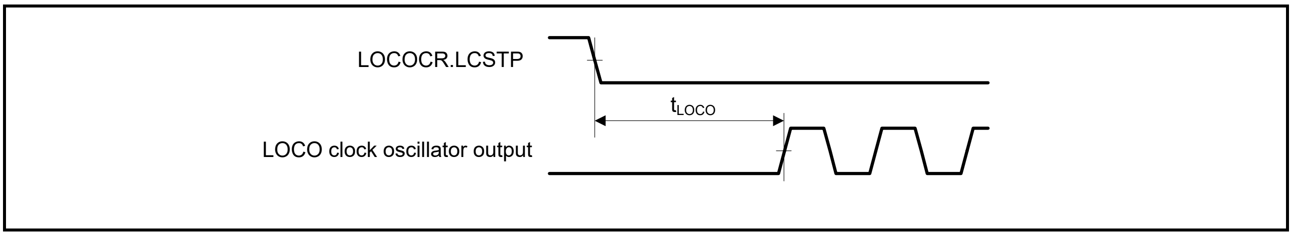


Figure 48.24 LOCO clock oscillation start timing

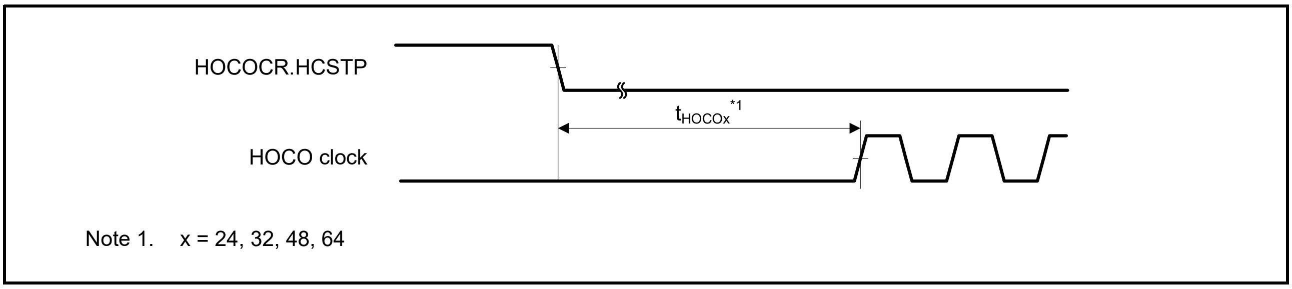

HOCOCR.HCSTP
t_{HOCOx}*1
HOCO clock
Note 1. x = 24, 32, 48, 64

Figure 48.25 HOCO clock oscillation start timing (started by setting HOCOCR.HCSTP bit)

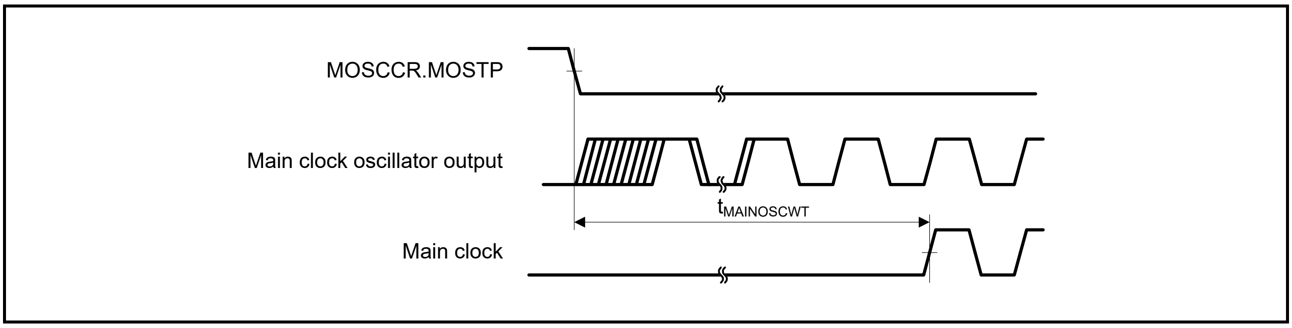

MOSCCR.MOSTP
Main clock oscillator output
$t_{MAINOSCWT}$
Main clock

Figure 48.26 Main clock oscillation start timing

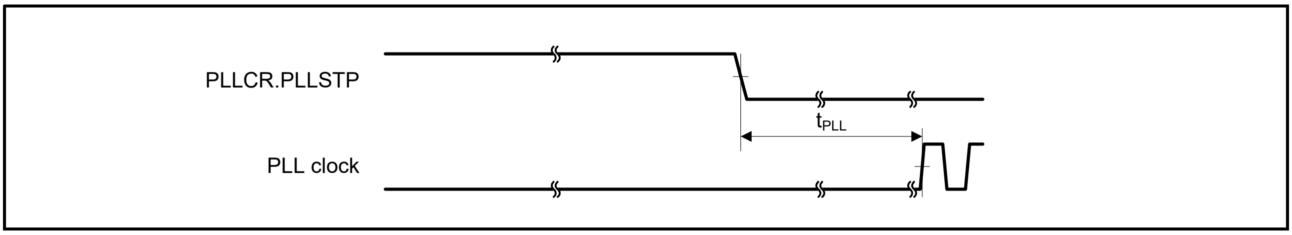

PLLCR.PLLSTP
t_{PLL}
PLL clock

Figure 48.27 PLL clock oscillation start timing (PLL is operated after main clock oscillation has settled)

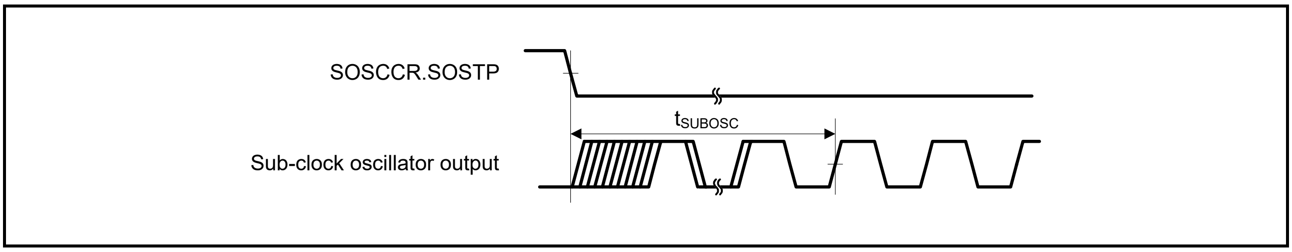

SOSCCR.SOSTP
t_{SUBOSC}
Sub-clock oscillator output

Figure 48.28 Sub-clock oscillation start timing

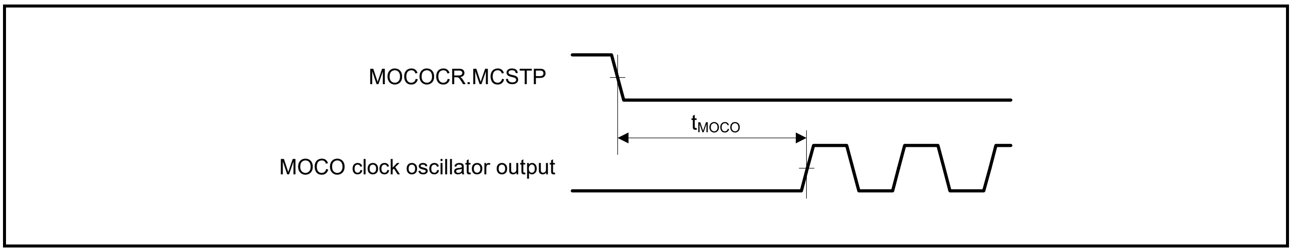

MOCOCR.MCSTP
t_{MOCO}
MOCO clock oscillator output

Figure 48.29 MOCO clock oscillation start timing

48.3.3 Reset Timing

Table 48.24 Reset timing

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
RES pulse width	At power-on	t_{RESWP}	3	-	-	ms	Figure 48.30
	Other than above	t_{RESW}	30	-	-	μs	Figure 48.31
Wait time after RES cancellation (at power-on)	LVD0: enable*1	t_{RESWT}	-	0.7	-	ms	Figure 48.30
	LVD0: disable*2		-	0.3	-		
Wait time after RES cancellation (during powered-on state)	LVD0: enable*1	t_{RESWT2}	-	0.5	-	ms	Figure 48.31
	LVD0: disable*2		-	0.05	-		
Internal reset cancellation time (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.

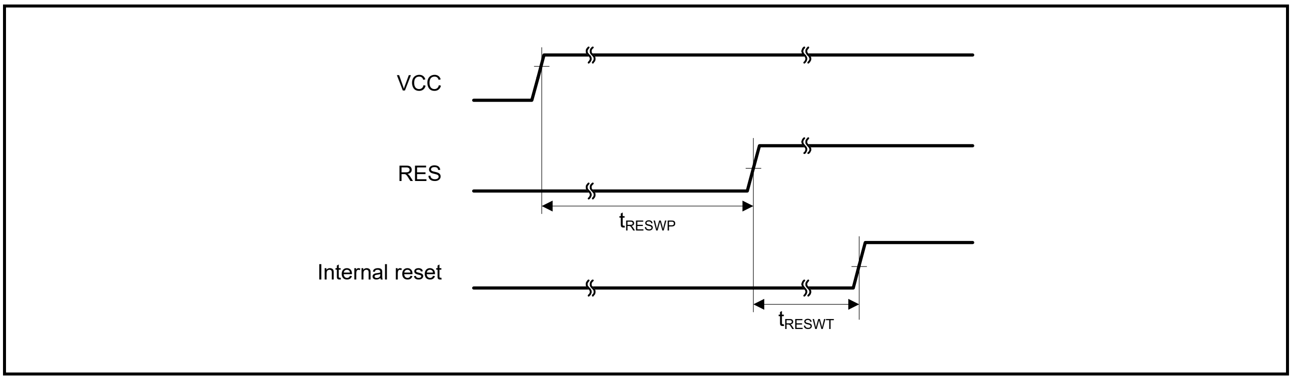


Figure 48.30 Reset input timing at power-on

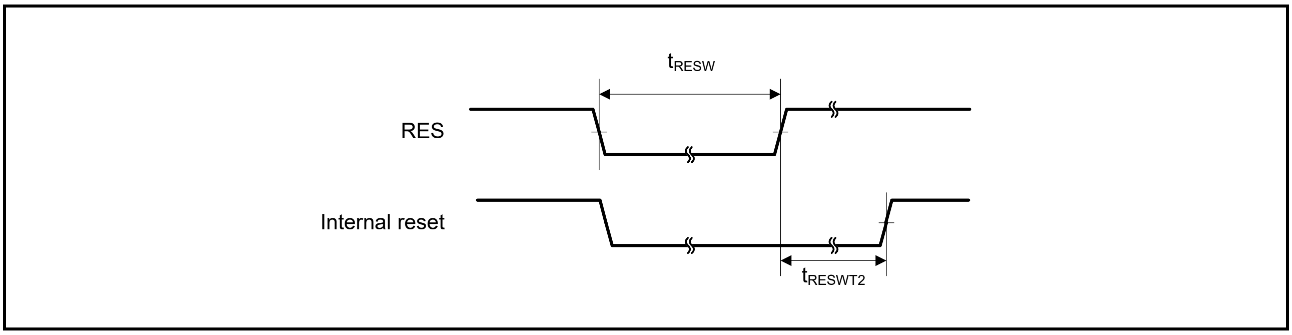


Figure 48.31 Reset input timing (1)

48.3.4 Wakeup Time

Table 48.25 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 48.32
			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 48.26 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 48.32
			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 (24 MHz)		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.

Table 48.27 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 48.32
		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 48.28 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 48.32
		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 48.29 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 48.32
		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.

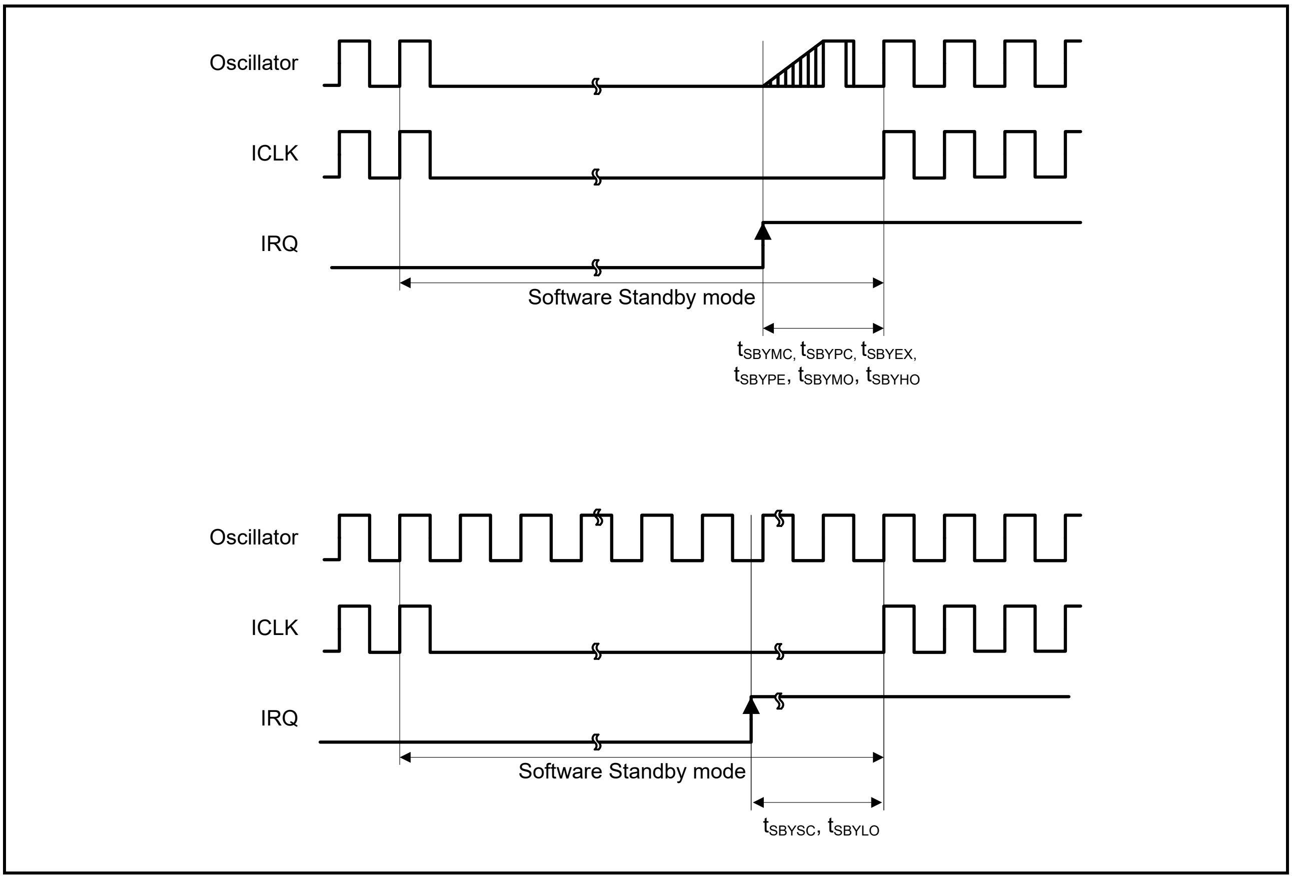
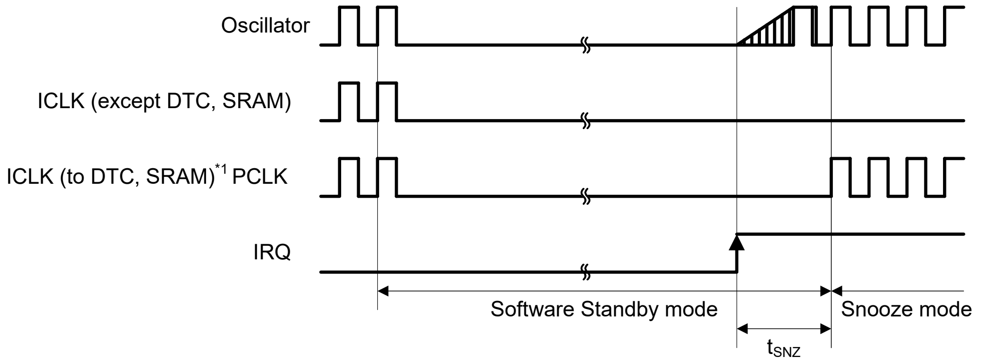

Oscillator
ICLK
IRQ
Software Standby mode
tSBYMC, tSBYPC, tSBYEX,
tSBYPE, tSBYMO, tSBYHO
Oscillator
ICLK
IRQ
Software Standby mode
tSBYSC, tSBYLO

Figure 48.32 Software Standby mode cancellation timing

Table 48.30 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 48.33
	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	


Oscillator
ICLK (except DTC, SRAM)
ICLK (to DTC, SRAM)*1 PCLK
IRQ
Software Standby mode
Snooze mode
tSNZ

Note 1. When SNZCR.SNZDTCEN is set to 1, ICLK is supplied to DTC and SRAM.

Figure 48.33 Recovery timing from Software Standby mode to Snooze mode

48.3.5 NMI and IRQ Noise Filter

Table 48.31 NMI and IRQ noise filter

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions	
NMI pulse width	t_{NMIW}	200	-	-	ns	NMI digital filter disabled	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2$*1	-	-			$t_{Pcyc} \times 2 > 200$ ns
		200	-	-		NMI digital filter enabled	$t_{NMICK} \times 3 \leq 200$ ns
		$t_{NMICK} \times 3.5$*2	-	-			$t_{NMICK} \times 3 > 200$ ns
IRQ pulse width	t_{IRQW}	200	-	-	ns	IRQ digital filter disabled	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2$*1	-	-			$t_{Pcyc} \times 2 > 200$ ns
		200	-	-		IRQ digital filter enabled	$t_{IRQCK} \times 3 \leq 200$ ns
		$t_{IRQCK} \times 3.5$*3	-	-			$t_{IRQCK} \times 3 > 200$ ns

Note: 200 ns minimum in Software Standby mode.
Note: If the clock source is switched, add 4 clock cycles of the switched source.
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).

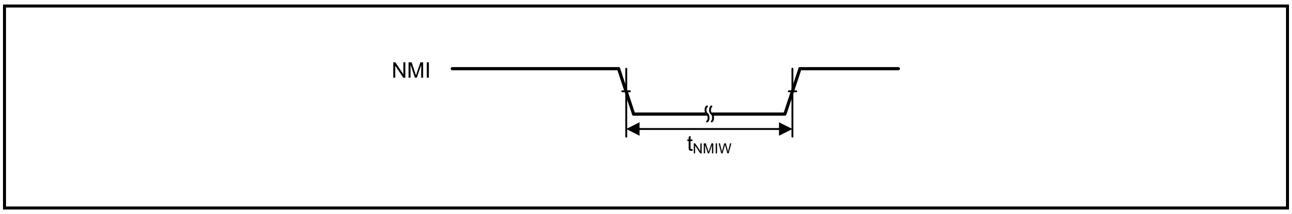


Figure 48.34 NMI interrupt input timing

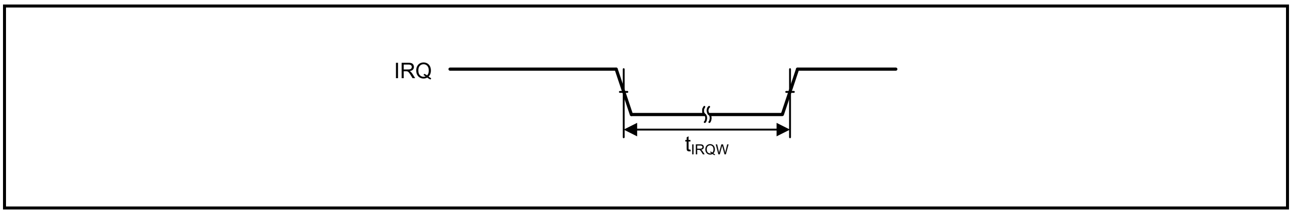


Figure 48.35 IRQ interrupt input timing

48.3.6 I/O Ports, POEG, GPT, AGT, KINT, and ADC14 Trigger Timing

Table 48.32 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 48.36
	Input/output data cycle (P004)		t_{POcyc}	10	-	us	
POEG	POEG input trigger pulse width		t_{POEW}	3	-	t_{Pcyc}	Figure 48.37
GPT	Input capture pulse width	Single edge	t_{GTICW}	1.5	-	t_{PDcyc}	Figure 48.38
		Dual edge		2.5	-		
AGT	AGTIO, AGTEE input cycle	2.7 V ≤ VCC ≤ 3.6 V	t_{ACYC}*1	250	-	ns	Figure 48.39
		2.4 V ≤ VCC < 2.7 V		500	-	ns	
		1.8 V ≤ VCC < 2.4 V		1000	-	ns	
	AGTIO, AGTEE input high level width, low-level width	2.7 V ≤ VCC ≤ 3.6 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	
	AGTIO, AGTO, AGTOB output cycle	2.7 V ≤ VCC ≤ 3.6 V	t_{ACYC2}	62.5	-	ns	Figure 48.39
		2.4 V ≤ VCC < 2.7 V		125	-	ns	
		1.8 V ≤ VCC < 2.4 V		250	-	ns	
ADC14	14-bit A/D converter trigger input pulse width		t_{TRGW}	1.5	-	t_{Pcyc}	Figure 48.40
KINT	KRn (n = 00 to 07) pulse width		t_{KR}	250	-	ns	Figure 48.41

Note 1. Constraints on input cycle:
When not switching the source clock: $t_{Pcyc} \times 2 < t_{ACYC}$ should be satisfied.
When switching the source clock: $t_{Pcyc} \times 6 < t_{ACYC}$ should be satisfied.

