User's Manual

32

Renesas RA2A1 Group

User's Manual: Hardware

32-bit MCU

Renesas Advanced (RA) Family
Renesas RA2 Series

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

Renesas Electronics
www.renesas.com

Rev.1.10 Jul 2023

Notice

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Contact Information

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

Trademarks

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

© 2023 Renesas Electronics Corporation. All rights reserved.

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

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

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

Preface

1. About this Document

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

2. Audience

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

3. Related documents

Renesas provides the following documents for this MCU.

Document type	Description
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

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
	NFLTEN	—	NFCLKSEL[1:0]		—	—	—	NMIMD
Value after reset:	0	0	0	0	0	0	0	0

(2) (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.

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

Contents

Features 46

1. Overview 47
   1.1 Function Outline 47
   1.2 Block Diagram 53
   1.3 Part Numbering 54
   1.4 Function Comparison 55
   1.5 Pin Functions 56
   1.6 Pin Assignments 59
   1.7 Pin Lists 62

2. CPU 64
   2.1 Overview 64
      2.1.1 CPU 64
      2.1.2 Debug 64
      2.1.3 Operating Frequency 64
   2.2 MCU Implementation Options 65
   2.3 Trace Interface 66
   2.4 SWD Interface 66
   2.5 Debug Mode 66
      2.5.1 Debug Mode Definition 66
      2.5.2 Debug Mode Effects 66
         2.5.2.1 Low power mode 66
         2.5.2.2 Reset 67
   2.6 Programmers Model 67
      2.6.1 Address Spaces 67
      2.6.2 Cortex-M23 Peripheral Address Map 68
      2.6.3 External Debug Address Map 68
      2.6.4 CoreSight ROM Table 68
         2.6.4.1 ROM entries 68
         2.6.4.2 CoreSight component registers 69
      2.6.5 DBGREG Module 69
         2.6.5.1 Debug Status Register (DBGSTR) 70
         2.6.5.2 Debug Stop Control Register (DBGSTOPCR) 70
         2.6.5.3 DBGREG CoreSight component registers 71
      2.6.6 OCDREG Module 71
         2.6.6.1 ID Authentication Code Register (IAUTH0 to 3) 72
         2.6.6.2 MCU Status Register (MCUSTAT) 72
         2.6.6.3 MCU Control Register (MCUCTRL) 73
         2.6.6.4 OCDREG CoreSight component registers 73
   2.7 SysTick System Timer 74

2.8 OCD Emulator Connection 74
2.8.1 Unlock ID Code 74
2.8.2 DBGEN 74
2.8.3 Restrictions on Connecting an OCD emulator 74
2.8.3.1 Starting connection while in low power mode 75
2.8.3.2 Changing low power mode while in OCD mode 75
2.8.3.3 Modify the unlock ID code in OSIS 75
2.8.3.4 Connecting sequence and SWD authentication 75
2.9 References 76
3. Operating Modes 77
3.1 Overview 77
3.2 Details of Operating Modes 77
3.2.1 Single-Chip Mode 77
3.2.2 SCI Boot Mode 77
3.2.3 USB Boot Mode 77
3.3 Operating Mode Transitions 77
3.3.1 Operating Mode Transitions as Determined by the Mode-Setting Pin 77
4. Address Space 79
4.1 Overview 79
5. Memory Mirror Function (MMF) 80
5.1 Overview 80
5.2 Register Descriptions 80
5.2.1 MemMirror Special Function Register (MMSFR) 80
5.2.2 MemMirror Enable Register (MMEN) 81
5.3 Operation 81
5.3.1 MMF Operation 81
5.3.2 Setting Example 84
6. Resets 86
6.1 Overview 86
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 97
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) 103
7.2.3 MPU Registers 104
7.2.4 Access Window Setting Control Register (AWSC) 105
7.2.5 Access Window Setting Register (AWS) 105
7.2.6 OCD/Serial Programmer ID Setting Register (OSIS) 107
7.3 Setting Option-Setting Memory 108
7.3.1 Allocation of Data in the Option-Setting Memory 108
7.3.2 Setting Data for Programming the Option-Setting Memory 108
7.4 Usage Notes 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) 112
8.2.3 Voltage Monitor 2 Circuit Control Register 1 (LVD2CR1) 112
8.2.4 Voltage Monitor 2 Circuit Status Register (LVD2SR) 113
8.2.5 Voltage Monitor Circuit Control Register (LVCMPCR) 113
8.2.6 Voltage Detection Level Select Register (LVDLVLR) 114
8.2.7 Voltage Monitor 1 Circuit Control Register 0 (LVD1CR0) 115
8.2.8 Voltage Monitor 2 Circuit Control Register 0 (LVD2CR0) 115
8.3 VCC Input Voltage Monitor 116
8.3.1 Monitoring V_{det0} 116
8.3.2 Monitoring V_{det1} 116
8.3.3 Monitoring V_{det2} 116
8.4 Reset from Voltage Monitor 0 117
8.5 Interrupt and Reset from Voltage Monitor 1 117
8.6 Interrupt and Reset from Voltage Monitor 2 119
8.7 Event Link Output 121
8.7.1 Interrupt Handling and Event Linking 121

9. Clock Generation Circuit 122
9.1 Overview 122
9.2 Register Descriptions 125
9.2.1 System Clock Division Control Register (SCKDIVCR) 125
9.2.2 System Clock Source Control Register (SCKSCR) 126

9.2.3 Memory Wait Cycle Control Register (MEMWAIT) 127
9.2.4 Main Clock Oscillator Control Register (MOSCCR) 129
9.2.5 Sub-Clock Oscillator Control Register (SOSCCR) 130
9.2.6 Low-Speed On-Chip Oscillator Control Register (LOCOCR) 131
9.2.7 High-Speed On-Chip Oscillator Control Register (HOCOCR) 131
9.2.8 Middle-Speed On-Chip Oscillator Control Register (MOCOCR) 132
9.2.9 Oscillation Stabilization Flag Register (OSCSF) 133
9.2.10 Oscillation Stop Detection Control Register (OSTDCR) 134
9.2.11 Oscillation Stop Detection Status Register (OSTDSR) 135
9.2.12 Main Clock Oscillator Wait Control Register (MOSCWTCR) 136
9.2.13 High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR) 137
9.2.14 Main Clock Oscillator Mode Oscillation Control Register (MOMCR) 138
9.2.15 Sub-Clock Oscillator Mode Control Register (SOMCR) 138
9.2.16 Clock Out Control Register (CKOCR) 139
9.2.17 LOCO User Trimming Control Register (LOCOUTCR) 140
9.2.18 MOCO User Trimming Control Register (MOCOUTCR) 140
9.2.19 HOCO User Trimming Control Register (HOCOUTCR) 141
9.2.20 24-bit Sigma-Delta A/D Converter Clock Control Register (SDADCCKCR) 141
9.3 Main Clock Oscillator 142
9.3.1 Connecting a Crystal Resonator 142
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 143
9.5 Oscillation Stop Detection Function 143
9.5.1 Oscillation Stop Detection and Operation after Detection 143
9.5.2 Oscillation Stop Detection Interrupts 144
9.6 Internal Clock 144
9.6.1 System Clock (ICLK) 145
9.6.2 Peripheral Module Clock (PCLKB, PCLKD) 146
9.6.3 Flash Interface Clock (FCLK) 147
9.6.4 USB Clock (UCLK) 147
9.6.5 CAN Clock (CANMCLK) 147
9.6.6 CAC Clock (CACCLK) 147
9.6.7 RTC-Dedicated Clock (RTCSCLK, RTCLCLK) 147
9.6.8 IWDT-Dedicated Clock (IWDTCLK) 147
9.6.9 AGT-Dedicated Clock (AGTSCLK, AGTLCLK) 147
9.6.10 SysTick Timer-Dedicated Clock (SYSTICCLK) 147
9.6.11 Clock/Buzzer Output Clock (CLKOUT) 147
9.6.12 24-bit Sigma-Delta A/D Converter Clock (SDADCCLK) 148
9.7 Usage Notes 148

9.7.1 Notes on Clock Generation Circuit 148
9.7.2 Notes on Resonator 148
9.7.3 Notes on Board Design 148
9.7.4 Notes on Resonator Connect Pin 148
10. Clock Frequency Accuracy Measurement Circuit (CAC) 149
10.1 Overview 149
10.2 Register Descriptions 150
10.2.1 CAC Control Register 0 (CACR0) 150
10.2.2 CAC Control Register 1 (CACR1) 151
10.2.3 CAC Control Register 2 (CACR2) 152
10.2.4 CAC Interrupt Control Register (CAICR) 153
10.2.5 CAC Status Register (CASTR) 154
10.2.6 CAC Upper-Limit Value Setting Register (CAULVR) 155
10.2.7 CAC Lower-Limit Value Setting Register (CALLVR) 155
10.2.8 CAC Counter Buffer Register (CACNTBR) 155
10.3 Operation 155
10.3.1 Measuring Clock Frequency 155
10.3.2 Digital Filtering of Signals on CACREF Pin 157
10.4 Interrupt Requests 157
10.5 Usage Notes 157
10.5.1 Settings for the Module-Stop Function 157
11. Low Power Modes 158
11.1 Overview 158
11.2 Register Descriptions 161
11.2.1 Standby Control Register (SBYCR) 161
11.2.2 Module Stop Control Register A (MSTPCRA) 162
11.2.3 Module Stop Control Register B (MSTPCRB) 162
11.2.4 Module Stop Control Register C (MSTPCRC) 163
11.2.5 Module Stop Control Register D (MSTPCRD) 164
11.2.6 Operating Power Control Register (OPCCR) 165
11.2.7 Sub Operating Power Control Register (SOPCCR) 166
11.2.8 Snooze Control Register (SNZCR) 167
11.2.9 Snooze End Control Register (SNZEDCR) 168
11.2.10 Snooze Request Control Register (SNZREQCR) 169
11.2.11 Flash Operation Control Register (FLSTOP) 170
11.2.12 System Control OCD Control Register (SYOCDCR) 171
11.3 Reducing Power Consumption by Switching Clock Signals 171
11.4 Module-Stop Function 171
11.5 Function for Lower Operating Power Consumption 172
11.5.1 Setting Operating Power Control Mode 172
11.5.2 Operating Range 173

11.6 Sleep Mode 176
11.6.1 Transitioning to Sleep Mode 176
11.6.2 Canceling Sleep Mode 176
11.7 Software Standby Mode 177
11.7.1 Transitioning to Software Standby Mode 177
11.7.2 Canceling Software Standby Mode 178
11.7.3 Example of Software Standby Mode Application 178
11.8 Snooze Mode 179
11.8.1 Transitioning to Snooze Mode 179
11.8.2 Canceling Snooze Mode 180
11.8.3 Returning to Software Standby Mode 181
11.8.4 Snooze Operation Example 182
11.9 Usage Notes 186
11.9.1 Register Access 186
11.9.2 I/O Port States 187
11.9.3 Module-Stop State of DTC 187
11.9.4 Internal Interrupt Sources 187
11.9.5 Transition to Low Power Modes 188
11.9.6 Timing of WFI Instruction 188
11.9.7 Writing WDT/IWDT Registers by DTC in Sleep Mode or Snooze Mode 188
11.9.8 Oscillators in Snooze Mode 188
11.9.9 Snooze Mode Entry by RXD0 Falling Edge 188
11.9.10 Using SCI0 in Snooze Mode 188
11.9.11 Conditions of A/D Conversion Start in Snooze Mode 188
11.9.12 Conditions of CTSU in Snooze Mode 188
11.9.13 ELC Event in Snooze Mode 188
11.9.14 Module-Stop Function for ADC160 189
11.9.15 Module-Stop Function for an Unused Circuit 189
12. Register Write Protection 190
12.1 Overview 190
12.2 Register Descriptions 190
12.2.1 Protect Register (PRCR) 190
13. Interrupt Controller Unit (ICU) 191
13.1 Overview 191
13.2 Register Descriptions 192
13.2.1 IRQ Control Register i (IRQCRi) (i = 0 to 7) 193
13.2.2 Non-Maskable Interrupt Status Register (NMISR) 194
13.2.3 Non-Maskable Interrupt Enable Register (NMIER) 196
13.2.4 Non-Maskable Interrupt Status Clear Register (NMICLR) 198
13.2.5 NMI Pin Interrupt Control Register (NMICR) 199
13.2.6 ICU Event Link Setting Register n (IELSRn) (n = 0 to 31) 200

13.2.7 SYS Event Link Setting Register (SELSR0) 201
13.2.8 Wake Up Interrupt Enable Register (WUPEN) 201
13.3 Vector Table 203
13.3.1 Interrupt Vector Table 203
13.3.2 Event Number 205
13.4 Interrupt Operation 208
13.4.1 Detecting Interrupts 208
13.4.2 Selecting Interrupt Request Destinations 210
13.4.2.1 CPU interrupt request 210
13.4.2.2 DTC activation 210
13.4.3 Digital Filter 211
13.4.4 External Pin Interrupts 211
13.5 Non-Maskable Interrupt Operation 212
13.6 Return from Low Power Modes 212
13.6.1 Return from Sleep Mode 212
13.6.2 Return from Software Standby Mode 212
13.6.3 Return from Snooze Mode 213
13.7 Using the WFI Instruction with Non-Maskable Interrupt 213
13.8 Reference 213
14. Buses 214
14.1 Overview 214
14.2 Description of Buses 215
14.2.1 Main Buses 215
14.2.2 Slave Interface 215
14.2.3 Parallel Operation 215
14.2.4 Restriction on Endianness 216
14.3 Register Descriptions 216
14.3.1 Master Bus Control Register (BUSMCNT<master>) 216
14.3.2 Slave Bus Control Register (BUSSCNT<slave>) 217
14.3.3 Bus Error Address Register (BUSnERRADD) (n = 3, 4) 218
14.3.4 BUS Error Status Register (BUSnERRSTAT) (n = 3, 4) 218
14.4 Bus Error Monitoring Section 219
14.4.1 Error Type that Occurs by Bus 219
14.4.2 Operation when a Bus Error Occurs 219
14.4.3 Conditions Leading to Illegal Address Access Errors 219
14.4.4 Timeout 220
14.5 Notes on using Flash Cache 220
14.6 References 220
15. Memory Protection Unit (MPU) 221
15.1 Overview 221
15.2 CPU Stack Pointer Monitor 221

15.2.1 Protection of Registers 224
15.2.2 Overflow/Underflow Error 224
15.2.3 Register Descriptions 224
15.2.3.1 Main Stack Pointer (MSP) Monitor Start Address Register (MSPMPUSA) 225
15.2.3.2 Main Stack Pointer (MSP) Monitor End Address Register (MSPMPUEA) 225
15.2.3.3 Process Stack Pointer (PSP) Monitor Start Address Register (PSPMPUSA) 226
15.2.3.4 Process Stack Pointer (PSP) Monitor End Address Register (PSPMPUEA) 226
15.2.3.5 Stack Pointer Monitor Operation After Detection Register (MSPMPUOAD, PSPMPUOAD) 227
15.2.3.6 Stack Pointer Monitor Access Control Register (MSPMPUCTL, PSPMPUCTL) 227
15.2.3.7 Stack Pointer Monitor Protection Register (MSPMPUPT, PSPMPUPT) 228
15.3 Arm MPU 229
15.4 Bus Master MPU 229
15.4.1 Register Descriptions 230
15.4.1.1 Group A Region n Start Address Register (MMPUSAn) (n = 0 to 3) 231
15.4.1.2 Group A Region n End Address Register (MMPUEAn) (n = 0 to 3) 231
15.4.1.3 Group A Region n Access Control Register (MMPUACAn) (n = 0 to 3) 231
15.4.1.4 Bus Master MPU Control Register (MMPUCTLA) 233
15.4.1.5 Group A Protection of Register (MMPUPTA) 233
15.4.2 Operation 234
15.4.2.1 Memory protection 234
15.4.2.2 Protecting the registers 236
15.4.2.3 Memory protection error 236
15.5 Bus Slave MPU 237
15.5.1 Register Descriptions 237
15.5.1.1 Access Control Register for Memory Bus 1 (SMPUMBIU) 238
15.5.1.2 Access Control Register for Internal Peripheral Bus 9 (SMPUFBIU) 238
15.5.1.3 Access Control Register for Memory Bus 4 (SMPUSRAM0) 239
15.5.1.4 Access Control Register for Internal Peripheral Bus 1 (SMPUP0BIU) 239
15.5.1.5 Access Control Register for Internal Peripheral Bus 3 (SMPUP2BIU) 240
15.5.1.6 Access Control Register for Internal Peripheral Bus 7 (SMPUP6BIU) 241
15.5.1.7 Slave MPU Control Register (SMPUCTL) 241
15.5.2 Operation 242
15.5.2.1 Memory protection 242
15.5.2.2 Protecting the registers 242
15.5.2.3 Memory protection error 242
15.6 Security MPU 242
15.6.1 Register Descriptions (Option-Setting Memory) 243
15.6.1.1 Security MPU Program Counter Start Address Register n (SECMPUPCSn) (n = 0, 1) 244

15.6.1.2 Security MPU Program Counter End Address Register n (SECMPUPCEn) (n = 0, 1) 244
15.6.1.3 Security MPU Region 0 Start Address Register (SECMPUS0) 245
15.6.1.4 Security MPU Region 0 End Address Register (SECMPUE0) 245
15.6.1.5 Security MPU Region 1 Start Address Register (SECMPUS1) 246
15.6.1.6 Security MPU Region 1 End Address Register (SECMPUE1) 246
15.6.1.7 Security MPU Region 2 Start Address Register (SECMPUS2) 247
15.6.1.8 Security MPU Region 2 End Address Register (SECMPUE2) 247
15.6.1.9 Security MPU Region 3 Start Address Register (SECMPUS3) 248
15.6.1.10 Security MPU Region 3 End Address Register (SECMPUE3) 248
15.6.1.11 Security MPU Access Control Register (SECMPUAC) 249
15.6.2 Memory Protection 250
15.6.3 Usage Notes 251
15.7 References 251
16. Data Transfer Controller (DTC) 252
16.1 Overview 252
16.2 Register Descriptions 253
16.2.1 DTC Mode Register A (MRA) 254
16.2.2 DTC Mode Register B (MRB) 254
16.2.3 DTC Transfer Source Register (SAR) 255
16.2.4 DTC Transfer Destination Register (DAR) 256
16.2.5 DTC Transfer Count Register A (CRA) 256
16.2.6 DTC Transfer Count Register B (CRB) 257
16.2.7 DTC Control Register (DTCCR) 257
16.2.8 DTC Vector Base Register (DTCVBR) 258
16.2.9 DTC Module Start Register (DTCST) 258
16.2.10 DTC Status Register (DTCSTS) 259
16.3 Activation Sources 259
16.3.1 Allocating Transfer Information and DTC Vector Table 260
16.4 Operation 261
16.4.1 Transfer Information Read Skip Function 264
16.4.2 Transfer Information Write-Back Skip Function 264
16.4.3 Normal Transfer Mode 265
16.4.4 Repeat Transfer Mode 266
16.4.5 Block Transfer Mode 267
16.4.6 Chain Transfer 268
16.4.7 Operation Timing 269
16.4.8 Execution Cycles of DTC 271
16.4.9 DTC Bus Mastership Release Timing 271
16.5 DTC Setting Procedure 271
16.6 Examples of DTC Usage 273

16.6.1 Normal Transfer 273
16.6.2 Chain transfer 273
16.6.3 Chain Transfer when Counter = 0 275
16.7 Interrupt Sources 276
16.8 Event Link 276
16.9 Snooze Control Interface 276
16.10 Module-Stop Function 276
16.11 Usage Notes 277
16.11.1 Transfer Information Start Address 277
17. Event Link Controller (ELC) 278
17.1 Overview 278
17.2 Register Descriptions 279
17.2.1 Event Link Controller Register (ELCR) 279
17.2.2 Event Link Software Event Generation Register n (ELSEGRn) (n = 0, 1) 279
17.2.3 Event Link Setting Register n (ELSRn) (n = 0 to 3, 8, 9, 12, 14, 15, 18 to 20, 22) 280
17.3 Operation 283
17.3.1 Relation between Interrupt Handling and Event Linking 283
17.3.2 Linking Events 283
17.3.3 Example Procedure for Linking Events 284
17.4 Usage Notes 284
17.4.1 Linking DTC Transfer End Signals as Events 284
17.4.2 Setting the Clocks 284
17.4.3 Setting the Module Stop Function 284
17.4.4 ELC Delay Time 284
18. I/O Ports 286
18.1 Overview 286
18.2 Register Descriptions 288
18.2.1 Port Control Register 1 (PCNTR1/PODR/PDR) 288
18.2.2 Port Control Register 2 (PCNTR2/EIDR/PIDR) 289
18.2.3 Port Control Register 3 (PCNTR3/PORR/POSR) 290
18.2.4 Port Control Register 4 (PCNTR4/EORR/EOSR) 291
18.2.5 Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) (m = 0 to 5, 9; n = 00 to 15) 292
18.2.6 Write-Protect Register (PWPR) 294
18.3 Operation 294
18.3.1 General I/O Ports 294
18.3.2 Port Function Select 295
18.3.3 Port Group Function for the ELC 295
18.3.3.1 Behavior when ELC_PORT1 or 2 is input from the ELC 295
18.3.3.2 Behavior when an event pulse is output to the ELC 296
18.4 Handling of Unused Pins 297
18.5 Usage Notes 298

18.5.1 Procedure for Specifying the Pin Functions 298
18.5.2 Procedure for Using Port Group Input 298
18.5.3 Port Output Data Register (PODR) Summary 298
18.5.4 Notes on Using Analog Functions 298
18.5.5 Selecting the USB_DP and USB_DM Pins 298
18.6 Peripheral Select Settings for each Product 299
19. Key Interrupt Function (KINT) 306
19.1 Overview 306
19.2 Register Descriptions 308
19.2.1 Key Return Control Register (KRCTL) 308
19.2.2 Key Return Flag Register (KRF) 308
19.2.3 Key Return Mode Register (KRM) 308
19.3 Operation 309
19.3.1 Operation When Not Using Key Interrupt Flag (KRMD = 0) 309
19.3.2 Operation When Using Key Interrupt Flag (KRMD = 1) 310
19.4 Usage Notes 311
20. Port Output Enable for GPT (POEG) 312
20.1 Overview 312
20.2 Register Descriptions 314
20.2.1 POEG Group n Setting Register (POEGGn) (n = A, B) 314
20.3 Output-Disable Control Operation 315
20.3.1 Pin Input Level Detection Operation 315
20.3.1.1 Digital filter 315
20.3.2 Output-Disable Request from the GPT 316
20.3.3 Comparator Interrupt Detection 316
20.3.4 Output-Disable Control Using Detection of Stopped Oscillation 316
20.3.5 Output-Disable Control Using Registers 316
20.3.6 Release from Output Disable 316
20.4 Interrupt Sources 317
20.5 External Trigger Output to the GPT 317
20.6 Usage Notes 318
20.6.1 Transition to Software Standby mode 318
20.6.2 Specifying Pins Associated with the GPT 318
21. General PWM Timer (GPT) 319
21.1 Overview 319
21.2 Register Descriptions 323
21.2.1 General PWM Timer Write Protection Register (GTWP) 324
21.2.2 General PWM Timer Software Start Register (GTSTR) 324
21.2.3 General PWM Timer Software Stop Register (GTSTP) 325
21.2.4 General PWM Timer Software Clear Register (GTCLR) 325
21.2.5 General PWM Timer Start Source Select Register (GTSSR) 326

21.2.6 General PWM Timer Stop Source Select Register (GTPSR) 328
21.2.7 General PWM Timer Clear Source Select Register (GTCSR) 330
21.2.8 General PWM Timer Up Count Source Select Register (GTUPSR) 333
21.2.9 General PWM Timer Down Count Source Select Register (GTDNSR) 335
21.2.10 General PWM Timer Input Capture Source Select Register A (GTICASR) 338
21.2.11 General PWM Timer Input Capture Source Select Register B (GTICBSR) 340
21.2.12 General PWM Timer Control Register (GTCR) 343
21.2.13 General PWM Timer Count Direction and Duty Setting Register (GTUDDTYC) 344
21.2.14 General PWM Timer I/O Control Register (GTIOR) 346
21.2.15 General PWM Timer Interrupt Output Setting Register (GTINTAD) 350
21.2.16 General PWM Timer Status Register (GTST) 351
21.2.17 General PWM Timer Buffer Enable Register (GTBER) 354
21.2.18 General PWM Timer Counter (GTCNT) 356
21.2.19 General PWM Timer Compare Capture Register n (GTCCRn) (n = A to F) 356
21.2.20 General PWM Timer Cycle Setting Register (GTPR) 357
21.2.21 General PWM Timer Cycle Setting Buffer Register (GTPBR) 357
21.2.22 General PWM Timer Dead Time Control Register (GTDTCR) 358
21.2.23 General PWM Timer Dead Time Value Register U (GTDVU) 358
21.2.24 Output Phase Switching Control Register (OPSCR) 359
21.3 Operation 361
21.3.1 Basic Operation 361
21.3.1.1 Counter operation 361
21.3.1.2 Waveform output by compare match 366
21.3.1.3 Input capture function 369
21.3.2 Buffer Operation 371
21.3.2.1 GTPR register buffer operation 371
21.3.2.2 Buffer operation for GTCCRA and GTCCRB 375
21.3.3 PWM Output Operating Mode 379
21.3.3.1 Saw-wave PWM mode 379
21.3.3.2 Saw-wave one-shot pulse mode 382
21.3.3.3 Triangle-wave PWM mode 1 (32-bit transfer at trough) 385
21.3.3.4 Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) 387
21.3.3.5 Triangle-wave PWM mode 3 (64-bit transfer at trough) 389
21.3.4 Automatic Dead Time Setting Function 392
21.3.5 Count Direction Changing Function 397
21.3.6 Function of Output Duty 0% and 100% 397
21.3.7 Hardware Count Start/Count Stop and Clear Operation 399
21.3.7.1 Hardware start operation 399
21.3.7.2 Hardware stop operation 400
21.3.7.3 Hardware clear operation 404
21.3.8 Synchronized Operation 406

21.3.8.1 Synchronized operation by software 406
21.3.8.2 Synchronized operation by hardware 409
21.3.9 PWM Output Operation Examples 411
21.3.10 Phase Counting Function 417
21.3.11 Output Phase Switching (GPT_OPS) 424
21.3.11.1 Input selection and synchronization of external input signal 426
21.3.11.2 Input sampling 427
21.3.11.3 Input phase decode 427
21.3.11.4 Output selection control 427
21.3.11.5 Output selection control (group output disable function) 429
21.3.11.6 Event Link Controller (ELC) output 429
21.3.11.7 GPT_OPS start operation setting flow 430
21.4 Interrupt Sources 430
21.4.1 DTC Activation 433
21.5 Operations Linked by the ELC 433
21.5.1 Event Signal Output to the ELC 433
21.5.2 Event Signal Inputs from the ELC 433
21.6 Noise Filter Function 433
21.7 Protection Function 434
21.7.1 Write-Protection for Registers 434
21.7.2 Disabling of Buffer Operation 434
21.7.3 GTIOC Pin Output Negate Control 435
21.8 Initialization Method of Output Pins 436
21.8.1 Pin Settings after Reset 436
21.8.2 Pin Initialization Caused by Error during Operation 437
21.9 Usage Notes 437
21.9.1 Module-Stop Function Setting 437
21.9.2 GTCCRn Settings during Compare Match Operation (n = A to F) 437
21.9.3 Setting Range for GTCNT Counter 438
21.9.4 Starting and Stopping the GTCNT Counter 438
21.9.5 Priority Order of Each Event 438
22. Low Power Asynchronous General Purpose Timer (AGT) 440
22.1 Overview 440
22.2 Register Descriptions 442
22.2.1 AGT Counter Register (AGT) 442
22.2.2 AGT Compare Match A Register (AGTCMA) 442
22.2.3 AGT Compare Match B Register (AGTCMB) 443
22.2.4 AGT Control Register (AGTCR) 443
22.2.5 AGT Mode Register 1 (AGTMR1) 445
22.2.6 AGT Mode Register 2 (AGTMR2) 446
22.2.7 AGT I/O Control Register (AGTIOC) 447

22.2.8 AGT Event Pin Select Register (AGTISR) 448
22.2.9 AGT Compare Match Function Select Register (AGTCMSR) 448
22.2.10 AGT Pin Select Register (AGTIOSEL) 449
22.3 Operation 449
22.3.1 Reload Register and Counter Rewrite Operation 449
22.3.2 Reload Register and Compare Register A/B Rewrite Operation 451
22.3.3 Timer Mode 452
22.3.4 Pulse Output Mode 453
22.3.5 Event Counter Mode 454
22.3.6 Pulse Width Measurement Mode 456
22.3.7 Pulse Period Measurement Mode 456
22.3.8 Compare Match function 457
22.3.9 Output Settings for Each Mode 458
22.3.10 Standby Mode 460
22.3.11 Interrupt Sources 460
22.3.12 Event Signal Output to ELC 461
22.4 Usage Notes 461
22.4.1 Count Operation Start and Stop Control 461
22.4.2 Access to Counter Register 461
22.4.3 When Changing Mode 461
22.4.4 Digital Filter 462
22.4.5 How to Calculate Event Number, Pulse Width, and Pulse Period 462
22.4.6 When Count is Forcibly Stopped by TSTOP Bit 462
22.4.7 When Selecting AGT0 Underflow as the Count Source 462
22.4.8 Reset of I/O Register 462
22.4.9 When Selecting PCLKB, PCLKB/8, or PCLKB/2 as the Count Source 462
22.4.10 When Selecting AGTLCLK or AGTSCLK as the Count Source 462
22.4.11 When Switching Source Clock 463
23. Realtime Clock (RTC) 464
23.1 Overview 464
23.2 Register Descriptions 466
23.2.1 64-Hz Counter (R64CNT) 466
23.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0) 466
23.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1) 467
23.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2) 468
23.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3) 469
23.2.6 Day Counter (RDAYCNT) 470
23.2.7 Month Counter (RMONCNT) 470
23.2.8 Year Counter (RYRCNT) 471
23.2.9 Second Alarm Register (RSECAR)/Binary Counter 0 Alarm Register (BCNT0AR) 471
23.2.10 Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR) 472

23.2.11 Hour Alarm Register (RHRAR)/Binary Counter 2 Alarm Register (BCNT2AR) 473
23.2.12 Day-of-Week Alarm Register (RWKAR)/Binary Counter 3 Alarm Register (BCNT3AR) 474
23.2.13 Date Alarm Register (RDAYAR)/Binary Counter 0 Alarm Enable Register (BCNT0AER) 475
23.2.14 Month Alarm Register (RMONAR)/Binary Counter 1 Alarm Enable Register (BCNT1AER) 476
23.2.15 Year Alarm Register (RYRAR)/Binary Counter 2 Alarm Enable Register (BCNT2AER) 477
23.2.16 Year Alarm Enable Register (RYRAREN)/ Binary Counter 3 Alarm Enable Register (BCNT3AER) 478
23.2.17 RTC Control Register 1 (RCR1) 479
23.2.18 RTC Control Register 2 (RCR2) 480
23.2.19 RTC Control Register 4 (RCR4) 483
23.2.20 Frequency Register (RFRH/RFRL) 484
23.2.21 Time Error Adjustment Register (RADJ) 485
23.3 Operation 485
23.3.1 Outline of Initial Settings of Registers after Power On 485
23.3.2 Clock and Count Mode Setting Procedure 486
23.3.3 Setting the Time 487
23.3.4 30-Second Adjustment 487
23.3.5 Reading 64-Hz Counter and Time 488
23.3.6 Alarm Function 489
23.3.7 Procedure for Disabling Alarm Interrupt 489
23.3.8 Time Error Adjustment Function 490
23.3.8.1 Automatic adjustment 490
23.3.8.2 Adjustment by software 491
23.3.8.3 Procedure for changing the mode of adjustment 491
23.3.8.4 Procedure for stopping adjustment 492
23.4 Interrupt Sources 492
23.5 Event Link Output 493
23.5.1 Interrupt Handling and Event Linking 493
23.6 Usage Notes 494
23.6.1 Register Writing during Counting 494
23.6.2 Use of Periodic Interrupts 494
23.6.3 RTCOUT (1-Hz/64-Hz) Clock Output 495
23.6.4 Transitions to Low Power Modes after Setting Registers 495
23.6.5 Notes on Writing to and Reading from Registers 495
23.6.6 Changing the Count Mode 495
23.6.7 Initialization Procedure when the RTC is not to be Used 495
23.6.8 When Switching Source Clock 496

24. Watchdog Timer (WDT) 497
24.1 Overview 497
24.2 Register Descriptions 498
24.2.1 WDT Refresh Register (WDTRR) 498
24.2.2 WDT Control Register (WDTCR) 499
24.2.3 WDT Status Register (WDTSR) 501
24.2.4 WDT Reset Control Register (WDTRCR) 502
24.2.5 WDT Count Stop Control Register (WDTCSTPR) 503
24.2.6 Option Function Select Register 0 (OFS0) 503
24.3 Operation 503
24.3.1 Count Operation in each Start Mode 503
24.3.1.1 Register start mode 503
24.3.1.2 Auto start mode 505
24.3.2 Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers 506
24.3.3 Refresh Operation 507
24.3.4 Reset Output 508
24.3.5 Interrupt Sources 508
24.3.6 Reading the Down-Counter Value 508
24.3.7 Association between Option Function Select Register 0 (OFS0) and WDT Registers 509
24.4 Link Operation by ELC 509
24.5 Usage Notes 509
24.5.1 ICU Event Link Setting Register n (IELSRn) Setting 509
25. Independent Watchdog Timer (IWDT) 510
25.1 Overview 510
25.2 Register Descriptions 511
25.2.1 IWDT Refresh Register (IWDTRR) 511
25.2.2 IWDT Status Register (IWDTSR) 512
25.2.3 Option Function Select Register 0 (OFS0) 513
25.3 Operation 515
25.3.1 Auto Start Mode 515
25.3.2 Refresh Operation 516
25.3.3 Status Flags 517
25.3.4 Reset Output 518
25.3.5 Interrupt Sources 518
25.3.6 Reading the Down-counter Value 518
25.4 Link Operation by the ELC 518
25.5 Usage Notes 519
25.5.1 Refresh Operations 519
25.5.2 Restrictions on the Clock Division Ratio Setting 519
26. USB 2.0 Full-Speed Module (USBFS) 520
26.1 Overview 520

26.2 Register Descriptions 522
26.2.1 System Configuration Control Register (SYSCFG) 522
26.2.2 System Configuration Status Register 0 (SYSSTS0) 523
26.2.3 Device State Control Register 0 (DVSTCTR0) 523
26.2.4 CFIFO Port Register (CFIFO/CFIFOL) 524
26.2.5 CFIFO Port Select Register (CFIFOSEL) 525
26.2.6 CFIFO Port Control Register (CFIFOCTR) 527
26.2.7 Interrupt Enable Register 0 (INTENB0) 528
26.2.8 BRDY Interrupt Enable Register (BRDYENB) 529
26.2.9 NRDY Interrupt Enable Register (NRDYENB) 529
26.2.10 BEMP Interrupt Enable Register (BEMPENB) 530
26.2.11 SOF Output Configuration Register (SOFCFG) 531
26.2.12 Interrupt Status Register 0 (INTSTS0) 531
26.2.13 BRDY Interrupt Status Register (BRDYSTS) 533
26.2.14 NRDY Interrupt Status Register (NRDYSTS) 534
26.2.15 BEMP Interrupt Status Register (BEMPSTS) 534
26.2.16 Frame Number Register (FRMNUM) 535
26.2.17 USB Request Type Register (USBREQ) 535
26.2.18 USB Request Value Register (USBVAL) 536
26.2.19 USB Request Index Register (USBINDX) 536
26.2.20 USB Request Length Register (USBLENG) 537
26.2.21 DCP Configuration Register (DCPCFG) 537
26.2.22 DCP Maximum Packet Size Register (DCPMAXP) 538
26.2.23 DCP Control Register (DCPCTR) 539
26.2.24 Pipe Window Select Register (PIPESEL) 541
26.2.25 Pipe Configuration Register (PIPECFG) 542
26.2.26 Pipe Maximum Packet Size Register (PIPEMAXP) 544
26.2.27 PIPEn Control Registers (PIPEnCTR) (n = 4 to 7) 544
26.2.28 PIPEn Transaction Counter Enable Register (PIPEnTRE) (n = 4 and 5) 550
26.2.29 PIPEn Transaction Counter Register (PIPEnTRN) (n = 4 and 5) 551
26.2.30 USB Module Control Register (USBMC) 552
26.2.31 BC Control Register 0 (USBBCCTRL0) 552
26.2.32 USB Clock Selection Register (UCKSEL) 554
26.3 Operation 554
26.3.1 System Control 554
26.3.1.1 Setting data to the USBFS-related registers 554
26.3.1.2 Controlling the USBFS data bus resistors 554
26.3.1.3 Example of USBFS power supply connection 555
26.3.1.4 Example of USB external connection circuits 557
26.3.2 Interrupts 562
26.3.3 Interrupt Descriptions 564

26.3.3.1 BRDY interrupt 564
26.3.3.2 NRDY interrupt 566
26.3.3.3 BEMP interrupt 567
26.3.3.4 Device state transition interrupt 568
26.3.3.5 Control transfer stage transition interrupt 569
26.3.3.6 Frame update interrupt 570
26.3.3.7 VBUS interrupt 570
26.3.3.8 Resume interrupt 571
26.3.4 Pipe Control 571
26.3.4.1 Pipe control register switching procedures 571
26.3.4.2 Transfer types 572
26.3.4.3 Endpoint number 572
26.3.4.4 Maximum packet size setting 572
26.3.4.5 Transaction counter for pipes 4 and 5 in the receiving direction 572
26.3.4.6 Response PID 573
26.3.4.7 Data PID sequence bit 573
26.3.4.8 Response PID = NAK function 573
26.3.4.9 Auto response mode 573
26.3.4.10 OUT-NAK mode 574
26.3.4.11 Null auto response mode 574
26.3.5 FIFO Buffer Memory 574
26.3.6 FIFO Buffer Clearing 575
26.3.7 FIFO Port Functions 575
26.3.8 Control Transfers Using DCP 576
26.3.8.1 Control Transfers 576
26.3.9 Bulk Transfers (Pipes 4 and 5) 577
26.3.10 Interrupt Transfers (Pipes 6 and 7) 577
26.3.11 Pipe Schedule 578
26.3.11.1 Transfer schedule 578
26.3.12 Battery Charging Detection Processing 578
26.3.12.1 Processing 578
26.4 Usage Notes 580
26.4.1 Settings for the Module-Stop State 580
26.4.2 Clearing the Interrupt Status Register on Exiting Software Standby Mode 580
26.4.3 Clearing the Interrupt Status Register after Setting the Port Function 580
26.4.4 Notes on 32-pin Products 580
27. Serial Communications Interface (SCI) 581
27.1 Overview 581
27.2 Register Descriptions 584
27.2.1 Receive Shift Register (RSR) 584
27.2.2 Receive Data Register (RDR) 584

27.2.3 Receive 9-Bit Data Register (RDRHL) 585
27.2.4 Receive FIFO Data Register H, L, HL (FRDRH, FRDRL, FRDRHL) 585
27.2.5 Transmit Data Register (TDR) 586
27.2.6 Transmit 9-Bit Data Register (TDRHL) 587
27.2.7 Transmit FIFO Data Register H, L, HL (FTDRH, FTDRL, FTDRHL) 587
27.2.8 Transmit Shift Register (TSR) 588
27.2.9 Serial Mode Register (SMR) for Non-Smart Card Interface mode (SCMR.SMIF = 0) 588
27.2.10 Serial Mode Register for Smart Card Interface Mode (SMR_SMCI) (SCMR.SMIF = 1) 590
27.2.11 Serial Control Register (SCR) for Non-Smart Card Interface Mode (SCMR.SMIF = 0) 591
27.2.12 Serial Control Register for Smart Card Interface Mode (SCR_SMCI) (SCMR.SMIF = 1) 593
27.2.13 Serial Status Register (SSR) for Non-Smart Card Interface and Non-FIFO Mode (SCMR.SMIF = 0 and FCR.FM = 0) 595
27.2.14 Serial Status Register for Non-Smart Card Interface and FIFO Mode (SSR_FIFO) (SCMR.SMIF = 0 and FCR.FM = 1) 597
27.2.15 Serial Status Register for Smart Card Interface Mode (SSR_SMCI) (SCMR.SMIF = 1) 600
27.2.16 Smart Card Mode Register (SCMR) 602
27.2.17 Bit Rate Register (BRR) 603
27.2.18 Modulation Duty Register (MDDR) 611
27.2.19 Serial Extended Mode Register (SEMR) 613
27.2.20 Noise Filter Setting Register (SNFR) 615
27.2.21 I2C Mode Register 1 (SIMR1) 615
27.2.22 I2C Mode Register 2 (SIMR2) 616
27.2.23 I2C Mode Register 3 (SIMR3) 617
27.2.24 I2C Status Register (SISR) 619
27.2.25 SPI Mode Register (SPMR) 619
27.2.26 FIFO Control Register (FCR) 621
27.2.27 FIFO Data Count Register (FDR) 622
27.2.28 Line Status Register (LSR) 623
27.2.29 Compare Match Data Register (CDR) 623
27.2.30 Data Compare Match Control Register (DCCR) 624
27.2.31 Serial Port Register (SPTR) 625
27.3 Operation in Asynchronous Mode 626
27.3.1 Serial Data Transfer Format 627
27.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode 628
27.3.3 Clock 629
27.3.4 Double-Speed Operation and Frequency of 6 Times the Bit Rate 630
27.3.5 CTS and RTS Functions 630
27.3.6 Address Match (Receive Data Match Detection) Function 631

27.3.7 SCI Initialization in Asynchronous Mode 634
27.3.8 Serial Data Transmission in Asynchronous Mode 636
27.3.9 Serial Data Reception in Asynchronous Mode 642
27.4 Multi-Processor Communications Function 649
27.4.1 Multi-Processor Serial Data Transmission 650
27.4.2 Multi-Processor Serial Data Reception 653
27.5 Operation in Clock Synchronous Mode 658
27.5.1 Clock 658
27.5.2 CTS and RTS Functions 659
27.5.3 SCI Initialization in Clock Synchronous Mode 659
27.5.4 Serial Data Transmission in Clock Synchronous Mode 661
27.5.5 Serial Data Reception in Clock Synchronous Mode 665
27.5.6 Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode 670
27.6 Operation in Smart Card Interface Mode 672
27.6.1 Example Connection 672
27.6.2 Data Format (Except in Block Transfer Mode) 673
27.6.3 Block Transfer Mode 674
27.6.4 Receive Data Sampling Timing and Reception Margin 674
27.6.5 SCI Initialization 675
27.6.6 Serial Data Transmission (Except in Block Transfer Mode) 677
27.6.7 Serial Data Reception (Except in Block Transfer Mode) 679
27.6.8 Clock Output Control 681
27.7 Operation in Simple IIC Mode 681
27.7.1 Generation of Start, Restart, and Stop Conditions 682
27.7.2 Clock Synchronization 684
27.7.3 SDAn Output Delay 684
27.7.4 SCI Initialization in Simple IIC Mode 685
27.7.5 Operation in Master Transmission in Simple IIC Mode 686
27.7.6 Master Reception in Simple IIC Mode 688
27.8 Operation in Simple SPI Mode 690
27.8.1 States of Pins in Master and Slave Modes 691
27.8.2 SS Function in Master Mode 691
27.8.3 SS Function in Slave Mode 691
27.8.4 Relationship between Clock and Transmit/Receive Data 691
27.8.5 SCI Initialization in Simple SPI Mode 692
27.8.6 Transmission and Reception of Serial Data in Simple SPI Mode 692
27.9 Bit Rate Modulation Function 692
27.10 Interrupt Sources 693
27.10.1 Buffer Operations for SCIn_TXI and SCIn_RXI Interrupts (non-FIFO selected) 693
27.10.2 Buffer Operations for SCIn_TXI and SCIn_RXI Interrupts (FIFO selected) 693
27.10.3 Interrupts in Asynchronous, Clock Synchronous, and Simple SPI Modes 693

27.10.4 Interrupts in Smart Card Interface Mode 695
27.10.5 Interrupts in Simple IIC Mode 695
27.11 Event Linking 696
27.12 Address Mismatch Event Output (SCI0_DCUF) 697
27.13 Noise Cancellation Function 697
27.14 Usage Notes 698
27.14.1 Settings for the Module-Stop State 698
27.14.2 SCI Operations during Low Power State 698
27.14.3 Break Detection and Processing 704
27.14.4 Mark State and Production of Breaks 704
27.14.5 Receive Error Flags and Transmit Operations in Clock Synchronous and Simple SPI Modes 704
27.14.6 Restrictions on Clock Synchronous Transmission in Clock Synchronous Mode and Simple SPI Mode 704
27.14.7 Restrictions on Using DTC 705
27.14.8 Notes on Starting Transfer 706
27.14.9 External Clock Input in Clock Synchronous Mode and Simple SPI Mode 706
27.14.10 Limitations on Simple SPI Mode 706
27.14.11 Notes on Transmit Enable Bit (SCR.TE) 707
27.14.12 Note on Stopping Reception When Using the RTS Function in Asynchronous Mode 707
28. I2C Bus Interface (IIC) 708
28.1 Overview 708
28.2 Register Descriptions 711
28.2.1 I2C Bus Control Register 1 (ICCR1) 711
28.2.2 I2C Bus Control Register 2 (ICCR2) 713
28.2.3 I2C Bus Mode Register 1 (ICMR1) 716
28.2.4 I2C Bus Mode Register 2 (ICMR2) 717
28.2.5 I2C Bus Mode Register 3 (ICMR3) 718
28.2.6 I2C Bus Function Enable Register (ICFER) 720
28.2.7 I2C Bus Status Enable Register (ICSER) 721
28.2.8 I2C Bus Interrupt Enable Register (ICIER) 722
28.2.9 I2C Bus Status Register 1 (ICSR1) 723
28.2.10 I2C Bus Status Register 2 (ICSR2) 726
28.2.11 I2C-Bus Wakeup Unit Register (ICWUR) 729
28.2.12 I2C Bus Wakeup Unit Register 2 (ICWUR2) 730
28.2.13 Slave Address Register Ly (SARLy) (y = 0 to 2) 731
28.2.14 Slave Address Register Uy (SARUy) (y = 0 to 2) 731
28.2.15 I2C Bus Bit Rate Low-Level Register (ICBRL) 732
28.2.16 I2C Bus Bit Rate High-Level Register (ICBRH) 733
28.2.17 I2C Bus Transmit Data Register (ICDRT) 734
28.2.18 I2C Bus Receive Data Register (ICDRR) 734

28.2.19 I2C Bus Shift Register (ICDRS) 735
28.3 Operation 735
28.3.1 Communication Data Format 735
28.3.2 Initial Settings 736
28.3.3 Master Transmit Operation 737
28.3.4 Master Receive Operation 740
28.3.5 Slave Transmit Operation 745
28.3.6 Slave Receive Operation 748
28.4 SCL Synchronization Circuit 750
28.5 SDA Output Delay Function 751
28.6 Digital Noise Filter Circuits 752
28.7 Address Match Detection 752
28.7.1 Slave-Address Match Detection 752
28.7.2 Detection of General Call Address 754
28.7.3 Device ID Address Detection 755
28.7.4 Host Address Detection 756
28.8 Wakeup Function 756
28.8.1 Normal Wakeup Mode 1 757
28.8.2 Normal Wakeup Mode 2 760
28.8.3 Command Recovery Mode and EEP Response Mode (Special Wakeup Modes) 762
28.9 Automatic Low-Hold Function for SCL 765
28.9.1 Function to Prevent Wrong Transmission of Transmit Data 765
28.9.2 NACK Reception Transfer Suspension Function 766
28.9.3 Function to Prevent Failure to Receive Data 767
28.10 Arbitration-Lost Detection Functions 768
28.10.1 Master Arbitration-Lost Detection (MALE Bit) 768
28.10.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit) 770
28.10.3 Slave Arbitration-Lost Detection (SALE Bit) 771
28.11 Start, Restart, and Stop Condition Issuing Function 772
28.11.1 Issuing a Start Condition 772
28.11.2 Issuing a Restart Condition 772
28.11.3 Issuing a Stop Condition 774
28.12 Bus Hanging 774
28.12.1 Timeout Function 774
28.12.2 Extra SCL Clock Cycle Output Function 775
28.12.3 IIC Reset and Internal Reset 776
28.13 SMBus Operation 776
28.13.1 SMBus Timeout Measurement 777
28.13.2 Packet Error Code (PEC) 778
28.13.3 SMBus Host Notification Protocol (Notify ARP Master Command) 778
28.14 Interrupt Sources 778

28.14.1 Buffer Operation for IICn_TXI and IICn_RXI Interrupts 779
28.15 State of Registers When Issuing Each Condition 779
28.16 Event Link Output 780
28.16.1 Interrupt Handling and Event Linking 781
28.17 Usage Notes 781
28.17.1 Settings for the Module-Stop State 781
28.17.2 Notes on Starting Transfer 781
29. Controller Area Network (CAN) Module 782
29.1 Overview 782
29.2 Register Descriptions 784
29.2.1 Control Register (CTLR) 784
29.2.2 Bit Configuration Register (BCR) 787
29.2.3 Mask Register k (MKRk) (k = 0 to 7) 789
29.2.4 FIFO Received ID Compare Registers 0 and 1 (FIDCR0 and FIDCR1) 790
29.2.5 Mask Invalid Register (MKIVLR) 791
29.2.6 Mailbox Register j (MBj_ID, MBj_DL, MBj_Dm, MBj_TS) (j = 0 to 31, m = 0 to 7) 791
29.2.7 Mailbox Interrupt Enable Register (MIER) 795
29.2.8 Mailbox Interrupt Enable Register for FIFO Mailbox Mode (MIER_FIFO) 796
29.2.9 Message Control Register for Transmit (MCTL_TXj) (j = 0 to 31) 797
29.2.10 Message Control Register for Receive (MCTL_RXj) (j = 0 to 31) 799
29.2.11 Receive FIFO Control Register (RFCR) 801
29.2.12 Receive FIFO Pointer Control Register (RFPCR) 803
29.2.13 Transmit FIFO Control Register (TFCR) 803
29.2.14 Transmit FIFO Pointer Control Register (TFPCR) 805
29.2.15 Status Register (STR) 805
29.2.16 Mailbox Search Mode Register (MSMR) 807
29.2.17 Mailbox Search Status Register (MSSR) 808
29.2.18 Channel Search Support Register (CSSR) 809
29.2.19 Acceptance Filter Support Register (AFSR) 809
29.2.20 Error Interrupt Enable Register (EIER) 810
29.2.21 Error Interrupt Factor Judge Register (EIFR) 811
29.2.22 Receive Error Count Register (RECR) 813
29.2.23 Transmit Error Count Register (TECR) 814
29.2.24 Error Code Store Register (ECSR) 814
29.2.25 Time Stamp Register (TSR) 815
29.2.26 Test Control Register (TCR) 816
29.3 Operation Modes 817
29.3.1 CAN Reset Mode 818
29.3.2 CAN Halt Mode 819
29.3.3 CAN Sleep Mode 820
29.3.4 CAN Operation Mode (Excluding Bus-Off State) 820

29.3.5 CAN Operation Mode (Bus-Off State) 821
29.4 Data Transfer Rate Configuration 821
29.4.1 Clock Setting 821
29.4.2 Bit Time Setting 822
29.4.3 Data Transfer Rate 822
29.5 Mailbox and Mask Register Structure 822
29.6 Acceptance Filtering and Masking Functions 824
29.7 Reception and Transmission 826
29.7.1 Reception 826
29.7.2 Transmission 828
29.8 Interrupts 829
29.9 Usage Notes 830
29.9.1 Settings for the Module-Stop State 830
29.9.2 Settings for the Operating Clock 830
30. Serial Peripheral Interface (SPI) 831
30.1 Overview 831
30.2 Register Descriptions 834
30.2.1 SPI Control Register (SPCR) 834
30.2.2 SPI Slave Select Polarity Register (SSLP) 836
30.2.3 SPI Pin Control Register (SPPCR) 836
30.2.4 SPI Status Register (SPSR) 837
30.2.5 SPI Data Register (SPDR/SPDR_HA) 839
30.2.6 SPI Bit Rate Register (SPBR) 841
30.2.7 SPI Data Control Register (SPDCR) 842
30.2.8 SPI Clock Delay Register (SPCKD) 843
30.2.9 SPI Slave Select Negation Delay Register (SSLND) 843
30.2.10 SPI Next-Access Delay Register (SPND) 844
30.2.11 SPI Control Register 2 (SPCR2) 844
30.2.12 SPI Command Register 0 (SPCMD0) 845
30.3 Operation 847
30.3.1 Overview of SPI Operations 847
30.3.2 Controlling the SPI Pins 848
30.3.3 SPI System Configuration Examples 849
30.3.3.1 Single master and single slave with the MCU as a master 849
30.3.3.2 Single master and single slave with the MCU as a slave 850
30.3.3.3 Single master and multi slave with the MCU as a master 851
30.3.3.4 Single master and multi-slave with the MCU as a slave 852
30.3.3.5 Multi-master and multi-slave with the MCU as a master 852
30.3.3.6 Master and slave in clock synchronous mode with the MCU as a master 853
30.3.3.7 Master and slave in clock synchronous mode with the MCU as a slave 854
30.3.4 Data Format 854

30.3.4.1 Operation when parity is disabled (SPCR2.SPPE = 0) 855
30.3.4.2 Operation when parity is enabled (SPCR2.SPPE = 1) 858
30.3.5 Transfer Format 862
30.3.5.1 Transfer format when CPHA = 0 862
30.3.5.2 When CPHA = 1 863
30.3.6 Data Transfer Modes 864
30.3.6.1 Full-duplex synchronous serial communications (SPCR.TXMD = 0) 864
30.3.6.2 Transmit-only operations (SPCR.TXMD = 1) 865
30.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts 865
30.3.8 Error Detection 867
30.3.8.1 Overrun errors 868
30.3.8.2 Parity errors 869
30.3.8.3 Mode fault errors 870
30.3.8.4 Underrun errors 870
30.3.9 Initializing the SPI 871
30.3.9.1 Initialization by clearing the SPE bit 871
30.3.9.2 Initialization by system reset 871
30.3.10 SPI Operation 871
30.3.10.1 Master mode operation 871
30.3.10.2 Slave mode operation 877
30.3.11 Clock Synchronous Operation 881
30.3.11.1 Master mode operation 881
30.3.11.2 Slave mode operation 883
30.3.12 Loopback Mode 884
30.3.13 Self-Diagnosis of Parity Bit Function 884
30.3.14 Interrupt Sources 886
30.4 Event Link Controller Event Output 887
30.4.1 Receive Buffer Full Event Output 887
30.4.2 Transmit Buffer Empty Event Output 887
30.4.3 Mode Fault, Underrun, Overrun, or Parity Error Event Output 887
30.4.4 SPI Idle Event Output 887
30.4.5 Transmission-Completed Event Output 888
30.5 Usage Notes 888
30.5.1 Settings for the Module-Stop State 888
30.5.2 Restrictions on Low Power Function 888
30.5.3 Restrictions on Starting Transfer 888
30.5.4 Restrictions on Mode Fault, Underrun, Overrun, or Parity Error Event Output 888
30.5.5 Restrictions on SPRF and SPTEF Flags 888
31. Cyclic Redundancy Check (CRC) Calculator 889
31.1 Overview 889
31.2 Register Descriptions 890

31.2.1 CRC Control Register 0 (CRCCR0) 890
31.2.2 CRC Control Register 1 (CRCCR1) 891
31.2.3 CRC Data Input Register (CRCDIR/CRCDIR_BY) 891
31.2.4 CRC Data Output Register (CRCDOR/CRCDOR_HA/CRCDOR_BY) 892
31.2.5 Snoop Address Register (CRCSAR) 892
31.3 Operation 893
31.3.1 Basic Operation 893
31.3.2 CRC Snoop 896
31.4 Usage Notes 897
31.4.1 Settings for the Module-Stop State 897
31.4.2 Notes on Transmission 897
32. 16-Bit A/D Converter (ADC16) 898
32.1 Overview 898
32.2 Register Descriptions 902
32.2.1 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) 902
32.2.2 A/D Self-Diagnosis Data Register (ADRD) 903
32.2.3 A/D Self-Diagnostic Status Register (ADRST) 903
32.2.4 A/D Control Register (ADCSR) 904
32.2.5 A/D Channel Select Register A0 (ADANSA0) 908
32.2.6 A/D Channel Select Register A1 (ADANSA1) 908
32.2.7 A/D Channel Select Register B0 (ADANSB0) 909
32.2.8 A/D Channel Select Register B1 (ADANSB1) 910
32.2.9 A/D-Converted Value Average Channel Select Register 0 (ADADS0) 910
32.2.10 A/D-Converted Value Average Channel Select Register 1 (ADADS1) 911
32.2.11 A/D-Converted Value Average Count Select Register (ADADC) 912
32.2.12 A/D Control Extended Register (ADCER) 913
32.2.13 A/D Conversion Start Trigger Select Register (ADSTRGR) 915
32.2.14 A/D Conversion Extended Input Control Register (ADEXICR) 916
32.2.15 A/D Sampling State Register n (ADSSTRn) (n = 00 to 08, L, T, O) 917
32.2.16 A/D Disconnection Detection Control Register (ADDISCR) 918
32.2.17 A/D Interrupt Control Register (ADICR) 920
32.2.18 A/D Group Scan Priority Control Register (ADGSPCR) 920
32.2.19 A/D Compare Function Control Register (ADCMPCR) 921
32.2.20 A/D Compare Function Window A Channel Select Register 0 (ADCMPANSR0) 923
32.2.21 A/D Compare Function Window A Channel Select Register 1 (ADCMPANSR1) 923
32.2.22 A/D Compare Function Window A Extended Input Select Register (ADCMPANSER) 924

32.2.23 A/D Compare Function Window A Comparison Condition Setting Register 0 (ADCMPLR0) 924
32.2.24 A/D Compare Function Window A Comparison Condition Setting Register 1 (ADCMPLR1) 926
32.2.25 A/D Compare Function Window A Extended Input Comparison Condition Setting Register (ADCMPLER) 926
32.2.26 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) 927
32.2.27 A/D Compare Function Window A Channel Status Register 0 (ADCMPSR0) 928
32.2.28 A/D Compare Function Window A Channel Status Register 1 (ADCMPSR1) 929
32.2.29 A/D Compare Function Window A Extended Input Channel Status Register (ADCMPSER) 930
32.2.30 A/D Compare Function Window B Channel Select Register (ADCMPBNSR) 931
32.2.31 A/D Compare Function Window B Status Register (ADCMPBSR) 933
32.2.32 A/D Compare Function Window A/B Status Monitor Register (ADWINMON) 933
32.2.33 A/D Dedicated Reference Voltage Circuit Control Register (VREFAMPCNT) 934
32.2.34 A/D Channel Input Mode Select Register (ADANIM) 935
32.2.35 A/D Calibration Execution Register (ADCALEXE) 936
32.3 Operation 937
32.3.1 Selection of Analog Input Channels 937
32.3.2 Results of A/D Conversion 937
32.3.3 Scanning Operation 941
32.3.4 Single Scan Mode 942
32.3.4.1 Basic operation 942
32.3.4.2 Channel selection and self-diagnosis 943
32.3.4.3 A/D conversion of temperature sensor output/internal reference voltage 944
32.3.4.4 A/D conversion in double trigger mode 945
32.3.4.5 Extended operations when double trigger mode is selected 946
32.3.5 Continuous Scan Mode 947
32.3.5.1 Basic operation 947
32.3.5.2 Channel selection and self-diagnosis 948
32.3.6 Group Scan Mode 949
32.3.6.1 Basic operation 949
32.3.6.2 A/D conversion in double trigger mode 950
32.3.6.3 Operation with group A priority control 951
32.3.7 Compare Function for Window A and Window B 958
32.3.7.1 Compare function 958
32.3.7.2 Event output of compare function 959
32.3.7.3 Restrictions on the compare function 961
32.3.8 Analog Input Sampling and Scan Conversion Time 962