Note: t_{Pcyc}: PCLKB cycle, t_{PDcyc}: PCLKD cycle

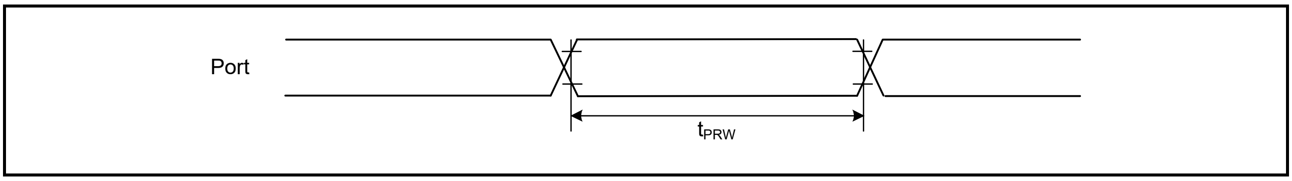


Figure 48.36 I/O ports input timing

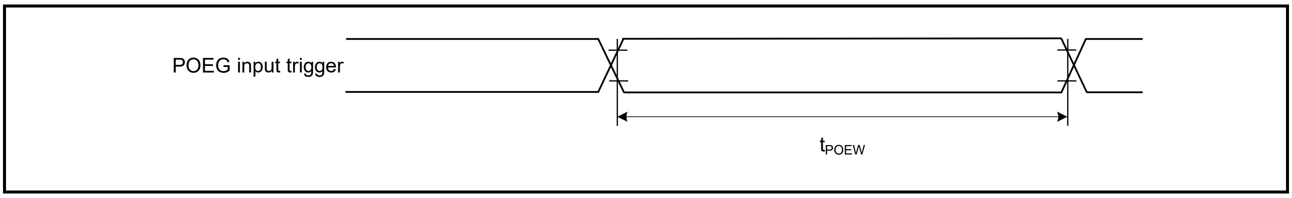


Figure 48.37 POEG input trigger timing

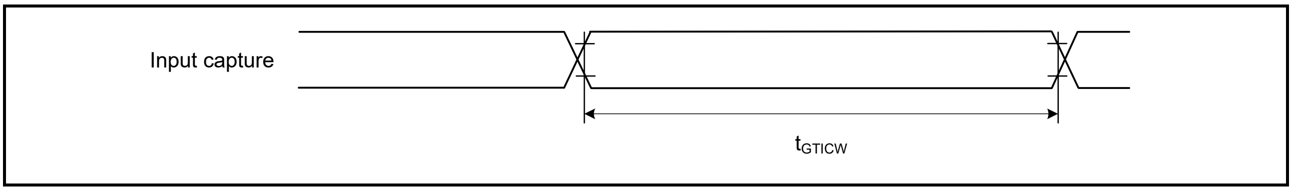


Figure 48.38 GPT input capture timing

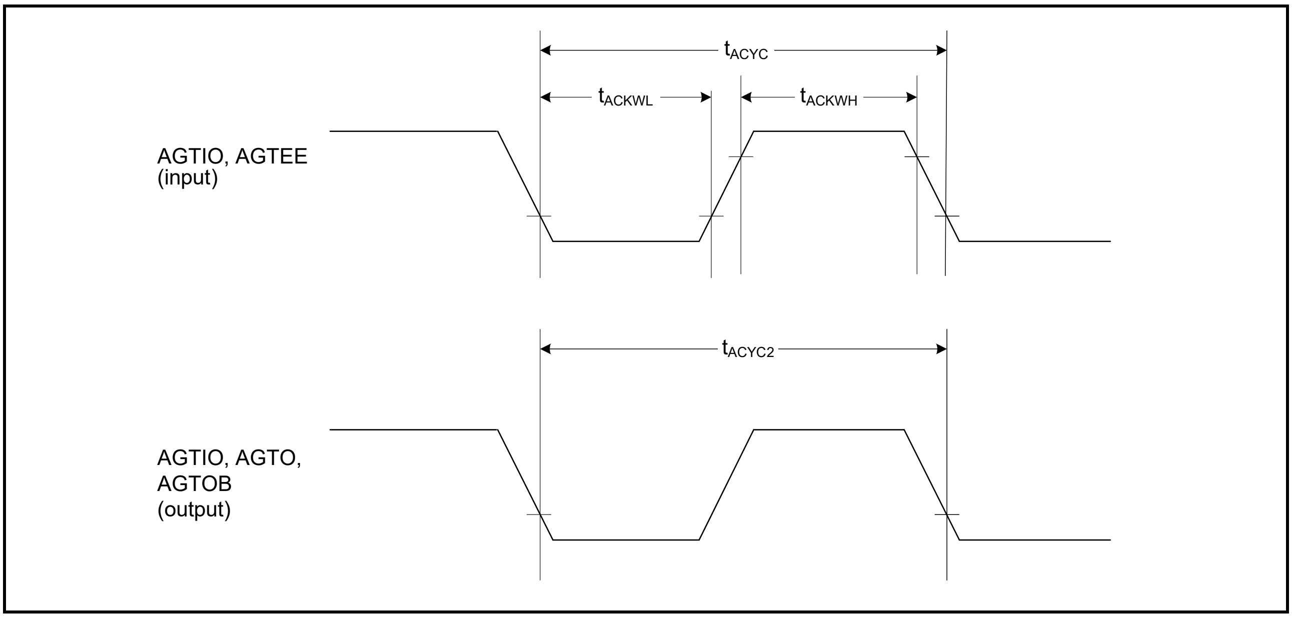

t_{ACYC}
t_{ACKWL}
t_{ACKWH}
AGTIO, AGTEE
(input)
t_{ACYC2}
AGTIO, AGTO,
AGTOB
(output)

Figure 48.39 AGT I/O timing

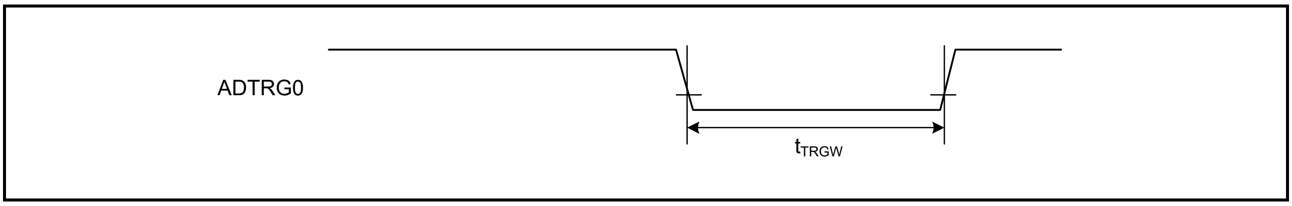

ADTRG0
t_{TRGW}

Figure 48.40 ADC14 trigger input timing

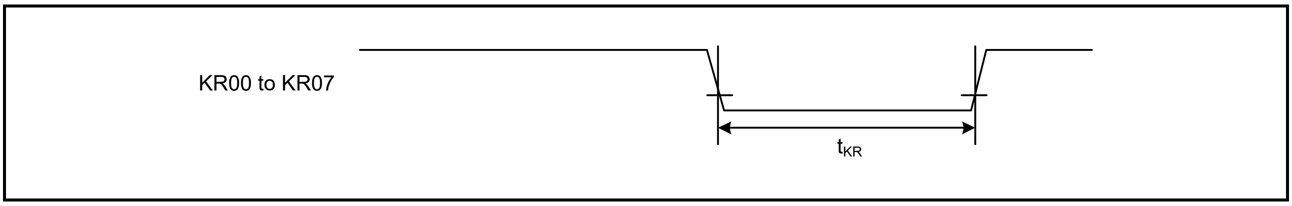

KR00 to KR07
t_{KR}

Figure 48.41 Key interrupt input timing

48.3.7 CAC Timing

Table 48.33 CAC timing

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
CAC	CACREF input pulse width	$t_{PBcyc}{}^{*1} \le t_{cac}{}^{*2}$	t_{CACREF}	$4.5 \times t_{cac} + 3 \times t_{PBcyc}{}^{*1}$	-	-	ns	-
		$t_{PBcyc}{}^{*1} > t_{cac}{}^{*2}$		$5 \times t_{cac} + 6.5 \times t_{PBcyc}{}^{*1}$	-	-	ns	

Note 1. t_{PBcyc}: PCLKB cycle.

Note 2. t_{cac}: CAC count clock source cycle.

48.3.8 SCI Timing

Table 48.34 SCI timing (1)

Parameter				Symbol	Min	Max	Unit[1]	Test conditions
SCI	Input clock cycle	Asynchronous		t_{Scyc}	4	-	t_{Pcyc}	Figure 48.42
		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	
	Output clock fall time		1.8 V or above	t_{SCKf}	-	20	ns	
	Transmit data delay (master)	Clock synchronous	1.8 V or above	t_{TXD}	-	40	ns	Figure 48.43
	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		
	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	-		
	Receive data setup time (slave)	Clock synchronous	2.7 V or above		40	-	ns	
			1.8 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.

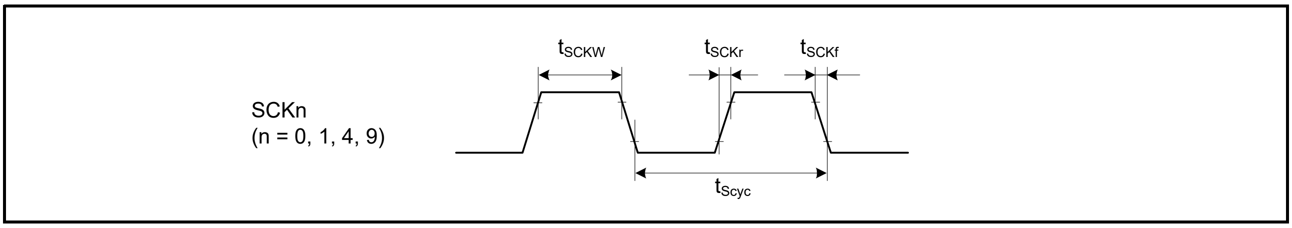


Figure 48.42 SCK clock input timing

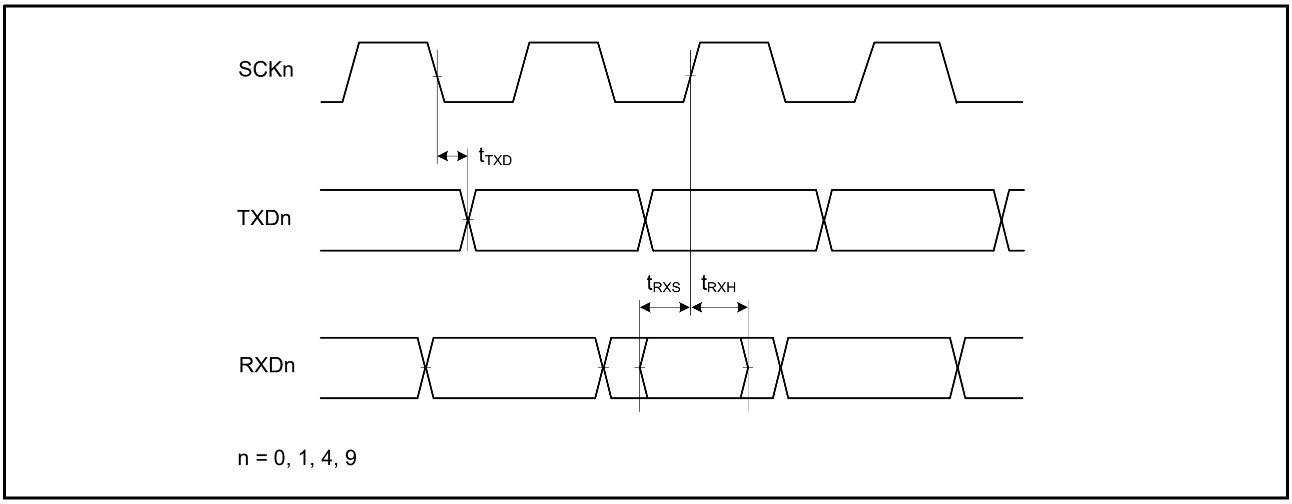
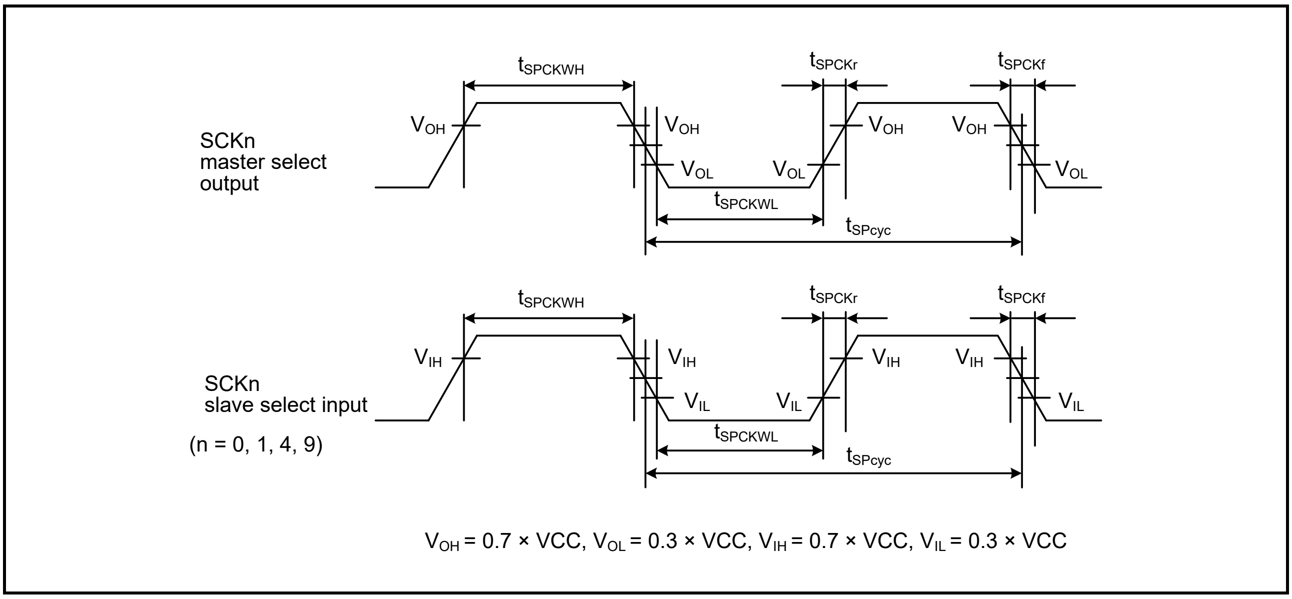


Figure 48.43 SCI input/output timing in clock synchronous mode

Table 48.35 SCI timing (2)

Parameter				Symbol	Min	Max	Unit	Test conditions
Simple SPI	SCK clock cycle output (master)			t_{SPcyc}	4	65536	t_{Pcyc}	Figure 48.44
	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	
	Data input setup time	Master	2.7 V or above	t_{SU}	45	-	ns	Figure 48.45 to Figure 48.48
			2.4 V or above		55	-		
			1.8 V or above		80	-		
		Slave	2.7 V or above		40	-		
			1.8 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	
		Slave	2.4 V or above		-	65		
			1.8 V or above		-	100		
	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	-		
		Slave			–10	-		
	Data rise and fall time	Master	1.8 V or above	t_{Dr}, t_{Df}	-	20	ns	
		Slave	1.8 V or above		-	20		
	Slave access time			t_{SA}	-	10 (PCLKA > 32 MHz), 6 (PCLKA ≤ 32 MHz)	t_{Pcyc}	Figure 48.47 and Figure 48.48
	Slave output release time			t_{REL}	-	10 (PCLKA > 32 MHz), 6 (PCLKA ≤ 32 MHz)	t_{Pcyc}	


SCKn
master select
output
SCKn
slave select input
(n = 0, 1, 4, 9)
tSPCKWH
tSPCKr
tSPCKf
tSPCKWL
tSPcyc
VOH
VOL
VIH
VIL

V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC

Figure 48.44 SCI simple SPI mode clock timing

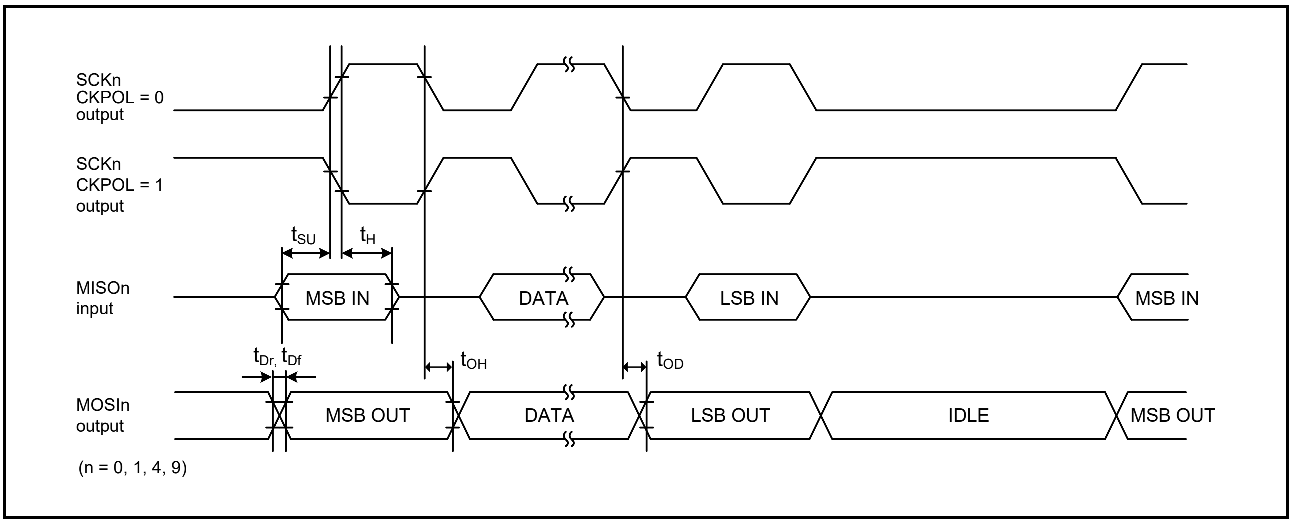

SCKn
CKPOL = 0
output
SCKn
CKPOL = 1
output
tSU
tH
MISOn
input
MSB IN
DATA
LSB IN
MSB IN
tDr, tDf
tOH
tOD
MOSIn
output
MSB OUT
DATA
LSB OUT
IDLE
MSB OUT
(n = 0, 1, 4, 9)

Figure 48.45 SCI simple SPI mode timing (master, CKPH = 1)

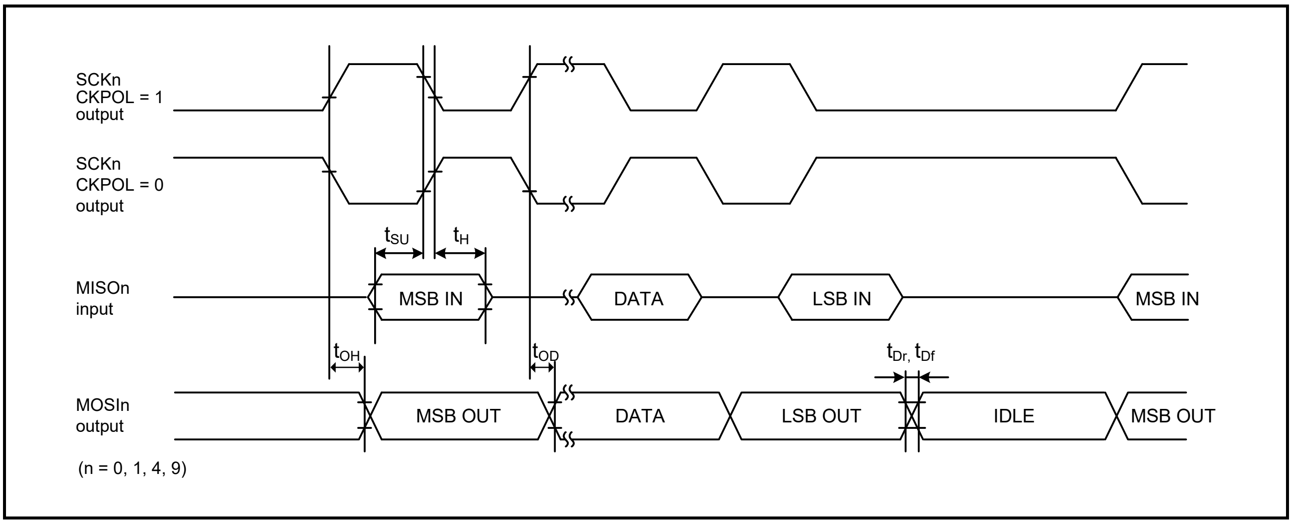

SCKn
CKPOL = 1
output
SCKn
CKPOL = 0
output
tSU
tH
MISOn
input
MSB IN
DATA
LSB IN
MSB IN
tOH
tOD
tDr, tDf
MOSIn
output
MSB OUT
DATA
LSB OUT
IDLE
MSB OUT
(n = 0, 1, 4, 9)

Figure 48.46 SCI simple SPI mode timing (master, CKPH = 0)