32.3.9 Usage Example of A/D Data Register Automatic Clearing Function 965
32.3.10 A/D-Converted Value Average Mode 965
32.3.11 Disconnection Detection Assist Function 965
32.3.12 Starting A/D Conversion with Asynchronous Trigger 967
32.3.13 Starting A/D Conversion with a Synchronous Trigger from Peripheral Module 967
32.3.14 Calibration Function 967
32.3.15 Calibration Time 969
32.4 Interrupt Sources and DTC Transfer Requests 969
32.4.1 Interrupt Requests 969
32.5 Event Link Function 970
32.5.1 Event Output to the ELC 970
32.5.2 ADC16 Operation through an Event from the ELC 970
32.6 Selecting Reference Voltage 970
32.7 Usage Notes 971
32.7.1 Notes on Reading Data Registers 971
32.7.2 Notes on Stopping A/D Conversion 971
32.7.3 A/D Conversion Restarting Timing and Termination Timing 972
32.7.4 Restrictions on Scan End Interrupt Handling 972
32.7.5 Settings for the Module-Stop State 972
32.7.6 Restrictions on Entering the Low Power States 973
32.7.7 Error in Absolute Accuracy when Disconnection Detection Assistance is in Use 973
32.7.8 Operating Modes and Status Bits 973
32.7.9 Notes on Board Design 973
32.7.10 Notes on Noise Reduction 973
32.7.11 Port Setting when Using the 16-bit A/D Converter Input 975
32.7.12 Relationship between the ADC16, OPAMP, ACMPHS, and SDADC24 976
32.7.13 Notes on Canceling Software Standby Mode 976
32.7.14 Notes on Calibration Function 976
33. 24-Bit Sigma-Delta A/D Converter (SDADC24) 977
33.1 Overview 977
33.2 Register Descriptions 979
33.2.1 Startup Control Register 1 (STC1) 979
33.2.2 Startup Control Register 2 (STC2) 980
33.2.3 Input Multiplexer n Setting Register (PGACn) (n = 0 to 4) 981
33.2.4 Sigma-Delta A/D Converter Control Register 1 (ADC1) 985
33.2.5 Sigma-Delta A/D Converter Control Register 2 (ADC2) 987
33.2.6 Sigma-Delta A/D Converter Conversion Result Register (ADCR) 987
33.2.7 Sigma-Delta A/D Converter Average Value Register (ADAR) 988
33.2.8 Calibration Control Register (CLBC) 989
33.2.9 Calibration Start Control Register (CLBSTR) 989
33.2.10 Calibration Status Register (CLBSSR) 990

33.2.11 Calibration Control Protection Release Register (CLBPR) 990
33.2.12 Gain Error Correction Factor Register n (GCVLRn) (n = 0 to 4) 991
33.2.13 Offset Error Correction Factor Register n (OCVLRn) (n = 0 to 4) 991
33.3 Operation 991
33.3.1 ADBGR 992
33.3.2 ADREG 992
33.3.3 SBIAS and VREFI 993
33.3.3.1 Description of the SBIAS and VREFI function 993
33.3.3.2 SBIAS independent operation 993
33.3.4 VBIAS 994
33.3.5 Input Multiplexers 994
33.3.5.1 Input Multiplexer Control Registers 995
33.3.6 Programmable Gain Instrumentation Amplifier (PGA) 995
33.3.6.1 Range of input voltage 997
33.3.6.2 Range of input voltage in differential input mode 997
33.3.6.3 Range of input voltage in single-ended input mode 999
33.3.6.4 Registers for controlling the PGA 1000
33.3.7 Input Voltage for the SDADC24 and Results of A/D Conversion 1000
33.3.8 Control of the Sigma-Delta A/D Converter (AUTOSCAN) 1001
33.3.9 Digital Filter 1004
33.3.9.1 Operation of the digital filter 1004
33.3.9.2 Configuration of the digital filter 1005
33.3.10 Calibration function 1006
33.3.10.1 Internal calibration operation mode 1006
33.3.10.2 External calibration operation mode 1007
33.3.10.3 Calibration Time 1007
33.3.11 Disconnection Detection Assist Function 1008
33.3.12 Self-Diagnosis Function of PGA Offset 1010
33.4 Control Flows 1010
33.4.1 Analog Power Supply Activation Flow 1011
33.4.2 Input Multiplexer Setting Flow 1012
33.4.3 Self-Diagnosis Flow of PGA Offset 1013
33.4.4 Disconnection Detection Assist Flow 1014
33.4.5 Internal Calibration Flow 1015
33.4.6 External Calibration Flow 1016
33.4.7 SDADC24 Conversion Flow 1017
33.4.8 SDADC24 Stop Setting Flow 1018
33.4.9 Recalibration Omitted Flow 1019
33.4.10 Flows for Independently Activating and for Switching/Stopping the Sensor Reference Voltage 1019
33.5 Usage Notes 1021

33.5.1 Notes on Reading Data Registers 1021
33.5.2 Settings for the Module-Stop State 1021
33.5.3 Restrictions on Entering the Low Power States 1021
33.5.4 A/D Conversion Operation Mode and SDADC24 Reference Clock Division Setting 1021
33.5.5 Restrictions on SBIAS Operation 1021
33.5.6 Oversampling Ratio, PGA Gain, and Offset Voltage (d_{OFR}) Setting 1021
33.5.7 Restrictions on the Multiplexer 4 (Internal OPAMP) 1022
33.5.8 Relationship between the SDADC24, ADC16 and ACMPHS 1022

34. 12-Bit D/A Converter (DAC12) 1023
34.1 Overview 1023
34.2 Register Descriptions 1024
34.2.1 D/A Data Register 0 (DADR0) 1024
34.2.2 D/A Control Register (DACR) 1024
34.2.3 DADR0 Format Select Register (DADPR) 1025
34.2.4 D/A A/D Synchronous Start Control Register (DAADSCR) 1025
34.2.5 D/A VREF Control Register (DAVREFCR) 1026
34.2.6 D/A Switch Charge Pump Control Register (DAPC) 1026
34.3 Operation 1027
34.3.1 MOCO Stop Procedure after D/A Conversion Disabled 1028
34.3.2 Reducing Interference between D/A and A/D Conversion 1028
34.4 Event Link Operation Setting Procedure 1029
34.5 Usage Notes on Event Link Operation 1029
34.6 Usage Notes 1029
34.6.1 Settings for the Module-Stop Function 1029
34.6.2 DAC12 Operation in Module-Stop State 1029
34.6.3 DAC12 Operation in Software Standby Mode 1030
34.6.4 Restriction on Usage when Interference Reduction between D/A and A/D Conversion is Enabled 1030
34.6.5 D/A Converter Output 1030
34.6.6 DAC12 Output Pin during Charge Pump Enabled 1030
34.6.7 Connection of D/A Converter Output 1030

35. 8-Bit D/A Converter (DAC8) 1031
35.1 Overview 1031
35.2 Register Descriptions 1032
35.2.1 D/A Conversion Value Setting Register n (DACSn) (n = 0, 1) 1032
35.2.2 D/A Converter Mode Register (DAM) 1032
35.2.3 D/A A/D Synchronous Start Control Register (DACADSCR) 1033
35.2.4 D/A Switch Charge Pump Control Register (DACPC) 1033
35.3 Operation 1034
35.3.1 Normal Operation Mode 1034
35.3.2 Real-Time Output Mode (Event Link) 1035

35.3.3 MOCO Stop Procedure after D/A Conversion Disabled 1035
35.3.4 Output Start Timing of the D/A Conversion Value 1036
35.3.5 Minimizing Interference between D/A and A/D Conversion 1036
35.4 Usage Notes 1037
35.4.1 Settings for the Module-Stop Function 1037
35.4.2 DAC8 Operation in Module-Stop State 1037
35.4.3 DAC8 Operation in Software Standby Mode 1037
35.4.4 Real-Time Output of the D/A Converter 1037
35.4.5 D/A Converter Output 1038
35.4.6 When Not Using the D/A Converter 1038
35.4.7 DAC8 Output Pin during Charge Pump Enabled 1038
35.4.8 Connection of D/A Converter Output 1038
36. Temperature Sensor (TSN) 1039
36.1 Overview 1039
36.2 Register Descriptions 1039
36.2.1 Temperature Sensor Calibration Data Register H (TSCDRH) 1039
36.2.2 Temperature Sensor Calibration Data Register L (TSCDRL) 1040
36.3 Using the Temperature Sensor 1040
36.3.1 Preparation for Using the Temperature Sensor 1040
36.3.2 Procedures for Using the Temperature Sensor 1041
37. Operational Amplifier (OPAMP) 1042
37.1 Overview 1042
37.2 Register Descriptions 1045
37.2.1 Operational Amplifier Mode Control Register (AMPMC) 1045
37.2.2 Operational Amplifier Trigger Mode Control Register (AMPTRM) 1046
37.2.3 Operational Amplifier Activation Trigger Select Register (AMPTRS) 1047
37.2.4 Operational Amplifier Control Register (AMPC) 1047
37.2.5 Operational Amplifier Monitor Register (AMPMON) 1048
37.2.6 Operational Amplifier 0 Output Select Register (AMP0OS) 1048
37.2.7 Operational Amplifier 0 Minus Input Select Register (AMP0MS) 1049
37.2.8 Operational Amplifier 0 Plus Input Select Register (AMP0PS) 1050
37.2.9 Operational Amplifier 1 Minus Input Select Register (AMP1MS) 1050
37.2.10 Operational Amplifier 1 Plus Input Select Register (AMP1PS) 1051
37.2.11 Operational Amplifier 2 Minus Input Select Register (AMP2MS) 1051
37.2.12 Operational Amplifier 2 Plus Input Select Register (AMP2PS) 1052
37.2.13 Operational Amplifier Switch Charge Pump Control Register (AMPCPC) 1052
37.2.14 Operational Amplifier User Offset Trimming Enable Register (AMPUOTE) 1053
37.2.15 Operational Amplifier n Offset Trimming Pch Register (AMPnOTP) (n = 0 to 2) 1053
37.2.16 Operational Amplifier n Offset Trimming Nch Register (AMPnOTN) (n = 0 to 2) 1054
37.3 Operation 1055
37.3.1 State Transitions 1055

37.3.2 OPAMP Control Operation 1055
37.4 Software Trigger Mode 1059
37.5 Activation Trigger Mode 1060
37.6 Activation and A/D Trigger Mode 1061
37.7 MOCO Stop Procedure after OPAMP Stopped 1062
37.8 Changing OPAMP Switches 1063
37.9 User Offset Trimming 1063
37.10 Using the OPAMP as a Configurable Amplifier 1066
37.10.1 Voltage Follower 1066
37.10.2 Programmable Non-Inverting Amplifier 1067
37.10.3 Programmable Trans-Impedance Amplifier 1067
37.10.4 Using the Configurable Amplifier as a D/A Converter Output Amplifier 1069
37.11 Usage Notes 1070
38. High-Speed Analog Comparator (ACMPHS) 1071
38.1 Overview 1071
38.2 Register Descriptions 1072
38.2.1 Comparator Control Register (CMPCTL) 1072
38.2.2 Comparator Input Select Register (CMPSEL0) 1073
38.2.3 Comparator Reference Voltage Select Register (CMPSEL1) 1074
38.2.4 Comparator Output Monitor Register (CMPMON) 1074
38.2.5 Comparator Output Control Register (CPIOC) 1075
38.3 Operation 1075
38.4 Noise Filter 1077
38.5 ACMPHS Interrupts 1077
38.6 ACMPHS Output to the Event Link Controller (ELC) 1077
38.7 ACMPHS Pin Output 1078
38.8 Usage Notes 1078
38.8.1 Settings for the Module-Stop Function 1078
38.8.2 Relationship with A/D Converter 1078
39. Low-Power Analog Comparator (ACMPLP) 1079
39.1 Overview 1079
39.2 Register Descriptions 1082
39.2.1 ACMPLP Mode Setting Register (COMPMDR) 1082
39.2.2 ACMPLP Filter Control Register (COMPFIR) 1083
39.2.3 ACMPLP Output Control Register (COMPOCR) 1083
39.2.4 Comparator Input Select Register (COMPSEL0) 1084
39.2.5 Comparator Reference Voltage Select Register (COMPSEL1) 1084
39.3 Operation 1085
39.4 Noise Filter 1087
39.5 ACMPLP Interrupts 1089
39.6 ELC Event Output 1089
39.7 Interrupt Handling and ELC Linking 1089

39.8 Comparator Pin Output 1089
39.9 Usage Notes 1089
39.9.1 Settings for the Module-Stop State 1089
40. Capacitive Touch Sensing Unit (CTSU) 1090
40.1 Overview 1090
40.2 Register Descriptions 1092
40.2.1 CTSU Control Register 0 (CTSUCR0) 1092
40.2.2 CTSU Control Register 1 (CTSUCR1) 1093
40.2.3 CTSU Synchronous Noise Reduction Setting Register (CTSUSDPRS) 1094
40.2.4 CTSU Sensor Stabilization Wait Control Register (CTSUSST) 1095
40.2.5 CTSU Measurement Channel Register 0 (CTSUMCH0) 1096
40.2.6 CTSU Measurement Channel Register 1 (CTSUMCH1) 1098
40.2.7 CTSU Channel Enable Control Register 0 (CTSUCHAC0) 1098
40.2.8 CTSU Channel Enable Control Register 1 (CTSUCHAC1) 1099
40.2.9 CTSU Channel Enable Control Register 2 (CTSUCHAC2) 1099
40.2.10 CTSU Channel Enable Control Register 3 (CTSUCHAC3) 1100
40.2.11 CTSU Channel Transmit/Receive Control Register 0 (CTSUCHTRC0) 1100
40.2.12 CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1) 1101
40.2.13 CTSU Channel Transmit/Receive Control Register 2 (CTSUCHTRC2) 1101
40.2.14 CTSU Channel Transmit/Receive Control Register 3 (CTSUCHTRC3) 1102
40.2.15 CTSU High-Pass Noise Reduction Control Register (CTSUDCLKC) 1102
40.2.16 CTSU Status Register (CTSUST) 1103
40.2.17 CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register (CTSUSSC) 1104
40.2.18 CTSU Sensor Offset Register 0 (CTSUSO0) 1105
40.2.19 CTSU Sensor Offset Register 1 (CTSUSO1) 1106
40.2.20 CTSU Sensor Counter (CTSUSC) 1107
40.2.21 CTSU Reference Counter (CTSURC) 1107
40.2.22 CTSU Error Status Register (CTSUERRS) 1108
40.3 Operation 1110
40.3.1 Principles of Measurement Operation 1110
40.3.2 Measurement Modes 1111
40.3.2.1 Initial settings flow 1112
40.3.2.2 Status counter 1113
40.3.2.3 Self-capacitance single scan mode operation 1114
40.3.2.4 Self-capacitance multi-scan mode operation 1116
40.3.2.5 Mutual-capacitance full scan mode operation 1118
40.3.3 Functions Common to Multiple Modes 1120
40.3.3.1 Sensor stabilization wait time and measurement time 1120
40.3.3.2 Interrupts 1121
40.4 Usage Notes 1122

40.4.1 Measurement Result Data (CTSUSC and CTSURC Counters) 1122
40.4.2 Constraint on Software Trigger 1123
40.4.3 Constraints on External Trigger 1123
40.4.4 Constraints on Forced Stops 1123
40.4.5 TSCAP Pin 1123
40.4.6 Restrictions on Measurement Operation (CTSUCR0.CTSUSTRT = 1) 1123
41. Data Operation Circuit (DOC) 1124
41.1 Overview 1124
41.2 Register Descriptions 1125
41.2.1 DOC Control Register (DOCR) 1125
41.2.2 DOC Data Input Register (DODIR) 1126
41.2.3 DOC Data Setting Register (DODSR) 1126
41.3 Operation 1126
41.3.1 Data Comparison Mode 1126
41.3.2 Data Addition Mode 1127
41.3.3 Data Subtraction Mode 1127
41.4 Interrupt Request and Output to the Event Link Controller (ELC) 1128
41.5 Usage Notes 1128
41.5.1 Settings for the Module-Stop State 1128
42. SRAM 1129
42.1 Overview 1129
42.2 Register Descriptions 1129
42.2.1 SRAM Parity Error Operation After Detection Register (PARIOAD) 1129
42.2.2 SRAM Protection Register (SRAMPRCR) 1130
42.2.3 ECC Operating Mode Control Register (ECCMODE) 1130
42.2.4 ECC 2-Bit Error Status Register (ECC2STS) 1131
42.2.5 ECC 1-Bit Error Information Update Enable Register (ECC1STSEN) 1131
42.2.6 ECC 1-Bit Error Status Register (ECC1STS) 1132
42.2.7 ECC Protection Register (ECCPRCR) 1132
42.2.8 ECC Protection Register 2 (ECCPRCR2) 1133
42.2.9 ECC Test Control Register (ECCETST) 1133
42.2.10 SRAM ECC Error Operation After Detection Register (ECCOAD) 1134
42.2.11 Trace Control (for the MTB) 1134
42.2.12 CoreSight™ (for MTB) 1134
42.3 Operation 1135
42.3.1 ECC Function 1135
42.3.2 ECC Error Generation 1135
42.3.3 ECC Decoder Testing 1136
42.3.4 Parity Calculation Function 1137
42.3.5 SRAM Error Sources 1138
42.3.6 Access Cycle 1139

42.4 Usage Notes 1139
42.4.1 Instruction Fetch from the SRAM Area 1139
42.4.2 Store Buffer of SRAM 1139
43. Flash Memory 1140
43.1 Overview 1140
43.2 Memory Structure 1141
43.3 Flash Cache 1142
43.3.1 Overview 1142
43.4 Register Descriptions 1143
43.4.1 Flash Cache Enable Register (FCACHEE) 1143
43.4.2 Flash Cache Invalidate Register (FCACHEIV) 1144
43.4.3 Data Flash Control Resister (DFLCTL) 1144
43.4.4 Factory MCU Information Flash Root Table (FMIFRT) 1145
43.4.5 Unique ID Register n (UIDRn) (n = 0 to 3) 1145
43.4.6 Part Numbering Register n (PNRn) (n = 0 to 3) 1146
43.4.7 MCU Version Register (MCUVER) 1146
43.5 Operation 1146
43.5.1 Notice to use Flash Cache 1147
43.6 Operating Modes Associated with the Flash Memory 1147
43.6.1 ID Code Protection 1147
43.7 Overview of Functions 1149
43.7.1 Configuration Area Bit Map 1150
43.7.2 Startup Area Select 1151
43.7.3 Protection by Access Window 1152
43.8 Programming Commands 1152
43.9 Suspend Operation 1152
43.10 Protection 1152
43.11 Serial Programming Mode 1153
43.11.1 SCI Boot Mode 1153
43.11.2 USB Boot Mode 1154
43.12 Using a Serial Programmer 1154
43.12.1 Serial Programming 1154
43.13 Self-Programming 1155
43.13.1 Overview 1155
43.13.2 Background Operation 1155
43.14 Reading the Flash Memory 1155
43.14.1 Reading the Code Flash Memory 1155
43.14.2 Reading the Data Flash Memory 1155
43.15 Usage Notes 1155
43.15.1 Erase Suspended Area 1155
43.15.2 Suspension by Erase Suspend Commands 1156
43.15.3 Restrictions on Additional Writes 1156

43.15.4 Reset during Programming and Erasure 1156
43.15.5 Non-Maskable Interrupt Disabled during Programming and Erasure 1156
43.15.6 Location of Interrupt Vectors during Programming and Erasure 1156
43.15.7 Programming and Erasure in Low-Speed Operating Mode 1156
43.15.8 Abnormal Termination during Programming and Erasure 1156
43.15.9 Actions Prohibited during Programming and Erasure 1156
44. AES Engine 1157
45. True Random Number Generator (TRNG) 1158
46. Internal Voltage Regulator 1159
46.1 Overview 1159
46.2 Operation 1159
47. Electrical Characteristics 1160
47.1 Absolute Maximum Ratings 1160
47.2 DC Characteristics 1162
47.2.1 Tj/Ta Definition 1162
47.2.2 I/O VIH, VIL 1163
47.2.3 I/O IOH, IOL 1164
47.2.4 I/O VOH, VOL, and Other Characteristics 1165
47.2.5 Output Characteristics for I/O Pins (Low Drive Capacity) 1168
47.2.6 Output Characteristics for I/O Pins (Middle Drive Capacity) 1170
47.2.7 Output Characteristics for P407, P408 and P409 I/O Pins (Middle Drive Capacity) 1173
47.2.8 Output Characteristics for IIC I/O Pins 1175
47.2.9 Operating and Standby Current 1176
47.2.10 VCC Rise and Fall Gradient and Ripple Frequency 1185
47.3 AC Characteristics 1186
47.3.1 Frequency 1186
47.3.2 Clock Timing 1188
47.3.3 Reset Timing 1190
47.3.4 Wakeup Time 1191
47.3.5 NMI and IRQ Noise Filter 1194
47.3.6 I/O Ports, POEG, GPT, AGT, KINT, and ADC16 Trigger Timing 1195
47.3.7 CAC Timing 1196
47.3.8 SCI Timing 1197
47.3.9 SPI Timing 1203
47.3.10 IIC Timing 1208
47.3.11 CLKOUT Timing 1209
47.4 USB Characteristics 1210
47.4.1 USBFS Timing 1210
47.4.2 USB External Supply 1211
47.5 ADC16 Characteristics 1212

47.6 SDADC24 Characteristics 1215
47.7 DAC12 Characteristics 1219
47.8 DAC8 Characteristics 1219
47.9 TSN Characteristics 1220
47.10 OSC Stop Detect Characteristics 1220
47.11 POR and LVD Characteristics 1221
47.12 CTSU Characteristics 1225
47.13 Comparator Characteristics 1225
47.14 OPAMP Characteristics 1226
47.15 Flash Memory Characteristics 1228
47.15.1 Code Flash Memory Characteristics 1228
47.15.2 Data Flash Memory Characteristics 1229
47.15.3 Serial Wire Debug (SWD) 1230
Appendix 1. Port States in each Processing Mode 1232
Appendix 2. Package Dimensions 1234
Appendix 3. I/O Registers 1241
3.1 Peripheral Base Addresses 1241
3.2 Access Cycles 1242
3.3 Register Descriptions 1244
Appendix 4. Connection Diagram of Analog Block 1282
Revision History 1287

Ultra-low power 48-MHz Arm® Cortex®-M23 core, up to 256-KB code flash memory, 32-KB SRAM, Capacitive Touch Sensing Unit, 16-bit A/D Converter, 24-bit sigma-delta A/D Converter, 12-bit D/A Converter, 8-bit D/A Converter, Operational Amplifier, security and safety features.

Features

■ Arm Cortex-M23 Core
- Armv8-M architecture
- Maximum operating frequency: 48 MHz
- Arm Memory Protection Unit (Arm MPU) with 8 regions
- Debug and Trace: DWT, FPB, and CoreSight™ MTB-M23
- CoreSight Debug Port: SW-DP

■ Memory
- Up to 256-KB code flash memory
- 8-KB data flash memory (100,000 program/erase (P/E) cycles)
- Up to 32-KB SRAM
- Flash Cache (FCACHE)
- Memory Protection Unit (MPU)
- Memory Mirror Function (MMF)
- 128-bit unique ID

■ Connectivity
- USB 2.0 Full-Speed (USBFS) module
  - On-chip transceiver with voltage regulator
  - Compliant with USB Battery Charging Specification 1.2
- Serial Communications Interface (SCI) × 3
  - UART
  - Simple IIC
  - Simple SPI
- Serial Peripheral Interface (SPI) × 2
- I^2C bus interface (IIC) × 2
- Controller Area Network (CAN) module

■ Analog
- 16-bit A/D Converter (ADC16)
  - 1.2 Msps
  - Differential input mode
  - Single-ended input mode
- 24-bit Sigma-Delta A/D Converter (SDADC24)
  - 15.6 ksps
  - Differential input mode
  - Single-ended input mode
- 12-bit D/A Converter (DAC12)
- 8-bit D/A Converter (DAC8) × 2
- High-Speed Analog Comparator (ACMPHS)
- Low-Power Analog Comparator (ACMPLP) × 2
- Operational Amplifier (OPAMP) × 3
- Temperature Sensor (TSN)

■ Timers
- General PWM Timer 32-bit (GPT32)
- General PWM Timer 16-bit (GPT16) × 6
- Low Power 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)
- Event Link Controller (ELC)
- Data Transfer Controller (DTC)
- Key Interrupt Function (KINT)
- Power-on reset
- Low Voltage Detection (LVD) with voltage settings

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

■ Human Machine Interface (HMI)
- Capacitive Touch Sensing Unit (CTSU)

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

■ General Purpose I/O Ports
- Up to 49 input/output pins
  - Up to 3 CMOS input
  - Up to 46 CMOS input/output
    - Up to 9 input/output 5 V tolerant
    - Up to 3 high current (20 mA)

■ Operating Voltage
- VCC: 1.6 to 5.5 V

■ Operating Temperature and Packages
- Ta = -40°C to +85°C
  - 36-pin BGA (5 mm × 5 mm, 0.8 mm pitch)
- Ta = -40°C to +105°C
  - 64-pin LQFP (10 mm × 10 mm, 0.5 mm pitch)
  - 32-pin LQFP (7 mm × 7 mm, 0.8 mm pitch)
  - 48-pin QFN (7 mm × 7 mm, 0.5 mm pitch)
  - 40-pin QFN (6 mm × 6 mm, 0.5 mm pitch)

1. Overview

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

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

- Up to 256-KB code flash memory
- 32-KB SRAM
- 16-bit A/D Converter (ADC16)
- 24-bit Sigma-Delta A/D Converter (SDADC24)
- 12-bit D/A Converter (DAC12)
- 8-bit D/A Converter (DAC8)
- Operational Amplifier (OPAMP) with configurable switches
- Security features.

1.1 Function Outline

Table 1.1 Arm core

Feature	Functional description
Arm Cortex-M23 core	• Maximum operating frequency: up to 48 MHz • Arm Cortex-M23 core: - Revision: r1p0-00rel0 - Armv8-M architecture profile - Single-cycle integer multiplier - 17-cycle integer divider. • Arm Memory Protection Unit (Arm MPU): - Armv8 Protected Memory System Architecture - 8 protect regions. • SysTick timer: - Driven by SYSTICCLK (LOCO) or ICLK.

Table 1.2 Memory

Feature	Functional description
Code flash memory	256 KB of code flash memory. See section 43, Flash Memory.
Data flash memory	8 KB of data flash memory. See section 43, Flash 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. Your 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).
Option-setting memory	The option-setting memory determines the state of the MCU after a reset. See section 7, Option-Setting Memory.
SRAM	On-chip high-speed SRAM with either parity bit or Error Correction Code (ECC). See section 42, SRAM.

Table 1.3 System (1 of 2)

Feature	Functional description
Operating modes	Two operating modes: • Single-chip mode • SCI or USB boot mode. See section 3, Operating Modes.

Table 1.3 System (2 of 2)

Feature	Functional description
Resets	13 resets: • RES pin reset • Power-on reset • Independent watchdog timer reset • Watchdog timer reset • Voltage monitor 0 reset • Voltage monitor 1 reset • Voltage monitor 2 reset • SRAM parity error reset • SRAM ECC error reset • Bus master MPU error reset • Bus slave MPU error reset • CPU stack pointer error reset • Software reset. See section 6, Resets.
Low Voltage Detection (LVD)	The Low Voltage Detection (LVD) function monitors the voltage level input to the VCC pin and the detection level can be selected using a software program. See section 8, Low Voltage Detection (LVD).
Clocks	• Main clock oscillator (MOSC) • Sub-clock oscillator (SOSC) • High-speed on-chip oscillator (HOCO) • Middle-speed on-chip oscillator (MOCO) • Low-speed on-chip oscillator (LOCO) • IWDT-dedicated on-chip oscillator • Clock out support. See section 9, Clock Generation Circuit.
Clock Frequency Accuracy Measurement Circuit (CAC)	The Clock Frequency Accuracy Measurement Circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock to be used as a measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range. When measurement is complete or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated. See section 10, Clock Frequency Accuracy Measurement Circuit (CAC).
Interrupt Controller Unit (ICU)	The Interrupt Controller Unit (ICU) controls which event signals are linked to the NVIC/DTC module. The ICU also controls NMI interrupts. See section 13, 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 19, Key Interrupt Function (KINT).
Low power modes	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.
Register write protection	The register write protection function protects important registers from being overwritten due to software errors. See section 12, Register Write Protection.
Memory Protection Unit (MPU)	Four Memory Protection Units (MPUs) and a CPU stack pointer monitor function are provided for memory protection. See section 15, 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. A refresh-permitted period can be set to refresh the counter and used as the condition to detect when the system runs out of control. See section 24, Watchdog Timer (WDT).
Independent Watchdog Timer (IWDT)	The Independent Watchdog Timer (IWDT) consists of a 14-bit down-counter that must be serviced periodically to prevent counter underflow. The IWDT provides functionality to reset the MCU or to generate a non-maskable interrupt/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 25, 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 17, 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 16, Data Transfer Controller (DTC).