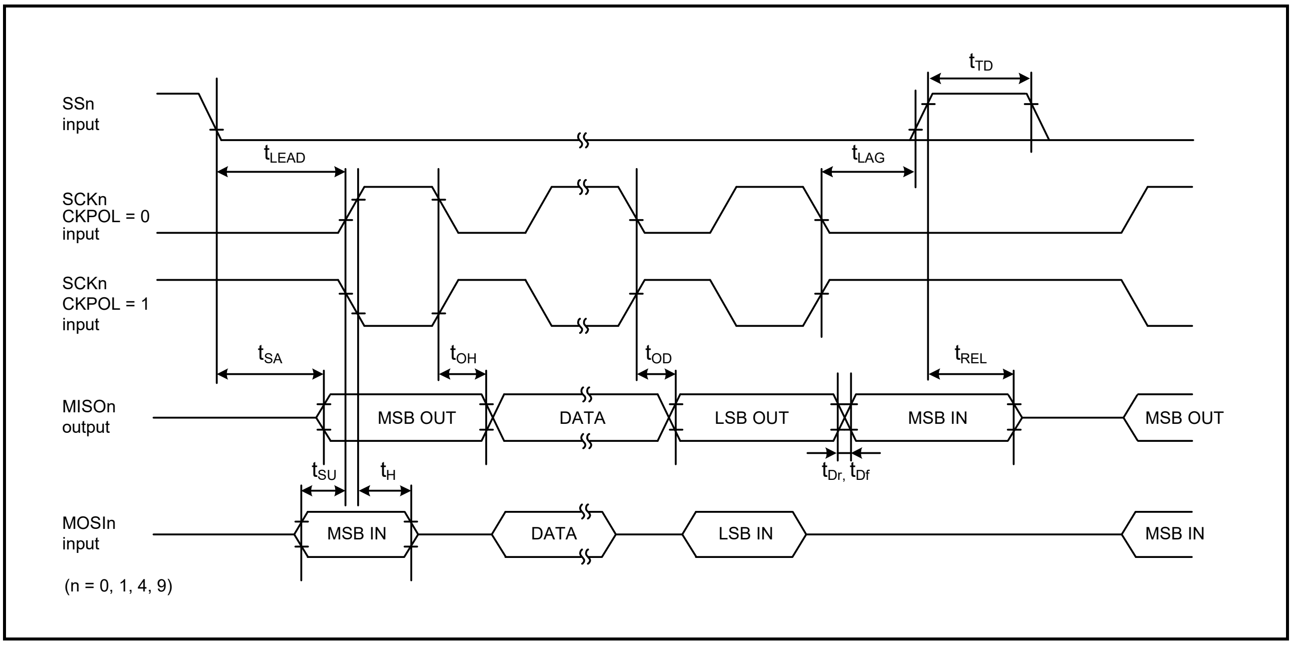

tTD
SSn
input
tLEAD
tLAG
SCKn
CKPOL = 0
input
SCKn
CKPOL = 1
input
tSA
tOH
tOD
tREL
MISOn
output
MSB OUT
DATA
LSB OUT
MSB IN
MSB OUT
tSU
tH
tDr, tDf
MOSIn
input
MSB IN
DATA
LSB IN
MSB IN
(n = 0, 1, 4, 9)

Figure 48.47 SCI simple SPI mode timing (slave, CKPH = 1)

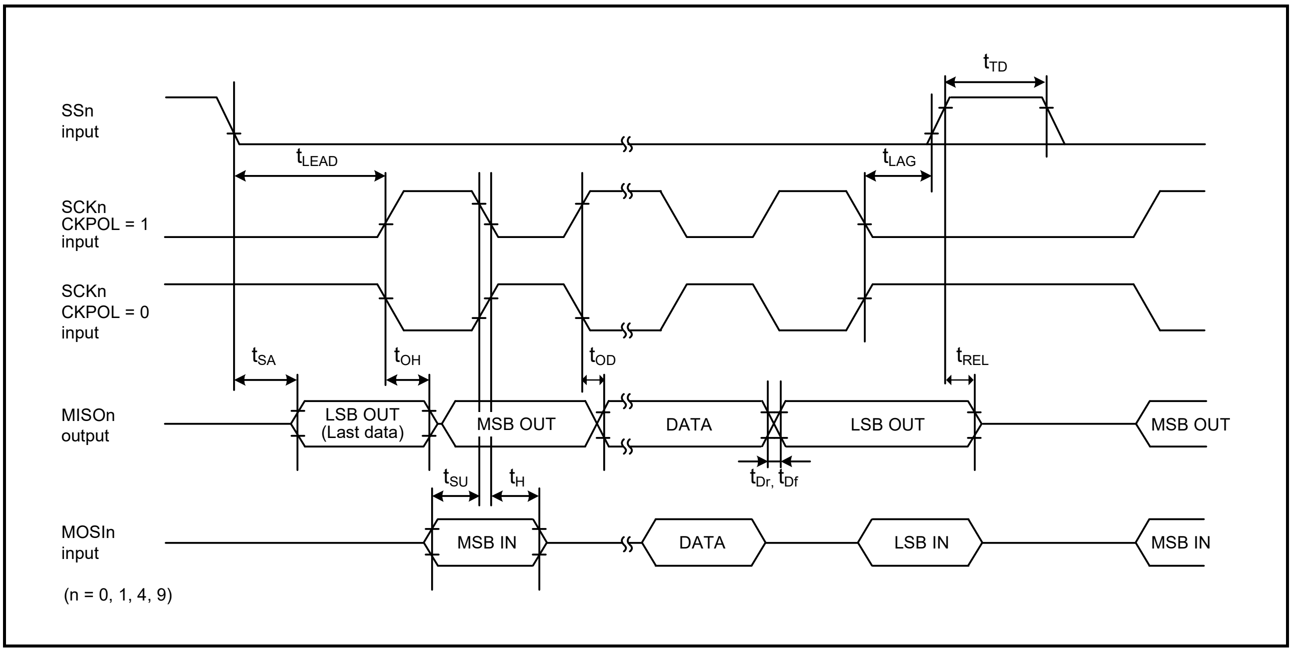

SSn
input
SCKn
CKPOL = 1
input
SCKn
CKPOL = 0
input
MISOn
output
MOSIn
input
(n = 0, 1, 4, 9)
tTD
tLEAD
tLAG
tSA
tOH
tOD
tREL
LSB OUT
(Last data)
MSB OUT
DATA
LSB OUT
MSB OUT
tSU
tH
tDr, tDf
MSB IN
DATA
LSB IN
MSB IN

Figure 48.48 SCI simple SPI mode timing (slave, CKPH = 0)

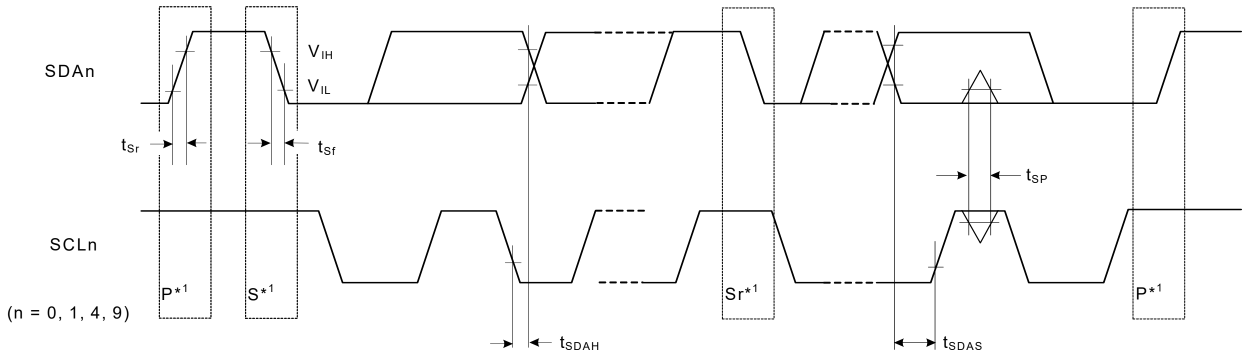
Table 48.36 SCI timing (3)

Conditions: VCC = 2.7 to 3.6 V

Parameter		Symbol	Min	Max	Unit	Test conditions
Simple IIC (Standard mode)	SDA input rise time	t_{Sr}	-	1000	ns	Figure 48.49
	SDA input fall time	t_{Sf}	-	300	ns	
	SDA input spike pulse removal time	t_{SP}	0	4 × t_{IICcyc}*1	ns	
	Data input setup time	t_{SDAS}	250	-	ns	
	Data input hold time	t_{SDAH}	0	-	ns	
	SCL, SDA capacitive load	C_b*2	-	400	pF	
Simple IIC (Fast mode)	SDA input rise time	t_{Sr}	-	300	ns	Figure 48.49 For all ports use PmnPFS.DSCR of middle drive.
	SDA input fall time	t_{Sf}	-	300	ns	
	SDA input spike pulse removal time	t_{SP}	0	4 × t_{IICcyc}*1	ns	
	Data input setup time	t_{SDAS}	100	-	ns	
	Data input hold time	t_{SDAH}	0	-	ns	
	SCL, SDA capacitive load	C_b*2	-	400	pF	

Note 1. t_{IICcyc}: Clock cycle selected by the SMR.CKS[1:0] bits.

Note 2. Cb indicates the total capacity of the bus line.


SDAn
V_{IH}
V_{IL}
t_{Sr}
t_{Sf}
t_{SP}
SCLn
(n = 0, 1, 4, 9)
P*1
S*1
Sr*1
P*1
t_{SDAH}
t_{SDAS}

Test conditions:
V_{IH} = VCC × 0.7, V_{IL} = VCC × 0.3
V_{OL} = 0.6 V, I_{OL} = 6 mA

Note 1. S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Restart condition.

Figure 48.49 SCI simple IIC mode timing

48.3.9 SPI Timing

Table 48.37 SPI timing (1 of 2)

Conditions: Middle drive output is selected in the Port Drive Capability in PmnPFS register

Parameter				Symbol	Min	Max	Unit*1	Test conditions
SPI	RSPCK clock cycle	Master		t_{SPcyc}	2^{*4}	4096	t_{Pcyc}	Figure 48.50
		Slave			6	4096		
	RSPCK clock high pulse width	Master		t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	-	ns	
		Slave			3 × t_{Pcyc}	-		
	RSPCK clock low pulse width	Master		t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	-	ns	
		Slave			3 × 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		
		Input			-	1	μs	
	Data input setup time	Master		t_{SU}	10	-	ns	Figure 48.51 to Figure 48.56
		Slave	2.4 V or above		10	-		
			1.8 V or above		15	-		
	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	1.8 V or above	t_{LEAD}	$-30 + N \times t_{Spcyc}^{*2}$	-	ns	
		Slave			6 × t_{Pcyc}	-		
	SSL hold time	Master		t_{LAG}	$-30 + N \times t_{Spcyc}^{*3}$	-		
		Slave			6 × t_{Pcyc}	-		
	Data output delay	Master	2.7 V or above	t_{OD}	-	14	ns	Figure 48.51 to Figure 48.56
			2.4 V or above		-	20		
			1.8 V or above		-	25		
		Slave	2.7 V or above		-	50		
			2.4 V or above		-	60		
			1.8 V or above		-	85		
	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 × 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		
		Input			-	1	μs	

Table 48.37 SPI timing (2 of 2)

Conditions: Middle drive output is selected in the Port Drive Capability in PmnPFS register

Parameter				Symbol	Min	Max	Unit*1	Test conditions
SPI	SSL rise and fall time	Output	2.7 V or above	t_{SSLr}, t_{SSLf}	-	10	ns	Figure 48.51 to Figure 48.56
			2.4 V or above		-	15		
			1.8 V or above		-	20		
		Input			-	1	μs	
	Slave access time		2.4 V or above	t_{SA}	-	2 × t_{Pcyc} + 100	ns	Figure 48.55 and Figure 48.56
			1.8 V or above		-	2 × t_{Pcyc} + 140		
	Slave output release time		2.4 V or above	t_{REL}	-	2 × t_{Pcyc} + 100	ns	
			1.8 V or above		-	2 × t_{Pcyc} + 140		

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.

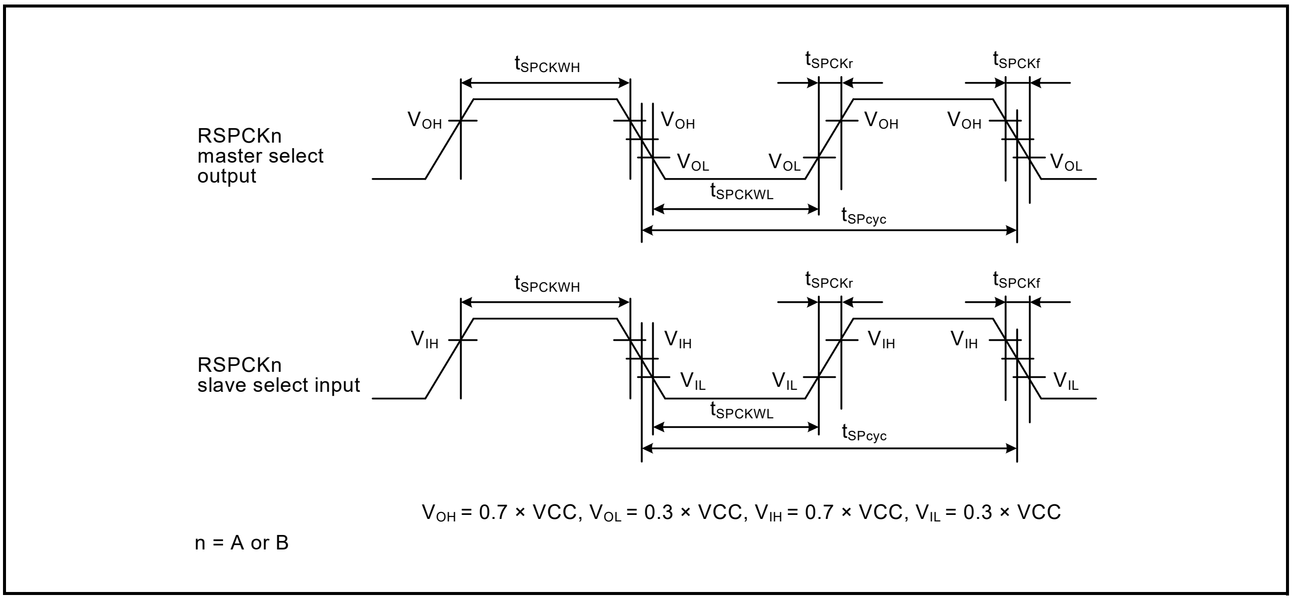
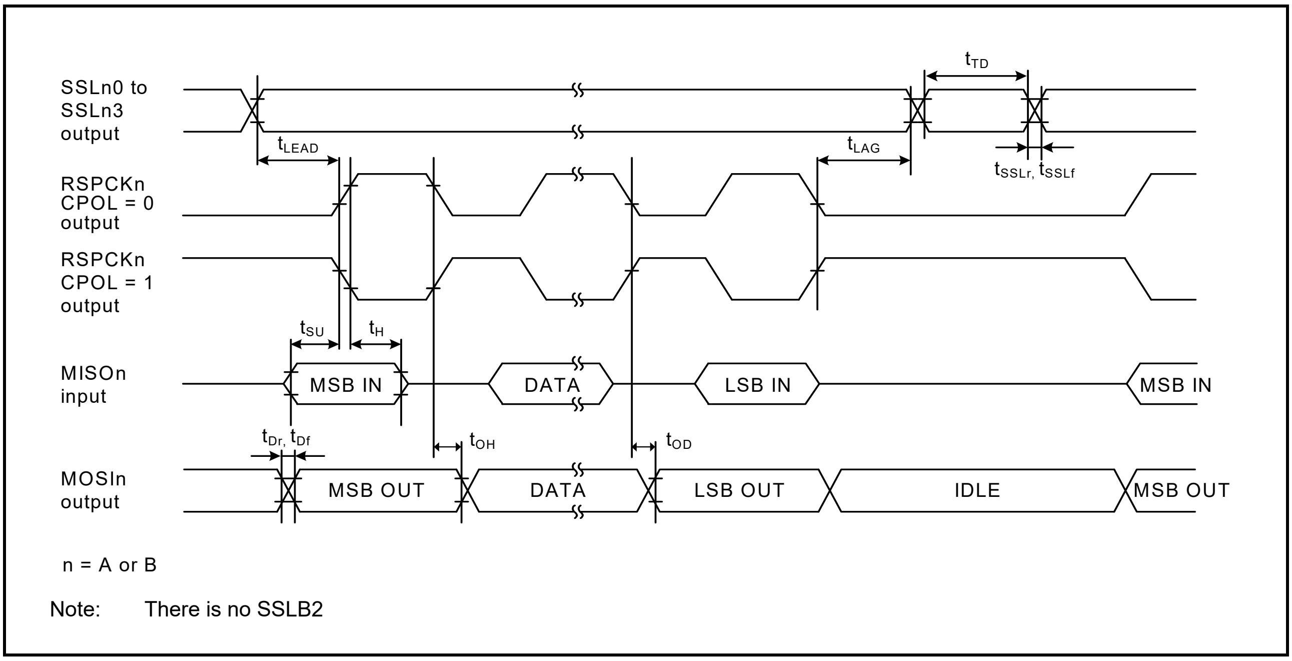

RSPCKn
master select
output
t_{SPCKWH}
t_{SPCKr}
t_{SPCKf}
V_{OH}
V_{OL}
t_{SPCKWL}
t_{SPcyc}
RSPCKn
slave select input
V_{IH}
V_{IL}
V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC
n = A or B

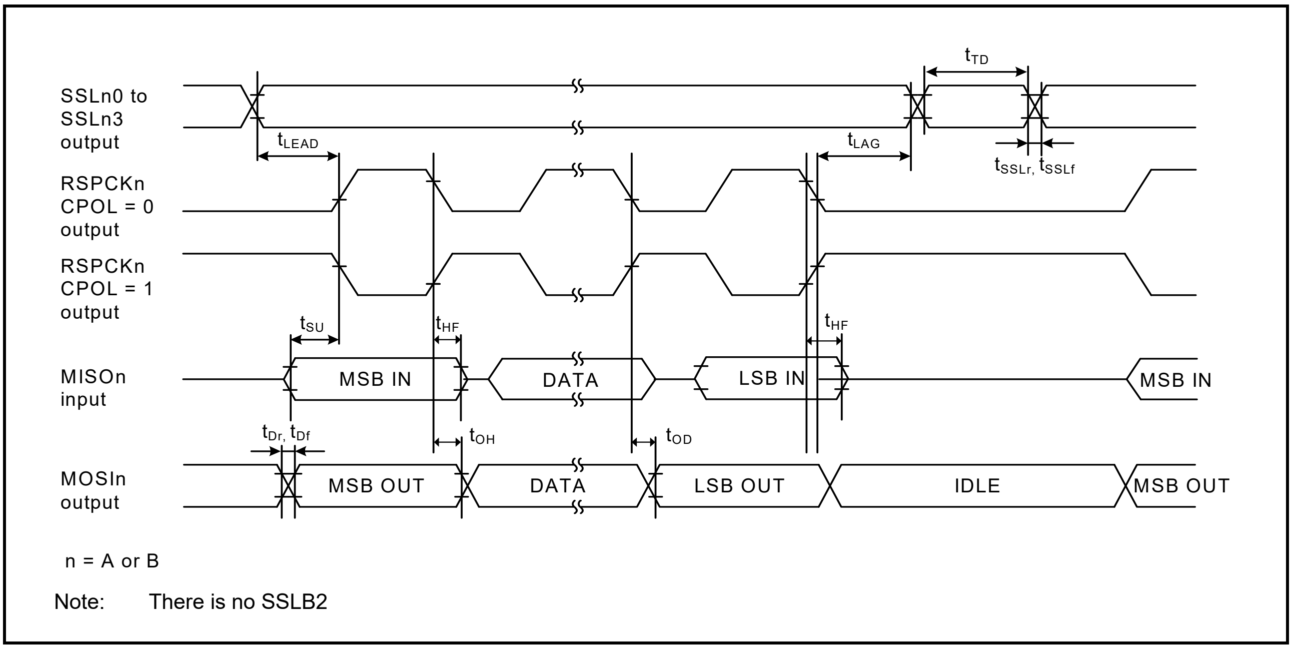
Figure 48.50 SPI clock timing


SSLn0 to
SSLn3
output
RSPCKn
CPOL = 0
output
RSPCKn
CPOL = 1
output
MISOn
input
MOSIn
output
tTD
tLEAD
tLAG
tSSLr, tSSLf
tSU
tH
MSB IN
DATA
LSB IN
MSB IN
tDr, tDf
tOH
tOD
MSB OUT
DATA
LSB OUT
IDLE
MSB OUT

n = A or B

Note: There is no SSLB2

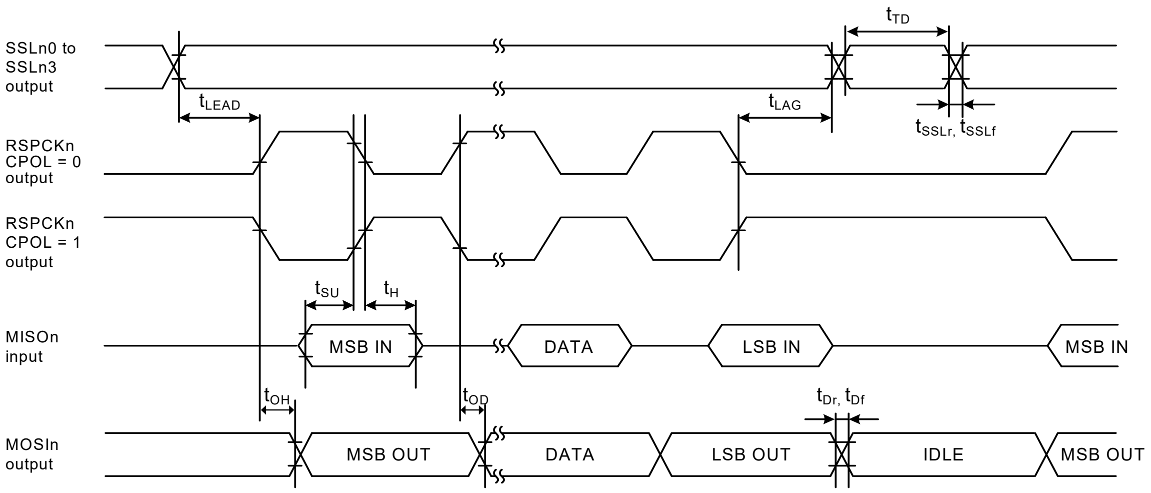
Figure 48.51 SPI timing (master, CPHA = 0) (bit rate: PCLKA division ratio is set to any value other than 1/2)


SSLn0 to
SSLn3
output
RSPCKn
CPOL = 0
output
RSPCKn
CPOL = 1
output
MISOn
input
MOSIn
output
tTD
tLEAD
tLAG
tSSLr, tSSLf
tSU
tHF
tHF
MSB IN
DATA
LSB IN
MSB IN
tDr, tDf
tOH
tOD
MSB OUT
DATA
LSB OUT
IDLE
MSB OUT

n = A or B

Note: There is no SSLB2

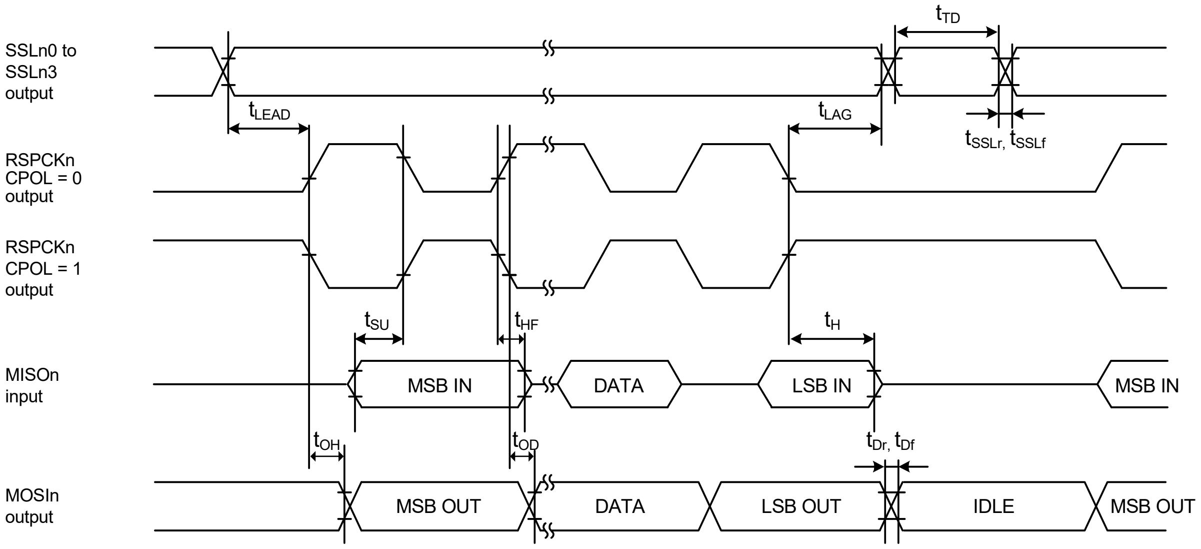
Figure 48.52 SPI timing (master, CPHA = 0) (bit rate: PCLKA division ratio is set to 1/2)


tTD
SSLn0 to SSLn3 output
tLEAD
tLAG
tSSLr, tSSLf
RSPCKn CPOL = 0 output
RSPCKn CPOL = 1 output
tSU
tH
MISOn input
MSB IN
DATA
LSB IN
MSB IN
tOH
tOD
tDr, tDf
MOSIn output
MSB OUT
DATA
LSB OUT
IDLE
MSB OUT

n = A or B

Note: There is no SSLB2

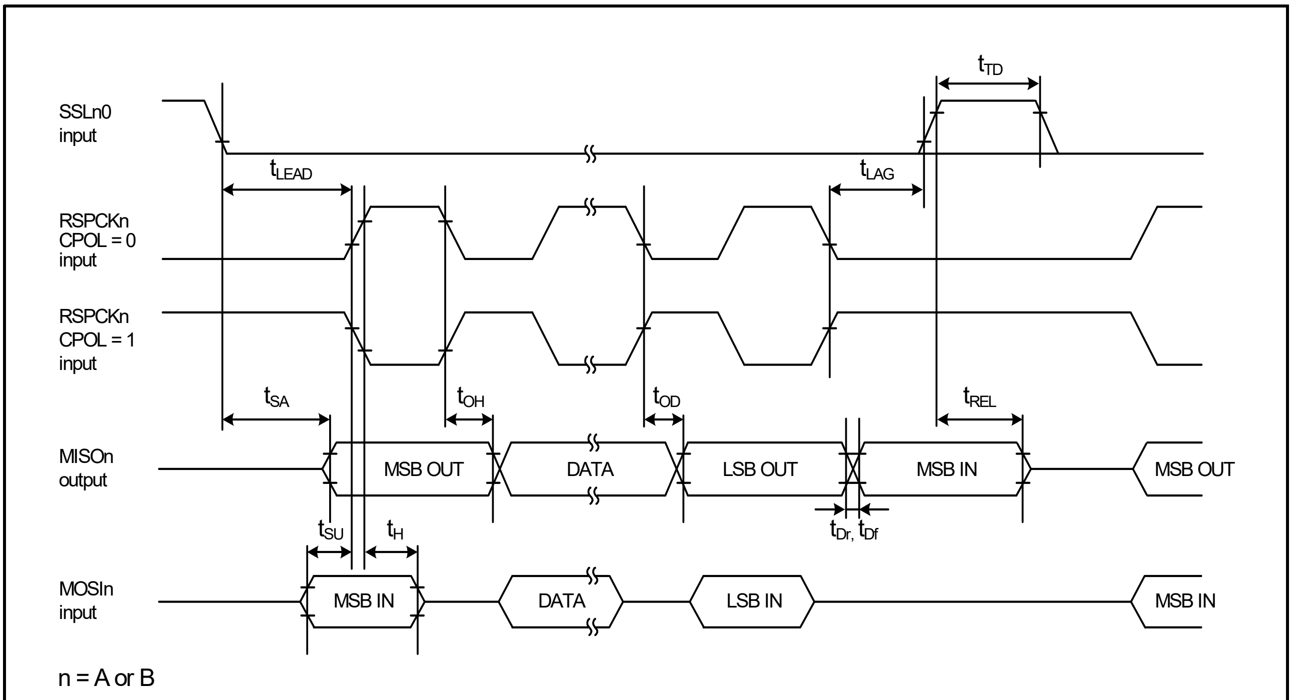
Figure 48.53 SPI timing (master, CPHA = 1) (bit rate: PCLKA division ratio is set to any value other than 1/2)


tTD
SSLn0 to SSLn3 output
tLEAD
tLAG
tSSLr, tSSLf
RSPCKn CPOL = 0 output
RSPCKn CPOL = 1 output
tSU
tHF
tH
MISOn input
MSB IN
DATA
LSB IN
MSB IN
tOH
tOD
tDr, tDf
MOSIn output
MSB OUT
DATA
LSB OUT
IDLE
MSB OUT

n = A or B

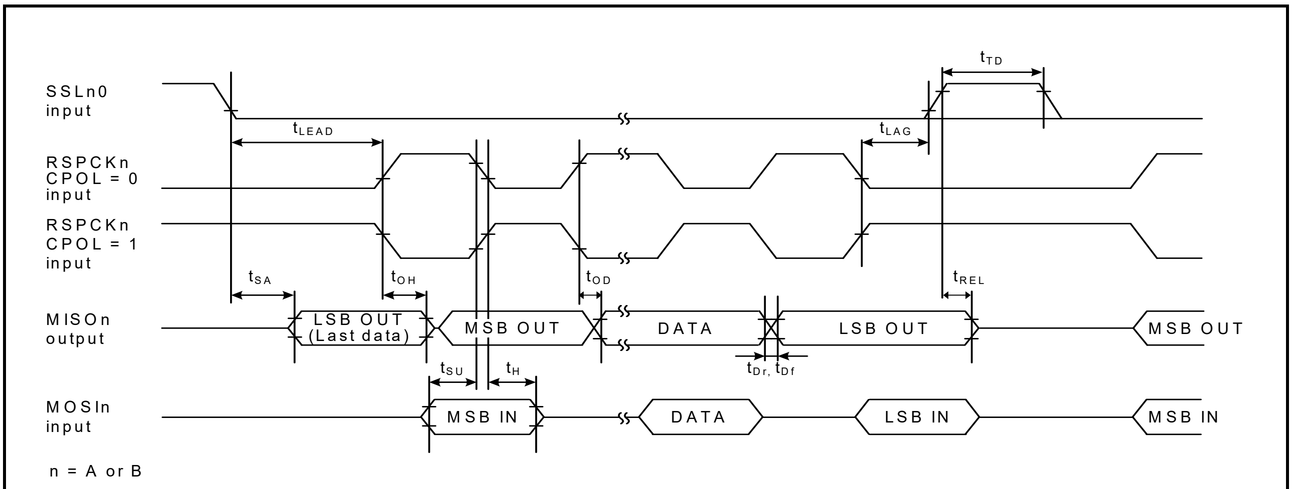
Note: There is no SSLB2

Figure 48.54 SPI timing (master, CPHA = 1) (bit rate: PCLKA division ratio is set to 1/2)


SSLn0
input
RSPCKn
CPOL = 0
input
RSPCKn
CPOL = 1
input
MISOn
output
MOSIn
input
t_{TD}
t_{LEAD}
t_{LAG}
t_{SA}
t_{OH}
t_{OD}
t_{REL}
t_{SU}
t_{H}
t_{Dr}, t_{Df}
MSB OUT
DATA
LSB OUT
MSB IN
MSB OUT
MSB IN
DATA
LSB IN
MSB IN
n = A or B

Note: There is no SSLB2

Figure 48.55 SPI timing (slave, CPHA = 0)


SSLn0
input
RSPCKn
CPOL = 0
input
RSPCKn
CPOL = 1
input
MISOn
output
MOSIn
input
t_{TD}
t_{LEAD}
t_{LAG}
t_{SA}
t_{OH}
t_{OD}
t_{REL}
t_{SU}
t_{H}
t_{Dr}, t_{Df}
LSB OUT
(Last data)
MSB OUT
DATA
LSB OUT
MSB OUT
MSB IN
DATA
LSB IN
MSB IN
n = A or B

Note: There is no SSLB2

Figure 48.56 SPI timing (slave, CPHA = 1)

48.3.10 IIC Timing

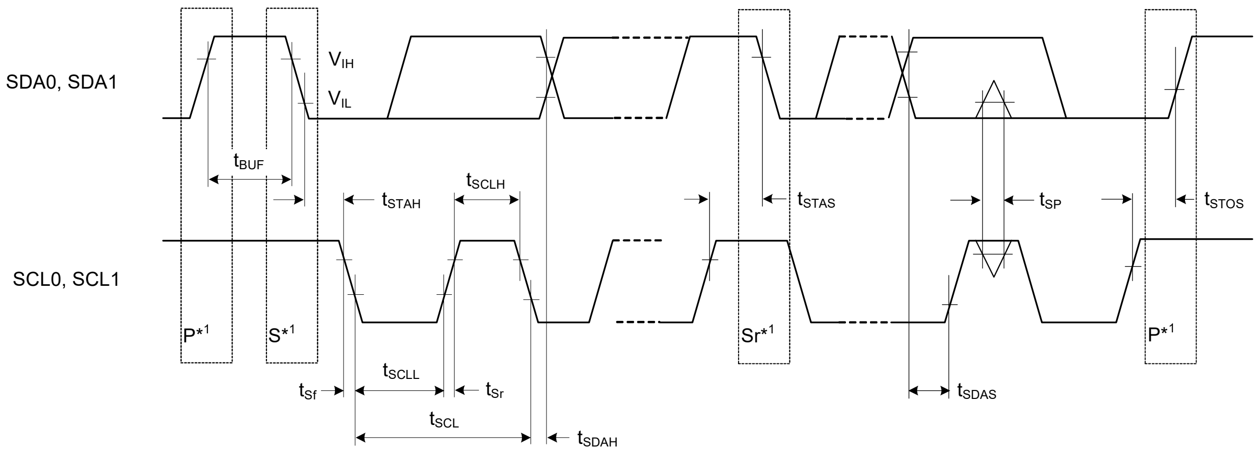
Table 48.38 IIC timing

Conditions: VCC = 2.7 to 3.6 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 48.57
	SCL input high pulse width	t_{SCLH}	3 (6) × t_{IICcyc} + 300	-	ns	
	SCL input low pulse width	t_{SCLL}	3 (6) × t_{IICcyc} + 300	-	ns	
	SCL, SDA input rise time	t_{Sr}	-	1000	ns	
	SCL, SDA input fall time	t_{Sf}	-	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA input bus free time (When wakeup function is disabled)	t_{BUF}	3 (6) × t_{IICcyc} + 300	-	ns	
	SDA input bus free time (When wakeup function is enabled)	t_{BUF}	3 (6) × t_{IICcyc} + 4 × t_{Pcyc} + 300	-	ns	
	START condition input hold time (When wakeup function is disabled)	t_{STAH}	t_{IICcyc} + 300	-	ns	
	START condition input hold time (When wakeup function is enabled)	t_{STAH}	1 (5) × t_{IICcyc} + t_{Pcyc} + 300	-	ns	
	Repeated START condition input setup time	t_{STAS}	1000	-	ns	
	STOP condition input setup time	t_{STOS}	1000	-	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	-	ns	
	Data input hold time	t_{SDAH}	0	-	ns	
	SCL, SDA capacitive load	C_b	-	400	pF	
IIC (Fast mode)	SCL input cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 600	-	ns	Figure 48.57 For all ports, use PmnPFS.DSCR of middle drive.
	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 1. S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 48.57 I^{2}C bus interface input/output timing

48.3.11 CLKOUT Timing

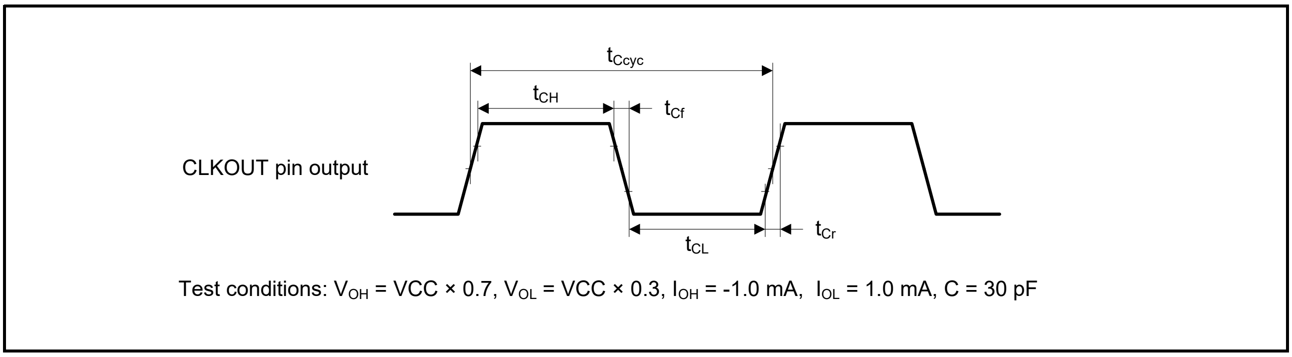
Table 48.39 CLKOUT timing

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 48.58
		VCC = 1.8 V or above		125	-		
	CLKOUT pin high pulse width*2	VCC = 2.7 V or above	t_{CH}	15	-	ns	
		VCC = 1.8 V or above		30	-		
	CLKOUT pin low pulse width*2	VCC = 2.7 V or above	t_{CL}	15	-	ns	
		VCC = 1.8 V or above		30	-		
	CLKOUT pin output rise time	VCC = 2.7 V or above	t_{Cr}	-	12	ns	
		VCC = 1.8 V or above		-	25		
	CLKOUT pin output fall time	VCC = 2.7 V or above	t_{Cf}	-	12	ns	
		VCC = 1.8 V or above		-	25		
CLKOUT_RF*3	CLKOUT_RF pin output cycle		t_{CRFcyc}	250	-	ns	Figure 48.59
	CLKOUT_RF pin high pulse width		t_{CRFH}	100	-	ns	
	CLKOUT_RF pin low pulse width		t_{CRFL}	100	-	ns	
	CLKOUT_RF pin output rise time		t_{CRFr}	-	5	ns	
	CLKOUT_RF pin output fall time		t_{CRFf}	-	5	ns	

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%.

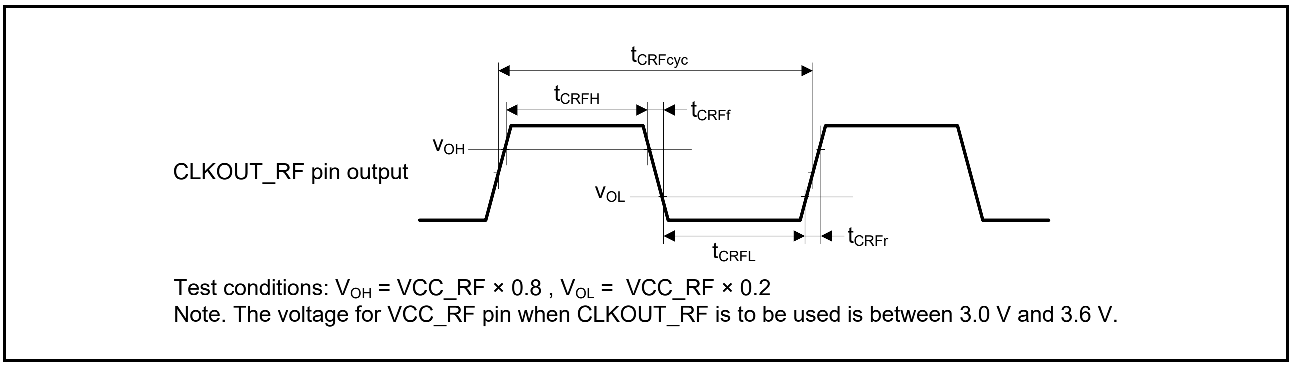
Note 2. When the MOCO is selected as the clock output source (the CKOCR.CKOSEL[2:0] bits are 001b), set the clock output division ratio selection to be divided by 2 (the CKOCR.CKODIV[2:0] bits are 001b).

Note 3. The voltage for VCC_RF when CLKOUT_RF pin is to be used is between 3.0 V and 3.6 V.


t_{Ccyc}
t_{CH}
t_{Cf}
CLKOUT pin output
t_{Cr}
t_{CL}

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 48.58 CLKOUT output timing


t_{CRFcyc}
t_{CRFH}
t_{CRFf}
V_{OH}
CLKOUT_RF pin output
V_{OL}
t_{CRFL}
t_{CRFr}

Test conditions: V_{OH} = VCC_RF × 0.8 , V_{OL} = VCC_RF × 0.2

Note. The voltage for VCC_RF pin when CLKOUT_RF is to be used is between 3.0 V and 3.6 V.

Figure 48.59 CLKOUT_RF Output Timing

48.4 USB Characteristics

48.4.1 USBFS Timing

Table 48.40 USB characteristics

Conditions: VCC = VCC_USB = 3.0 to 3.6 V, Ta = -20 to +85°C (USBCLKSEL = 1)

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 48.60, Figure 48.61, Figure 48.62
	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 required.)
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 Ver 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

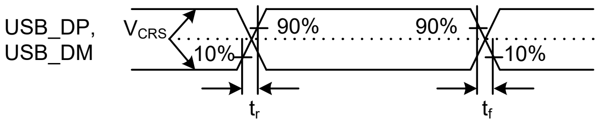


Figure 48.60 USB_DP and USB_DM output timing

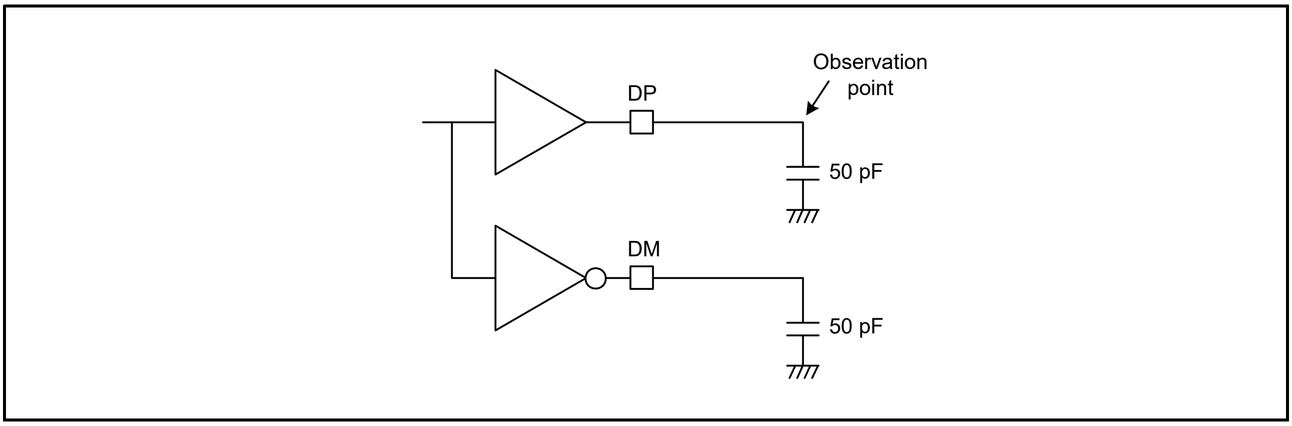

Observation
point
DP
50 pF
DM
50 pF

Figure 48.61 Test circuit for Full-Speed (FS) connection

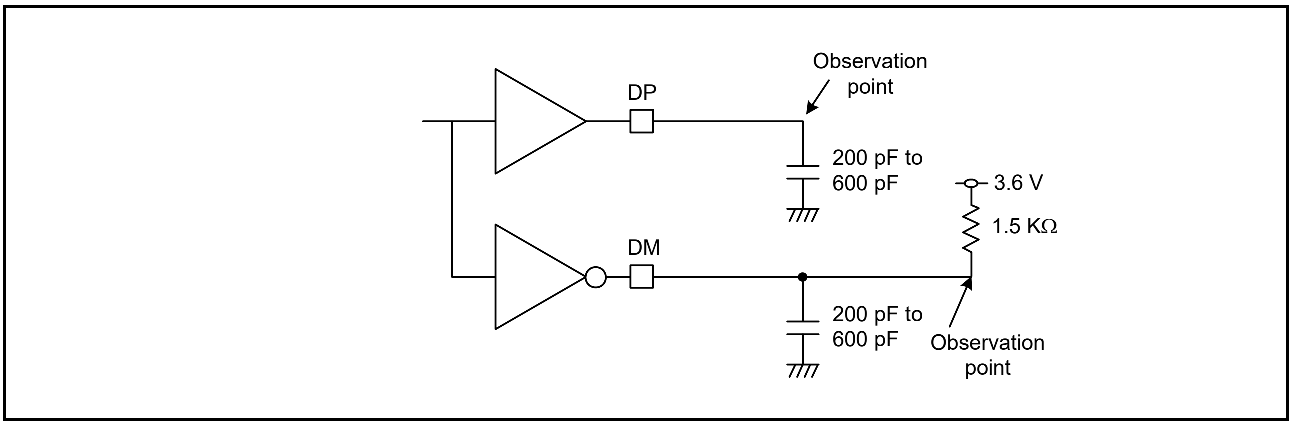

Observation
point
DP
200 pF to
600 pF
3.6 V
1.5 KΩ
DM
200 pF to
600 pF
Observation
point

Figure 48.62 Test circuit for Low-Speed (LS) connection

48.5 ADC14 Characteristics

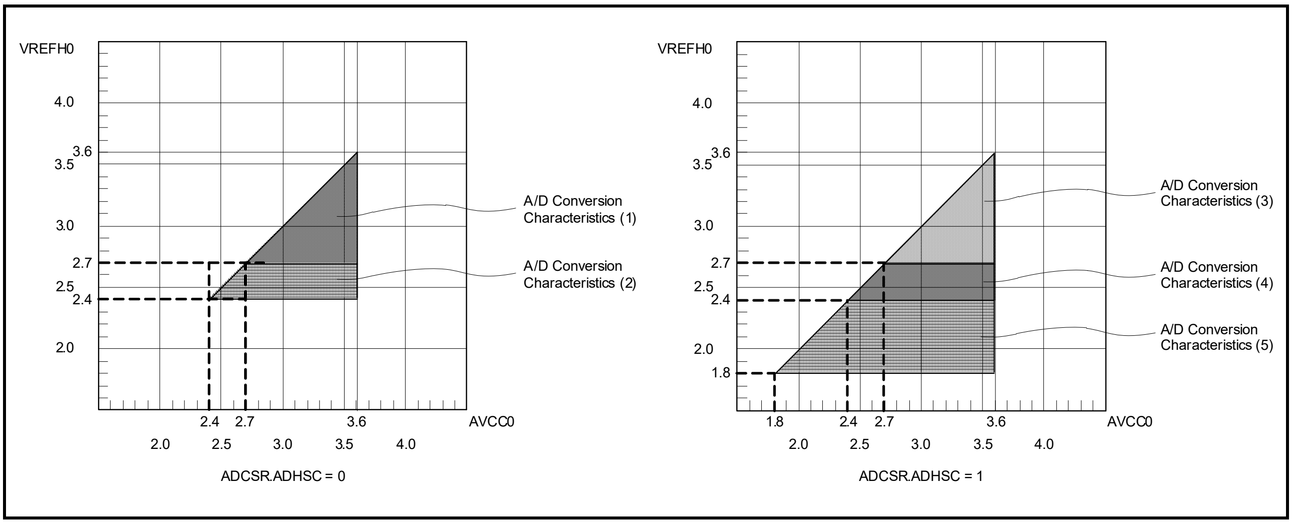


Figure 48.63 AVCC0 to VREFH0 voltage range

Table 48.41 A/D conversion characteristics (1) in high-speed A/D conversion mode (1 of 2)
Conditions: VCC = AVCC0 = 2.7 to 3.6 V, VREFH0 = 2.7 to 3.6 V
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	-
INL integral nonlinearity error		-	±1.0	±3.0	LSB	-
14-bit mode						
Resolution		-	-	14	Bit	-

Table 48.41 A/D conversion characteristics (1) in high-speed A/D conversion mode (2 of 2)
Conditions: VCC = AVCC0 = 2.7 to 3.6 V, VREFH0 = 2.7 to 3.6 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
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 (Cin), see section 48.2.4, I/O V_{OH}, V_{OL}, and Other Characteristics.