Table 1.6 Timers

Feature	Functional description
General PWM Timer (GPT)	The General PWM Timer (GPT) is a 32-bit timer with one channel and a 16-bit timer with six 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 21, 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 20, Port Output Enable for GPT (POEG).
Low Power Asynchronous General Purpose Timer (AGT)	The Low Power 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 to the same address, and they can be accessed with the AGT register. See section 22, Low Power 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 23, 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 has FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator. See section 27, Serial Communications Interface (SCI).
I^2C bus interface (IIC)	The 2-channel I^2C bus interface (IIC) conforms with and provides a subset of the NXP I^2C (Inter-Integrated Circuit) bus interface functions. See section 28, I^2C 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 30, Serial Peripheral Interface (SPI).

Table 1.7 Communication interfaces (2 of 2)

Feature	Functional description
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 29, Controller Area Network (CAN) Module.
USB 2.0 Full-Speed (USBFS) module	The USB 2.0 Full-Speed (USBFS) module can operate as a device controller. The module supports full-speed and low-speed 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 five pipes. Pipe 0 and pipe 4 to pipe 7 can be assigned any endpoint number based on the peripheral devices used for communication or based on your system. The MCU supports Battery Charging Specification revision 1.2. Because the MCU can be powered at 5 V, the USB LDO regulator provides the internal USB transceiver power supply 3.3 V. See section 26, USB 2.0 Full-Speed Module (USBFS).

Table 1.8 Analog (1 of 2)

Feature	Functional description
16-bit A/D Converter (ADC16)	A successive approximation 16-bit A/D Converter (ADC16) is provided. Up to 17 single-ended/ 4 differential analog input channels are selectable. Reference voltage of SDADC24, temperature sensor output, and internal reference voltage are selectable for conversion. The calibration function calculates capacitor array DAC and gain/offset correction values under the usage conditions to enable accurate A/D conversion. See section 32, 16-Bit A/D Converter (ADC16).
24-bit Sigma-Delta A/D Converter (SDADC24)	A 24-bit Sigma-Delta A/D Converter (SDADC24) with a programmable gain instrumentation amplifier is provided. Up to 10 single-ended/5 differential analog input channels are selectable. The 2 single-ended/1 differential analog input channels of these analog input channels are inputs from internal OPAMP. Analog input multiplexer is input to the sigma-delta A/D converter by the programmable gain instrumentation amplifier (PGA). The A/D conversion result is filtered by the SINC3 digital filter, and then stored in an output register. The calibration function calculates gain error and offset error correction values under the usage conditions to enable accurate A/D conversion. See section 33, 24-Bit Sigma-Delta A/D Converter (SDADC24).
12-bit D/A Converter (DAC12)	A 12-bit D/A Converter (DAC12) is provided. See section 34, 12-Bit D/A Converter (DAC12).
8-bit D/A Converter (DAC8)	An 8-bit D/A Converter (DAC8) is provided. See section 35, 8-Bit D/A Converter (DAC8).
Temperature Sensor (TSN)	The on-chip Temperature Sensor (TSN) determines and monitors the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is linear. The output voltage is provided to the ADC16 for conversion and can be further used by the end application. See section 36, Temperature Sensor (TSN).
High-Speed Analog Comparator (ACMPHS)	The High-Speed Analog Comparator (ACMPHS) compares a reference voltage with an analog input voltage. The comparison result can be read by software and also be output externally. The reference voltage can be selected from either an input to the IVREFi (i = 0 to 2) pin, an output from internal D/A converter, or from the internal reference voltage (Vref) generated internally in the MCU. Such flexibility is useful in applications that require go/no-go comparisons to be performed between analog signals without necessarily requiring A/D conversion. See section 38, High-Speed Analog Comparator (ACMPHS).
Low-Power Analog Comparator (ACMPLP)	The Low-Power Analog Comparator (ACMPLP) compares a reference voltage with an analog input voltage. The comparison result can be read by software and also be output externally. The reference voltage can be selected from either 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 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. See section 39, Low-Power Analog Comparator (ACMPLP).

Table 1.8 Analog (2 of 2)

Feature	Functional description
Operational Amplifier (OPAMP)	The Operational Amplifier (OPAMP) can be used to amplify small analog input voltages and output the amplified voltages. A total of three differential operational amplifier units with two input pins and one output pin are provided. All units have switches that can select input signals. Additionally, operational amplifier 0 has a switch that can select the output pin. See section 37, Operational Amplifier (OPAMP).

Table 1.9 Human machine interfaces

Feature	Functional description
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 fingers do not come into direct contact with the electrodes. 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-generating 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 31, 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
AES	See section 44, AES Engine
True Random Number Generator (TRNG)	See section 45, True Random Number Generator (TRNG)

Table 1.12 I/O ports

Feature	Functional description
I/O ports	• I/O ports for the 64-pin LQFP - I/O pins: 46 - Input pins: 3 - Pull-up resistors: 44 - N-ch open-drain outputs: 24 - 5-V tolerance: 9 • I/O ports for the 48-pin QFN - I/O pins: 30 - Input pins: 3 - Pull-up resistors: 28 - N-ch open-drain outputs: 17 - 5-V tolerance: 6 • I/O ports for the 40-pin QFN - I/O pins: 22 - Input pins: 3 - Pull-up resistors: 20 - N-ch open- drain outputs: 13 - 5-V tolerance: 3 • I/O ports for the 36-pin BGA - I/O pins: 19 - Input pins: 3 - Pull-up resistors: 17 - N-ch open-drain outputs: 13 - 5-V tolerance: 3 • I/O ports for the 32-pin LQFP - I/O pins: 19 - Input pins: 1 - Pull-up resistors: 19 - N-ch open-drain outputs: 15 - 5-V tolerance: 4

1.2 Block Diagram

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

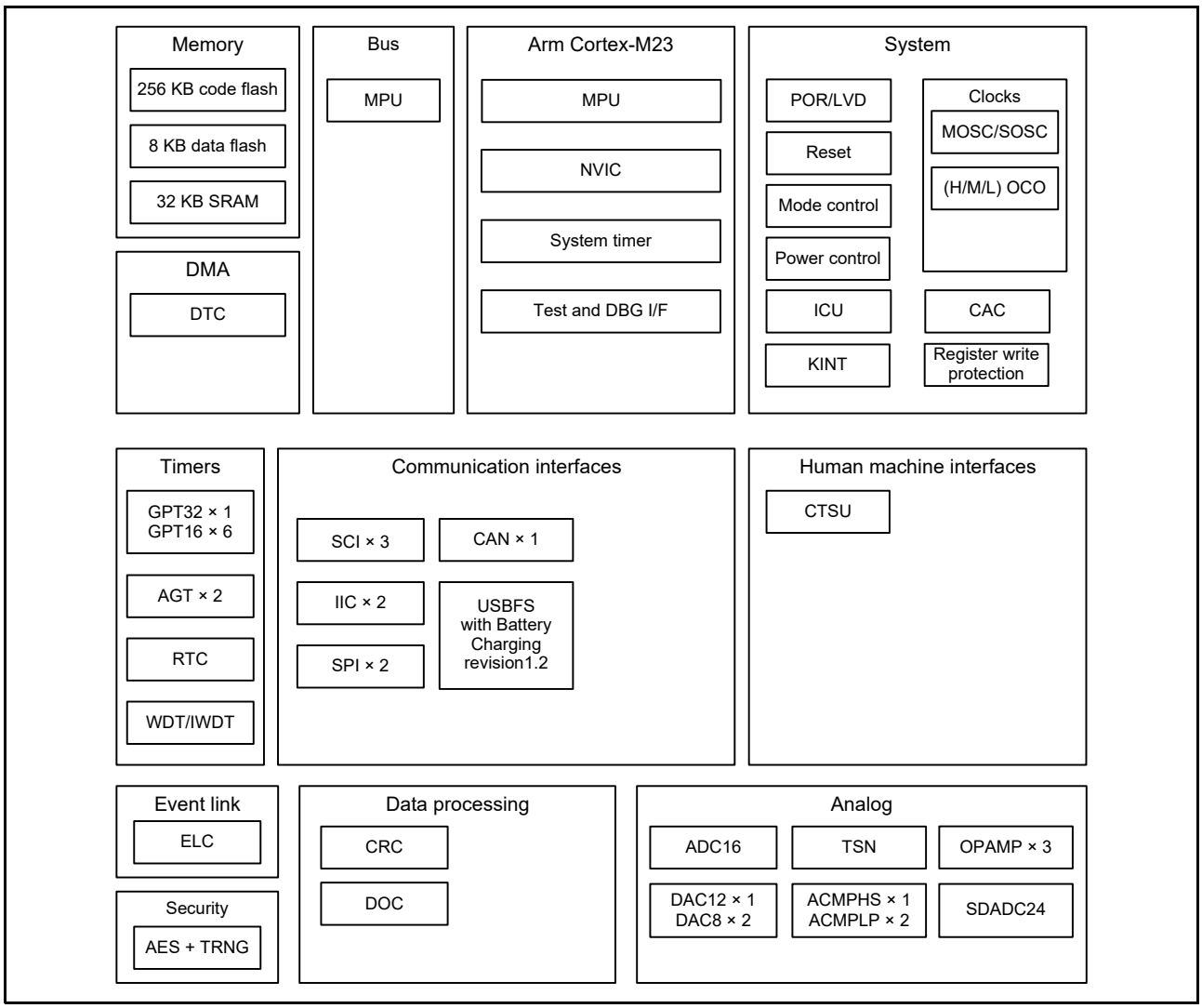
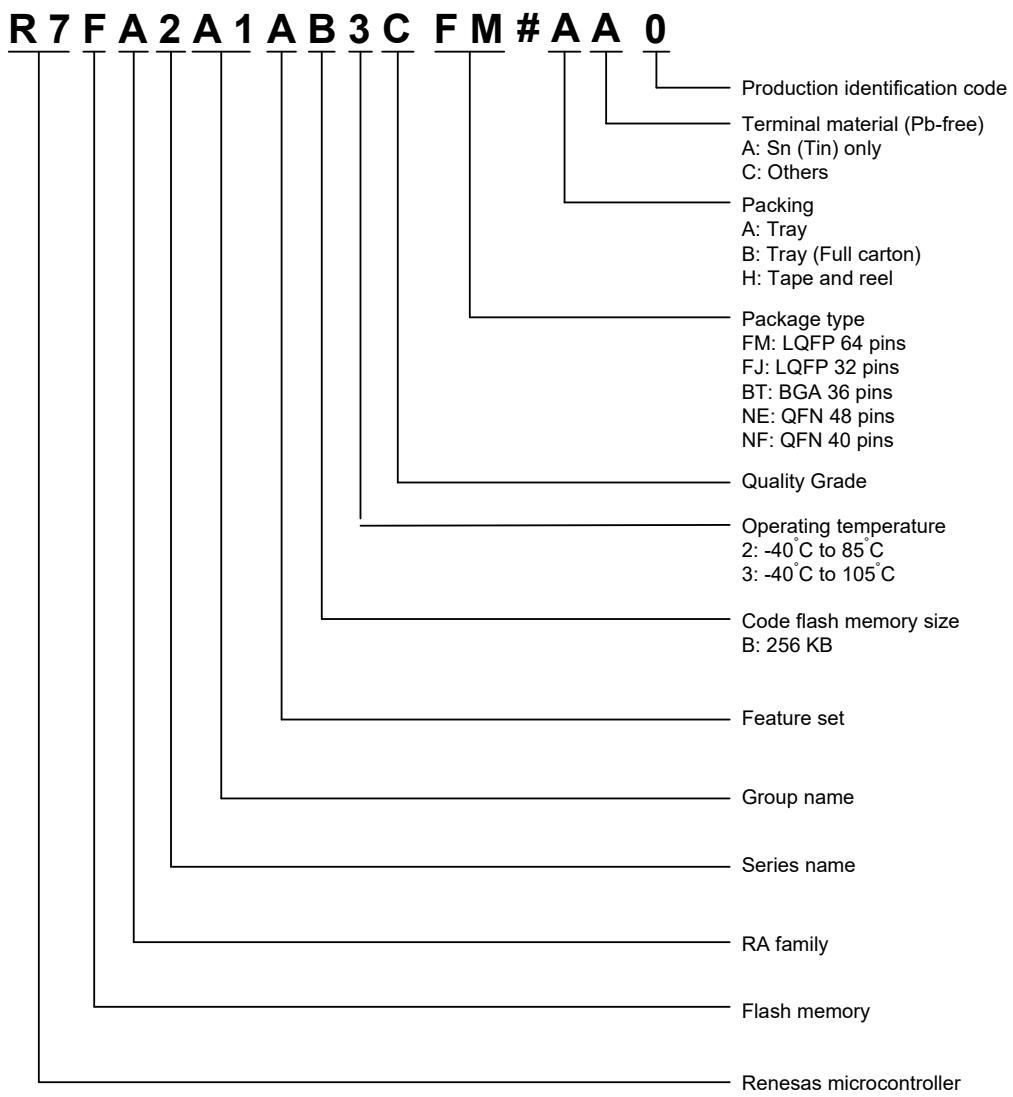


Figure 1.1 Block diagram

1.3 Part Numbering

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



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

Figure 1.2 Part numbering scheme

Table 1.13 Product list

Product part number	Package code	Code flash	Data flash	SRAM	Operating temperature
R7FA2A1AB3CFM	PLQP0064KB-C	256 KB	8 KB	32 KB	-40 to +105°C
R7FA2A1AB3CNE	PWQN0048KB-A				-40 to +105°C
	PWQN0048KC-A				
R7FA2A1AB3CNF	PWQN0040KC-A				-40 to +105°C
	PWQN0040KD-A				
R7FA2A1AB2CBT	PLBG0036GA-A				-40 to +85°C
R7FA2A1AB3CFJ	PLQP0032GB-A				-40 to +105°C

1.4 Function Comparison

Table 1.14 Function comparison

Part numbers		R7FA2A1AB3CFM	R7FA2A1AB3CNE	R7FA2A1AB3CNF	R7FA2A1AB2CBT	R7FA2A1AB3CFJ
Pin count		64	48	40	36	32
Package		LQFP	QFN	QFN	BGA	LQFP
Code flash memory		256 KB				
Data flash memory		8 KB				
SRAM		32 KB				
	Parity	16 KB				
	ECC	16 KB				
System	CPU clock	48 MHz				
	Sub-clock oscillator	Yes				No
	ICU	Yes				
	KINT	8	6	4	4	3
Event control	ELC	Yes				
DMA	DTC	Yes				
Timers	GPT32	1				
	GPT16	6	6	4	3	4
	AGT	2				
	RTC	Yes				
	WDT/IWDT	Yes				
Communication	SCI	3				
	IIC	2				
	SPI	2			1	2
	CAN	Yes				
	USBFS	Yes				No
Analog	ADC16	17 (4*1)	12 (3*1)	8 (1*1)	5 (1*1)	5 (1*1)
	SDADC24	8 (4*1)	6 (3*1)	4 (2*1)	2 (1*1)	2 (1*1)
	DAC12	1				
	DAC8	2	2*2		2*3	
	ACMPHS	1				
	ACMPLP	2				
	OPAMP	3	2	1	1	1
	TSN	Yes				
HMI	CTSU	26	16	11	9	11
Data processing	CRC	Yes				
	DOC	Yes				
Security		AES and TRNG				
I/O ports	I/O pins	46	30	22	19	19
	Input pins	3	3	3	3	1
	Pull-up resistors	44	28	20	17	19
	N-ch open-drain outputs	24	17	13	13	15
	5-V tolerance	9	6	3	3	4

Note 1. The number of channels of the differential analog input.
Note 2. Pin output function of DA8_1 cannot be used.
Note 3. Pin output function of DA8_0 and DA8_1 cannot be used.

1.5 Pin Functions

Table 1.15 Pin functions (1 of 4)

Function	Signal	I/O	Description
Power supply	VCC	Input	Power supply pin. Connect this pin to the system power supply. Connect it to VSS by a 0.1-μF capacitor. Place the capacitor close to the pin.
	VCL	I/O	Connect this pin to VSS through a smoothing capacitor used to stabilize the internal power supply. Place the capacitor close to the pin.
	VSS	Input	Ground pin. Connect to the system power supply (0 V).
Clock	XTAL	Output	Pins for a crystal resonator. An external clock signal can be input through the EXTAL pin.
	EXTAL	Input	
	XCIN	Input	Input/output pins for the sub-clock oscillator. Connect a crystal resonator between XCOUT and XCIN.
	XCOUT	Output	
	CLKOUT	Output	Clock output pin
Operating mode control	MD	Input	Pins for setting the operating mode. The signal level on this pin must not be changed during operation mode transition on release from the reset state.
System control	RES	Input	Reset signal input pin. The MCU enters the reset state when this signal goes low.
CAC	CACREF	Input	Measurement reference clock input pin
On-chip debug	SWDIO	I/O	Serial wire debug data input/output pin
	SWCLK	Input	Serial wire clock pin
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQ0 to IRQ7	Input	Maskable interrupt request pins
GPT	GTETRGA, GTETRGB	Input	External trigger input pin
	GTIOC0A to GTIOC6A, GTIOC0B to GTIOC6B	I/O	Input capture, output compare, 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
	AGTOA0, AGTOA1	Output	Output compare match A output
	AGTOB0, AGTOB1	Output	Output compare match B output
RTC	RTCOUT	Output	Output pin for 1-Hz/64-Hz clock

Table 1.15 Pin functions (2 of 4)

Function	Signal	I/O	Description
SCI	SCK0, SCK1, SCK9	I/O	Input/output pins for the clock (clock synchronous mode)
	RXD0, RXD1, RXD9	Input	Input pins for received data (asynchronous mode/clock synchronous mode)
	TXD0, TXD1, TXD9	Output	Output pins for transmitted data (asynchronous mode/clock synchronous mode)
	CTS0_RTS0, CTS1_RTS1, CTS9_RTS9	I/O	Input/output pins for controlling the start of transmission and reception (asynchronous mode/clock synchronous mode), active-low
	SCL0, SCL1, SCL9	I/O	Input/output pins for the IIC clock (simple IIC)
	SDA0, SDA1, SDA9	I/O	Input/output pins for the IIC data (simple IIC)
	SCK0, SCK1, SCK9	I/O	Input/output pins for the clock (simple SPI)
	MISO0, MISO1, MISO9	I/O	Input/output pins for slave transmission of data (simple SPI)
	MOSI0, MOSI1, MOSI9	I/O	Input/output pins for master transmission of data (simple SPI)
	SS0, SS1, SS9	Input	Chip-select input pins (simple SPI), active-low
IIC	SCL0, SCL1	I/O	Input/output pins for clock
	SDA0, 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 to SSLA3, SSLB1 to 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 LDO regulator
	VCC_USB	I/O	Input: Power supply pin for USB transceiver. Output: USB LDO regulator output pin. This pin should be connected to an external capacitor.
	USB_DP	I/O	D+ I/O pin of the USB on-chip transceiver. 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.

Table 1.15 Pin functions (3 of 4)

Function	Signal	I/O	Description
Analog power supply	AVCC0	Input	Analog voltage supply pin for the ADC16, DAC12, DAC8, ACMPHS, ACMPLP, and OPAMP
	AVSS0	Input	Analog ground pin for the ADC16, DAC12, DAC8, ACMPHS, ACMPLP, and OPAMP
	AVCC1	Input	Analog voltage supply pin for the SDADC24
	AVSS1	Input	Analog ground pin for the SDADC24
	VREFH0	Input	Analog reference voltage supply pin for the ADC16. Connect this pin to AVCC0 when not using the ADC16.
	VREFL0	Input	Analog reference ground pin for the ADC16. Connect this pin to AVSS0 when not using the ADC16.
	VREFH	Input	Analog reference voltage supply pin for the DAC12
	VREFL	Input	Analog reference ground pin for the DAC12
ADC16	AN000 to AN008, AN016 to AN023	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
SDADC24	ANSD0P to ANSD3P	Input	Input pins for the analog signals to be processed by the SDADC24
	ANSD0N to ANSD3N	Input	Input pins for the analog signals to be processed by the SDADC24
	ADREG	Output	Regulator capacitance for the SDADC24
	SBIAS	Output	Sensor power supply
	VREFI	Input	External reference voltage supply pin for the SDADC24
DAC12	DA12_0	Output	Output pin for the analog signals to be processed by the 12-bit D/A converter
DAC8	DA8_0, DA8_1	Output	Output pins for the analog signals to be processed by the 8-bit D/A converter
Comparator output	VCOUT	Output	Comparator output pin
ACMPHS	IVREF0 to IVREF2	Input	Reference voltage input pin
	IVCMP0 to IVCMP2	Input	Analog voltage input pin
ACMPLP	CMPREF0, CMPREF1	Input	Reference voltage input pins
	CMPIN0, CMPIN1	Input	Analog voltage input pins
OPAMP	AMP0+ to AMP2+	Input	Analog voltage input pins
	AMP0- to AMP2-	Input	Analog voltage input pins
	AMP0O to AMP2O	Output	Analog voltage output pins
CTSU	TS00 to TS25	Input	Capacitive touch detection pins (touch pins)
	TSCAP	-	Secondary power supply pin for the touch driver
KINT	KR00 to KR07	Input	Key interrupt input pins

Table 1.15 Pin functions (4 of 4)

Function	Signal	I/O	Description
I/O ports	P000 to P003, P012 to P015	I/O	General-purpose input/output pins
	P100 to P112	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 to P304	I/O	General-purpose input/output pins
	P400 to P403, P407 to P411	I/O	General-purpose input/output pins
	P500 to P502	I/O	General-purpose input/output pins
	P914, P915	I/O	General-purpose input/output pins

1.6 Pin Assignments

Figure 1.3 to Figure 1.7 show the pin assignments.

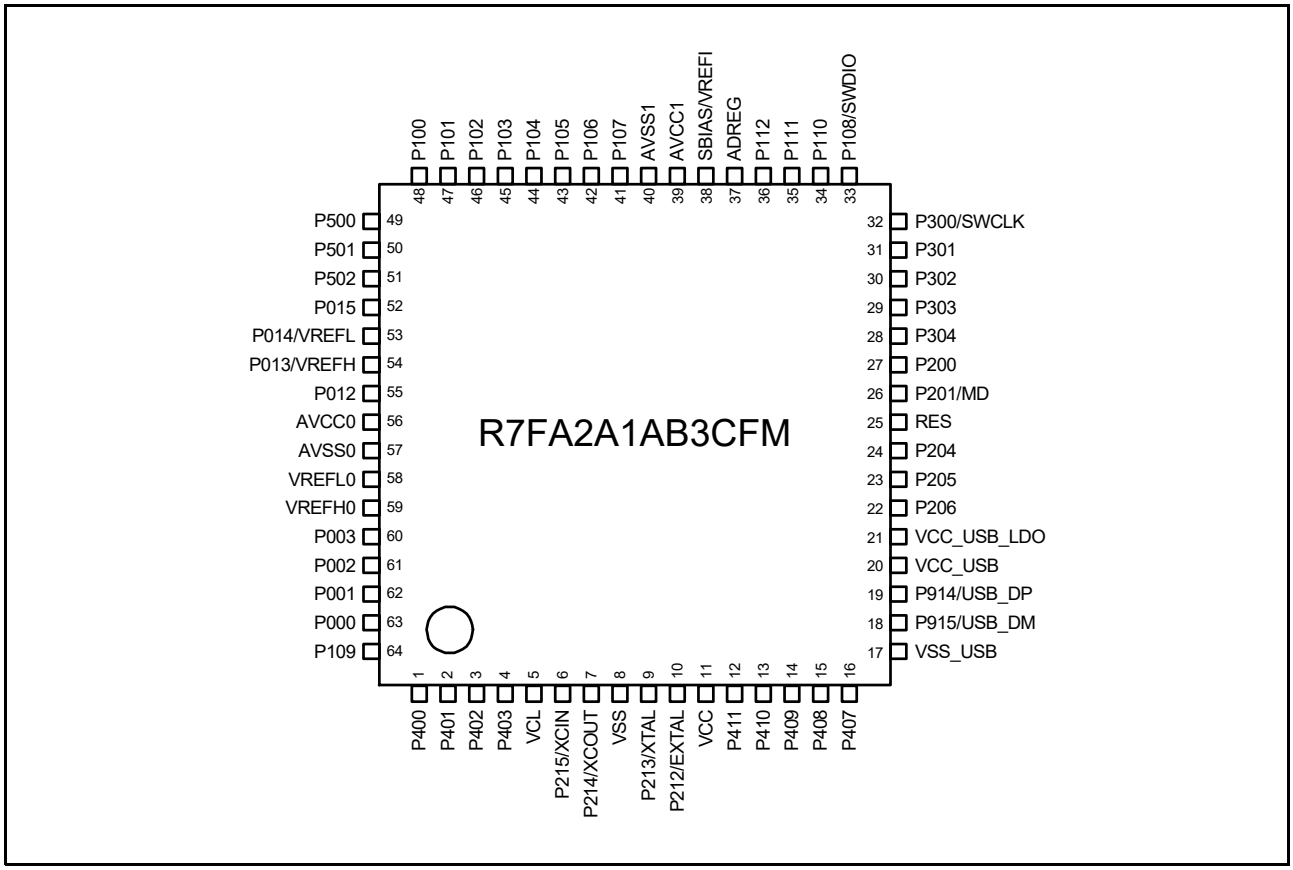
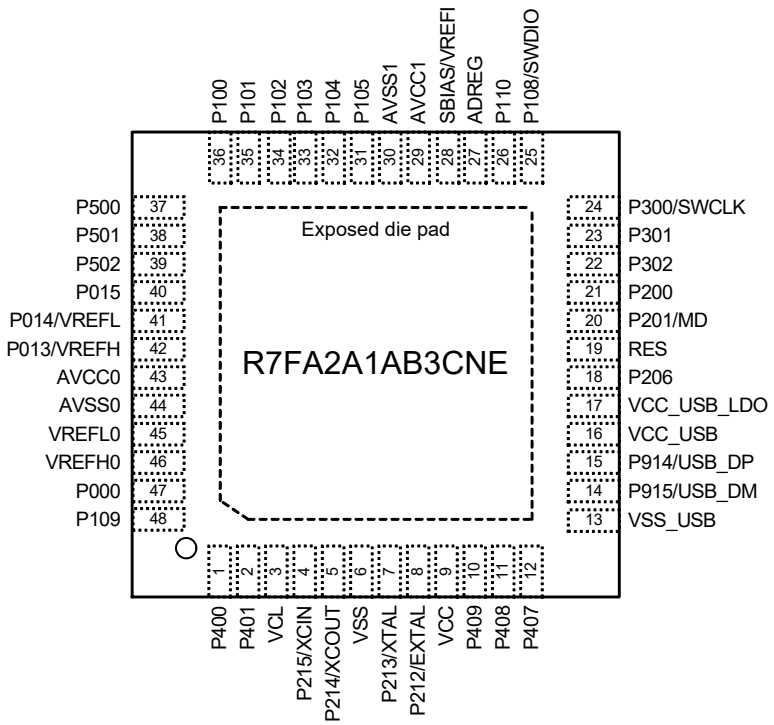
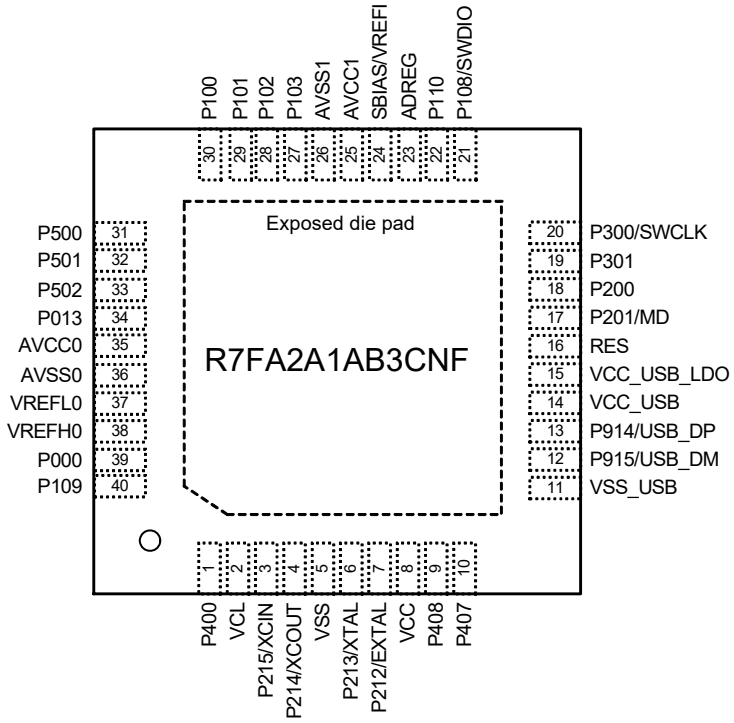


Figure 1.3 Pin assignment for LQFP 64-pin



Note: Exposed die pad is recommended to connect to VSS.

Figure 1.4 Pin assignment for QFN 48-pin



Note: Exposed die pad is recommended to connect to VSS.