Table 48.42 A/D conversion characteristics (2) in high-speed A/D conversion mode (1 of 2)
Conditions: VCC = AVCC0 = 2.4 to 3.6 V, VREFH0 = 2.4 to 3.6 V
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
DNL differential nonlinearity error		-	±1.0	-	LSB	-
INL integral nonlinearity error		-	±1.0	±3.0	LSB	-
14-bit mode						

Table 48.42 A/D conversion characteristics (2) in high-speed A/D conversion mode (2 of 2)

Conditions: VCC = AVCC0 = 2.4 to 3.6 V, VREFH0 = 2.4 to 3.6 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
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 (Cin), see section 48.2.4, I/O V_{OH}, V_{OL}, and Other Characteristics.

Table 48.43 A/D conversion characteristics (3) in low power A/D conversion mode (1 of 2)

Conditions: VCC = AVCC0 = 2.7 to 3.6 V, VREFH0 = 2.7 to 3.6 V
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	-
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	-

Table 48.43 A/D conversion characteristics (3) in low power A/D conversion mode (2 of 2)
Conditions: VCC = AVCC0 = 2.7 to 3.6 V, VREFH0 = 2.7 to 3.6 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
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 (Cin), see section 48.2.4, I/O V_{OH}, V_{OL}, and Other Characteristics.

Table 48.44 A/D conversion characteristics (4) in low power A/D conversion mode (1 of 2)
Conditions: VCC = AVCC0 = 2.4 to 3.6 V, VREFH0 = 2.4 to 3.6 V
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 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 = 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
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 48.44 A/D conversion characteristics (4) in low power A/D conversion mode (2 of 2)
Conditions: VCC = AVCC0 = 2.4 to 3.6 V, VREFH0 = 2.4 to 3.6 V
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 = 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 (Cin), see section 48.2.4, I/O V_{OH}, V_{OL}, and Other Characteristics.

Table 48.45 A/D conversion characteristics (5) in low power A/D conversion mode (1 of 2)
Conditions: VCC = AVCC0 = 1.8 to 3.6 V (AVCC0 = VCC when VCC < 2.0 V), VREFH0 = 1.8 to 3.6 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
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

Table 48.45 A/D conversion characteristics (5) in low power A/D conversion mode (2 of 2)
Conditions: VCC = AVCC0 = 1.8 to 3.6 V (AVCC0 = VCC when VCC < 2.0 V), VREFH0 = 1.8 to 3.6 V
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 = 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 (Cin), see section 48.2.4, I/O V_{OH}, V_{OL}, and Other Characteristics.

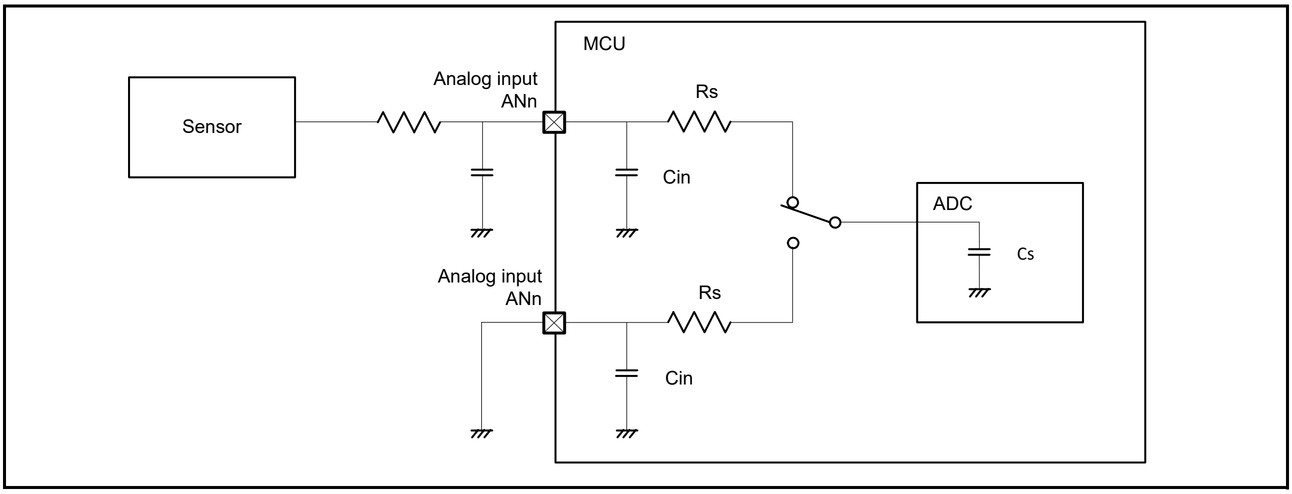


Figure 48.64 Equivalent circuit for analog input

Table 48.46 14-bit A/D converter channel classification (1 of 2)

Classification	Channel	Conditions	Remarks
High-precision channel	AN004 to AN006, AN009, AN010	AVCC0 = 1.8 to 3.6 V	Pins AN004 to AN006, AN009 and AN010 cannot be used as general I/O, IRQ3 inputs, and TS transmission, when the A/D converter is in use
Normal-precision channel	AN017, AN019, AN020		
Internal reference voltage input channel	Internal reference voltage	AVCC0 = 2.0 to 3.6 V	-

Table 48.46 14-bit A/D converter channel classification (2 of 2)

Classification	Channel	Conditions	Remarks
Temperature sensor input channel	Temperature sensor output	AVCC0 = 2.0 to 3.6 V	-

Table 48.47 A/D internal reference voltage characteristics

Conditions: VCC = AVCC0 = VREFH0 = 2.0 to 3.6 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 a 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.

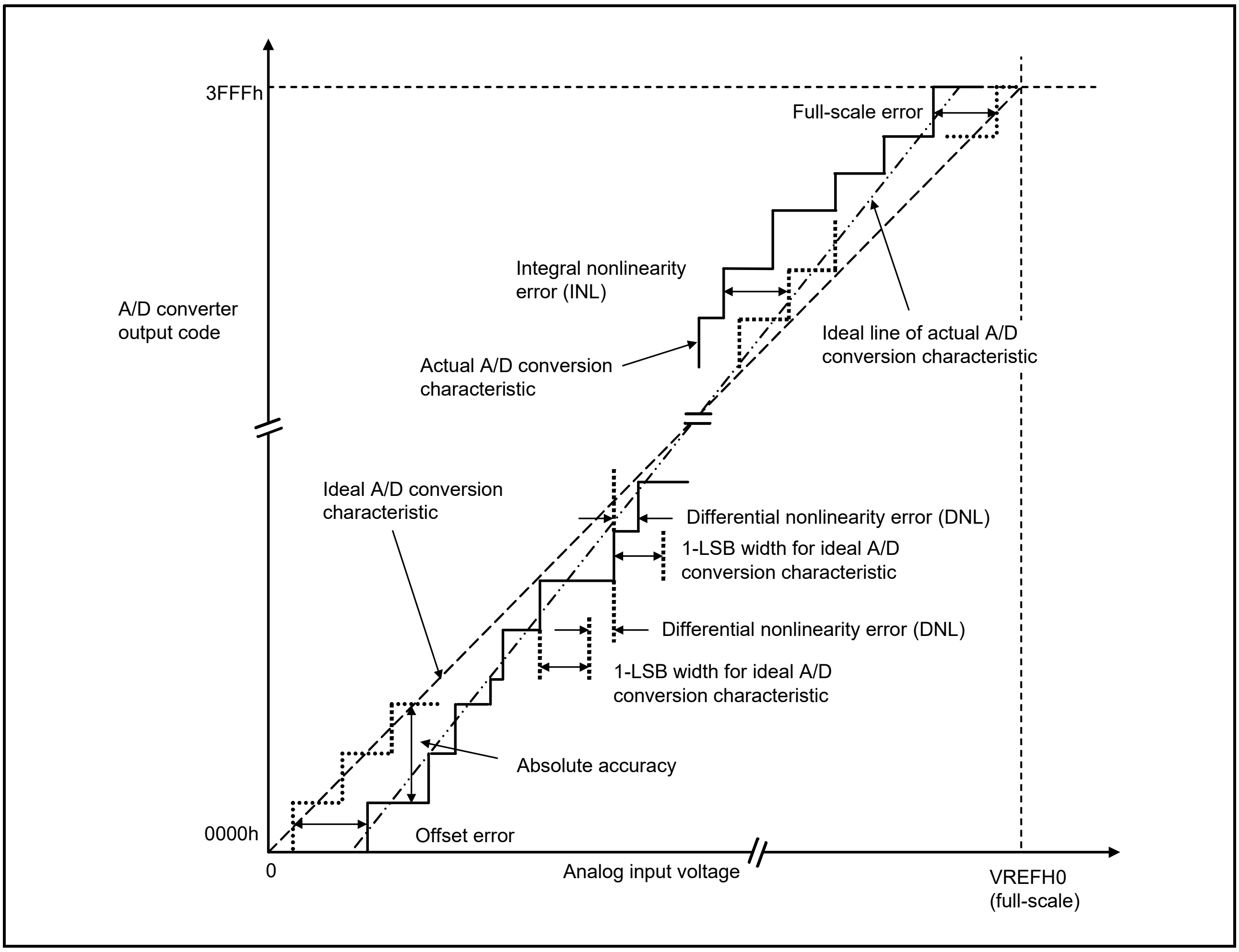

3FFFh
0000h
0
A/D converter output code
Analog input voltage
VREFH0 (full-scale)
Full-scale error
Integral nonlinearity error (INL)
Ideal line of actual A/D conversion characteristic
Actual A/D conversion characteristic
Ideal A/D conversion characteristic
Differential nonlinearity error (DNL)
1-LSB width for ideal A/D conversion characteristic
Differential nonlinearity error (DNL)
1-LSB width for ideal A/D conversion characteristic
Absolute accuracy
Offset error

Figure 48.65 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, and 1.5 mV are used as the analog input voltages. If analog input voltage is 6 mV, an absolute accuracy of ±5 LSB means that the actual A/D conversion result is in the range of 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.

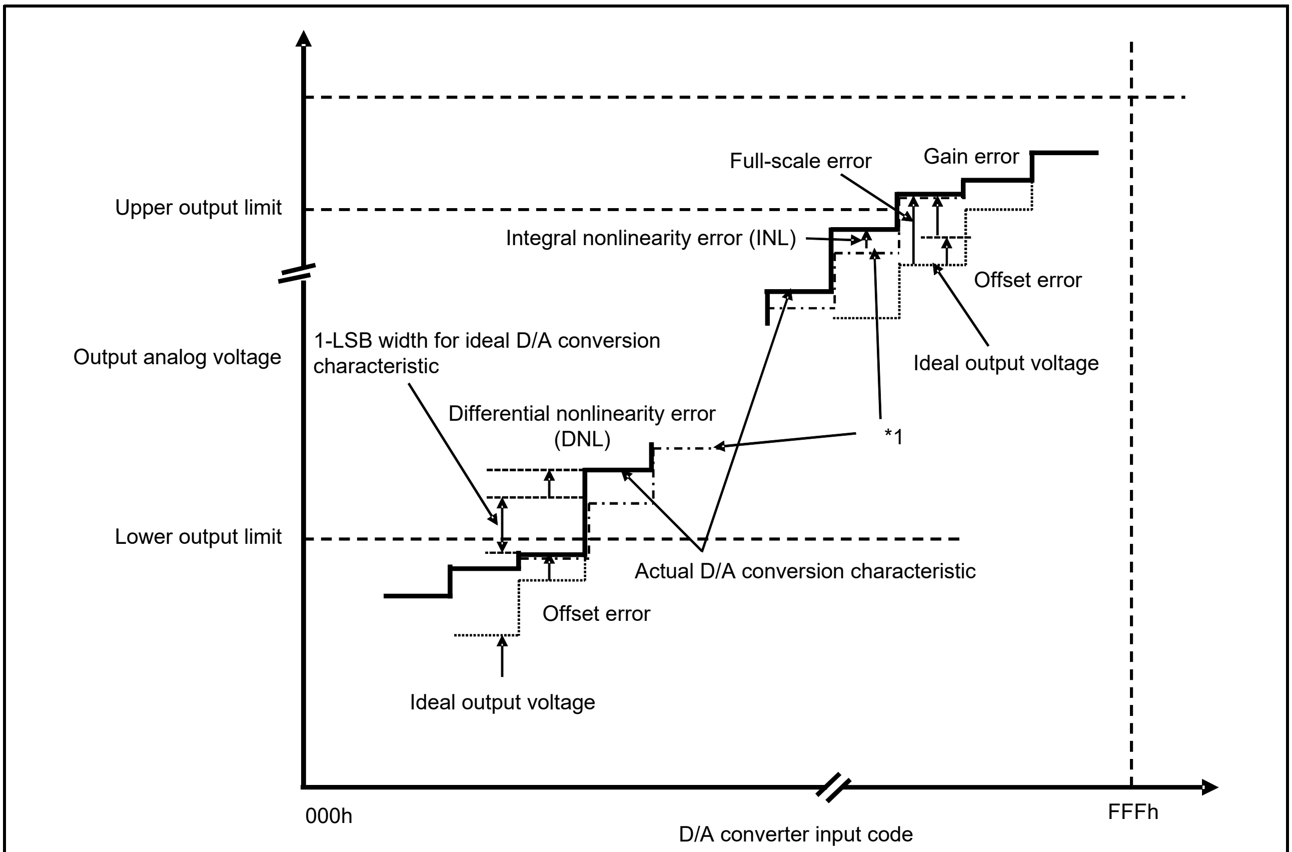
48.6 DAC12 Characteristics

Table 48.48 D/A conversion characteristics (1)
Conditions: VCC = AVCC0 = 1.8 to 3.6 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 48.49 D/A conversion characteristics (2)
Conditions: VCC = AVCC0 = 1.8 to 3.6 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	-


Full-scale error
Gain error
Upper output limit
Integral nonlinearity error (INL)
Offset error
Output analog voltage
1-LSB width for ideal D/A conversion characteristic
Ideal output voltage
Differential nonlinearity error (DNL)
*1
Lower output limit
Actual D/A conversion characteristic
Offset error
Ideal output voltage
000h
FFFh
D/A converter input code

Note 1. Ideal D/A conversion output voltage that is adjusted so that offset and full scale errors are zeroed.

Figure 48.66 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.

48.7 TSN Characteristics

Table 48.50 TSN characteristics

Conditions: VCC = AVCC0 = 2.0 to 3.6 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	-

48.8 OSC Stop Detect Characteristics

Table 48.51 Oscillation stop detection circuit characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Detection time	t_{dr}	-	-	1	ms	Figure 48.67

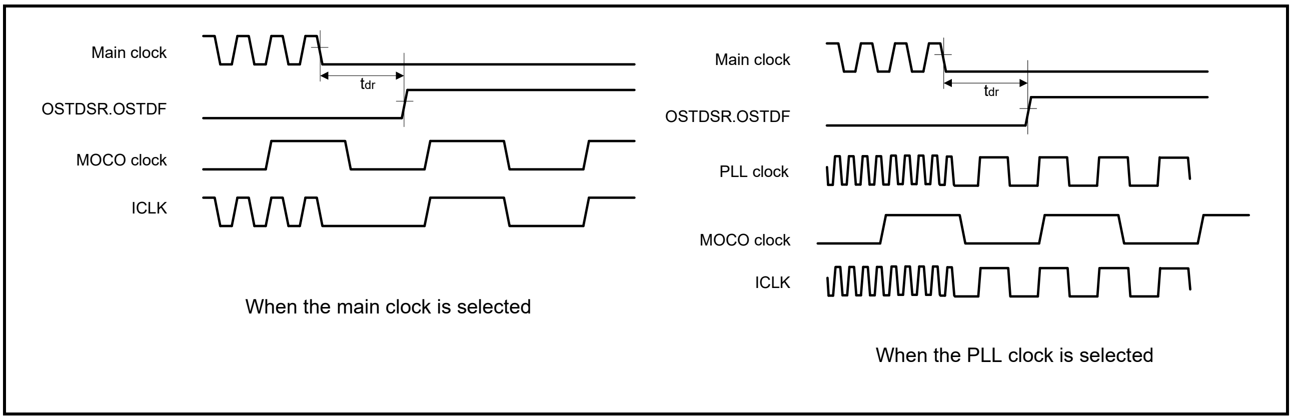

Main clock
OSTDSR.OSTDF
MOCO clock
ICLK
tdr
When the main clock is selected
Main clock
OSTDSR.OSTDF
PLL clock
MOCO clock
ICLK
tdr
When the PLL clock is selected

Figure 48.67 Oscillation stop detection timing

48.9 POR and LVD Characteristics

Table 48.52 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 48.68, Figure 48.69
	Voltage detection circuit (LVD0)*2	V_{det0_1}	2.68	2.85	2.96	V	Figure 48.70 At falling edge VCC
		V_{det0_2}	2.38	2.53	2.64		
		V_{det0_3}	1.78	1.90	2.02		
	Voltage detection circuit (LVD1)*3	V_{det1_4}	2.98	3.10	3.22	V	Figure 48.71 At falling edge VCC
		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		

Note 1. These characteristics apply when noise is not superimposed on the power supply.
Note 2. # in the symbol Vdet0_# denotes the value of the OFS1.VDSEL1[2:0] bits.
Note 3. # in the symbol Vdet1_# denotes the value of the LVDLVLR.LVD1LVL[4:0] bits.

Table 48.53 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 reset cancellation	LVD0:enable*1	$t_{LVD0,1}$	-	0.6	-	ms	-
	LVD0:disable*2	t_{LVD1}	-	0.2	-	ms	-
Response delay*3		t_{det}	-	-	350	μs	Figure 48.68, Figure 48.69
Minimum VCC down time		t_{VOFF}	450	-	-	μs	Figure 48.68, VCC = 1.0 V or above
Power-on reset enable time		$t_{W\ (POR)}$	1	-	-	ms	Figure 48.69, VCC = below 1.0 V
LVD operation stabilization time (after LVD is enabled)		$t_{d\ (E\text{-}A)}$	-	-	300	μs	Figure 48.71
Hysteresis width (POR)		V_{PORH}	-	110	-	mV	-
Hysteresis width (LVD0 and LVD1)		V_{LVH}	-	60	-	mV	LVD0 selected
			-	60	-		V_{det1_4} to V_{det1_9} selected
			-	50	-		V_{det1_A} or V_{det1_B} selected
			-	40	-		V_{det1_C} or V_{det1_F} 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} and V_{det1} for the POR/LVD.