Figure 1.5 Pin assignment for QFN 40-pin

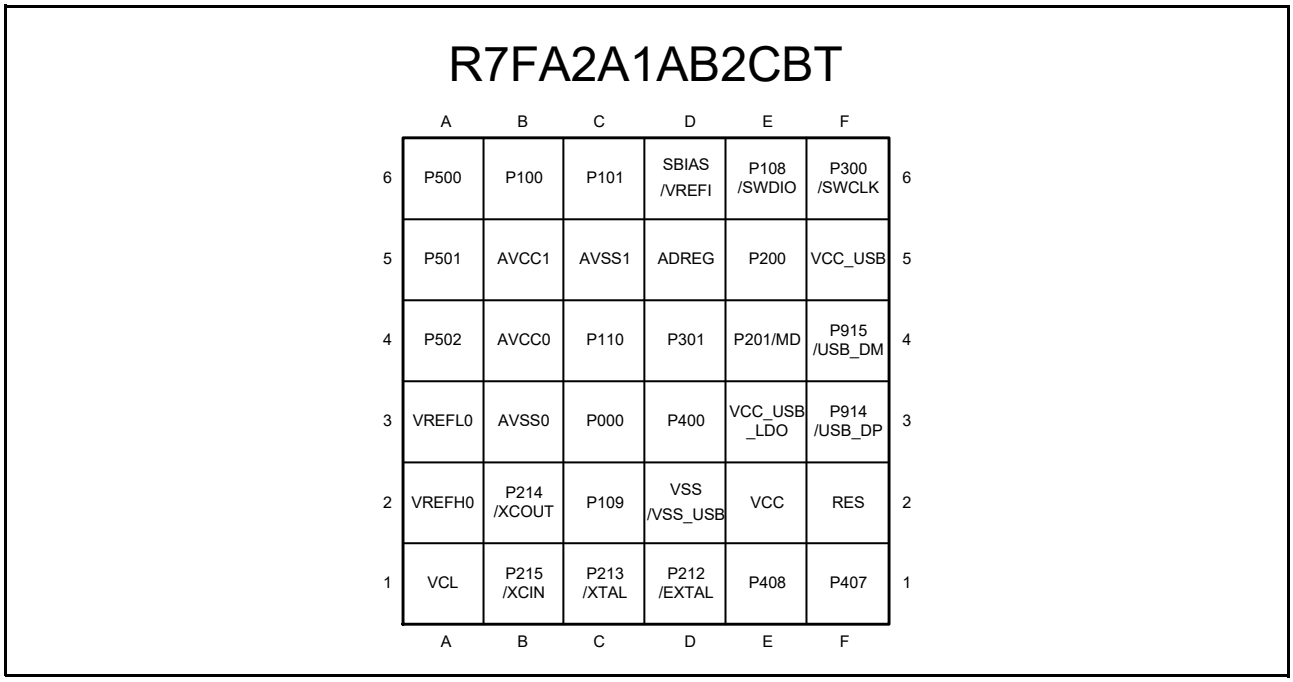


Figure 1.6 Pin assignment for BGA 36-pin (top view, pad side down)

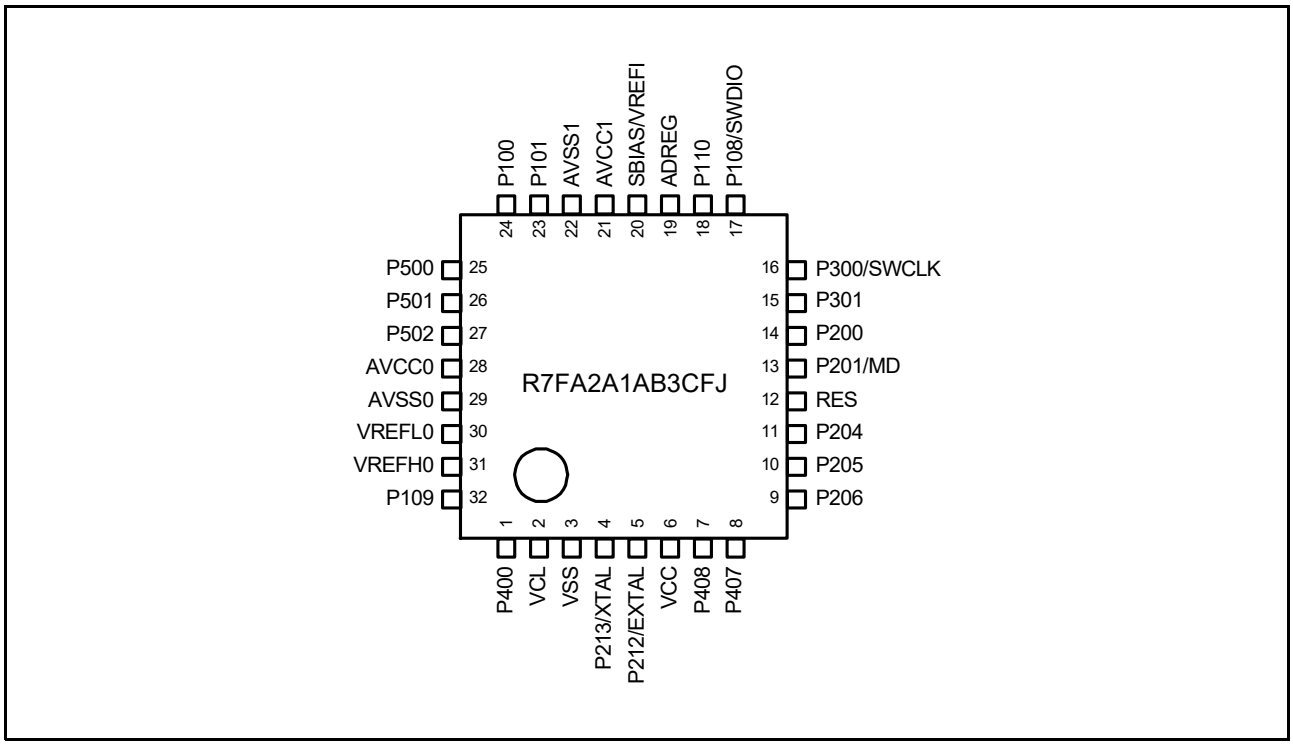


Figure 1.7 Pin assignment for LQFP 32-pin

1.7 Pin Lists

Pin number					Power, System, Clock, Debug, CAC	I/O ports	Timers				Communication Interfaces				Analogs					HMI	
LQFP64	QFN48	QFN40	BGA36	LQFP32			AGT	GPT_OPS, POEG	GPT	RTC	USBFS, CAN	SCI	IIC	SPI	ADC16	SDADC24	DAC12, DAC8	ACMPHS, ACMPLP	OPAMP	CTSU	Interrupt
1	1	1	D3	1		P400	AGTEE0_A	GTETRGA_A	GTIOC1A_A	RTCOUT_C		CTS0_RTS0_D/ SS0_D/ RXD1_C/ MISO1_C/ SCL1_C	SDA1_A	MOSIA_A				CMPIN0		TS00	KR02/ IRQ0_A
2	2	-	-	-		P401	AGTEE1_A	GTIU_A	GTIOC4A_A			SCK0_D/ SCK9_A	SDA0_C	SSLB1_A				VCOUT_B		TS01	KR03/ IRQ5_B
3	-	-	-	-		P402		GTIV_A	GTIOC0A_D			CTS9_RTS9_C/ SS9_C		SSLB2_A						TS02	
4	-	-	-	-		P403		GTIW_A	GTIOC0B_C			SCK1_B		SSLB3_A						TS03	
5	3	2	A1	2	VCL																
6	4	3	B1	-	XCIN	P215															
7	5	4	B2	-	XCOUT	P214															
8	6	5	D2	3	VSS																
9	7	6	C1	4	XTAL	P213	AGTEE1_B	GTETRGA_B	GTIOC0A_B			RXD1_D/ MISO1_D/ SCL1_D									IRQ2_B
10	8	7	D1	5	EXTAL	P212	AGTIO0_A	GTETRGB_B	GTIOC0B_B			TXD1_D/ MOSI1_D/ SDA1_D									IRQ3_B
11	9	8	E2	6	VCC																
12	-	-	-	-		P411			GTIOC5A_A			TXD0_F/ MOSI0_F/ SDA0_F/ RXD1_B/ MISO1_B/ SCL1_B		SSLA3_A						TS04	
13	-	-	-	-		P410			GTIOC5B_A			CTS0_RTS0_A/ SS0_A/ TXD1_B/ MOSI1_B/ SDA1_B		SSLA2_A						TS05	
14	10	-	-	-		P409	AGTO1_A		GTIOC0A_C		CTX0_B	SCK0_A/ CTS1_RTS1_B/ SS1_B	SCL0_B	SSLA1_A						TSCAP_E	IRQ7_A
15	11	9	E1	7		P408	AGTO0_A	GTOUUP_A	GTIOC0A_A		CRX0_B	RXD0_A/ MISO0_A/ SCL0_A/ TXD1_C/ MOSI1_C/ SDA1_C	SDA0_B	SSLA0_A				CMPIN1		TS06	IRQ1_A
16	12	10	F1	8	CACREF_B	P407	AGTIO0_C	GTOULO_A	GTIOC0B_A		USB_VBUS/ CTX0_D	TXD0_A/ MOSI0_A/ SDA0_A/ TXD9_A/ MOSI9_A/ SDA9_A	SCL0_A	RSPCKB_B						TSCAP_D	IRQ1_B
17	13	11	D2	-	VSS_USB																
18	14	12	F4	-		P915					USB_DM										
19	15	13	F3	-		P914					USB_DP										
20	16	14	F5	-	VCC_USB																
21	17	15	E3	-	VCC_USB_LDO																
22	18	-	-	9		P206	AGTIO0_B	GTOVUP_A	GTIOC3A_A			CTS0_RTS0_C/ SS0_C/ TXD1_A/ MOSI1_A/ SDA1_A	SCL1_B	SSLB0_A						TS07	IRQ6_A
23	-	-	-	10		P205		GTOVLO_A	GTIOC3B_A			TXD0_C/ MOSI0_C/ SDA0_C/ CTS1_RTS1_A/ SS1_A	SDA1_B	MISOB_B						TS08	IRQ0_C
24	-	-	-	11		P204						RXD0_C/ MISO0_C/ SCL0_C/ SCK9_B		MOSIB_B						TS09	
25	19	16	F2	12	RES																
26	20	17	E4	13	MD	P201															
27	21	18	E5	14		P200															NMI
28	-	-	-	-		P304			GTIOC6A_A		CTX0_A	SCK0_B/ TXD9_C/ MOSI9_C/ SDA9_C		MISOA_B						TS10	KR07
29	-	-	-	-		P303			GTIOC6B_A		CRX0_A	CTS0_RTS0_B/ SS0_B/ SCK1_A		MOSIA_B						TS11	KR06
30	22	-	-	-	CACREF_A	P302	AGTOA1_A	GTOVLO_B	GTIOC3B_B			TXD0_B/ MOSI0_B/ SDA0_B/ RXD1_A/ MISO1_A/ SCL1_A		RSPCKB_A						TS12	KR05/ IRQ4_B

Pin number					Power, System, Clock, Debug, CAC	I/O ports	Timers				Communication Interfaces				Analogs					HMI	
LQFP64	QFN48	QFN40	BGA36	LQFP32			AGT	GPT_OPS, POEG	GPT	RTC	USBFS, CAN	SCI	IIC	SPI	ADC16	SDADC24	DAC12, DAC8	ACMPHS, ACMPLP	OPAMP	CTSU	Interrupt
31	23	19	D4	15		P301	AGTOB1_A	GTOWUP_A	GTIOC2A_B	RTCOUT_A		RXD0_B/ MISO0_B/ SCL0_B/ CTS9_RTS9_B/ SS9_B	SDA0_A	MOSIB_A						TS13	KR04/ IRQ5_A
32	24	20	F6	16	SWCLK	P300															
33	25	21	E6	17	SWDIO	P108															
34	26	22	C4	18	CLKOUT_A	P110	AGTOB0_A	GTOWLO_A	GTIOC2B_B		CTX0_C	TXD0_D/ MOSI0_D/ SDA0_D/ RXD9_B/ MISO9_B/ SCL9_B	SDA1_D	RSPCKA_A	ADTRG0_A			CMPREF1		TSCAP_A	IRQ2_A
35	-	-	-	-		P111				RTCOUT_B			SCL1_C	RSPCKA_B						TS14	IRQ6_B
36	-	-	-	-	CLKOUT_B	P112							SDA1_C	SSLA0_B						TSCAP_B	IRQ7_B
37	27	23	D5	19	ADREG																
38	28	24	D6	20	SBIAS/ VREFI																
39	29	25	B5	21	AVCC1																
40	30	26	C5	22	AVSS1																
41	-	-	-	-		P107									AN023	ANSD3N					
42	-	-	-	-		P106									AN022	ANSD3P					
43	31	-	-	-		P105								MOSIB_C	AN021	ANSD2N				TS18	IRQ7_C
44	32	-	-	-		P104								MISOB_C	AN020	ANSD2P				TS19	IRQ6_C
45	33	27	-	-		P103			GTIOC6A_B					RSPCKB_C	AN019	ANSD1N				TS20	
46	34	28	-	-		P102			GTIOC6B_B			CTS9_RTS9_D/ SS9_D		SSLB0_C	AN018	ANSD1P				TS21	
47	35	29	C6	23		P101			GTIOC5A_B			RXD9_C/ MISO9_C/ SCL9_C			AN017	ANSD0N		IVREF2		TS22	IRQ5_C
48	36	30	B6	24		P100			GTIOC5B_B			TXD9_D/ MOSI9_D/ SDA9_D			AN016	ANSD0P		IVCMP2		TS23	IRQ4_C
49	37	31	A6	25		P500			GTIOC5A_C			RXD0_D/ MISO0_D/ SCL0_D			AN000		DA12_0	IVCMP0	AMP0+	TS24	IRQ3_C
50	38	32	A5	26		P501			GTIOC5B_C			TXD0_E/ MOSI0_E/ SDA0_E			AN001			IVREF0	AMP0-	TS25	IRQ2_C
51	39	33	A4	27		P502						CTS0_RTS0_E/ SS0_E			AN002				AMP0O		IRQ1_C
52	40	-	-	-		P015									AN003				AMP1O		
53	41	-	-	-	VREFL	P014			GTIOC6A_C						AN004			IVREF1	AMP1-		
54	42	34	-	-	VREFH	P013			GTIOC6B_C						AN005		DA8_0	IVCMP1	AMP1+		
55	-	-	-	-		P012									AN008				AMP2O		
56	43	35	B4	28	AVCC0																
57	44	36	B3	29	AVSS0																
58	45	37	A3	30	VREFL0																
59	46	38	A2	31	VREFH0																
60	-	-	-	-		P003									AN006				AMP2-		
61	-	-	-	-		P002									AN007		DA8_1		AMP2+		
62	-	-	-	-		P001				RTCOUT_D		CTS9_RTS9_A/ SS9_A		RSPCKB_D						TS15	IRQ0_B
63	47	39	C3	-		P000	AGTIO1_A		GTIOC4B_B			RXD9_A/ MISO9_A/ SCL9_A	SCL0_C	MISOB_A						TS16	KR00/ IRQ4_A
64	48	40	C2	32		P109	AGTOA0_A	GTETRGB_A	GTIOC1B_B			SCK0_C/ TXD9_B/ MOSI9_B/ SDA9_B	SCL1_A	MISOA_A	ADTRG0_B			CMPREF0/ VCOUT_A		TS17	KR01/ IRQ3_A

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

2. CPU

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

2.1 Overview

2.1.1 CPU

- Arm Cortex-M23
  - Revision: r1p0-00rel0
  - Armv8-M architecture profile
  - Single-cycle integer multiplier
  - 17-cycle integer divider.
- Memory Protection Unit (MPU)
  - Armv8 Protected Memory System Architecture
  - 8 protected regions.
- SysTick timer
  - Driven by SYSTICCLK (LOCO) or ICLK.

See reference 1. and reference 2. in section 2.9 for details.

2.1.2 Debug

- Arm CoreSight™ MTB-M23
  - Revision: r0p0-00rel0
  - Buffer size: 1 KB of 16-KB MTB SRAM.
- Data Watchpoint Unit (DWT)
  - 2 comparators for watchpoints.
- Flash Patch and Breakpoint Unit (FPB)
  - 4 instruction comparators.
- CoreSight Debug Access Port (DAP)
  - Serial Wire-Debug Port (SW-DP).
- Debug Register Module (DBGREG)
  - Reset control
  - Halt control.

See reference 1. and reference 2. in section 2.9 for details.

2.1.3 Operating Frequency

The operating frequencies for the MCU are as follows:

- CPU: maximum 48 MHz
- Serial Wire Debug (SWD) interface: maximum 12.5 MHz.

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

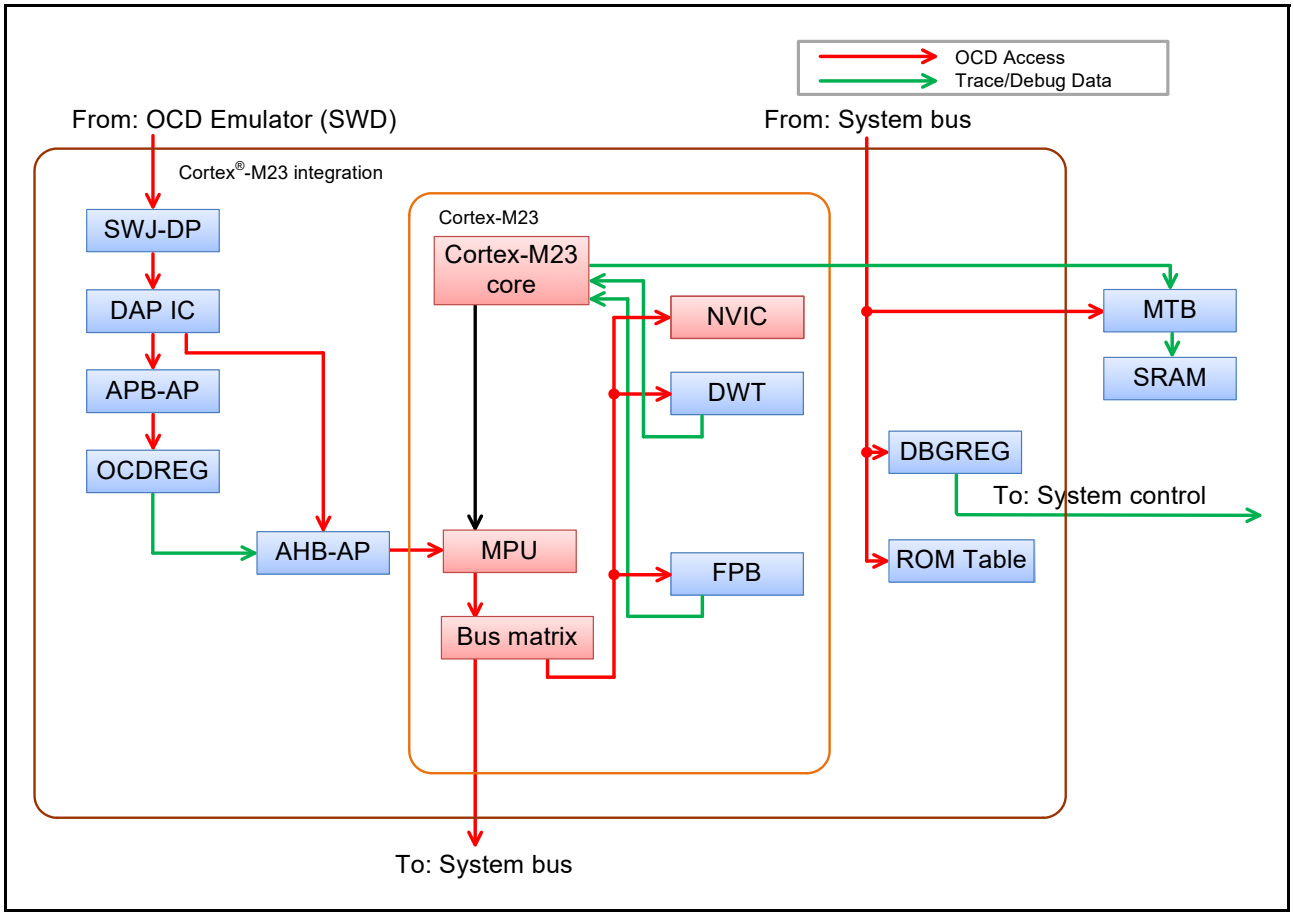


Figure 2.1 Cortex-M23 CPU block diagram

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 (1 of 2)

Option	Implementation
Non-secure MPU	Included, 8 protect regions
Secure MPU	Not included
Security Extension	Not included
Single-cycle multiplier	Included
Divider	Included, 17 cycles
Number of interrupts	32
Number of Wakeup Interrupt Controllers (WIC)	Not included
Cross Trigger Interface (CTI)	Not included
Micro Trace Buffer (MTB)	Included
Embedded Trace Macrocell (ETM)	Not included
Multi-drop support for serial wire	Not supported
Sleep mode power saving	Sleep mode and other low power modes are supported. For more details, see section 11, Low Power Modes. SCB.SCR.SLEEPDEEP is ignored.
Endianness	Little-endian

Table 2.1 Implementation options (2 of 2)

Option	Implementation
SysTick SYST_CALIB register	Included SYST_CALIB = 4000 0147h Bit [31] = 0 Reference clock provided Bit [30] = 1 TERMS value is inexact Bits [29:24] = 00h Reserved Bits [23:0] = 000147h TERM: (32768 × 10 ms) - 1 / 32.768 kHz = 326.66 decimal = 327 with skew = 000147h
Event input/output	Not implemented
System reset request output	The SYSRESETREQ bit in the Application Interrupt and Reset Control Register causes a CPU reset
Auxiliary fault inputs (AUXFAULT)	Not implemented

2.3 Trace Interface

The MCU does not provide a dedicated trace output interface.

2.4 SWD Interface

Table 2.2 shows the SWD pins.

Table 2.2 SWD pins

Name	I/O	P/N	Width	Function	When not in use
SWCLK	Input	Positive	1 bit	SWD clock pin	Pull-up
SWDIO	I/O	Negative	1 bit	SWD I/O pin	Pull-up

2.5 Debug Mode

2.5.1 Debug Mode Definition

In single chip mode, the debugger connection state is defined as On-Chip Debugger (OCD) mode, and the non-connected debugger state is defined as User mode.

Table 2.3 shows the CPU debug modes and usage conditions.

Table 2.3 CPU debug mode and conditions

Conditions		Mode	
OCD connect	**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 Armv8-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 2.6.6.3, MCU Control Register (MCUCTRL).

2.5.2.2 Reset

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

Table 2.4 Reset or interrupt and mode setting

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

Note: In OCD break mode, the CPU is halted. In OCD run mode, the CPU is in OCD mode and the CPU is not halted.
Note 1. The IWDT and WDT always stop in this mode.
Note 2. The IWDT and WDT operation depends on the DBGSTOPCTR setting.
Note 3. Reset or interrupt masking depends on the DBGSTOPCTR 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.

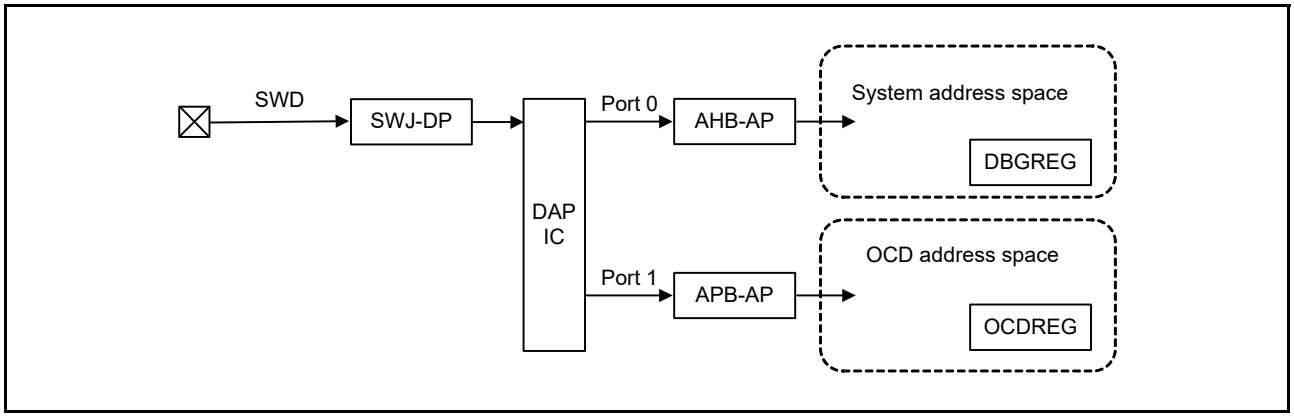


Figure 2.2 SWD authentication block diagram

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-M23 Peripheral Address Map

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

Table 2.5 Cortex-M23 peripheral address map

Component name	Start address	End address	Note
DWT	E000 1000h	E000 1FFFh	See reference 2.
FPB	E000 2000h	E000 2FFFh	See reference 2.
SCS	E000 E000h	E000 EFFFh	See reference 2.

2.6.3 External Debug Address Map

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

Table 2.6 External debug address map

Component name	Start address	End address	Note
MTB (SRAM area)	2000 4000h	2000 7FFFh	MTB uses 1 KB of the 16 KB as trace buffer See reference 5.
MTB (SFR area)	4001 9000h	4001 9FFFh	See reference 5.
ROM Table	4001 A000h	4001 AFFFh	See reference 5.

2.6.4 CoreSight ROM Table

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

2.6.4.1 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 4. for details.

Table 2.7 CoreSight ROM Table

#	Address	Access size	R/W	Value	Target module pointer
0	4001 A000h	32 bits	R	9FFF 4003h	SCS
1	4001 A004h	32 bits	R	9FFE 7003h	DWT
2	4001 A008h	32 bits	R	9FFE 8003h	FPB
3	4001 A00Ch	32 bits	R	FFFF F003h	MTB
4	4001 A010h	32 bits	R	0000 0000h	End of entries

2.6.4.2 CoreSight component registers

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

Table 2.8 shows the registers. See reference 4. for details of each register.

Table 2.8 CoreSight component registers in the CoreSight ROM Table

Name	Address	Access size	R/W	Initial value
DEVTYPE	E00F FFCCh	32 bits	R	0000 0001h
PID4	E00F FFD0h	32 bits	R	0000 0004h
PID5	E00F FFD4h	32 bits	R	0000 0000h
PID6	E00F FFD8h	32 bits	R	0000 0000h
PID7	E00F FFDCh	32 bits	R	0000 0000h
PID0	E00F FFE0h	32 bits	R	0000 001Bh
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

2.6.5 DBGREG Module

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

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

Table 2.9 Non-CoreSight DBGREG registers

Name		DAP port	Address	Access size	R/W
Debug Status Register	DBGSTR	Port 0	4001 B000h	32 bits	R
Debug Stop Control Register	DBGSTOPCTR	Port 0	4001 B010h	32 bits	R/W

2.6.5.1 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

2.6.5.2 Debug Stop Control Register (DBGSTOPCR)

Address(es): DBG.DBGSTOPCR 4001 B010h

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

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

Bit	Symbol	Bit name	Description	R/W
b0	DBGSTOP_IWDT	Mask bit for IWDT reset/interrupt in the OCD run mode	In the OCD break mode, the reset/interrupt is masked and IWDT counter is stopped, regardless of this bit value. 0: Enable IWDT reset/interrupt 1: Mask IWDT reset/interrupt and stop IWDT counter.	R/W
b1	DBGSTOP_WDT	Mask bit for WDT reset/interrupt in the OCD run mode	In the OCD break mode, the reset/interrupt is masked and WDT counter is stopped, regardless of this bit value. 0: Enable WDT reset/interrupt 1: Mask WDT reset/interrupt and stop WDT counter.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	DBGSTOP_LVD[2:0]	Mask bit for LVD0 reset	0: Enable LVD0 reset 1: Mask LVD0 reset.	R/W
b17		Mask bit for LVD1 reset/interrupt	0: Enable LVD1 reset/interrupt 1: Mask LVD1 reset/interrupt.	R/W
b18		Mask bit for LVD2 reset/interrupt	0: Enable LVD2 reset/interrupt 1: Mask LVD2 reset/interrupt.	R/W

Bit	Symbol	Bit name	Description	R/W
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	DBGSTOP_RPER	Mask bit for SRAM parity error reset/interrupt	0: Enable SRAM parity error reset/interrupt 1: Mask SRAM parity error reset/interrupt.	R/W
b25	DBGSTOP_RECCR	Mask bit for SRAM ECC error reset/interrupt	0: Enable SRAM ECC error reset/interrupt 1: Mask SRAM ECC error reset/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) controls the functional stop in OCD mode. All bits in the register are regarded as 0 when the MCU is not in OCD mode.

2.6.5.3 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 4. 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	0000 0004h
PID5	4001 BFD4h	32 bits	R	0000 0000h
PID6	4001 BFD8h	32 bits	R	0000 0000h
PID7	4001 BFDCh	32 bits	R	0000 0000h
PID0	4001 BFE0h	32 bits	R	0000 0005h
PID1	4001 BFE4h	32 bits	R	0000 0030h
PID2	4001 BFE8h	32 bits	R	0000 001Ah
PID3	4001 BFECh	32 bits	R	0000 0000h
CID0	4001 BFF0h	32 bits	R	0000 000Dh
CID1	4001 BFF4h	32 bits	R	0000 00F0h
CID2	4001 BFF8h	32 bits	R	0000 0005h
CID3	4001 BFFCh	32 bits	R	0000 00B1h

2.6.6 OCDREG Module

The OCDREG 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 of the system address map. See section 2.6.2, Cortex-M23 Peripheral Address Map.

2.6.6.1 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 SWD access is initially permitted when the ID code in the OSIS register has the initial value. See section 2.8.1, Unlock ID Code.

Address(es): IAUTH0 8000 0000h

	b31																															b0
	IAUTH0: AID 31 to 0 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): IAUTH1 8000 0100h

	b31																															b0
	IAUTH1: AID 63 to 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 to 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 to 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

2.6.6.2 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.

2.6.6.3 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.