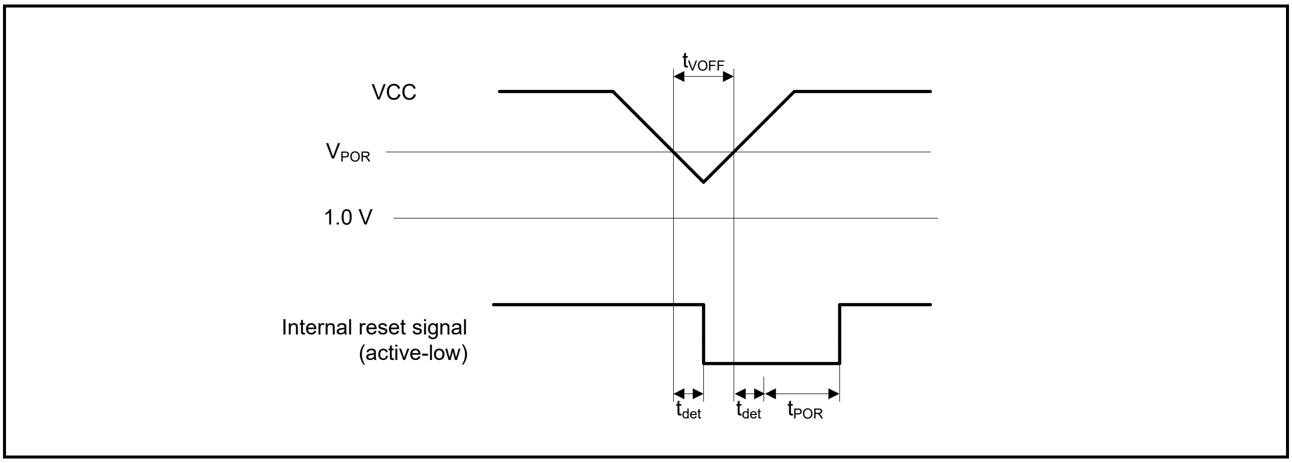
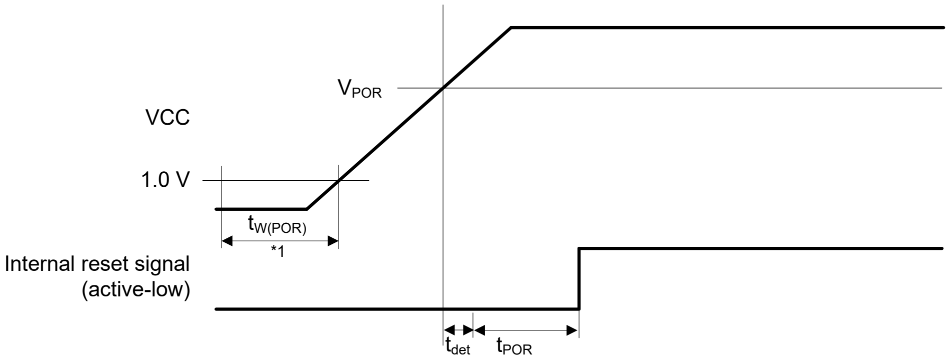


Figure 48.68 Voltage detection reset timing


V_{POR}
VCC
1.0 V
$t_{W(POR)}$
*1
Internal reset signal
(active-low)
t_{det}
t_{POR}

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 48.69 Power-on reset timing

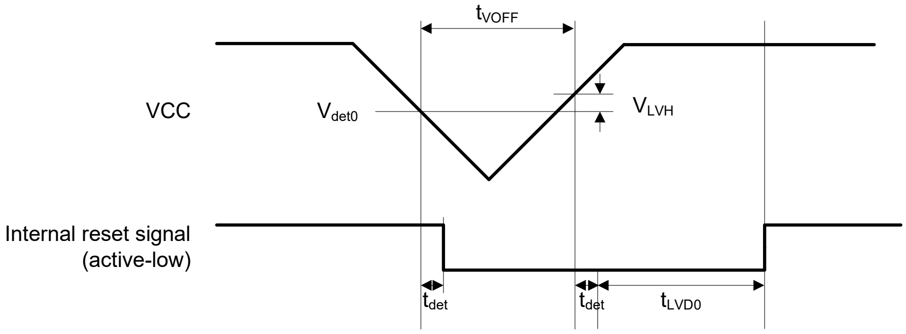

t_{VOFF}
VCC
V_{det0}
V_{LVH}
Internal reset signal
(active-low)
t_{det}
t_{det}
t_{LVD0}

Figure 48.70 Voltage detection circuit timing (V_{det0})

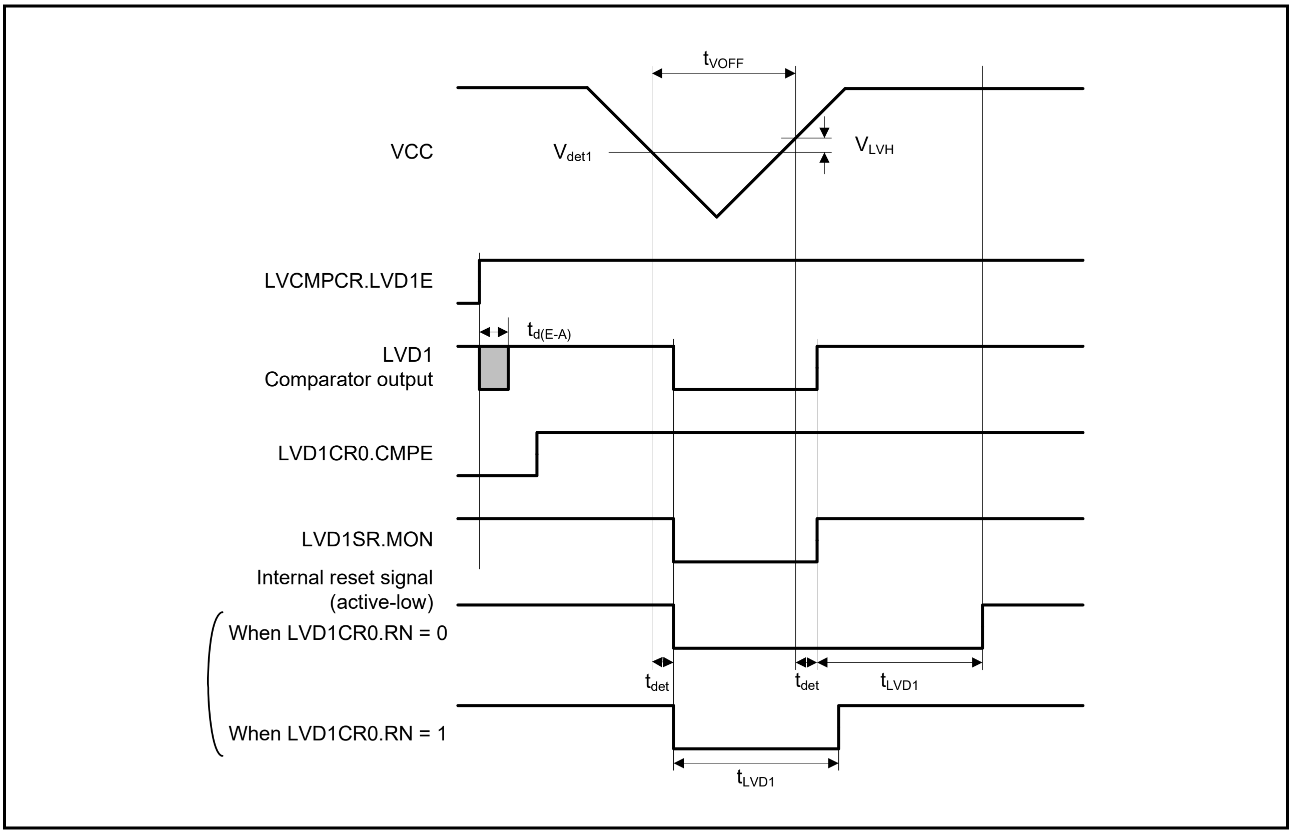

tVOFF
VCC
Vdet1
VLVH
LVCMPCR.LVD1E
td(E-A)
LVD1
Comparator output
LVD1CR0.CMPE
LVD1SR.MON
Internal reset signal
(active-low)
When LVD1CR0.RN = 0
tdet
tdet
tLVD1
When LVD1CR0.RN = 1
tLVD1

Figure 48.71 **Voltage detection circuit timing (V_{det1})**

48.10 VBATT Characteristics

Table 48.54 Battery backup function characteristics

Conditions: VCC = AVCC0 = 1.8V to 3.6V, VBATT = 1.6 to 3.6 V

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 48.72, Figure 48.73
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 48.72, Figure 48.73
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 48.74
	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 48.74
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 ($V_{DETBATT}$).

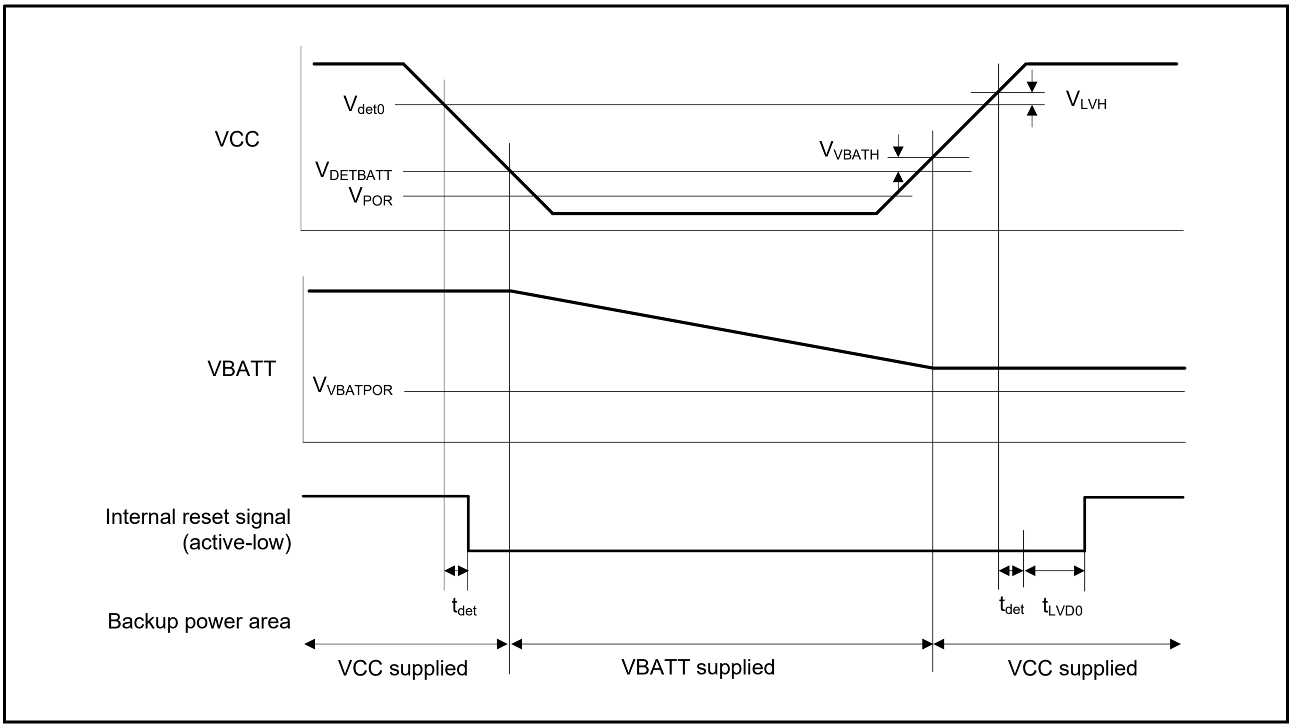

VCC
V_{det0}
V_{LVH}
V_{VBATH}
$V_{DETBATT}$
V_{POR}
VBATT
$V_{VBATPOR}$
Internal reset signal (active-low)
t_{det}
t_{det}
t_{LVD0}
Backup power area
VCC supplied
VBATT supplied
VCC supplied

Figure 48.72 Power supply switching and LVD0 reset timing

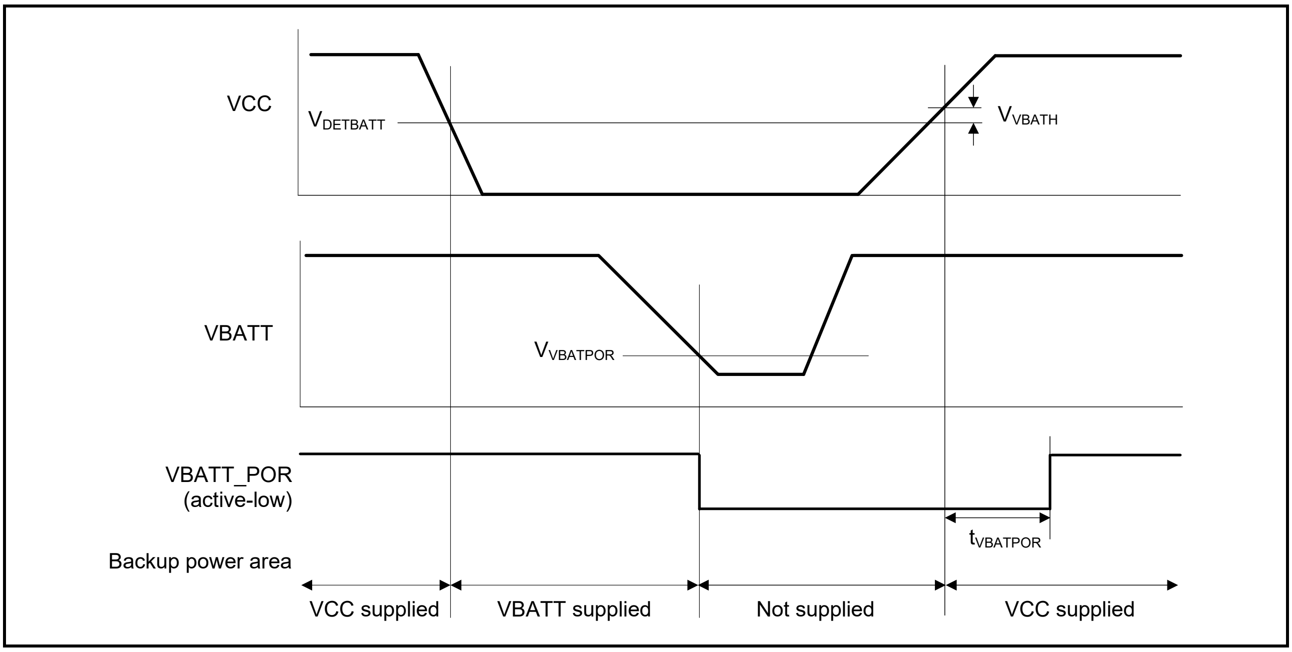

VCC
$V_{DETBATT}$
V_{VBATH}
VBATT
$V_{VBATPOR}$
VBATT_POR
(active-low)
$t_{VBATPOR}$
Backup power area
VCC supplied
VBATT supplied
Not supplied
VCC supplied

Figure 48.73 VBATT_POR reset timing

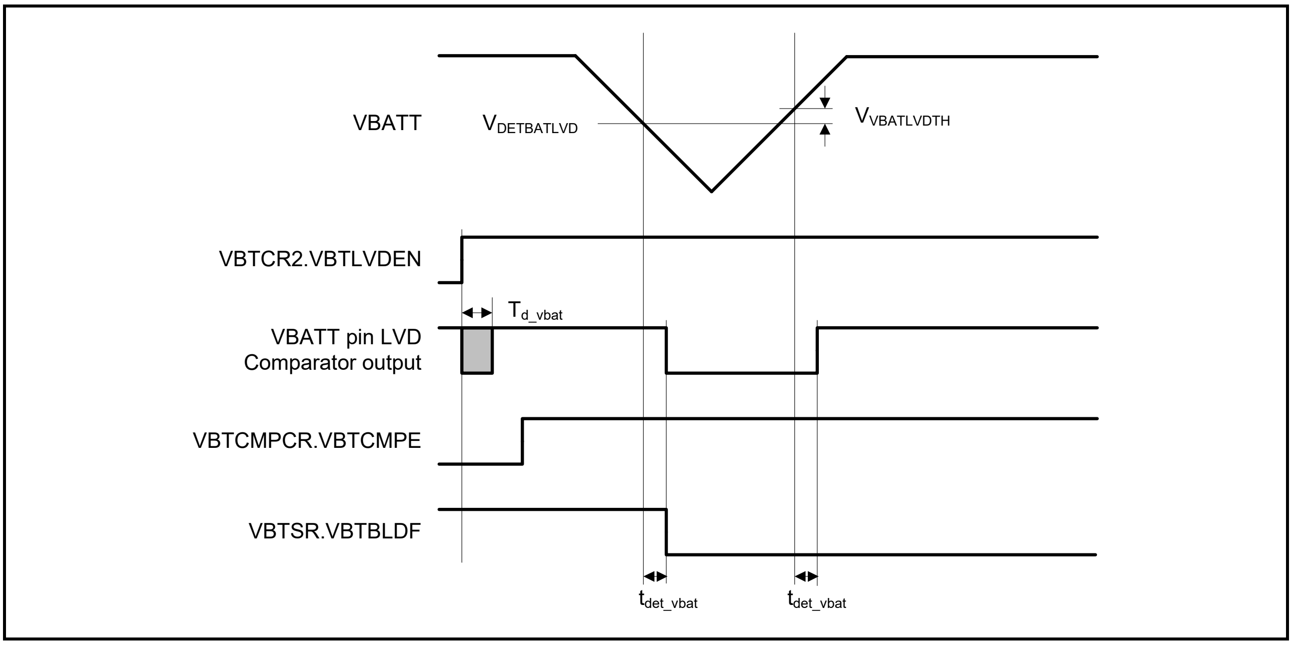

VBATT
$V_{DETBATLVD}$
$V_{VBATLVDTH}$
VBTCR2.VBTLVDEN
T_{d_vbat}
VBATT pin LVD
Comparator output
VBTCMPCR.VBTCMPE
VBTSR.VBTBLDF
t_{det_vbat}
t_{det_vbat}

Figure 48.74 VBATT pin voltage detection circuit timing

Table 48.55 VBATT-I/O characteristics

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
VBATWIOn I/O output characteristics (n = 0)	VCC > $V_{DETBATT}$	VCC = 2.7 to 3.6 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 = 1.8 to 2.7 V	V_{OH}	V_{BATT} - 0.3	-	-		I_{OH} = -50 µA
			V_{OL}	-	-	0.3		I_{OL} = 50 µA

48.11 CTSU Characteristics

Table 48.56 CTSU characteristics
Conditions: VCC = AVCC0 = 1.8 to 3.6 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

48.12 Segment LCD Controller Characteristics

48.12.1 Resistance Division Method

[Static Display Mode]

Table 48.57 Resistance division method LCD characteristics (1)
Conditions: VL4 ≤ VCC ≤ 3.6 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 48.58 Resistance division method LCD characteristics (2)
Conditions: VL4 ≤ VCC ≤ 3.6 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
LCD drive voltage	V_{L4}	2.7	-	VCC	V	-

[1/3 Bias Method]

Table 48.59 Resistance division method LCD characteristics (3)
Conditions: VL4 ≤ VCC ≤ 3.6 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
LCD drive voltage	V_{L4}	2.5	-	VCC	V	-

48.13 Comparator Characteristics

Table 48.60 ACMPLP characteristics
Conditions: VCC = 1.8 to 3.6 V

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Reference voltage range	Standard mode	IVREFn (n=0,1)	VREF	0	-	VCC–1.4	V	-
	Window mode*2	IVREF1	VREFH	1.4	-	VCC	V	-
		IVREF0	VREFL	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*1	High-speed mode		-	-	-	50	mV	-
	Low-speed mode		-	-	-	40	mV	-
	Window mode		-	-	-	60	mV	-
Operation stabilization wait time			T_{cmp}	100	-	-	μs	-

Note 1. When 8-bit DAC output is used as the reference voltage, the offset voltage increases up to 2.5 x VCC/256.
Note 2. In window mode, be sure to satisfy the following condition: IVREF1 - IVREF0 ≥ 0.2 V.

48.14 OPAMP Characteristics

Table 48.61 OPAMP characteristics

Conditions: VCC = AVCC0 = 1.8 to 3.6 V (AVCC0 = VCC when VCC < 2.0 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.

48.15 Flash Memory Characteristics

48.15.1 Code Flash Memory Characteristics

Table 48.62 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 erasures for each block. When the reprogram/erase cycle is n times (n = 1,000), erasing can be done 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 48.63 Code flash characteristics (2)

High-speed operating mode
Conditions: VCC = 2.7 to 3.6 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 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 48.64 Code flash characteristics (3)

Middle-speed operating mode

Conditions: VCC = 1.8 to 3.6 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 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.

48.15.2 Data Flash Memory Characteristics

Table 48.65 Data flash characteristics (1)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Reprogramming/erasure cycle*1		N_{DPEC}	100000	1000000	-	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 48.66 Data flash characteristics (2)

High-speed operating mode

Conditions: VCC = 2.7 to 3.6 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: Does not include the time until each operation of the flash memory is started after instructions are executed by software.

Note: The lower-limit frequency of 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 48.67 Data flash characteristics (3)

Middle-speed operating mode

Conditions: VCC = 1.8 to 3.6 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: Does not include the time until each operation of the flash memory is started after instructions are executed by software.

Note: The lower-limit frequency of 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.