2.6.6.4 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 4. 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	R	0000 0004h
PID5	8000 0FD4h	32 bits	R	0000 0000h
PID6	8000 0FD8h	32 bits	R	0000 0000h
PID7	8000 0FDCh	32 bits	R	0000 0000h
PID0	8000 0FE0h	32 bits	R	0000 0004h
PID1	8000 0FE4h	32 bits	R	0000 0030h
PID2	8000 0FE8h	32 bits	R	0000 000Ah
PID3	8000 0FECh	32 bits	R	0000 0000h
CID0	8000 0FF0h	32 bits	R	0000 000Dh
CID1	8000 0FF4h	32 bits	R	0000 00F0h
CID2	8000 0FF8h	32 bits	R	0000 0005h
CID3	8000 0FFCh	32 bits	R	0000 00B1h

2.7 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 clock is SYSTICCLK (LOCO), and the processor clock is ICLK.

2.8 OCD Emulator Connection

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

Figure 2.3 shows a block diagram of the authentication mechanism.

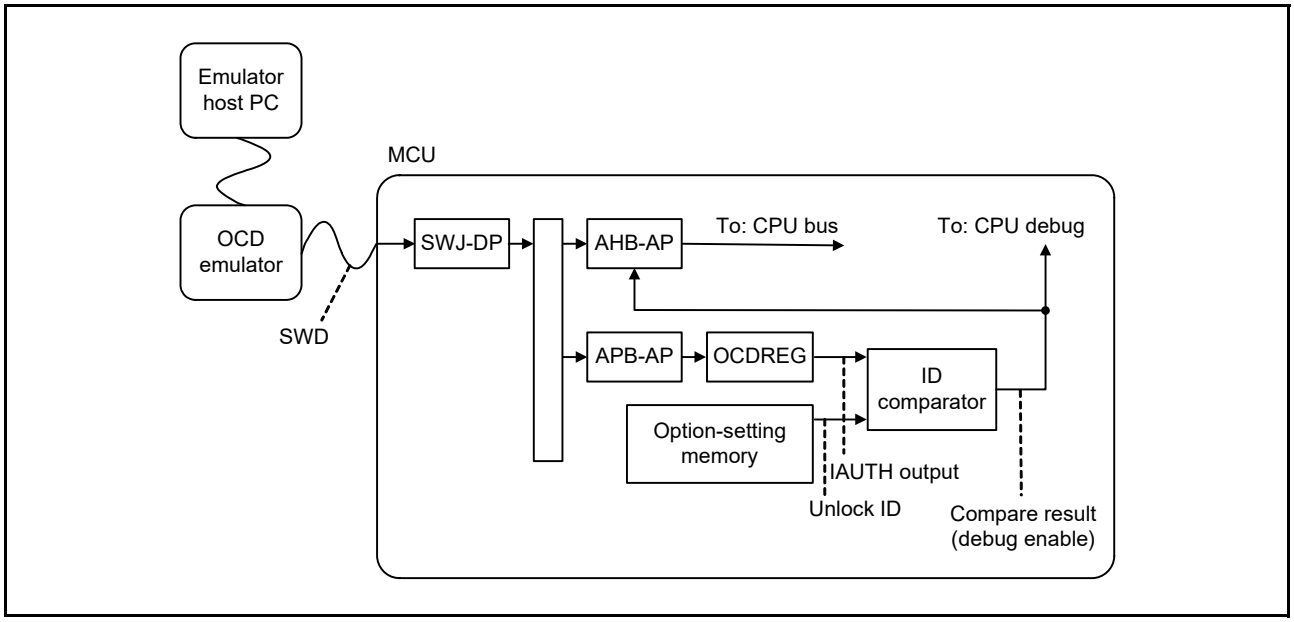


Figure 2.3 Authentication mechanism block diagram

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

2.8.1 Unlock ID Code

The unlock ID code is used for checking permission for debug and access to on-chip resources. If the unlock ID code matches the 128-bit data written in the ID Authentication Code Registers 0 to 3, the SWD debugger obtains access permission. Unlock ID code is written in the OCD/Serial Programmer ID Setting Register (OSIS) in the option-setting memory. The initial value of the unlock ID code is all 1s (FFFFFFFF_FFFFFFFF_FFFFFFFF_FFFFFFFFh). See section 7, Option-Setting Memory for details.

2.8.2 DBGEN

After the OCD emulator gets access permission, the OCD emulator must set the DBGEN bit in the System Control OCD Control Register (SYOCDCR). In addition, the OCD emulator must clear the DBGEN bit before disconnecting it. See section 11, Low Power Modes for details.

2.8.3 Restrictions on Connecting an OCD emulator

This section describes the restrictions on emulator access.

2.8.3.1 Starting connection while in low power mode

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

2.8.3.2 Changing low power mode while in OCD mode

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

Table 2.13 Restrictions by mode

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

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

2.8.3.3 Modify the unlock ID code in OSIS

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

2.8.3.4 Connecting sequence and SWD authentication

Because the OCD emulator is protected by the SWD authentication mechanism, the OCD might be required to input the ID code to the authentication registers. The OSIS value in the option-setting memory determines whether the code is required. After 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)

An emulator connection will be refused when OSIS bit [127] is set as 0 however the ALeRASE command will be accepted. When the ALeRASE command is executed, the User memory region and Option memory region are erased. The OSIS register value is also erased, so that the emulator can be connected again.

When OSIS bit [127] = 0, disabling acceptance of the ALeRASE command needs some additional settings as below.

User can select between two equivalent workarounds.

A) Setting SECMPUAC (when boot swap is set, the address of SECMPUAC shifts by 2000h.)

- Please set data as 0xFEFF at SECMPUAC
- Please set 0xFFFF_FFFC at SECMPUPCS0 and set 0xFFFF_FFFF at SECMPUPCE0.

Or

B) Setting AWSC

Please set AWSC bit [14] = 0.

AWSC bit [14] cannot be changed to 1 once it is set to 0. After clearing the AWSC bits, the access window and startup area selection options are permanently fixed and cannot be used again. In this case, the self-programming is prohibited because the startup area cannot be exchanged.

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

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

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

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

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

1. Connect the OCD debugger to the MCU through the SWD interface.
2. Set up SWJ-DP to access the DAP bus. In the setup, the OCD emulator must assert CDBGPWRUPREQ in SWJ-DP Control Status Register, then wait until CDBGPWRUPACK in the same register is asserted.
3. Set up the APB-AP to access OCDREG. The APB-AP is connected to the DAP bus port 1.
4. Write the 128-bit ID code to IAUTH registers 0 to 3 in 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] = 11b

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

When IATUH registers 0 to 3 are written with “ALeRASE” in ASCII code (414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFFh), the contents of the code flash, data flash, and configuration area are erased at once. See section 43, Flash Memory for details.

The ALeRASE sequence is as follows:

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

2.9 References

1. *ARM®v8-M Architecture Reference Manual* (ARM DDI 0553B.a).
2. *ARM® Cortex®-M23 Processor Technical Reference Manual* (ARM DDI 0550C).
3. *ARM® Cortex®-M23 Processor User Guide* (ARM DUI 0963B).
4. *ARM® CoreSight™ Architecture Specification* (ARM IHI 0029D).
5. *ARM® CoreSight™ MTB-M23 Technical Reference Manual* (ARM DDI 0564C).

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*1	Enable

Note 1. USB boot mode does not exist in 32-pin products.

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 code flash memory and data flash memory, can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (UART) SCI. For details, see section 43, Flash Memory. The MCU starts up in SCI boot mode if the MD pin is held low on release from the reset state.

3.2.3 USB Boot Mode

In this mode, the on-chip flash memory programming routine (USB boot program), stored in the boot area within the MCU, is used. The on-chip flash, including the code flash memory and data flash memory, can be modified from outside the MCU by using the USB. For details, see section 43, Flash Memory. The MCU starts in USB boot mode if the MD pin is held low on release from the reset state.

Note: USB boot mode does not exist in 32-pin products.

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.

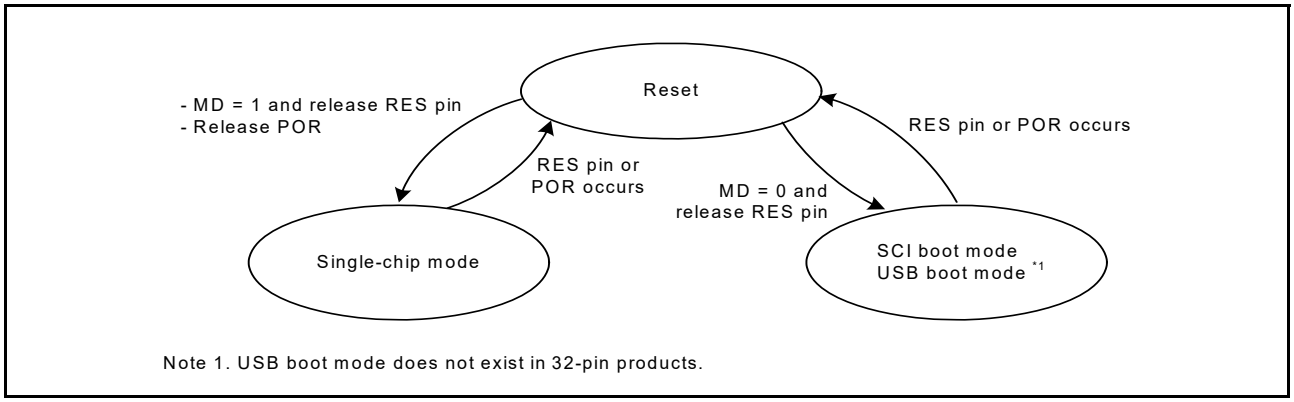
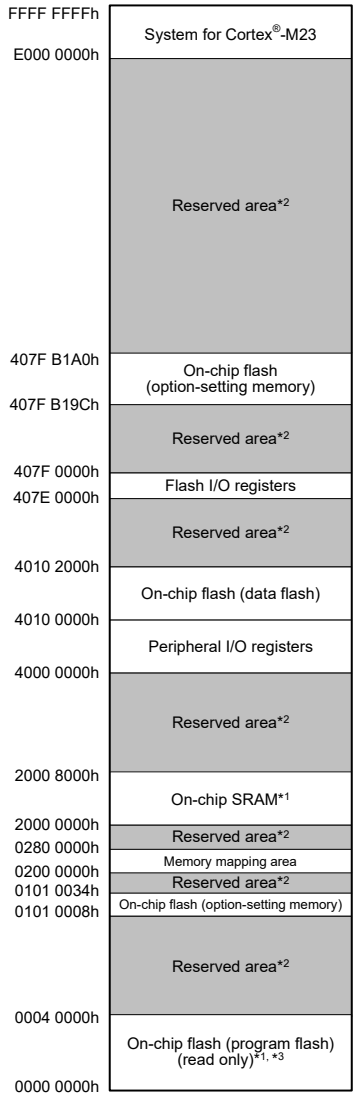


Figure 3.1 Mode-setting pin level and operating mode

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 programs and data. Figure 4.1 shows the memory map.



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

Figure 4.1 Memory map

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[15:0] 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[15:0] bits. Data written to the KEY[7:0] bits is not saved. These bits are read as 0. The KEY code and MEMMIRADDR[15:0] bits must be written 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[15:0] bits specify where the starting address of the memory mirror space addresses (0200 0000h) is linked to. Figure 5.2, Figure 5.3, and Figure 5.4 show the MMF operation. Figure 5.5 shows the setting procedure of the MMF.

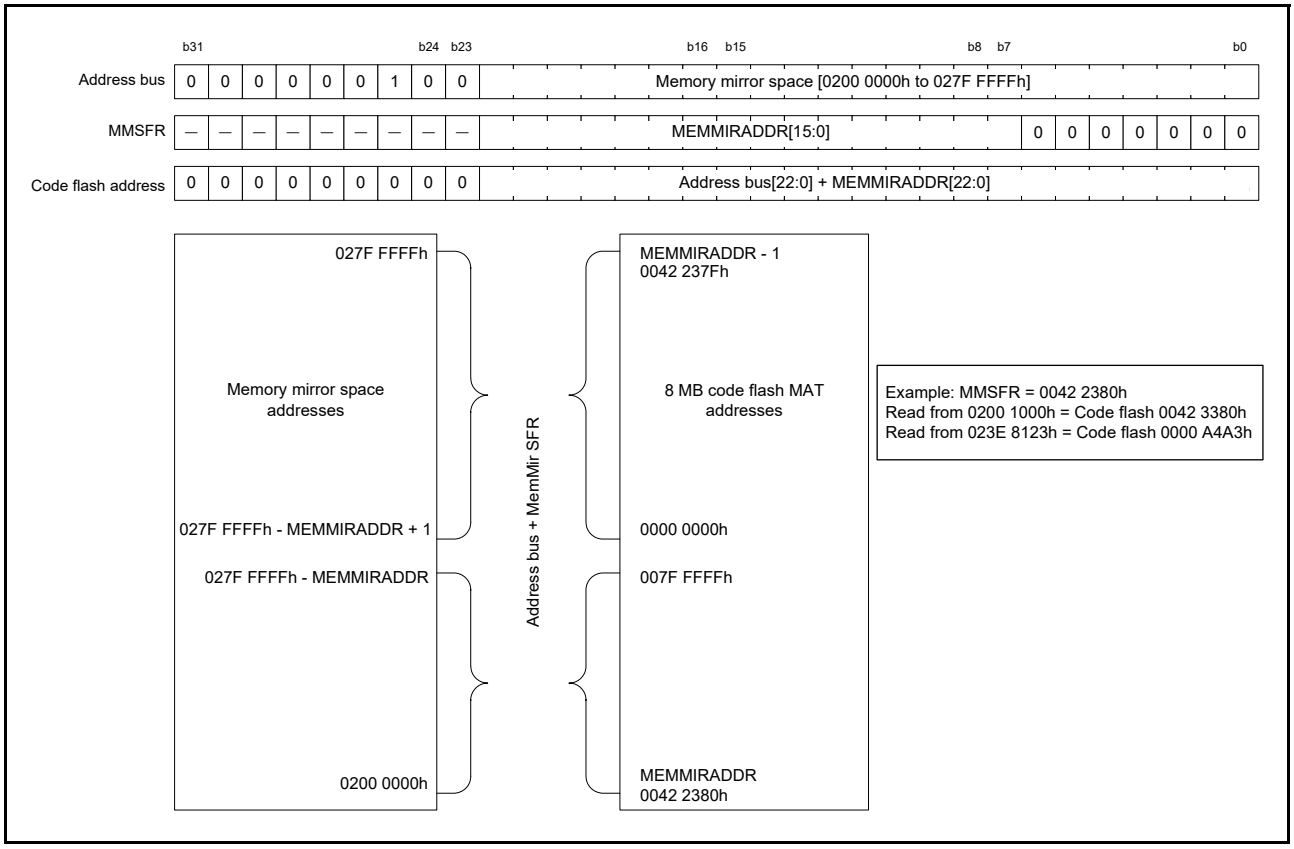
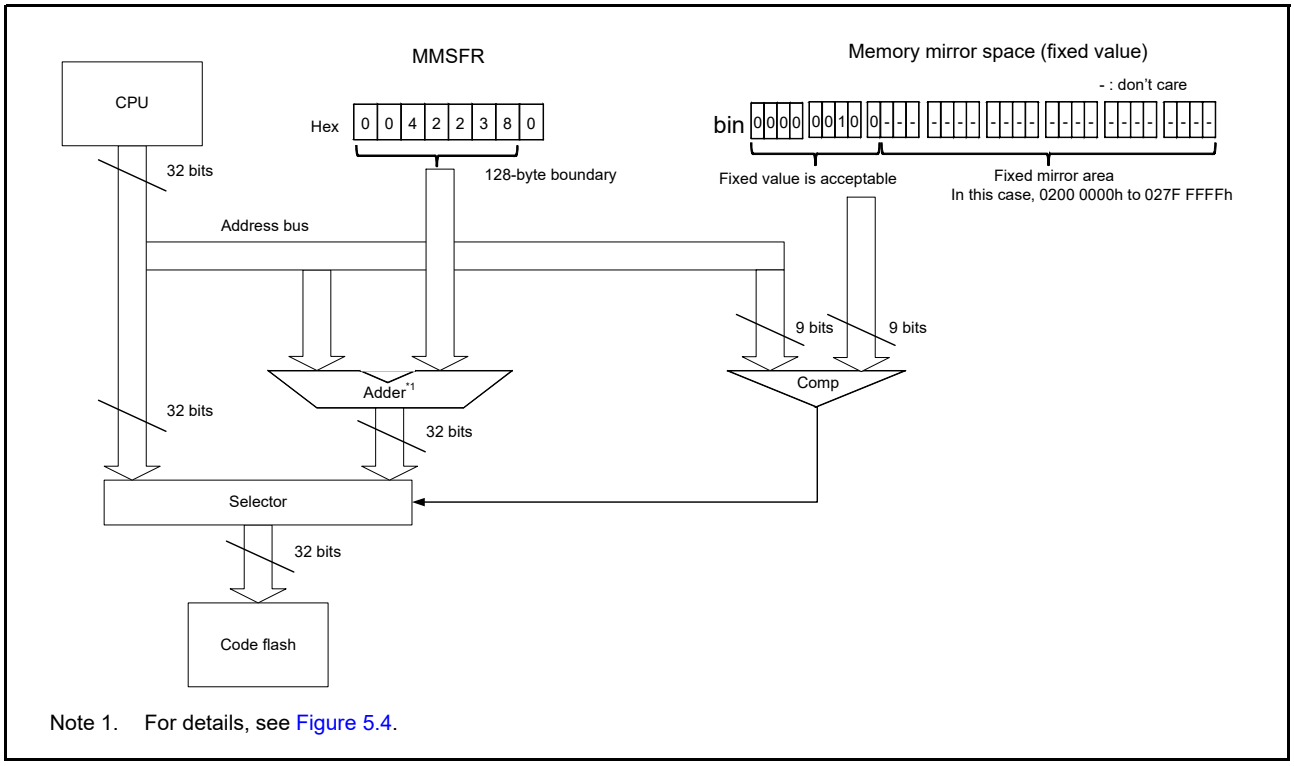


Figure 5.1 MMF operation



Note 1. For details, see Figure 5.4.

Figure 5.2 MMF block diagram

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

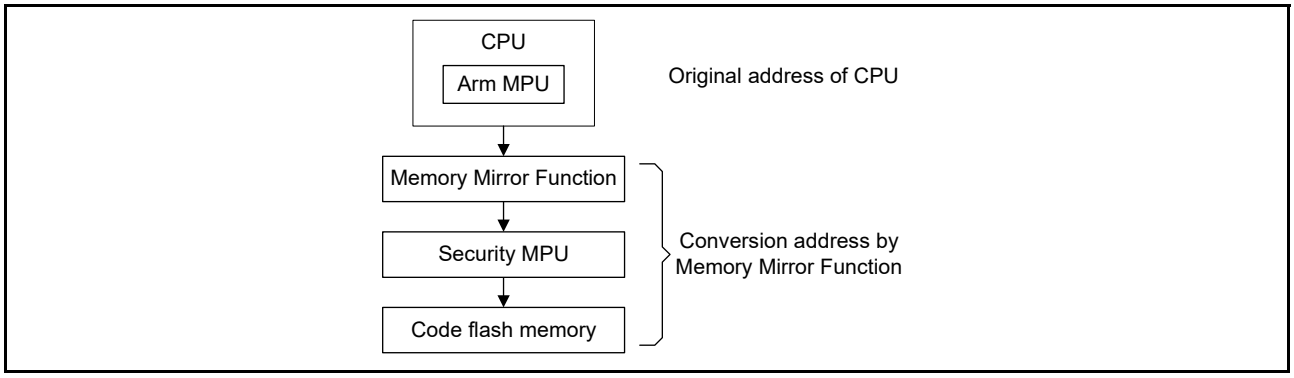


Figure 5.3 MMF address handling

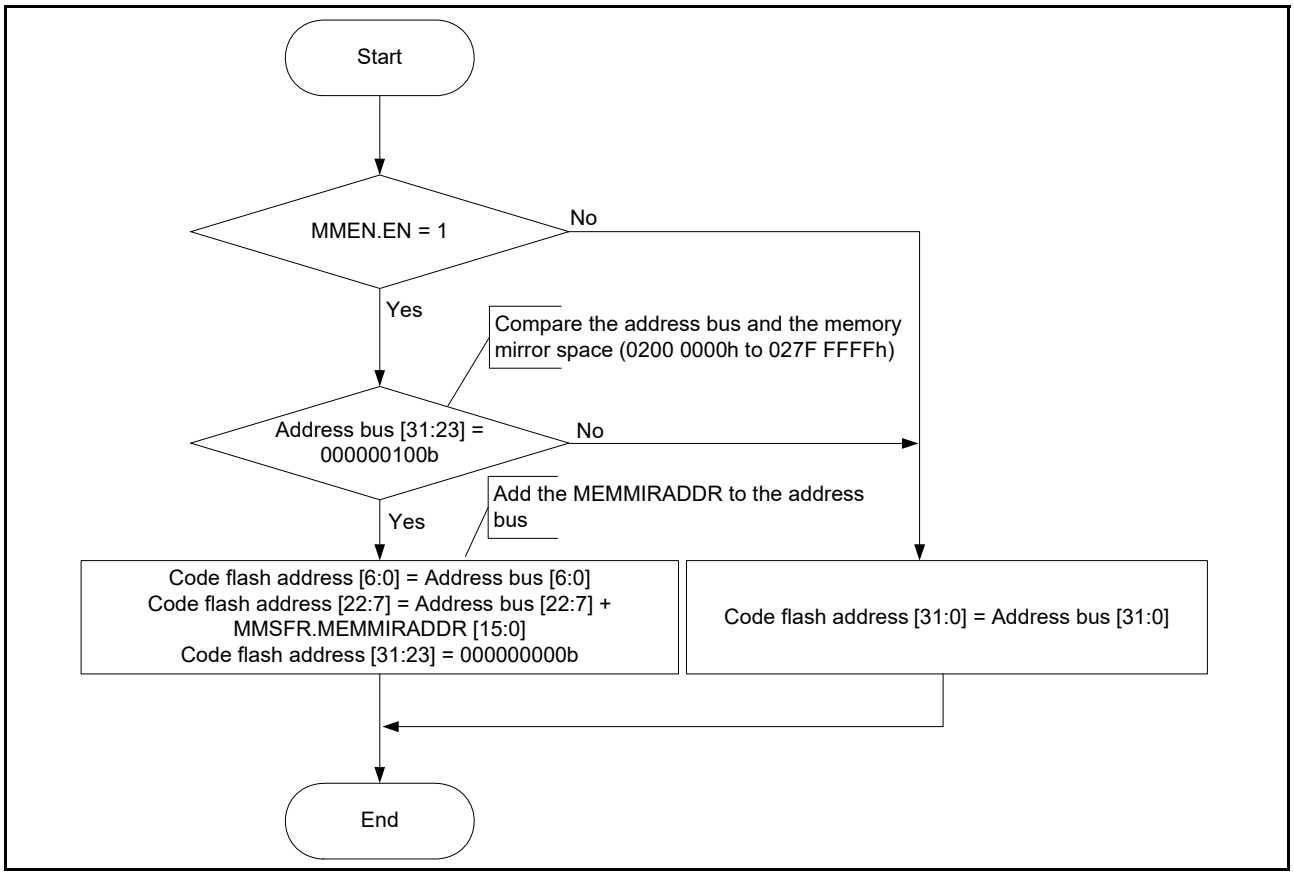


Figure 5.4 MMF operation flow

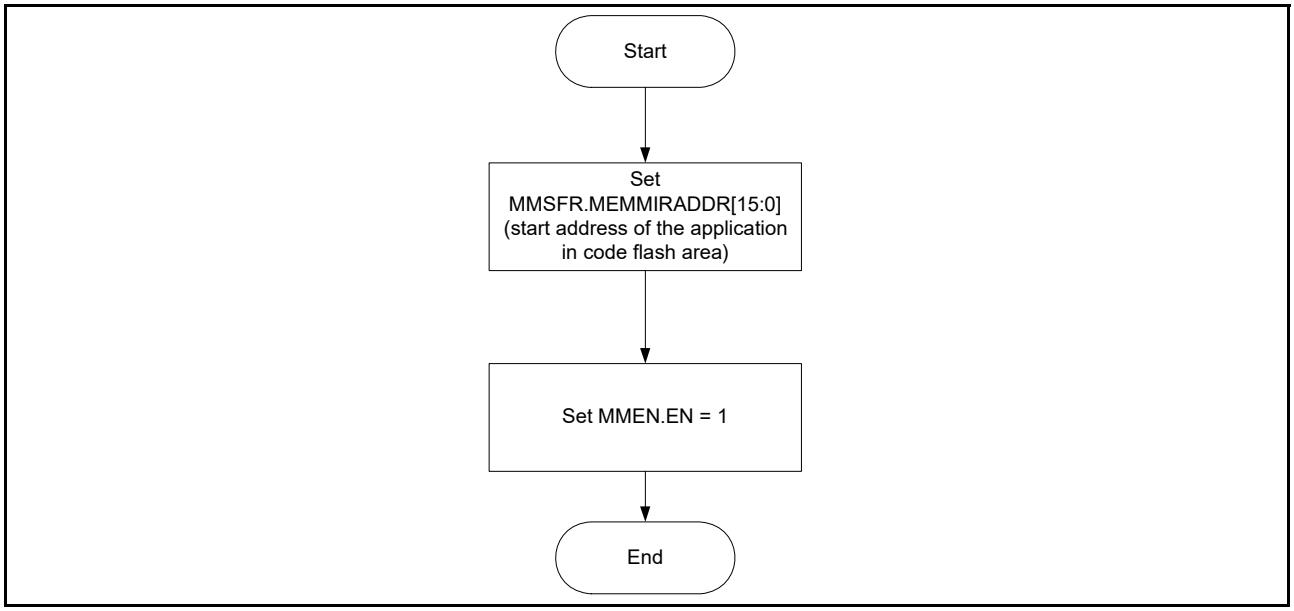


Figure 5.5 MMF setup flow

5.3.2 Setting Example

The target application code on the code flash can be accessed from the address 0200 0000h on the memory mirror space by setting up the code flash starting address in MMSFR.MEMMIRADDR[15:0] and setting MMEN.EN = 1.

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

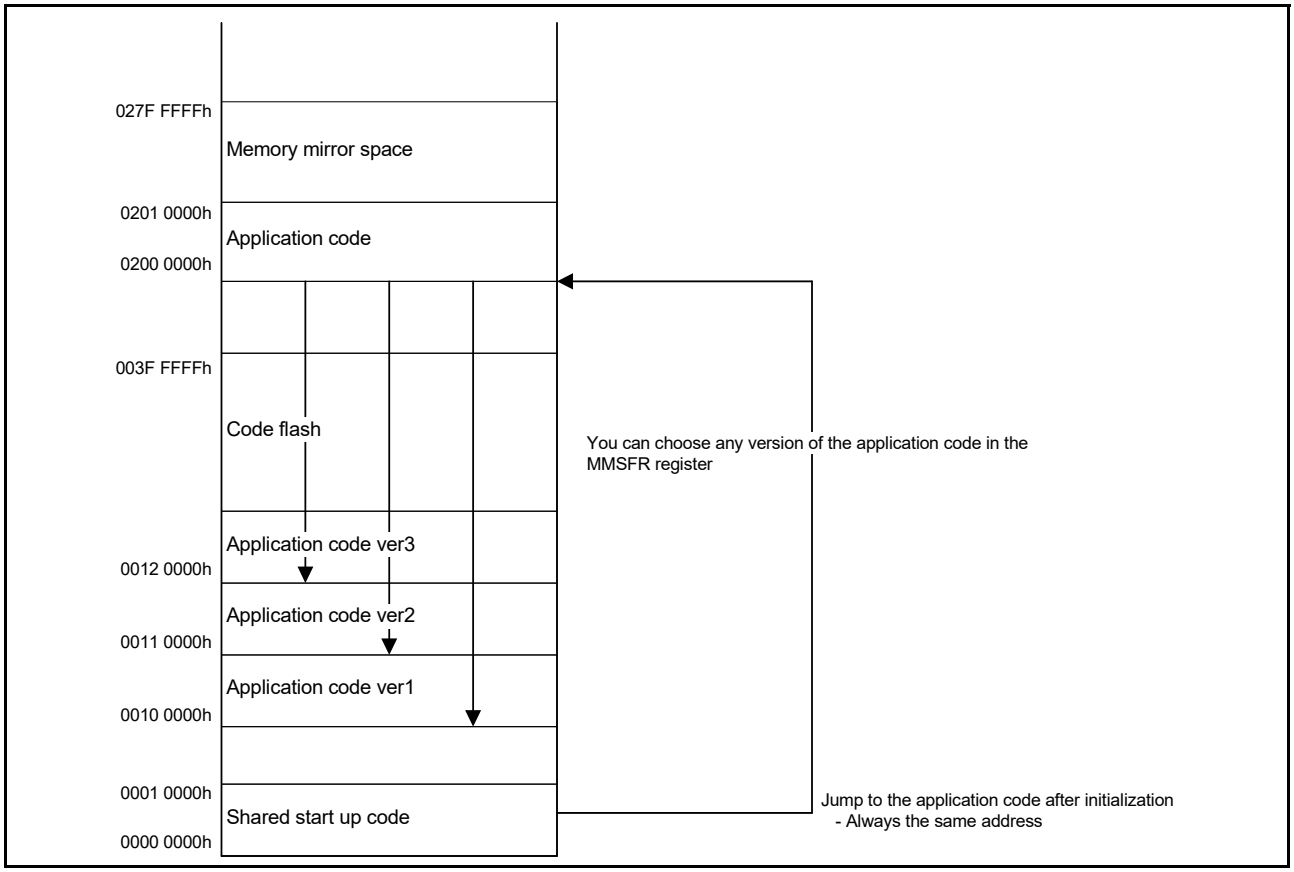


Figure 5.6 MMF setting example

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 13 resets:

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

Table 6.1 lists the reset names and sources.

Table 6.1 Reset names and sources

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

Note 1. For details on the voltages to be monitored (V_{POR}, V_{det0}, V_{det1}, and V_{det2}), see section 8, Low Voltage Detection (LVD) and section 47, Electrical Characteristics.

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

Table 6.2 Reset detect flags initialized by each reset source

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

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

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

Flags to be initialized	Reset source		
	Bus master MPU error reset	Bus slave MPU error reset	CPU stack pointer error reset
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	x	x	x

✓: Initialized to 0
x: Not initialized

Table 6.3 Module-related registers initialized by each reset source

Registers to be initialized		Reset source						
		RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	Voltage monitor 1 reset	Voltage monitor 2 reset
Watchdog timer registers	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓	✓	✓	✓	✓	✓
Voltage monitor function 1 registers	LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL	✓	✓	✓	✓	✓	x	x
	LVD1CR1/LVD1SR	✓	✓	✓	✓	✓	x	x
Voltage monitor function 2 registers	LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL	✓	✓	✓	✓	✓	x	x
	LVD2CR1/LVD2SR	✓	✓	✓	✓	✓	x	x
SOSC registers	SOSCCR	x	✓	x	x	x	x	x
	SOMCR	x	✓	x	x	x	x	x
LOCO registers	LOCOCR	✓	✓	✓	✓	✓	✓	✓
	LOCOUTCR	x	✓	✓	x	x	✓	✓
MOSC register	MOMCR	✓	✓	✓	✓	✓	✓	✓
Realtime Clock (RTC) register*1		x	x	x	x	x	x	x
AGT register		x	✓	✓	x	x	✓	✓
MPU register		✓	✓	✓	✓	✓	✓	✓
Pin state (except XCIN/XCOUT pin)		✓	✓	✓	✓	✓	✓	✓
Pin state (XCIN/XCOUT pin)		x	✓	x	x	x	x	x
Registers other than those shown, CPU, and internal state		✓	✓	✓	✓	✓	✓	✓

Registers to be initialized		Reset source					
		Software reset	SRAM parity error reset	SRAM ECC error reset	Bus master MPU error reset	Bus slave MPU error reset	CPU stack pointer error reset
Watchdog timer registers	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓	✓	✓	✓	✓
Voltage monitor function 1 registers	LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL	x	x	x	x	x	x
	LVD1CR1/LVD1SR	x	x	x	x	x	x
Voltage monitor function 2 registers	LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL	x	x	x	x	x	x
	LVD2CR1/LVD2SR	x	x	x	x	x	x
SOSC registers	SOSCCR	x	x	x	x	x	x
	SOMCR	x	x	x	x	x	x
LOCO registers	LOCOCR	✓	✓	✓	✓	✓	✓
	LOCOUTCR	x	x	x	x	x	x
MOSC register	MOMCR	✓	✓	✓	✓	✓	✓
Realtime Clock (RTC) register*1		x	x	x	x	x	x
AGT register		x	x	x	x	x	x
MPU register		✓	✓	✓	x	x	x
Pin state (except XCIN/XCOUT pin)		✓	✓	✓	✓	✓	✓
Pin state (XCIN/XCOUT pin)		x	x	x	x	x	x
Registers other than those shown, CPU, and internal state		✓	✓	✓	✓	✓	✓

✓: Initialized
x: Not initialized

Note 1. The RTC has a software reset. RCR1.RTCOS, RCR1.CIE, RCR2.RTCOE, RCR2.ADJ30, and RCR2.RESET are initialized.
Note 2. For details on the target bits, see section 23, Realtime Clock (RTC).

The RTC is not initialized by any reset source. SOSC and LOCO can be selected as the clock sources of RTC. 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	
		POR	**Other**
SOSC	Enable or disable	Initialized to disable	Continue with the state that was selected before the reset occured
	Drive capability	Initialized to normal mode	Continue with the state that was selected before the reset occured
	XCIN/XCOUT	Initialized to general-purpose input pins	Continue with the state that was selected before the reset occured

Table 6.5 States of LOCO when a reset occurs

		Reset source	
		POR/LVD0/LVD1/LVD2	**Other**
LOCO	Enable or disable	Initialized to enable	
	Oscillation accuracy*1	Initialized to accuracy before trimming by power-on (accuracy: ± 15%)	Continue with the accuracy that was trimmed by LOCOUTCR

Note 1. The LOCO User Trimming Control Register (LOCOUTCR) is reset by POR, LVD0, LVD1, and LVD2 resets, returning the LOCO to the default oscillation accuracy. This can affect RTC accuracy if the RTC uses the LOCO (with a user trimming value in LOCOUTCR) as the RTC source clock. To restore the pre-reset LOCO oscillation accuracy, reload the required trimming value into LOCOUTCR after any of these resets.

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
	—	—	—	—	LVD2RF	LVD1RF	LVD0RF	PORF
Value after reset:	0	0	0	0	x*1	x*1	x*1	x*1

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

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

PORF flag (Power-On Reset Detect Flag)

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

[Setting condition]

- When a power-on reset occurs.

[Clearing conditions]

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

LVD0RF flag (Voltage Monitor 0 Reset Detect Flag)

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

[Setting condition]

- When a voltage monitor 0 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read and then 0 is written to LVD0RF.

LVD1RF flag (Voltage Monitor 1 Reset Detect Flag)

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

[Setting condition]

- When a voltage monitor 1 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs

- When 1 is read and then 0 is written to LVD1RF.

LVD2RF flag (Voltage Monitor 2 Reset Detect Flag)

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

[Setting condition]

- When a voltage monitor 2 reset occurs.

[Clearing conditions]

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

6.2.2 Reset Status Register 1 (RSTSR1)

Address(es): SYSTEM.RSTSR1 4001 E0C0h

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

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	IWDTRF	Independent Watchdog Timer Reset Detect Flag	0: Independent watchdog timer reset not detected 1: Independent watchdog timer reset detected.	R(/W) *2
b1	WDTRF	Watchdog Timer Reset Detect Flag	0: Watchdog timer reset not detected 1: Watchdog timer reset detected.	R(/W) *2
b2	SWRF	Software Reset Detect Flag	0: Software reset not detected 1: Software reset detected.	R(/W) *2
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	RPERF	SRAM Parity Error Reset Detect Flag	0: SRAM parity error reset not detected 1: SRAM parity error reset detected.	R(/W) *2
b9	REERF	SRAM ECC Error Reset Detect Flag	0: SRAM ECC error reset not detected 1: SRAM ECC error reset detected.	R(/W) *2
b10	BUSSRF	Bus Slave MPU Error Reset Detect Flag	0: Bus slave MPU error reset not detected 1: Bus slave MPU error reset detected.	R(/W) *2
b11	BUSMRF	Bus Master MPU Error Reset Detect Flag	0: Bus master MPU error reset not detected 1: Bus master MPU error reset detected.	R(/W) *2
b12	SPERF	CPU Stack Pointer Error Reset Detect Flag	0: CPU stack pointer error reset not detected 1: CPU stack pointer error reset detected.	R(/W) *2
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 occurs.

[Setting condition]

- When an independent watchdog timer reset occurs.

[Clearing conditions]

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

WDTRF flag (Watchdog Timer Reset Detect Flag)

The WDTRF flag indicates that a watchdog timer reset occurs.

[Setting condition]

- When a watchdog timer reset occurs.

[Clearing conditions]

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

SWRF flag (Software Reset Detect Flag)

The SWRF flag indicates that a software reset occurs.

[Setting condition]

- When a software reset occurs.

[Clearing conditions]

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

RPERF flag (SRAM Parity Error Reset Detect Flag)

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

[Setting condition]

- When an SRAM parity error reset occurs.

[Clearing conditions]

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

REERF flag (SRAM ECC Error Reset Detect Flag)

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

[Setting condition]

- When an SRAM ECC error reset occurs.

[Clearing conditions]

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

[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 and then 0 is written to BUSSRF.

BUSMRF flag (Bus Master MPU Error Reset Detect Flag)

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

[Setting condition]

- When a bus master MPU error reset occurs.

[Clearing conditions]

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

SPERF flag (CPU Stack Pointer Error Reset Detect Flag)

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

[Setting condition]

- When a stack pointer error reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read 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 47, Electrical Characteristics.

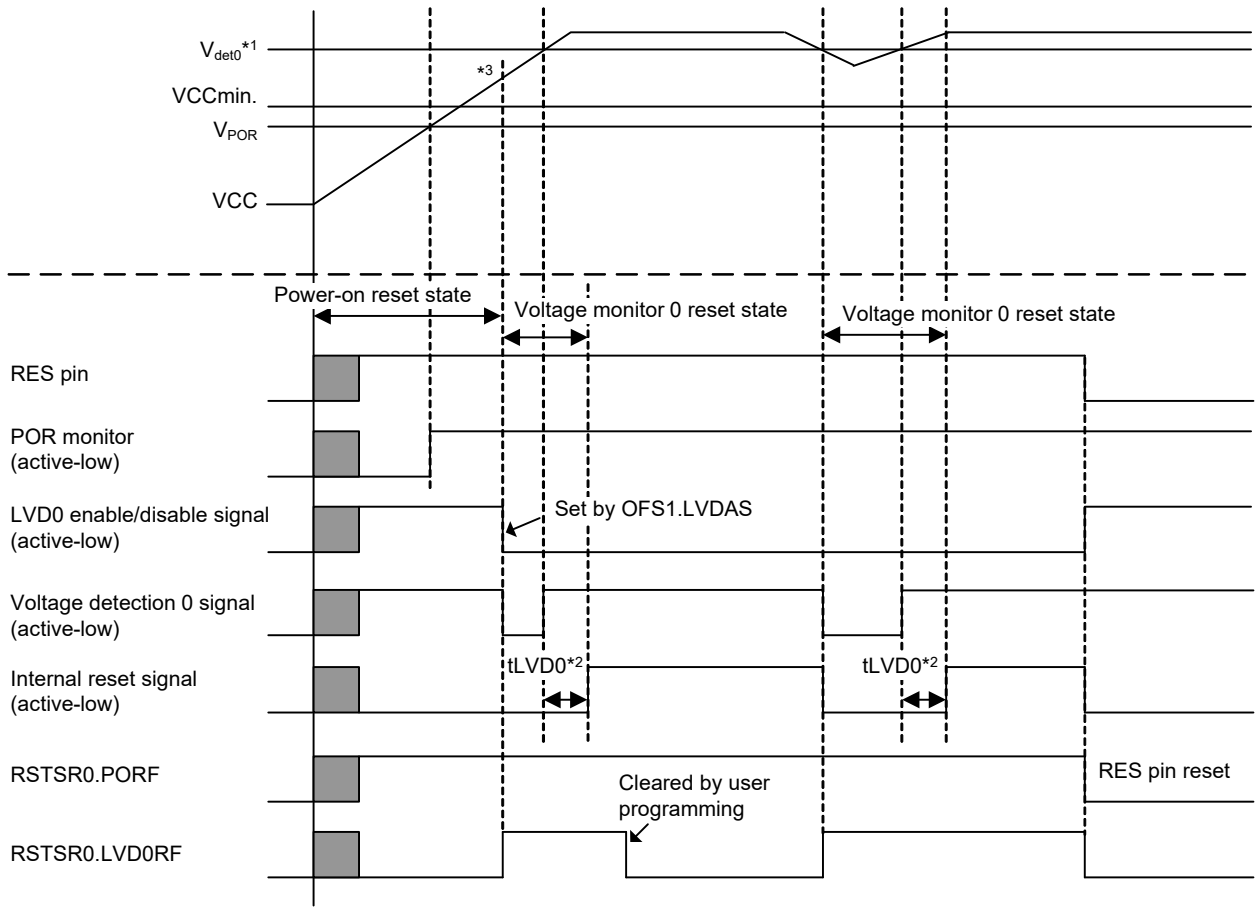
6.3.2 Power-On Reset

The power-on reset (POR) is an internal reset generated by the power-on reset circuit. If the RES pin is in a high level state when power is supplied, a power-on reset is generated. After VCC exceeds V_{POR} and the specified power-on reset time elapses, the internal reset is canceled and the CPU starts the reset exception handling. The power-on reset time is a stabilization period of the external power supply and the MCU circuit. After a power-on reset is generated, the PORF flag in the RSTSR0 is set to 1. The PORF flag is initialized by the RES pin reset.

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 the Option Function Select Register 1 (OFS1) is 0 (voltage monitor 0 reset is enabled after a reset) and VCC falls below V_{det0}, the RSTSR0.LVD0RF flag 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 V_{det0} and the voltage monitor 0 reset time (tLVD0) elapses, the internal reset is canceled and the CPU starts the reset exception handling. The V_{det0} voltage detection level can be changed by the setting in the VDSEL1[2:0] bits in the Option Function Select Register 1 (OFS1).

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



Note: For details on the electrical characteristics, see section 47, Electrical Characteristics.
Note 1. V_{det0} shows a voltage monitor 0 reset detection level, V_{POR} 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 rise to the minimum guaranteed voltage before the POR reset is released.

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

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 the Option Function Select Register 1 (OFS1) is 0 (voltage monitor 0 reset is enabled after a reset) and VCC falls below V_{det0}, the RSTSR0.LVD0RF flag becomes 1 and the voltage detection circuit generates voltage monitor 0 reset. Clear the OFS1.LVDAS bit to 0 if the voltage monitor 0 reset is to be used. After VCC exceeds V_{det0} and the voltage monitor 0 reset time (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 (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 1 Circuit Control Register 0 (LVD1CR0), the RSTSR0.LVD1RF flag is set to 1 and the voltage detection circuit generates a voltage monitor 1 reset if VCC falls to or below V_{det1}.

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

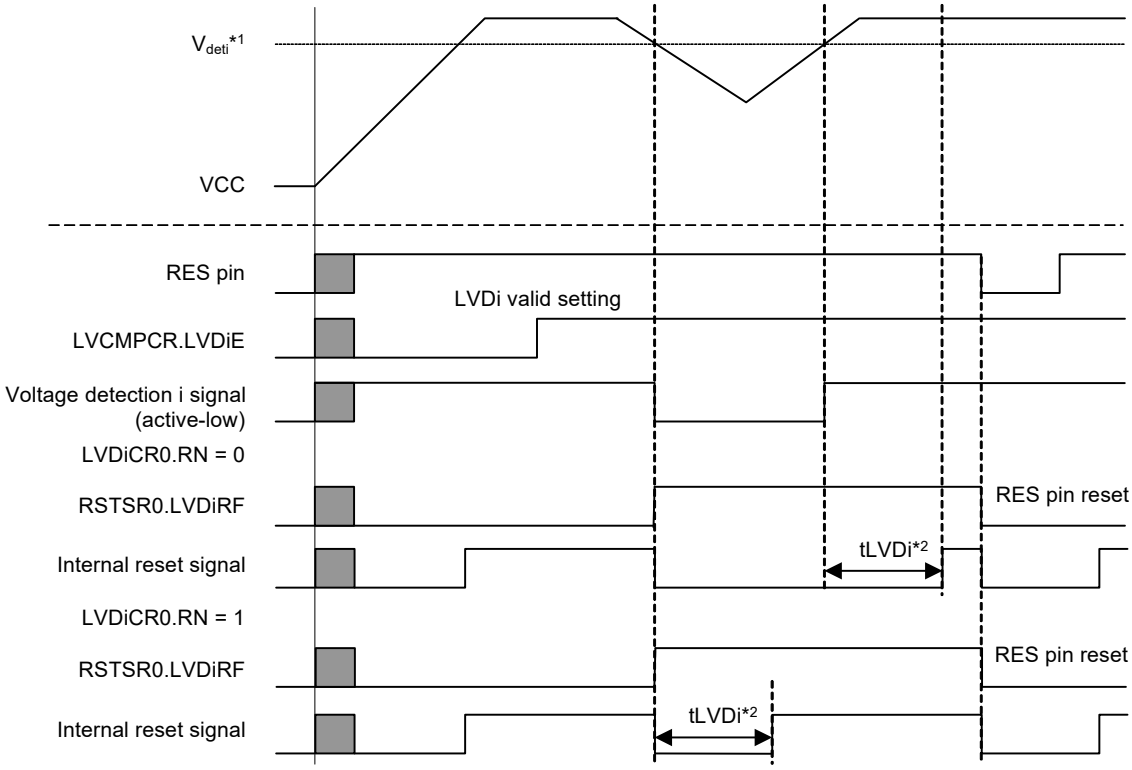
Similarly, timing for release from the voltage monitor 1 reset state is selectable with the Voltage Monitor 1 Reset Negate

Select bit (RN) in the LVD1CR0 register. When the LVD1CR0.RN bit is 0 and VCC has fallen 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 V_{det1}. 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.

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

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

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



Note: For details on the electrical characteristics, see section 47, Electrical Characteristics.
Note 1. V_{deti} indicates the detection level of voltage monitor 1 reset and voltage monitor 2 reset (i = 1, 2).
Note 2. tLVDi indicates the time for voltage monitor 1 reset and voltage monitor 2 reset (i = 1, 2).

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

6.3.4 Independent Watchdog Timer Reset

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

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

For details on the independent watchdog timer reset, see section 25, 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 (t_{RESW2}) elapses after the watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the watchdog timer reset, see section 24, Watchdog Timer (WDT).

6.3.6 Software Reset

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

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

6.3.7 Determination of Cold/Warm Start

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

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

Figure 6.3 shows an example of cold/warm start determination operation.

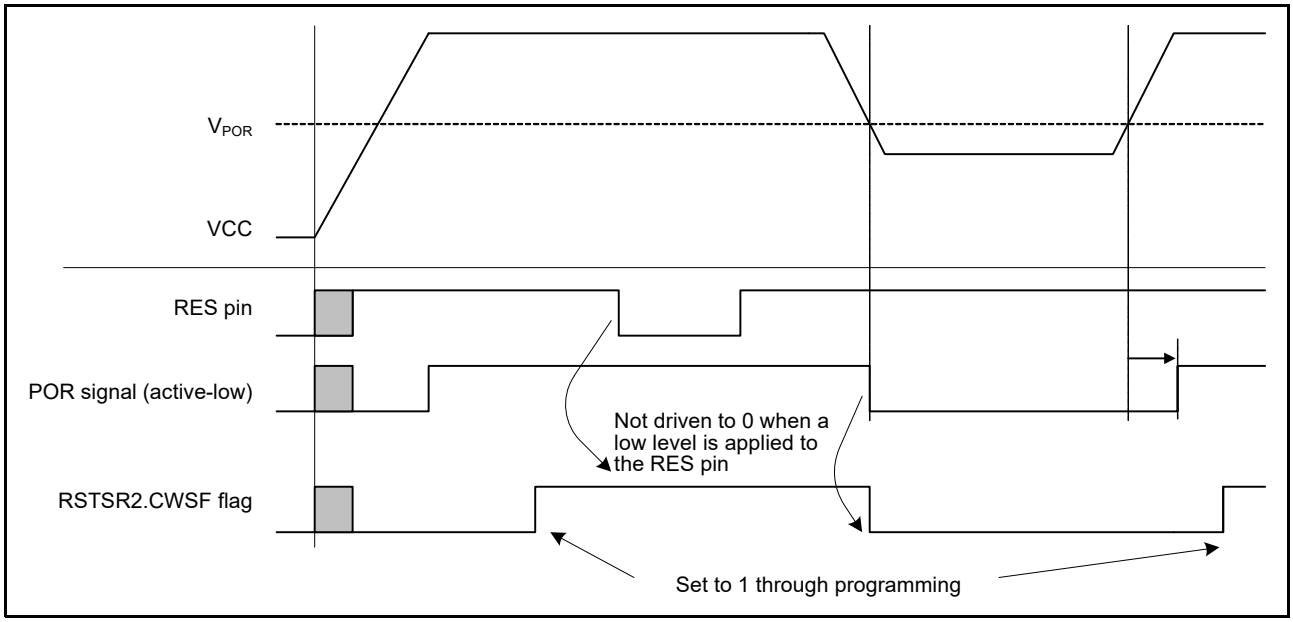
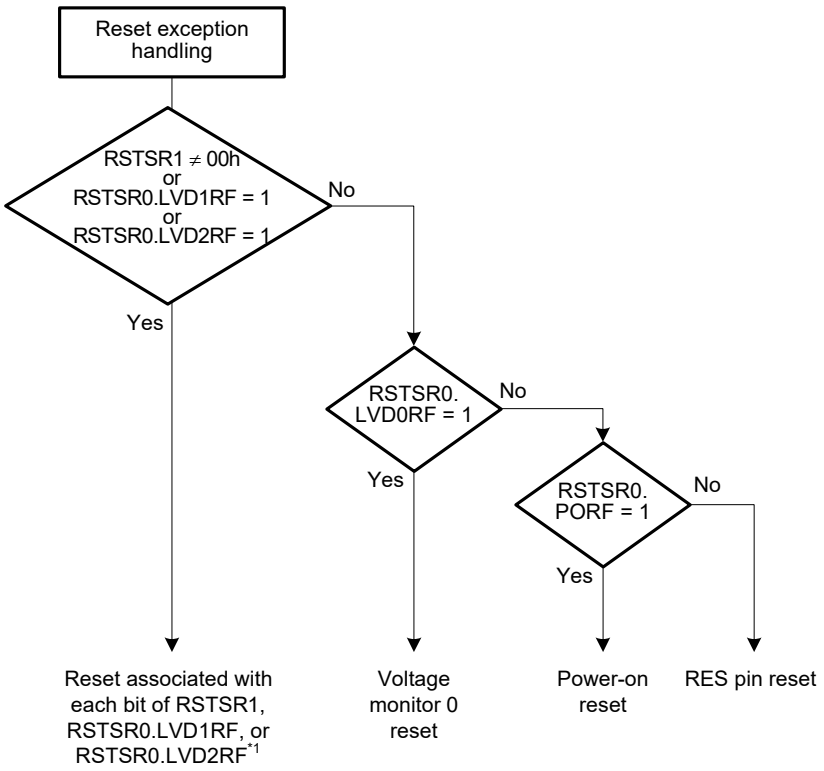


Figure 6.3 Example of cold/warm start determination operation

6.3.8 Determination of Reset Generation Source

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

Figure 6.4 shows an example flow to identify a reset generation source. The reset flag must be written with 0 after it is read as 1.



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

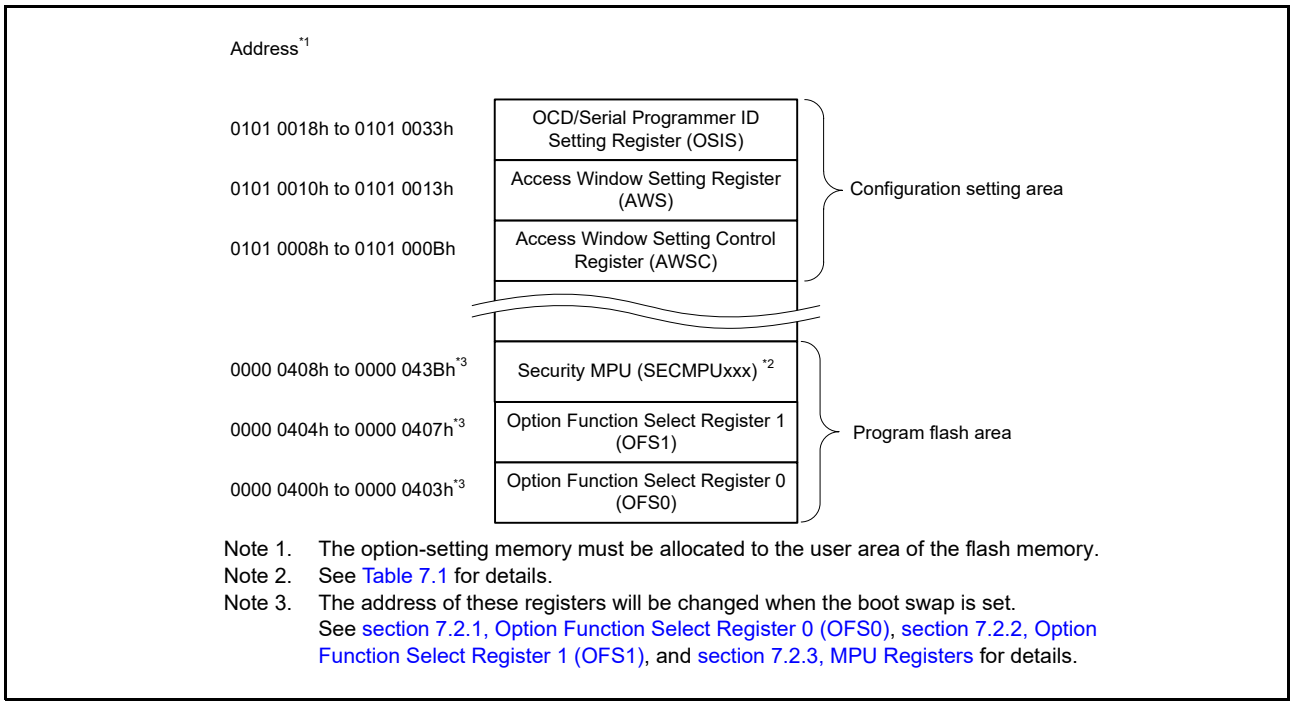
Figure 6.4 Example of reset generation source determination flow

7. Option-Setting Memory

7.1 Overview

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

Figure 7.1 shows the option-setting memory area.



Note 1. The option-setting memory must be allocated to the user area of the flash memory.
Note 2. See Table 7.1 for details.
Note 3. The address of these registers will be changed when the boot swap is set.
See section 7.2.1, Option Function Select Register 0 (OFS0), section 7.2.2, Option Function Select Register 1 (OFS1), and section 7.2.3, MPU Registers for details.

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/0000 2400h*1

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	WDTST PCTL	—	WDTRS TIRQS	WDTRPSS[1:0]		WDTRPES[1:0]		WDTCKS[3:0]				WDTTOPS[1:0]		WDTST RT	—
Value after reset:	The value set by the user*2															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	IWDTST PCTL	—	IWDTRS TIRQS	IWDTRPSS[1:0]		IWDTRPES[1:0]		IWDTCKS[3:0]				IWDTTOPS[1:0]		IWDTST RT	—
Value after reset:	The value set by the user*2															

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 after a reset.	R

Bit	Symbol	Bit name	Description	R/W
b3, b2	IWDTTOPS[1:0]	IWDT Timeout Period Select	b3 b2 0 0: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh).	R
b7 to b4	IWDTCKS[3:0]	IWDT-Dedicated Clock Frequency Division Ratio Select	b7 b4 0 0 0 0: × 1 0 0 1 0: × 1/16 0 0 1 1: × 1/32 0 1 0 0: × 1/64 1 1 1 1: × 1/128 0 1 0 1: × 1/256. Other settings are prohibited.	R
b9, b8	IWDTRPES[1:0]	IWDT Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (no window end position setting).	R
b11, b10	IWDTRPSS[1:0]	IWDT Window Start Position Select	b11 b10 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (no window start position setting).	R
b12	IWDTRSTIRQS	IWDT Reset Interrupt Request Select	0: Interrupt 1: 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, Snooze, 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: Interrupt 1: Reset.	R
b29	—	Reserved	When read, this bit returns the written value. The write value should be 1.	R

Bit	Symbol	Bit name	Description	R/W
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. When the boot swap is set, the address of this register changes.
Therefore, set 0000 2400h and 0000 0400h to the same value if boot swap is used.
Note 2. The value in a blank product is FFFF FFFFh. It is set to the value written by your application.

IWDTSTRT bit (IWDT Start Mode Select)

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

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

The IWDTTOPS[1:0] bits select the timeout period, that is, the time it takes for the down counter to underflow, as 128, 512, 1024, or 2048 cycles of the frequency-divided clock set in the IWDTCKS[3:0] bits. The 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 25, Independent Watchdog Timer (IWDT).

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

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

For details, see section 25, Independent Watchdog Timer (IWDT).

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

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

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

For details, see section 25, Independent Watchdog Timer (IWDT).

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

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

For details, see section 25, 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 25, Independent Watchdog Timer (IWDT).

IWDTSTPCTL bit (IWDT Stop Control)

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

For details, see section 25, 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 select the timeout period, that is, the time it takes for the down counter to underflow, as 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set in the WDTCKS[3:0] bits. The number of 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 24, Watchdog Timer (WDT).

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

The WDTCKS[3:0] bits specify the division ratio of the prescaler for dividing the frequency of PCLKB as 1/4, 1/64, 1/128, 1/512, 1/2048, and 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 24, 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 24, Watchdog Timer (WDT).

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

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

For details, see section 24, 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 24, Watchdog Timer (WDT).

WDTSTPCTL bit (WDT Stop Control)

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

For details, see section 24, Watchdog Timer (WDT).

7.2.2 Option Function Select Register 1 (OFS1)

Address(es): OFS1 0000 0404h/0000 2404h*1

	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*2															

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

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	When read, these bits return the written value. The write value should be 1.	R
b2	LVDAS	Voltage Detection 0 Circuit Start	0: Enable voltage monitor 0 reset after a reset 1: Disable voltage monitor 0 reset after a reset.	R
b5 to b3	VDSEL1[2:0]	Voltage Detection 0 Level Select	b5 b3 0 0 0: Selects 3.84 V 0 0 1: Selects 2.82 V 0 1 0: Selects 2.51 V 0 1 1: Selects 1.90 V 1 0 0: Selects 1.70 V. Other settings are prohibited.	R
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. When the boot swap is set, the address of this register changes.
Therefore, set 0000 2404h and 0000 0404h to the same value if boot swap is used.
Note 2. The value in a 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, set the OFS1.HOCOFRQ1[2:0] bits to an optimum value.

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

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

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

7.2.3 MPU Registers

Table 7.1 shows the registers related to the MPU function. For details, see 15. Memory Protection Unit (MPU).