48.16 Joint Test Action Group (JTAG)

Table 48.68 JTAG (debug) characteristics (1)

Conditions: VCC = 2.4 to 3.6 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
TCK clock cycle time	t_{TCKcyc}	80	-	-	ns	Figure 48.75
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 48.76
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 48.69 JTAG (debug) characteristics (2)

Conditions: VCC = 1.8 to 2.4 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
TCK clock cycle time	t_{TCKcyc}	250	-	-	ns	Figure 48.75
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 48.76
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	

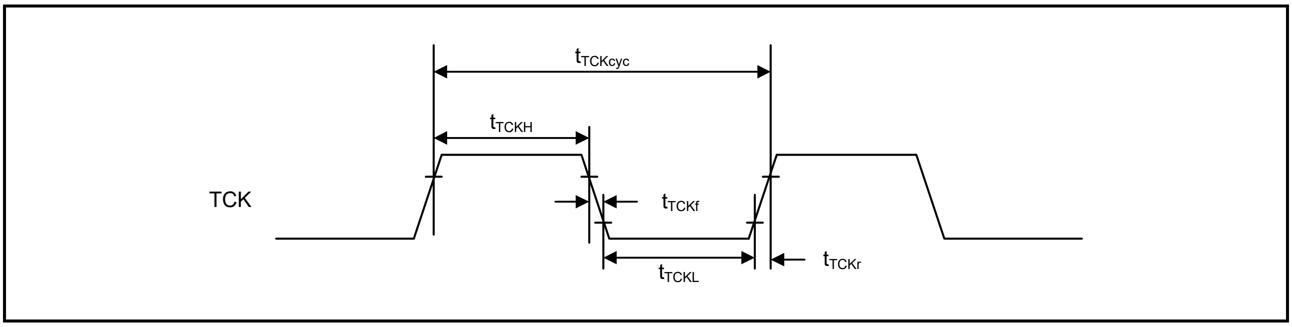

t_{TCKcyc}
t_{TCKH}
TCK
t_{TCKf}
t_{TCKL}
t_{TCKr}

Figure 48.75 JTAG TCK timing

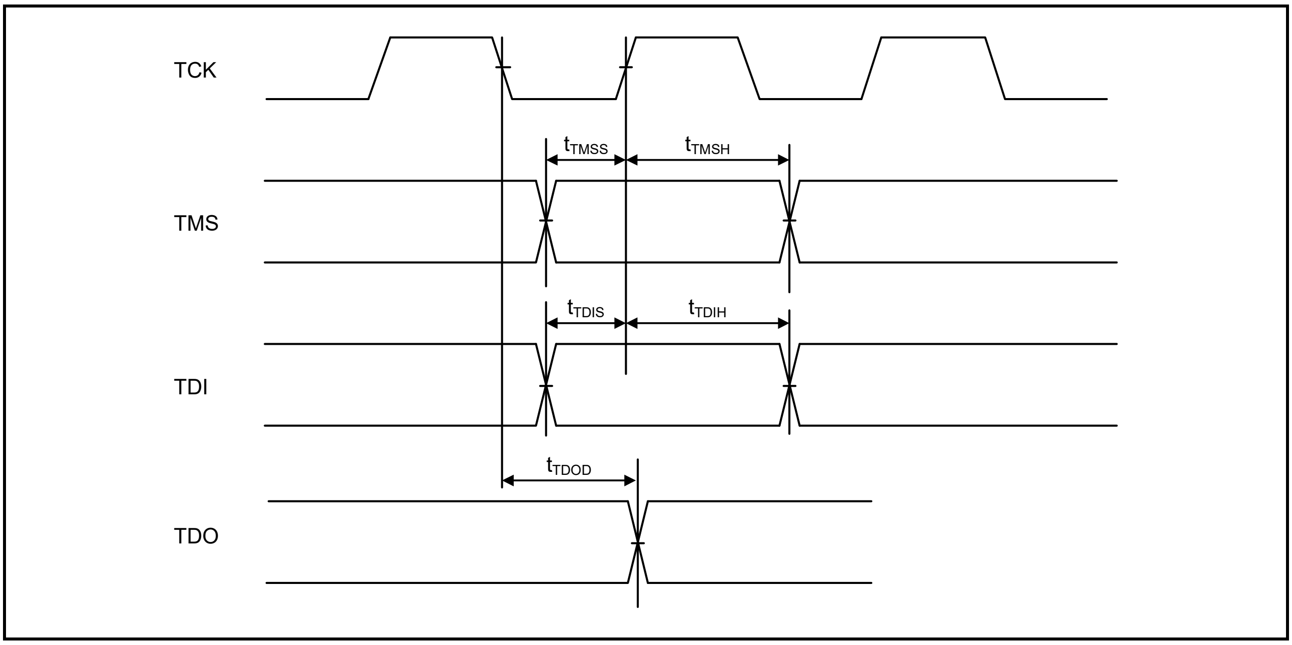

TCK
t_{TMSS}
t_{TMSH}
TMS
t_{TDIS}
t_{TDIH}
TDI
t_{TDOD}
TDO

Figure 48.76 JTAG input/output timing

48.16.1 Serial Wire Debug (SWD)

Table 48.70 SWD characteristics (1)

Conditions: VCC = 2.4 to 3.6 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
SWCLK clock cycle time	$t_{SWCKcyc}$	80	-	-	ns	Figure 48.77
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 48.78
SWDIO hold time	t_{SWDH}	16	-	-	ns	
SWDIO data delay time	t_{SWDD}	2	-	70	ns	

Table 48.71 SWD characteristics (2)

Conditions: VCC = 1.8 to 2.4 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
SWCLK clock cycle time	$t_{SWCKcyc}$	250	-	-	ns	Figure 48.77
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 48.78
SWDIO hold time	t_{SWDH}	50	-	-	ns	
SWDIO data delay time	t_{SWDD}	2	-	150	ns	

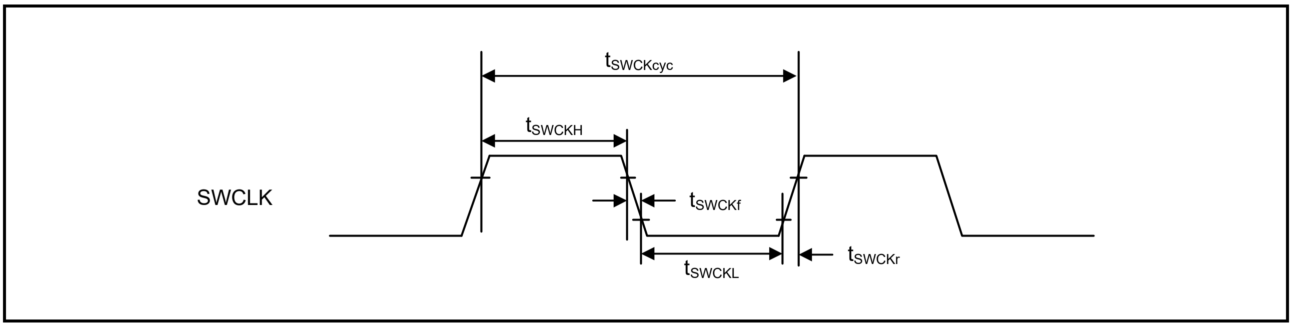


Figure 48.77 SWD SWCLK timing

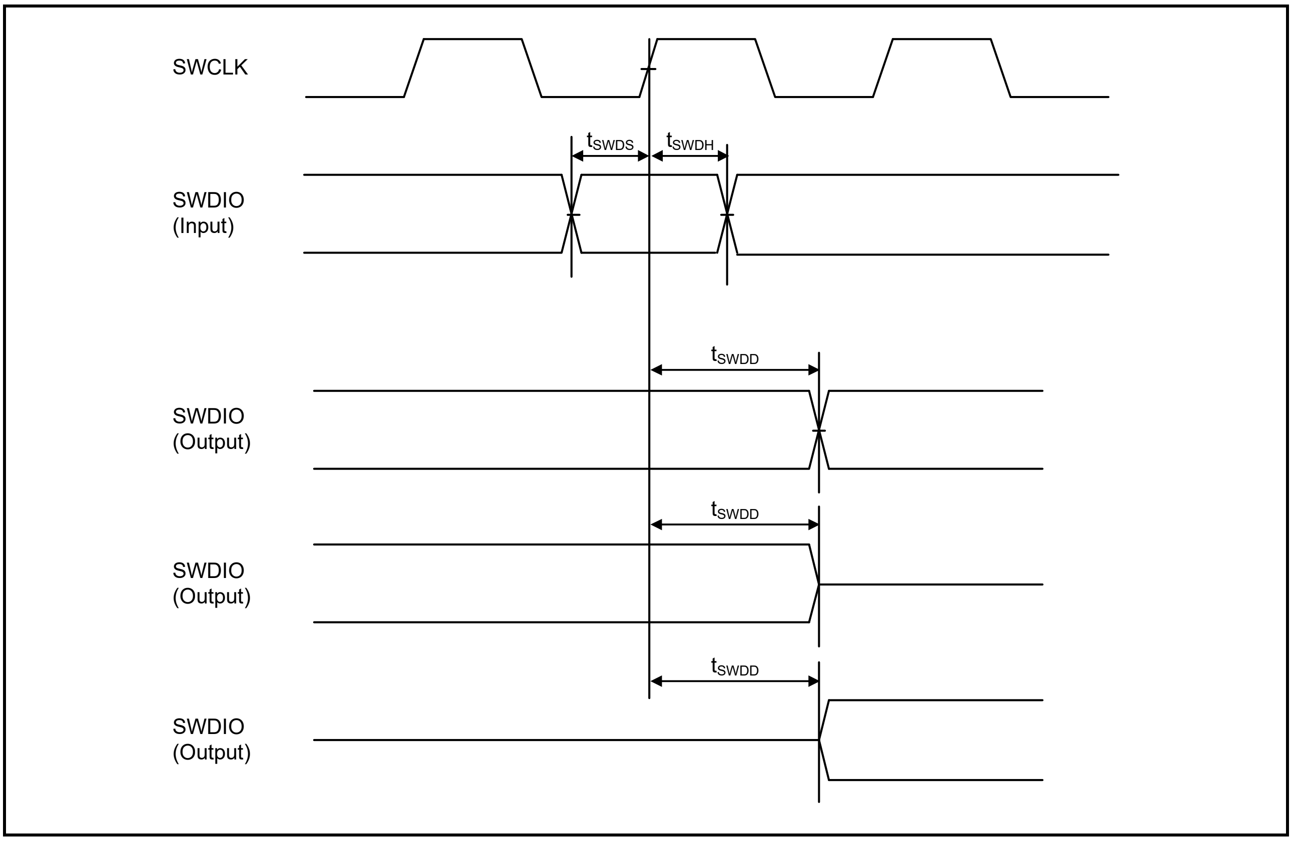


Figure 48.78 SWD input/output timing

48.17 BLE Characteristics

48.17.1 Transmission Characteristics

Table 48.72 Transmission Characteristics

Conditions: VCC = VCC_RF = AVCC_RF = 3.3 V, VSS = VSS_RF = 0 V, T_a = +25°C

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Range of frequency	RF_{CF}	2402	-	2480	MHz	
Data rate	RF_{DATA_2M}	-	2	-	Mbps	
	RF_{DATA_1M}	-	1	-	Mbps	
	RF_{DATA_500k}	-	500	-	kbps	
	RF_{DATA_125k}	-	125	-	kbps	
Maximum transmitted output power	RF_{POWER}	-	0	2	dBm	0 dBm output mode
		-	4	6	dBm	4 dBm output mode
Output frequency error	RF_{TXFERR}	–10	-	10	ppm	*1

Note: The characteristics are based on pins and functions other than those for the BLE interface not being in use.

Note 1. This does not take frequency errors due to manufacturing irregularities, drift with temperature, or deterioration of the crystal over time into account.

48.17.2 Reception Characteristics (2 Mbps)

Table 48.73 Reception Characteristics

Conditions:VCC = VCC_RF = AVCC_RF = 3.3 V, VSS = VSS_RF = 0 V, T$_a$ = +25°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Input frequency	RF$_{RXFIN\_2M}$	2402	—	2480	MHz		
Maximum input level	RF$_{LEVL\_2M}$	–10	4	—	dBm	*1	
Receiver sensitivity	RF$_{STY\_2M}$	—	–92	—	dBm	*1	
Secondary emission strength	RF$_{RXSP\_2M}$	—	–72	–57	dBm	30 MHz to 1 GHz	
		—	–54	–47	dBm	1 GHz to 12 GHz	
Co-channel rejection ratio	RF$_{CCR\_2M}$	—	–8	—	dB	Prf = –67 dBm*1	
Adjacent channel rejection ratio	RF$_{ADCR\_2M}$	—	2	—	dB	Prf = –67 dBm*1	±2 MHz
		—	35	—	dB		±4 MHz
		—	39	—	dB		±6 MHz
Blocking	RF$_{BLK\_2M}$	—	–1	—	dBm	Prf = –67 dBm*1	30 MHz to 2000 MHz
		—	–25	—	dBm		2000 MHz to 2399 MHz
		—	–21	—	dBm		2484 MHz to 3000 MHz
		—	–10	—	dBm		> 3000 MHz
Allowable frequency deviation*2	RF$_{RXFER\_2M}$	–120	—	120	ppm	*1	
RSSI accuracy	RF$_{RSSIS\_2M}$	—	±4	—	dB	–70 dBm ≤ Prf ≤ –10 dBm	

Note: The characteristics are based on pins and functions other than those for the BLE interface not being in use.

Note 1. PER ≤ 30.8%, and a 37-byte payload

Note 2. Allowable range of difference between the center frequency for the RF input signals and the carrier frequency generated within the chip

48.17.3 Reception Characteristics (1 Mbps)

Table 48.74 Reception Characteristics

Conditions:VCC = VCC_RF = AVCC_RF = 3.3 V, VSS = VSS_RF = 0 V, T$_a$ = +25°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Input frequency	RF$_{RXFIN\_1M}$	2402	—	2480	MHz		
Maximum input level	RF$_{LEVL\_1M}$	–10	4	—	dBm	*1	
Receiver sensitivity	RF$_{STY\_1M}$	—	–95	—	dBm	*1	
Secondary emission strength	RF$_{RXSP\_1M}$	—	–72	–57	dBm	30MHz to 1GHz	
		—	–54	–47	dBm	1GHz to 12GHz	
Co-channel rejection ratio	RF$_{CCR\_1M}$	—	–7	—	dB	Prf = –67dBm*1	
Adjacent channel rejection ratio	RF$_{ADCR\_1M}$	—	–1	—	dB	Prf = –67dBm*1	±1MHz
		—	34	—	dB		±2MHz
		—	35	—	dB		±3MHz
Blocking	RF$_{BLK\_1M}$	—	0	—	dBm	Prf = –67dBm*1	30MHz to 2000MHz
		—	–24	—	dBm		2000MHz to 2399MHz
		—	–20	—	dBm		2484MHz to 3000MHz
		—	–4	—	dBm		> 3000MHz
Allowable frequency deviation*2	RF$_{RXFER\_1M}$	–120	—	120	ppm	*1	
RSSI accuracy	RF$_{RSSIS\_1M}$	—	±4	—	dB	–70dBm ≤ Prf ≤ –10dBm	

Note: The characteristics are based on pins and functions other than those for the BLE interface not being in use.

Note 1. PER ≤ 30.8%, and a 37-byte payload

Note 2. Allowable range of difference between the center frequency for the RF input signals and the carrier frequency generated within the chip

48.17.4 Reception Characteristics (500 kbps)

Table 48.75 Reception Characteristics
Conditions:VCC = VCC_RF = AVCC_RF = 3.3 V, VSS = VSS_RF = 0 V, T_a = +25°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Input frequency	RF_{RXFIN_500k}	2402	—	2480	MHz		
Maximum input level	RF_{LEVL_500k}	–10	4	—	dBm	*1	
Receiver sensitivity	RF_{STY_500k}	—	–100	—	dBm	*1	
Secondary emission strength	RF_{RXSP_500k}	—	–72	–57	dBm	30MHz to 1GHz	
		—	–54	–47	dBm	1GHz to 12GHz	
Co-channel rejection ratio	RF_{CCR_500k}	—	–4	—	dB	Prf = –72dBm*1	
Adjacent channel rejection ratio	RF_{ADCR_500k}	—	6	—	dB	Prf = –72dBm*1	±1MHz
		—	36	—	dB		±2MHz
		—	42	—	dB		±3MHz
Blocking	RF_{BLK_500k}	—	0	—	dBm	Prf = –72dBm*1	30MHz to 2000MHz
		—	–23	—	dBm		2000MHz to 2399MHz
		—	–20	—	dBm		2484MHz to 3000MHz
		—	–7	—	dBm		> 3000MHz
Allowable frequency deviation*2	RF_{RXFER_500k}	–120	—	120	ppm	*1	
RSSI accuracy	RF_{RSSIS_500k}	—	±4	—	dB	–70dBm ≤ Prf ≤ –10dBm	

Note: The characteristics are based on pins and functions other than those for the BLE interface not being in use.
Note 1. PER ≤ 30.8%, and a 37-byte payload
Note 2. Allowable range of difference between the center frequency for the RF input signals and the carrier frequency generated within the chip

48.17.5 Reception Characteristics (125 kbps)

Table 48.76 Reception Characteristics
Conditions:VCC = VCC_RF = AVCC_RF = 3.3 V, VSS = VSS_RF = 0 V, T_a = +25°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Input frequency	RF_{RXFIN_125k}	2402	—	2480	MHz		
Maximum input level	RF_{LEVL_125k}	–10	4	—	dBm	*1	
Receiver sensitivity	RF_{STY_125k}	—	–105	—	dBm	*1	
Secondary emission strength	RF_{RXSP_125k}	—	–72	–57	dBm	30 MHz to 1 GHz	
		—	–54	–47	dBm	1 GHz to 12 GHz	
Co-channel rejection ratio	RF_{CCR_125k}	—	–2	—	dB	Prf = –79 dBm*1	
Adjacent channel rejection ratio	RF_{ADCR_125k}	—	12	—	dB	Prf = –79 dBm*1	±1 MHz
		—	39	—	dB		±2 MHz
		—	45	—	dB		±3 MHz
Blocking	RF_{BLK_125k}	—	0	—	dBm	Prf = –79 dBm*1	30 MHz to 2000 MHz
		—	–23	—	dBm		2000 MHz to 2399 MHz
		—	–20	—	dBm		2484 MHz to 3000 MHz
		—	–1	—	dBm		> 3000MHz
Allowable frequency deviation*2	RF_{RXFER_125k}	–120	—	120	ppm	*1	
RSSI accuracy	RF_{RSSIS_125k}	—	±4	—	dB	T_a = +25°C, –70 dBm ≤ Prf ≤ –10 dBm	

Note: The characteristics are based on pins and functions other than those for the BLE interface not being in use.
Note 1. PER ≤ 30.8%, and a 37-byte payload
Note 2. Allowable range of difference between the center frequency for the RF input signals and the carrier frequency generated within the chip

Appendix 1. Port States in each Processing Mode

Table 1.1 Port states in each processing mode (1 of 2)

Port name	Reset	Software Standby Mode	
		OPE = 0	OPE = 1
P004/IRQ3	Hi-Z	Keep-O*1	
P010/IRQ14	Hi-Z	Keep-O*1	
P011/IRQ15	Hi-Z	Keep-O*1	
P014/DA0	Hi-Z	[DA0 output (DAOE0 = 1)] DA output retained [All other (DAOE0 = 0)] Keep-O	
P015/IRQ7	Hi-Z	Keep-O*1	
P100/RXD0/CMPIN0/KR00/IRQ2/AGTIO0	Hi-Z	[AGTIO0 selected] AGTIO0 output*2 [All other] Keep-O*1	
P101/CMPREF0/KR01/IRQ1	Hi-Z	[All other] Keep-O*1	
P102/CMPIN1/KR02/AGTO0	Hi-Z	[AGTO0 selected] AGTO0 output*2 [All other] Keep-O*1	
P103/CMPREF1/KR03	Hi-Z	[All other] Keep-O*1	
P104/RXD0/KR04/IRQ1	Hi-Z	[All other] Keep-O*1	
P105/KR05/IRQ0	Hi-Z	[All other] Keep-O*1	
P106/KR06	Hi-Z	[All other] Keep-O*1	
P107/KR07	Hi-Z	[All other] Keep-O*1	
P108/TMS	Pull-up	Keep-O	
P109/TDO/CLKOUT	TDO output	[CLKOUT selected] CLKOUT output [All other] Keep-O	
P110/IRQ3/TDI/VCOUT	Pull-up	[ACMPLP selected] VCOUT output [All other] Keep-O*1	
P200/NMI	Hi-Z	Hi-Z	
P201	Pull-up	Keep-O	
P204/SCL0/USB_OVRCURB/AGTIO1	Hi-Z	[AGTIO1 selected] AGTIO1 output*2 [All other] Keep-O*1	
P205/USB_OVRCURA/IRQ1/CLKOUT/AGTO1	Hi-Z	[CLKOUT selected] CLKOUT output [AGTO1 selected] AGTO1 output*2 [All other] 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 [All other] Hi-Z	
P215/XCIN	Hi-Z	[Sub-clock oscillator selected] Sub-clock oscillator is operating [All other] Hi-Z	

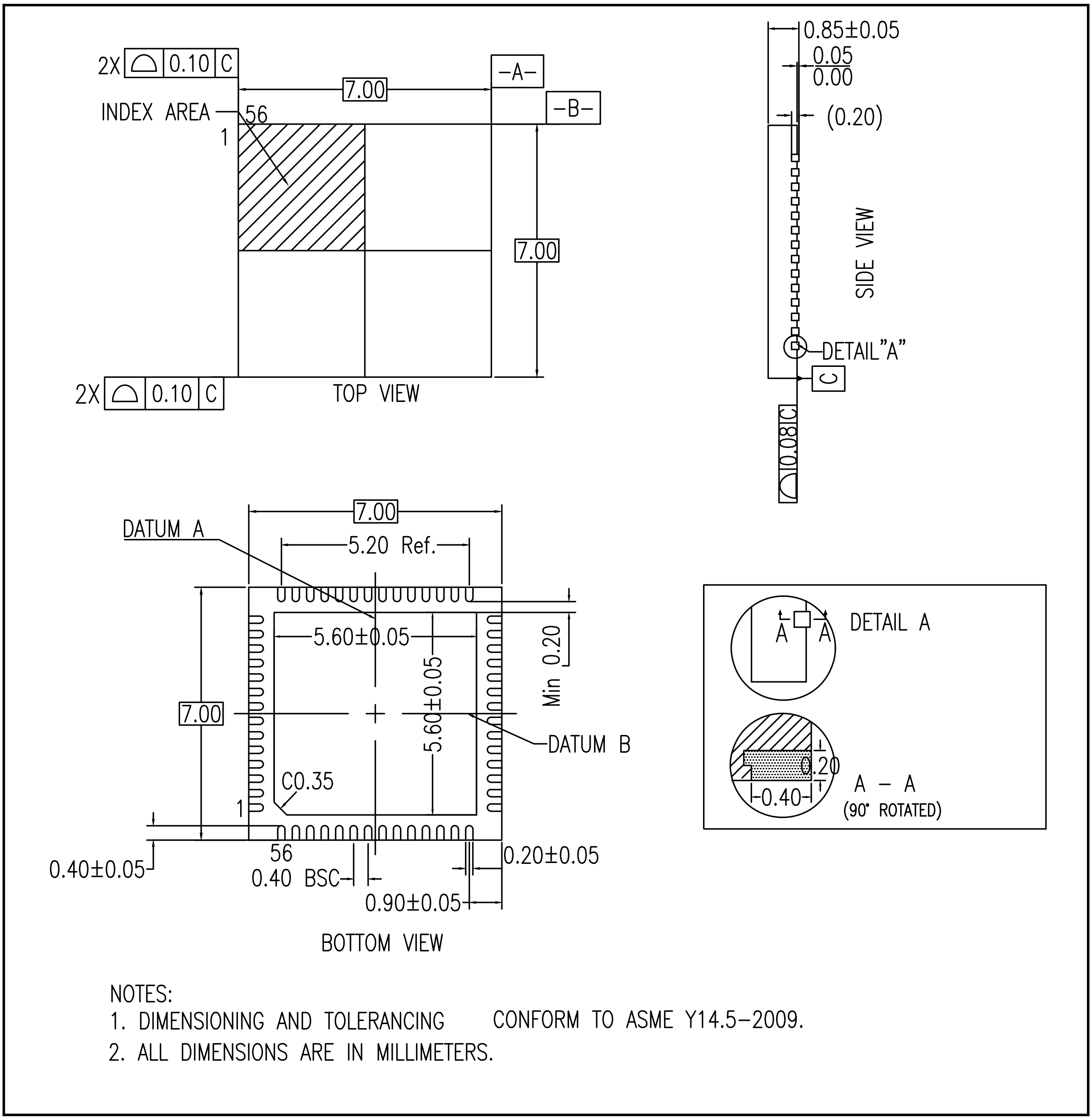
Table 1.1 Port states in each processing mode (2 of 2)

Port name	Reset	Software Standby Mode	
		OPE = 0	OPE = 1
P300/TCK	Pull-up	Keep-O	
P402/RTCIC0/IRQ4	Hi-Z	[All other] Keep-O*1	
P404	Hi-Z	Keep-O*1	
P407/SDA0/USB_VBUS/RTCOUT/AGTIO0	Hi-Z	[RTCOUT selected] RTCOUT output [AGTIO0 selected] AGTIO0 output*2 [All other] Keep-O*1	
P409/IRQ6	Hi-Z	Keep-O*1	
P414/IRQ9	Hi-Z	Keep-O*1	
P501/CMPIN1/USB_OVRCURA/IRQ11/ AGTOB0	Hi-Z	[AGTOB0 selected] AGTOB0 output*2 [All other] Keep-O*1	
P914/USB_DP	Hi-Z	Keep-O	
P915/USB_DM	Hi-Z	Keep-O	

H: High-level
L: Low-level
Hi-Z: High-impedance
Keep-O: Output pins retain their previous values. Input pins become high-impedance.

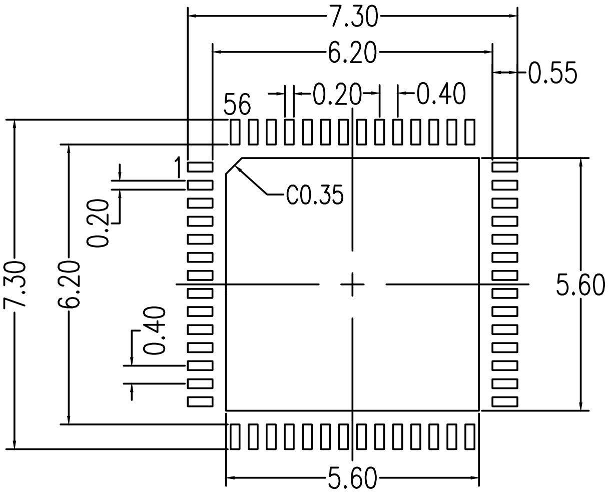
Note: Retains LCD output when the LCD controller/driver pin functions (COM0 to COM3 and SEG6, SEG9, SEG11, SEG12, SEG20, SEG23, SEG49, SEG52, SEG51) are set and LOCO or SOSC is selected in the SLCDSCKCR.LCDSCKSEL[2:0] bit.
Note 1. Input is enabled if the pin is specified as the software standby canceling 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


2X ⌓ 0.10 C
INDEX AREA
56
1
7.00
−A−
−B−
7.00
2X ⌓ 0.10 C
TOP VIEW
0.85±0.05
0.05
0.00
(0.20)
SIDE VIEW
DETAIL"A"
C
⏥ 0.08 C
DATUM A
7.00
5.20 Ref.
5.60±0.05
5.60±0.05
Min 0.20
7.00
DATUM B
C0.35
1
56
0.40±0.05
0.40 BSC
0.20±0.05
0.90±0.05
BOTTOM VIEW
DETAIL A
A
A
0.20
0.40
A − A
(90° ROTATED)

NOTES:
1. DIMENSIONING AND TOLERANCING CONFORM TO ASME Y14.5−2009.
2. ALL DIMENSIONS ARE IN MILLIMETERS.

Figure 2.1 QFN 56-pin


7.30
6.20
0.55
56
0.20
0.40
1
C0.35
0.20
7.30
6.20
5.60
0.40
5.60

RECOMMENDED LAND PATTERN DIMENSION

NOTES:
1. ALL DIMENSIONS ARE IN MM. ANGLES IN DEGREES.
2. TOP DOWN VIEW. AS VIEWED ON PCB.

Figure 2.2 **Land Pattern**

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 the 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
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
CAN0	CAN0 Module	0x40050000
IIC0	Inter-Integrated Circuit 0	0x40053000
IIC1	Inter-Integrated Circuit 1	0x40053100
DOC	Data Operation Circuit	0x40054100
ADC140	14-bit A/D Converter	0x4005C000
DAC12	12-bit D/A Converter	0x4005E000

Table 3.1 Peripheral base address (2 of 2)

Name	Description	Base address
SCI0	Serial Communication Interface 0	0x40070000
SCI1	Serial Communication Interface 1	0x40070020
SCI4	Serial Communication Interface 4	0x40070080
SCI9	Serial Communication Interface 9	0x40070120
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
GPT164	General PWM Timer 4 (16-bit)	0x40078400
GPT165	General PWM Timer 5 (16-bit)	0x40078500
GPT168	General PWM Timer 8 (16-bit)	0x40078800
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
ACMPLP	Low-Power Analog Comparator	0x40085E00
OPAMP	Operational Amplifier	0x40086000
USBFS	USB 2.0 Full-Speed Module	0x40090000
DAC8	8-bit D/A converter	0x4009E000
FLCN	Flash I/O Register	0x407EC000
TSN	Temperature Sensor	0x407EC000

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

Peripherals	Address		Number of access cycles					
			ICLK = PCLK		ICLK > PCLK*1		Cycle unit	Related function
	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 ~ 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 ~ 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
CAN0, IICn, DOC, ADC140, DAC12	4004 E000h	4005 EFFFh	3		2 ~ 3		PCLKB	Controller Area Network Module, I^{2}C Bus Interface, Data Operation Circuit, 14-Bit A/D Converter, 12-Bit D/A Converter
SCIn	4007 0000h	4007 0EFFh	5*2		2 ~ 3*2		PCLKA	Serial Communications Interface
SPIn	4007 2000h	4007 2FFFh	5*3		2 ~ 3*3		PCLKA	Serial Peripheral Interface
CRC	4007 4000h	4007 4FFFh	3		2 ~ 3		PCLKA	CRC Calculator
GPT32n, GPT_OPS	4007 8000h	4007 8FFFh	See Table 3.3*4				PCLKA	General PWM Timer
FLCN	407E C000h	407E CFFFh	7		7		ICLK	Flash I/O Register
TSN	407E C000h	407E CFFFh	7		7		ICLK	Temperature Sensor
KINT, CTSU, SLCDC	4008 0000h	4008 1FFFh	2		1 ~ 2		PCLKB	Key Interrupt Function, Capacitive Touch Sensing Unit, Segment LCD Controller
AGTn	4008 4000h	4008 4FFFh	3		2 ~ 3		PCLKB	Asynchronous General Purpose Timer
ACMPLP, OPAMP	4008 5000h	4008 6FFFh	2		1 ~ 2		PCLKB	Low-Power Analog Comparator, Operational Amplifier
USBFS	4009 0000h	4009 03FFh	4		3 ~ 4		PCLKB	USB 2.0 Full-Speed Module
USBFS	4009 0400h	4009 04FFh	3		2 ~ 3		PCLKB	USB 2.0 Full-Speed Module

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 ~ 2. 5 is 1 ~ 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.

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 ~ 6	3 ~ 4	PCLKA
ICLK > PCLKD > PCLKA	3 ~ 4	2 ~ 3	PCLKA
PCLKD = ICLK = PCLKA	6	4	PCLKA
PCLKD = ICLK > PCLKA	2 ~ 3	1 ~ 2	PCLKA
PCLKD > ICLK = PCLKA	4	3	PCLKA
PCLKD > ICLK > PCLKA	2 ~ 3	1 ~ 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 25)

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
				SMPUSRAM0	Access Control Register for SRAM0	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

Table 3.4 Register description (2 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
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
				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
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

Table 3.4 Register description (3 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
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
				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	Snooze 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
	32	0x4	0-31	IELSR%s	ICU Event Link Setting Register %s	0x300	32	read/ write	0x00000000	0xFFFFFFFF
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

Table 3.4 Register description (4 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SYSTEM	-	-	-	SBYCR	Standby Control Register	0x00C	16	read/ write	0x4000	0xFFFF
				MSTPCRA	Module Stop Control Register A	0x01C	32	read/ write	0xFFBFFFBE	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
				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
				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
				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

Table 3.4 Register description (5 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SYSTEM	-	-	-	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
				BKRACR	Backup Register Access Control Register	0x0C6	8	read/ write	0x06	0xFF
				USBCKCR	USB Clock Control register	0x0D0	8	read/ write	0x00	0xFF
	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
				VBTCMPCR	VBATT Comparator Control Register	0x4B2	8	read/ write	0x00	0xFF

Table 3.4 Register description (6 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SYSTEM	-	-	-	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
				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	0x0000
				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
				PIDR	Input Data Register	0x06	16	read-only	0x0000	0x0000
				PCNTR3	Port Control Register 3	0x08	32	write-only	0x00000000	0xFFFFFFFF
				PORR	Output Set Register	0x08	16	write-only	0x0000	0xFFFF
				POSR	Output Reset Register	0x0A	16	write-only	0x0000	0xFFFF
PORT1-4	-	-	-	PCNTR4	Port Control Register 4	0x0C	32	read/ write	0x00000000	0xFFFFFFFF
				EORR	Event Output Set Register	0x0C	16	read/ write	0x0000	0xFFFF
				EOSR	Event Output Reset Register	0x0E	16	read/ write	0x0000	0xFFFF

Table 3.4 Register description (7 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
PFS	-	-	-	P004PFS	P00%s Pin Function Control Register	0x004	32	read/ write	0x00000000	0xFFFFFFFD
	-	-	-	P004PFS_HA	P00%s Pin Function Control Register	0x006	16	read/ write	0x0000	0xFFFD
	-	-	-	P004PFS_BY	P00%s Pin Function Control Register	0x007	8	read/ write	0x00	0xFD
	4	0x4	10, 11, 14, 15	P0%sPFS	P0%s Pin Function Control Register	0x028	32	read/ write	0x00000000	0xFFFFFFFD
	4	0x4	10, 11, 14, 15	P0%sPFS_HA	P0%s Pin Function Control Register	0x02A	16	read/ write	0x0000	0xFFFD
	4	0x4	10, 11, 14, 15	P0%sPFS_BY	P0%s Pin Function Control Register	0x02B	8	read/ write	0x00	0xFD
	8	0x4	0-7	P10%sPFS	P10%s Pin Function Control Register	0x040	32	read/ write	0x00000000	0xFFFFFFFD
	8	0x4	0-7	P10%sPFS_HA	P10%s Pin Function Control Register	0x042	16	read/ write	0x0000	0xFFFD
	8	0x4	0-7	P10%sPFS_BY	P10%s Pin Function Control Register	0x043	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
				P110PFS_BY	P110 Pin Function Control Register	0x06B	8	read/ write	0x10	0xFD
				P111PFS	P1%s Pin Function Control Register	0x06C	32	read/ write	0x00000000	0xFFFFFFFD
				P111PFS_HA	P1%s Pin Function Control Register	0x06E	16	read/ write	0x0000	0xFFFD
				P111PFS_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

Table 3.4 Register description (8 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
PFS	-	-	-	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
	3	0x4	4-6	P20%sPFS	P20%s Pin Function Control Register	0x088	32	read/ write	0x00000000	0xFFFFFFFD
	3	0x4	4-6	P20%sPFS_HA	P20%s Pin Function Control Register	0x08A	16	read/ write	0x0000	0xFFFD
	3	0x4	4-6	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
	4	0x4	2, 4, 7	P40%sPFS	P40%s Pin Function Control Register	0x100	32	read/ write	0x00000000	0xFFFFFFFD
	4	0x4	2, 4, 7	P40%sPFS_HA	P40%s Pin Function Control Register	0x102	16	read/ write	0x0000	0xFFFD
	4	0x4	2, 4, 7	P40%sPFS_BY	P40%s Pin Function Control Register	0x103	8	read/ write	0x00	0xFD
	-	-	-	P409PFS	P409 Pin Function Control Register	0x124	32	read/ write	0x00000000	0xFFFFFFFD
				P409PFS_HA	P409 Pin Function Control Register	0x126	16	read/ write	0x0000	0xFFFD
				P409PFS_BY	P409 Pin Function Control Register	0x127	8	read/ write	0x00	0xFD
	-	-	-	P414PFS	P4%s Pin Function Control Register	0x128	32	read/ write	0x00000000	0xFFFFFFFD
				P414PFS_HA	P4%s Pin Function Control Register	0x12A	16	read/ write	0x0000	0xFFFD
				P414PFS_BY	P4%s Pin Function Control Register	0x12B	8	read/ write	0x00	0xFD
PFS	-	-	-	P501PFS	P50%s Pin Function Control Register	0x140	32	read/ write	0x00000000	0xFFFFFFFD
				P501PFS_HA	P50%s Pin Function Control Register	0x142	16	read/ write	0x0000	0xFFFD
				P501PFS_BY	P50%s Pin Function Control Register	0x143	8	read/ write	0x00	0xFD
PFS	2	0x4	14,15	P9%sPFS	P9%s Pin Function Control Register	0x278	32	read/ write	0x00010000	0xFFFFFFFD
	2	0x4	14,15	P9%sPFS_HA	P9%s Pin Function Control Register	0x27A	16	read/ write	0x0000	0xFFFD
	2	0x4	14,15	P9%sPFS_BY	P9%s Pin Function Control Register	0x27B	8	read/ write	0x00	0xFD
PMISC	-	-	-	PWPR	Write-Protect Register	0x03	8	read/ write	0x80	0xFF

Table 3.4 Register description (9 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
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
	-	-	-	ELSR12	Event Link Setting Register 12	0x40	16	read/ write	0x0000	0xFFFF
	5	0x4	14-18	ELSR%s	Event Link Setting Register %s	0x48	16	read/ write	0x0000	0xFFFF
POEG	2	0x100	A,B	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
				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

Table 3.4 Register description (10 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
RTC	-	-	-	RDAYAR	Date Alarm Register	0x18	8	read/ write	0x00	0x00
				BCNT0AER	Binary Counter 0 Alarm Enable Register	0x18	8	read/ write	0x00	0x00
				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
				IWDTRR	IWDT Refresh Register	0x00	8	read/ write	0xFF	0xFF

Table 3.4 Register description (11 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
WDT	-	-	-	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
				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
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
	32	0x10	0-31	MB%s_D7	Mailbox Register	0x20D	8	read/ write	0x00	0x00

Table 3.4 Register description (12 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
CAN0	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	I^{2}C Bus Control Register 1	0x00	8	read/ write	0x1F	0xFF
				ICCR2	I^{2}C Bus Control Register 2	0x01	8	read/ write	0x00	0xFF
				ICMR1	I^{2}C Bus Mode Register 1	0x02	8	read/ write	0x08	0xFF

Table 3.4 Register description (13 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
IIC0	-	-	-	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	I²C Bus Wake up Unit Register 2	0x17	8	read/ write	0xFD	0xFF
IIC1	-	-	-	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
	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

Table 3.4 Register description (14 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
IIC1	-	-	-	ICDRR	I^2C 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
				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

Table 3.4 Register description (15 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
ADC140	-	-	-	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
				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
	5	0x1	4-6, 9, 10	ADSSTR%s	A/D Sampling State Register %s	0x0E0	8	read/ write	0x0D	0xFF
DAC12	-	-	-	DADR0	D/A Data Register 0	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

Table 3.4 Register description (16 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
DAC12	-	-	-	DAVREFCR	D/A VREF Control Register	0x07	8	read/ write	0x00	0xFF
SCI0,1	-	-	-	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=0)	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^{2}C Mode Register 1	0x09	8	read/ write	0x00	0xFF
				SIMR2	I^{2}C Mode Register 2	0x0A	8	read/ write	0x00	0xFF
				SIMR3	I^{2}C Mode Register 3	0x0B	8	read/ write	0x00	0xFF
				SISR	I^{2}C Status Register	0x0C	8	read-only	0x00	0xCB
				SPMR	SPI Mode Register	0x0D	8	read/ write	0x00	0xFF
				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

Table 3.4 Register description (17 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SCI0,1	-	-	-	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
SCI4,9	-	-	-	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=0)	0x04	8	read/ write	0x84	0xFF
				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^{2}C Mode Register 1	0x09	8	read/ write	0x00	0xFF
				SIMR2	I^{2}C Mode Register 2	0x0A	8	read/ write	0x00	0xFF
				SIMR3	I^{2}C Mode Register 3	0x0B	8	read/ write	0x00	0xFF
				SISR	I^{2}C Status Register	0x0C	8	read-only	0x00	0xCB
				SPMR	SPI Mode Register	0x0D	8	read/ write	0x00	0xFF
				TDRHL	Transmit 9-bit Data Register	0x0E	16	read/ write	0xFFFF	0xFFFF
				RDRHL	Receive 9-bit Data Register	0x10	16	read-only	0x0000	0xFFFF
				MDDR	Modulation Duty Register	0x12	8	read/ write	0xFF	0xFF
				DCCR	Data Compare Match Control Register	0x13	8	read/ write	0x40	0xFF
				CDR	Compare Match Data Register	0x1A	16	read/ write	0x0000	0xFFFF
				SPTR	Serial Port Register	0x1C	8	read/ write	0x03	0xFF
SPI0	-	-	-	SPCR	SPI Control Register	0x00	8	read/ write	0x00	0xFF

Table 3.4 Register description (18 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SPI0	-	-	-	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
				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
SPI1	-	-	-	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
				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
				SPCR2	SPI Control Register 2	0x0F	8	read/ write	0x00	0xFF
				SPCMD0	SPI Command Register 0	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

Table 3.4 Register description (19 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
CRC	-	-	-	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
GPT320-3	-	-	-	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
				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

Table 3.4 Register description (20 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
GPT320-3	-	-	-	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
GPT164, 5, 8	-	-	-	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
				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

Table 3.4 Register description (21 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
GPT164, 5, 8	-	-	-	GTCCRA	General PWM Timer Compare Capture Register A	0x4C	32	read/ write	0x0000FFFF	0xFFFFFFFF
				GTCCRB	General PWM Timer Compare Capture Register B	0x50	32	read/ write	0x0000FFFF	0xFFFFFFFF
				GTCCRC	General PWM Timer Compare Capture Register C	0x54	32	read/ write	0x0000FFFF	0xFFFFFFFF
				GTCCRE	General PWM Timer Compare Capture Register E	0x58	32	read/ write	0x0000FFFF	0xFFFFFFFF
				GTCCRD	General PWM Timer Compare Capture Register D	0x5C	32	read/ write	0x0000FFFF	0xFFFFFFFF
				GTCCRF	General PWM Timer Compare Capture Register F	0x60	32	read/ write	0x0000FFFF	0xFFFFFFFF
				GTPR	General PWM Timer Cycle Setting Register	0x64	32	read/ write	0x0000FFFF	0xFFFFFFFF
				GTPBR	General PWM Timer Cycle Setting Buffer Register	0x68	32	read/ write	0x0000FFFF	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	0x0000FFFF	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
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

Table 3.4 Register description (22 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
CTSU	-	-	-	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
				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
	9	0x1	6, 9, 11, 12, 20, 23, 49, 52, 53	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

Table 3.4 Register description (23 of 25)

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
ACMPLP	-	-	-	COMPSEL0	Comparator Input Select Register	0x04	8	read/ write	0x11	0xFF
				COMPSEL1	Comparator Reference Voltage Select Register	0x05	8	read/ write	0x91	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

Table 3.4 Register description (24 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
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
				SOFCFG	SOF Output Configuration Register	0x03C	16	read/ write	0x0000	0xFFFF
				INTSTS0	Interrupt Status Register 0	0x040	16	read/ write	0x0000	0xFF7F

Table 3.4 Register description (25 of 25)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
USBFS	-	-	-	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
DAC8	2	0x01	0,1	DACS%s	D/A Conversion Value Setting Register %s	0x00	8	read/ write	0x00	0xFF
	-	-	-	DAM	D/A Converter Mode Register	0x03	8	read/ write	0x00	0xFF
FLCN	-	-	-	DFLCTL	Data flash Control Register	0x90	8	read/ write	0x00	0xFF
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

Peripheral name = Name of peripheral
Dim = Number of elements in an array of registers
Dim inc = Address increment between two simultaneous registers of a register array in the address map

Dim index = Sub string that replaces the %s placeholder within 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	RA4W1 Group User's Manual

Rev.	Date	Chapter	Description
1.00	Mar 31, 2020	-	First release

RA4W1 Group Hardware User’s Manual

Publication Date: Rev.1.00 Mar 31, 2020

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. Milpitas Campus
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics America Inc. San Jose Campus
6024 Silver Creek Valley Road, San Jose, CA 95138, USA
Tel: +1-408-284-8200, Fax: +1-408-284-2775

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

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 101-T01, Floor 1, Building 7, Yard No. 7, 8th Street, Shangdi, Haidian District, Beijing 100085, 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, 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, #06-02 Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit No 3A-1 Level 3A Tower 8 UOA Business Park, No 1 Jalan Pengaturcara U1/51A, Seksyen U1, 40150 Shah Alam, Selangor, Malaysia
Tel: +60-3-5022-1288, Fax: +60-3-5022-1290

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700

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

© 2020 Renesas Electronics Corporation. All rights reserved.

Colophon 7.0

Renesas RA Family
RA4W1 Group

Renesas Electronics Corporation