The security MPU is disabled on erasure of the flash memory. If improper data is written to an MPU register, the MCU might fail to operate. See section 15, Memory Protection Unit (MPU) to set the proper data.

Table 7.1 MPU registers

Register name	Symbol	Function	Address*1	Size (byte)
Security MPU Program Counter Start Address Register 0	SECMPUPCS0	Specifies the security fetch region of the code flash or SRAM	0000 0408h	4
Security MPU Program Counter End Address Register 0	SECMPUPCE0	Specifies the security fetch region of the code flash or SRAM	0000 040Ch	4
Security MPU Program Counter Start Address Register 1	SECMPUPCS1	Specifies the security fetch region of the code flash or SRAM	0000 0410h	4
Security MPU Program Counter End Address Register 1	SECMPUPCE1	Specifies the security fetch region of the code flash or SRAM	0000 0414h	4
Security MPU Region 0 Start Address Register	SECMPUS0	Specifies the secure program and the code flash data	0000 0418h	4
Security MPU Region 0 End Address Register	SECMPUE0	Specifies the secure program and the code flash data	0000 041Ch	4
Security MPU Region 1 Start Address Register	SECMPUS1	Specifies the secure program and data of the SRAM	0000 0420h	4
Security MPU Region 1 End Address Register	SECMPUE1	Specifies the secure program and data of the SRAM	0000 0424h	4
Security MPU Region 2 Start Address Register	SECMPUS2	Specifies the secure data of the security functions	0000 0428h	4
Security MPU Region 2 End Address Register	SECMPUE2	Specifies the secure data of the security functions	0000 042Ch	4
Security MPU Region 3 Start Address Register	SECMPUS3	Specifies the secure data of the security functions	0000 0430h	4
Security MPU Region 3 End Address Register	SECMPUE3	Specifies the secure data of the security functions	0000 0434h	4
Security MPU Access Control Register	SECMPUAC	Specifies the security enabled/disabled region	0000 0438h	4

Note 1. When the boot swap is set, the address of MPU registers change.Therefore, set (0000 2408h to 0000 243Bh) and (0000 0408h to 0000 043Bh) to the same value if boot swap is used.

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 written value. The write value should be 1.	R
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]											
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]											
Value after reset:	The value set by the user															

Bit	Symbol	Bit name	Description	R/W
b11 to b0	FAWS[11:0]	Access Window Start Block Address	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

Bit	Symbol	Bit name	Description	R/W
b15 to b12	—	Reserved	When read, these bits return the written value. The write value should be 1.	R
b27 to b16	FAWE[11:0]	Access Window End Block Address	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

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 settings for the bits are as follows:

- 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.

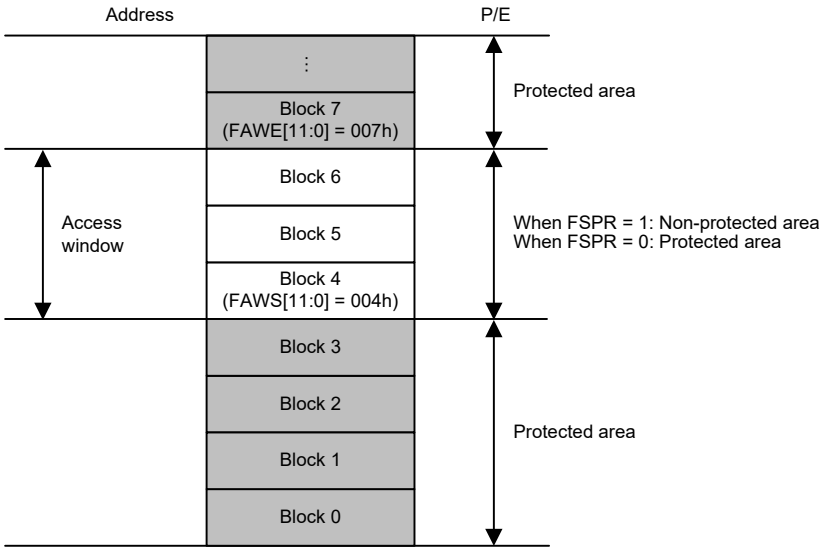


Figure 7.2 Access window overview

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 with the OCD/serial programmer is permitted, if not, connection with the OCD/serial programmer is not possible. The OSIS register must be set in 32-bit words.

Address(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 bits [127] and [126] determine whether the ID code protection is enabled, and the authentication method to use with the host. Table 7.2 shows how the ID code determines the authentication method.

Setting bit [127] to 0 prevents Renesas from accessing the test mode. Therefore, Renesas cannot perform failure analysis unless provided with bits [126:0]. To process any warranty claim, Renesas must be able to perform failure analysis.

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 boot mode) On-chip debug mode (SWD boot mode)	FFh, …, FFh (all bytes are FFh)	Protection disabled	The ID code is not checked, the ID code always matches, and connection to the programmer or on-chip debugger is permitted
	Bit [127] = 1, bit [126] = 1, and at least one of the 16 bytes is not FFh	Protection enabled	Matching ID code indicates that authentication is complete and connection to the programmer or the on-chip debugger is permitted. Mismatching ID code indicates that 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 contents of the user flash (code and data) area, and configuration area are erased. However, forced erasure is not executed when the FSPR bit is 0.
	Bit [127] = 1 and bit [126] = 0	Protection enabled	Matching ID code indicates that authentication is complete and connection to the programmer or the on-chip debugger is permitted. Mismatching ID code indicates transition to the ID code protection wait state.
	Bit [127] = 0	Protection enabled	The ID code is not checked, the ID code is always mismatching, the connection to the programmer or the on-chip debugger is prohibited, but the ALeRASE command will be accepted. For the prohibition of the ALeRASE command, see section 2.8.3.4 (1), When MSB of OSIS is 0 (bit [127] = 0). Renesas cannot access the test mode.

7.3 Setting Option-Setting Memory

7.3.1 Allocation of Data in the 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 the Option-Setting Memory

Allocating data according to the procedure described in section 7.3.1, Allocation of Data in the 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 on 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, so see the tool manual for details.

The MCU provides two setting procedures as follows:

- Read the data allocated as described in section 7.3.1, Allocation of Data in the Option-Setting Memory, from an object file or Motorola S-format file generated by the compiler, and write the data to the MCU
- Use the GUI interface of the tool to program the same data allocated as described in section 7.3.1, Allocation of Data in the Option-Setting Memory.

7.4 Usage Notes

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 available for programming, write 1 to all bits in the reserved areas and all reserved bits. If 0 is written to these bits, normal operation cannot be guaranteed.

8. Low Voltage Detection (LVD)

8.1 Overview

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

Each voltage level detector has a voltage monitor associated with it, for example voltage monitor 0, 1, and 2. 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, 1, and 2, Figure 8.2 shows a block diagram of the voltage monitor 1 interrupt/reset circuit, and Figure 8.3 shows a block diagram of the voltage monitor 2 interrupt/reset circuit.

Table 8.1 LVD specifications

Parameter		Voltage monitor 0	Voltage monitor 1	Voltage monitor 2
VCC monitoring	Monitored voltage	V_{det0}	V_{det1}	V_{det2}
	Detected event	Voltage falls below V_{det0}	Voltage rises or falls past V_{det1}	Voltage rises or falls past V_{det2}
	Detected voltage	Selectable from five different levels in the OFS1.VDSEL1[2:0] bits	Selectable from 16 different levels in the LVDLVLR.LVD1LVL[4:0] bits	Selectable from four different levels in the LVDLVLR.LVD2LVL[2:0] bits
	Monitor flag	None	LVD1SR.MON flag: Monitors whether voltage is higher or lower than V_{det1}	LVD2SR.MON flag: Monitors whether voltage is higher or lower than V_{det2}
			LVD1SR.DET flag: V_{det1} crossing detection	LVD2SR.DET flag: V_{det2} crossing detection
Process on voltage detection	Reset	Voltage monitor 0 reset	Voltage monitor 1 reset	Voltage monitor 2 reset
		Reset when V_{det0} > VCC CPU restart after specified time with VCC > V_{det0}	Reset when V_{det1} > VCC CPU restart timing selectable: after specified time with VCC > V_{det1} or V_{det1} > VCC	Reset when V_{det2} > VCC CPU restart timing selectable: after specified time with VCC > V_{det2} or V_{det2} > VCC
	Interrupt	No interrupt	Voltage monitor 1 interrupt	Voltage monitor 2 interrupt
			Non-maskable interrupt or maskable interrupt selectable	Non-maskable interrupt or maskable interrupt selectable
			Interrupt request issued when V_{det1} > VCC or VCC > V_{det1}	Interrupt request issued when V_{det2} > VCC or VCC > V_{det2}
Event linking		None	Available Output of event signals on detection of V_{det1} crossings	Available Output of event signals on detection of V_{det2} crossings

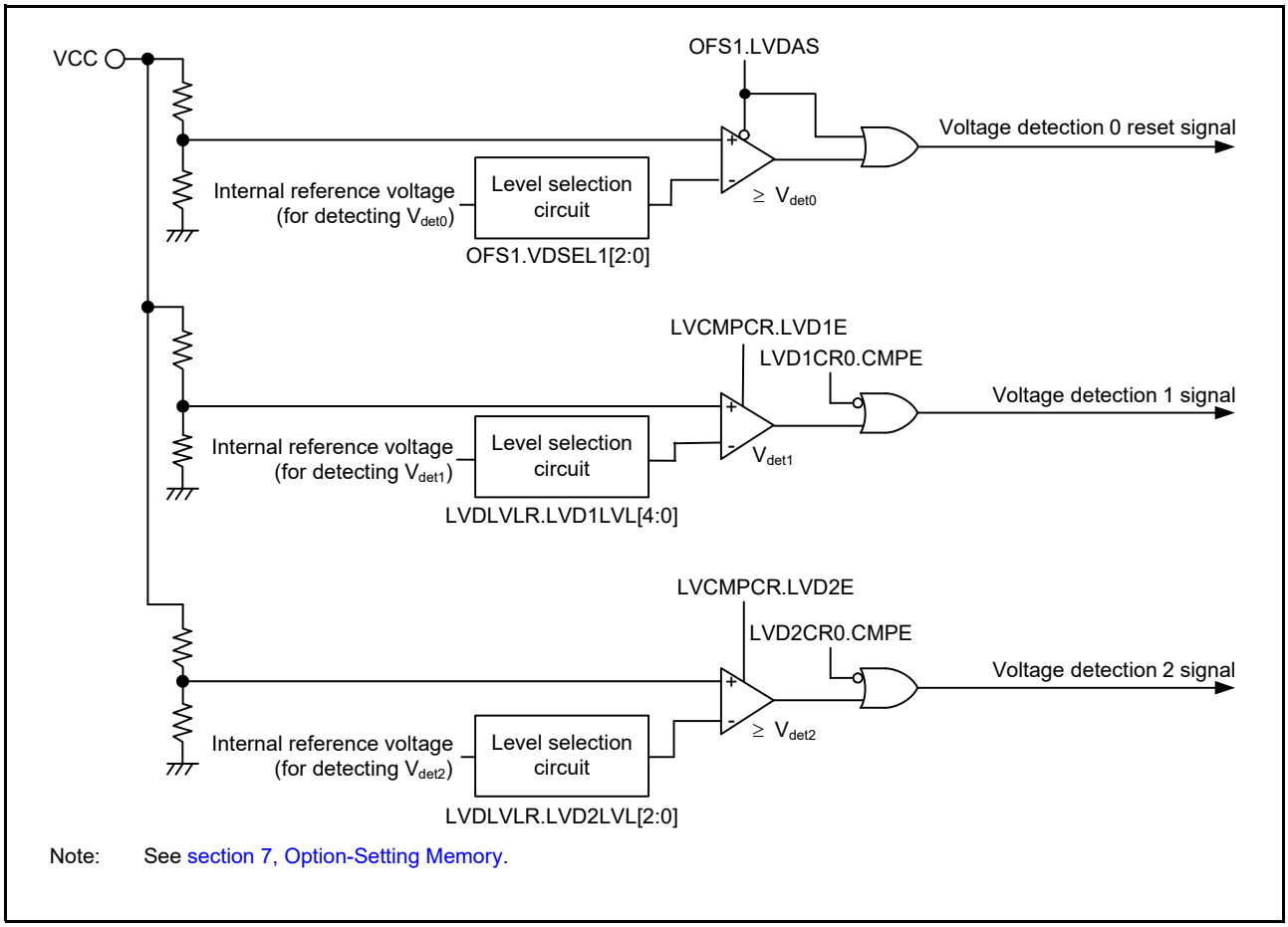


Figure 8.1 Voltage detection 0, 1, and 2 block diagram

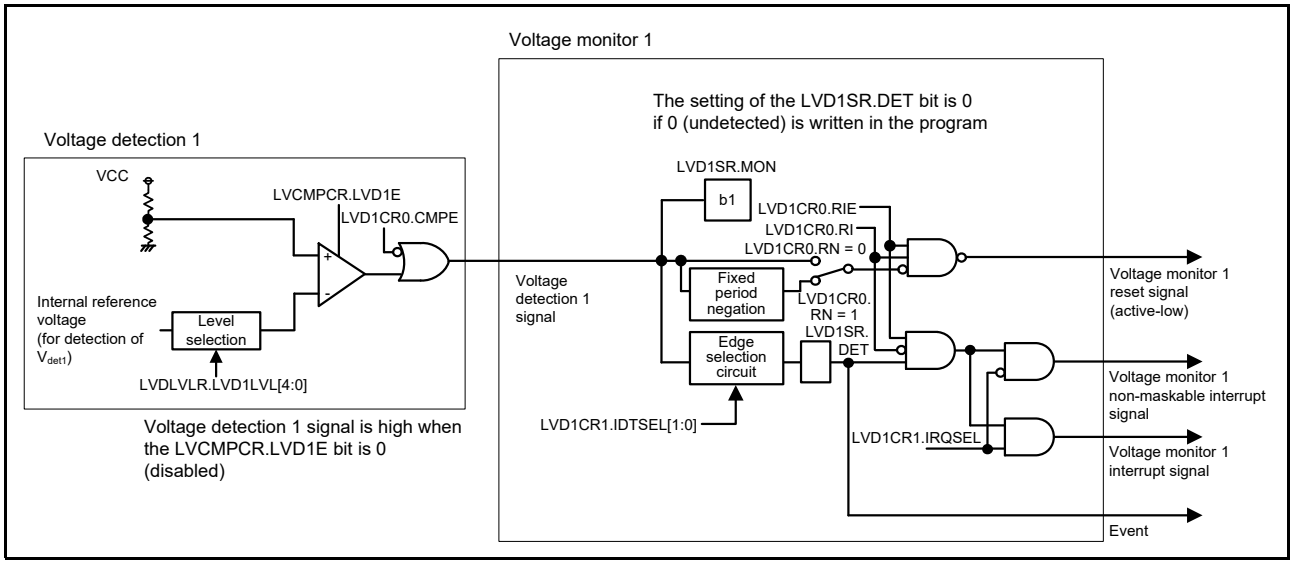


Figure 8.2 Voltage monitor 1 interrupt/reset circuit block diagram

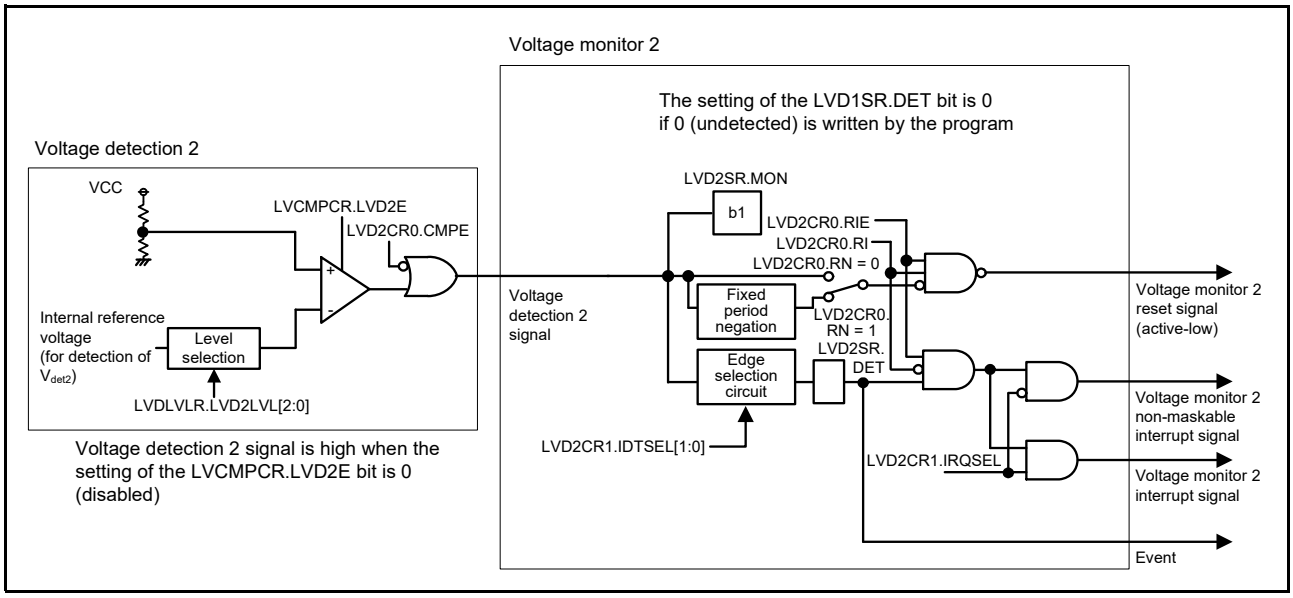


Figure 8.3 Voltage monitor 2 interrupt/reset circuit block diagram

8.2 Register Descriptions

8.2.1 Voltage Monitor 1 Circuit Control Register 1 (LVD1CR1)

Address(es): SYSTEM.LVD1CR1 4001 E0E0h

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

Bit	Symbol	Bit name	Description	R/W
b1, b0	IDTSEL[1:0]	Voltage Monitor 1 Interrupt Generation Condition Select	b1 b0 0 0: When VCC ≥ V_{det1} (rise) is detected 0 1: When VCC < V_{det1} (fall) is detected 1 0: When fall and rise are detected 1 1: Settings prohibited.	R/W
b2	IRQSEL	Voltage Monitor 1 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt*1.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

Note 1. When enabling maskable interrupts, do not change the NMIER.LVD1EN bit value in the ICU from the reset state.

8.2.2 Voltage Monitor 1 Circuit Status Register (LVD1SR)

Address(es): SYSTEM.LVD1SR 4001 E0E1h

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

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

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

Note 1. Only 0 can be written to this bit. After writing 0 to this bit, 2 system clock cycles are required for the bit to be read as 0.

DET flag (Voltage Monitor 1 Voltage Change Detection Flag)

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

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 clock cycles elapse.

MON flag (Voltage Monitor 1 Signal Monitor Flag)

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

8.2.3 Voltage Monitor 2 Circuit Control Register 1 (LVD2CR1)

Address(es): SYSTEM.LVD2CR1 4001 E0E2h

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

Bit	Symbol	Bit name	Description	R/W
b1, b0	IDTSEL[1:0]	Voltage Monitor 2 Interrupt Generation Condition Select	b1 b0 0 0: When VCC ≥ V_{det2} (rise) is detected 0 1: When VCC < V_{det2} (fall) is detected 1 0: When fall and rise are detected 1 1: Settings prohibited.	R/W
b2	IRQSEL	Voltage Monitor 2 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt.*1	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

Note 1. When enabling maskable interrupts, do not change the NMIER.LVD2EN bit value in the ICU from the reset state.

8.2.4 Voltage Monitor 2 Circuit Status Register (LVD2SR)

Address(es): SYSTEM.LVD2SR 4001 E0E3h

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

Bit	Symbol	Bit name	Description	R/W
b0	DET	Voltage Monitor 2 Voltage Change Detection Flag	0: Not detected 1: V_{det2} passage detected.	R(/W)*1
b1	MON	Voltage Monitor 2 Signal Monitor Flag	0: VCC < V_{det2} 1: VCC ≥ V_{det2} or MON is disabled.	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

Note 1. Only 0 can be written to this bit. After writing 0 to this bit, 2 system clock cycles are required for the bit to be read as 0.

DET flag (Voltage Monitor 2 Voltage Change Detection Flag)

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

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

MON flag (Voltage Monitor 2 Signal Monitor Flag)

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

8.2.5 Voltage Monitor Circuit Control Register (LVCMPCR)

Address(es): SYSTEM.LVCMPCR 4001 E417h

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

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

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

LVD1E bit (Voltage Detection 1 Enable)

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

LVD2E bit (Voltage Detection 2 Enable)

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

8.2.6 Voltage Detection Level Select Register (LVDLVLR)

Address(es): SYSTEM.LVDLVLR 4001 E418h

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD2LVL[2:0]			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 0 0 0: 4.29 V (Vdet1_0) 0 0 0 0 1: 4.14 V (Vdet1_1) 0 0 0 1 0: 4.02 V (Vdet1_2) 0 0 0 1 1: 3.84 V (Vdet1_3) 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) 0 1 1 1 0: 1.75 V (Vdet1_E) 0 1 1 1 1: 1.65 V (Vdet1_F). Other settings are prohibited.	R/W
b7 to b5	LVD2LVL[2:0]	Voltage Detection 2 Level Select (standard voltage during fall in voltage)	b7 b5 0 0 0: 4.29 V (Vdet2_0) 0 0 1: 4.14 V (Vdet2_1) 0 1 0: 4.02 V (Vdet2_2) 0 1 1: 3.84 V (Vdet2_3) 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited.	R/W

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

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

8.2.7 Voltage Monitor 1 Circuit Control Register 0 (LVD1CR0)

Address(es): SYSTEM.LVD1CR0 4001 E41Ah

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

Bit	Symbol	Bit name	Description	R/W
b0	RIE	Voltage Monitor 1 Interrupt/Reset Enable	0: Disable 1: Enable.	R/W
b1	—	Reserved	The read value is 0. The write value should be 0.	R/W
b2	CMPE	Voltage Monitor 1 Circuit Comparison Result Output Enable	0: Disable voltage monitor 1 circuit comparison result output 1: Enable voltage monitor 1 circuit comparison result output.	R/W
b3	—	Reserved	The read value is undefined. The write value should be 1.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	RI	Voltage Monitor 1 Circuit Mode Select	0: Generate voltage monitor 1 interrupt on V_{det1} passage 1: Enable voltage monitor 1 reset when the voltage falls to and below V_{det1}.	R/W
b7	RN	Voltage Monitor 1 Reset Negate Select	0: Negate after a stabilization time (t_{LVD1}) when VCC > V_{det1} is detected 1: Negate after a stabilization time (t_{LVD1}) on assertion of the LVD1 reset.	R/W

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

RIE bit (Voltage Monitor 1 Interrupt/Reset Enable)

The RIE bit enables or disables voltage monitor 1 interrupt/reset. 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 on assertion of the LVD1 reset signal), set the MOCOCR.MCSTP bit to 0 (the MOCO operates). Additionally, for a transition to Software Standby mode, the only possible value for the RN bit is 0 (negation follows a stabilization time when VCC > V_{det1} is detected). Do not set the RN bit to 1 (negation follows a stabilization time on assertion of the LVD1 reset signal) when this is the case.

8.2.8 Voltage Monitor 2 Circuit Control Register 0 (LVD2CR0)

Address(es): SYSTEM.LVD2CR0 4001 E41Bh

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

Bit	Symbol	Bit name	Description	R/W
b0	RIE	Voltage Monitor 2 Interrupt/Reset Enable	0: Disable 1: Enable.	R/W
b1	—	Reserved	The read value is 0. The write value should be 0.	R/W
b2	CMPE	Voltage Monitor 2 Circuit Comparison Result Output Enable	0: Disable voltage monitor 2 circuit comparison result output 1: Enable voltage monitor 2 circuit comparison result output.	R/W

Bit	Symbol	Bit name	Description	R/W
b3	—	Reserved	The read value is undefined. The write value should be 1.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	RI	Voltage Monitor 2 Circuit Mode Select	0: Generate voltage monitor 2 interrupt on V_{det2} passage 1: Enable voltage monitor 2 reset when the voltage falls to and below V_{det2}.	R/W
b7	RN	Voltage Monitor 2 Reset Negate Select	0: Negate after a stabilization time (t_{LVD2}) when VCC > V_{det2} is detected 1: Negate after a stabilization time (t_{LVD2}) on assertion of the LVD2 reset.	R/W

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

RIE bit (Voltage Monitor 2 Interrupt/Reset Enable)

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

RN bit (Voltage Monitor 2 Reset Negate Select)

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

8.3 VCC Input Voltage Monitor

8.3.1 Monitoring V_{det0}

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

8.3.2 Monitoring V_{det1}

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

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

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

8.3.3 Monitoring Vdet2

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

Table 8.3 Procedure to set up monitoring against V_{det2} (1 of 2)

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

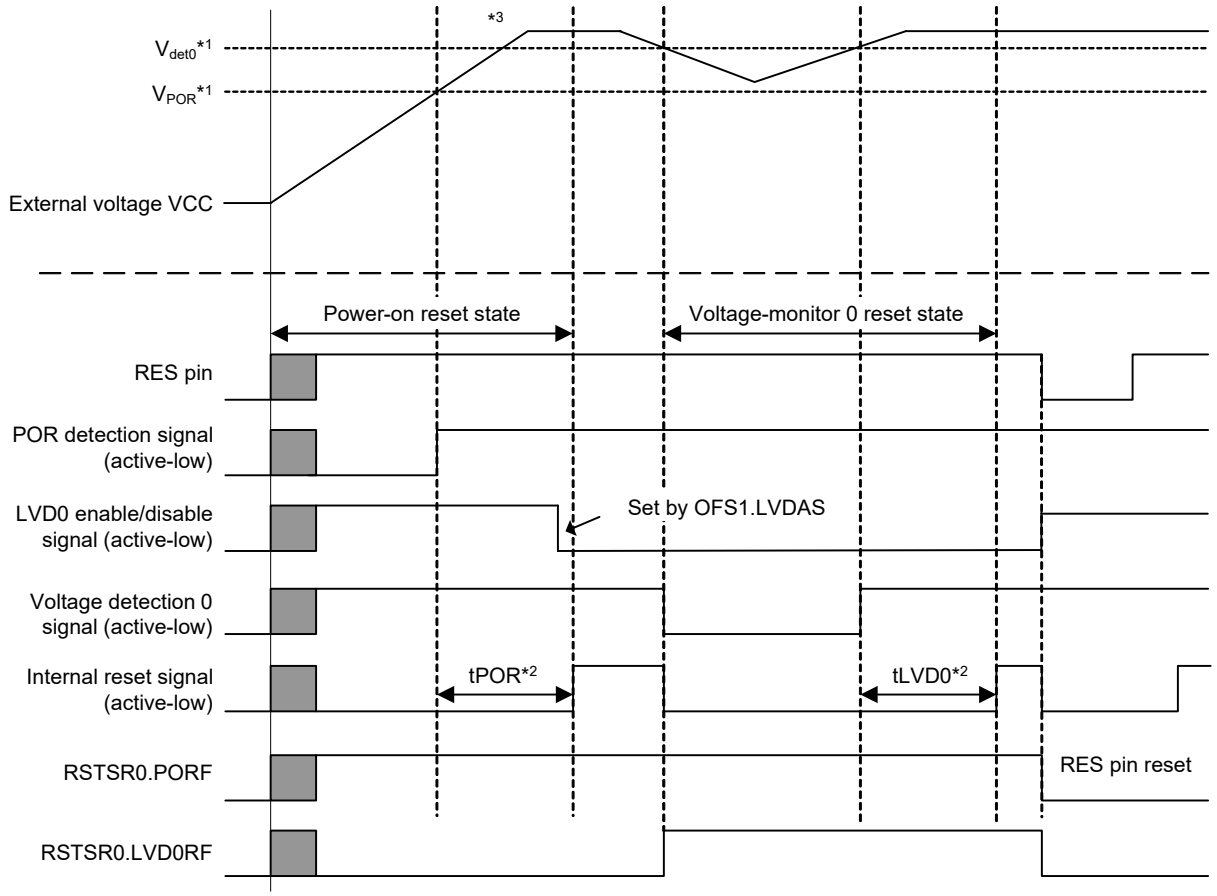
Table 8.3 Procedure to set up monitoring against V_{det2} (2 of 2)

Step		Monitoring the comparison results from voltage monitor 2
Enabling output	5	Set LVD2CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 2.

8.4 Reset from Voltage Monitor 0

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

Figure 8.4 shows an example of operations for a voltage monitor 0 reset.



Note: For details of the electrical characteristics, see section 47, Electrical Characteristics.
Note 1. V_{POR} indicates the detection level for a power-on reset and V_{det0} indicates the detection level for a voltage monitor 0 reset.
Note 2. t_{POR} indicates the time until the power-on reset is released and t_{LVD0} indicates the time until the LVD0 reset is released.
Note 3. At power-on, the VCC should rise to the minimum guaranteed voltage before the POR reset is released.

Figure 8.4 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 comparison results from the voltage monitor 1 circuit.

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

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

(1) Setting in Software Standby mode

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

Table 8.4 Procedure for setting bits related to the voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring 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[4:0] bits.	
	3	Set LVCMPCR.LVD1E = 1 to enable the voltage detection 1 circuit.	
	4	Wait for at least $t_{d(E-A)}$ for the LVD operation stabilization time after LVD is enabled.*1	
Setting the voltage monitor 1 interrupt or reset	5	Set LVD1CR0.RI = 0 to select the voltage monitor 1 interrupt.	• Set LVD1CR0.RI = 1 to select the voltage monitor 1 reset • Select the type of reset negation 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 $t_{d(E-A)}$, see section 47, Electrical Characteristics.

Note 2. Step 8 is not required if only the ELC event signal is to be output.

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

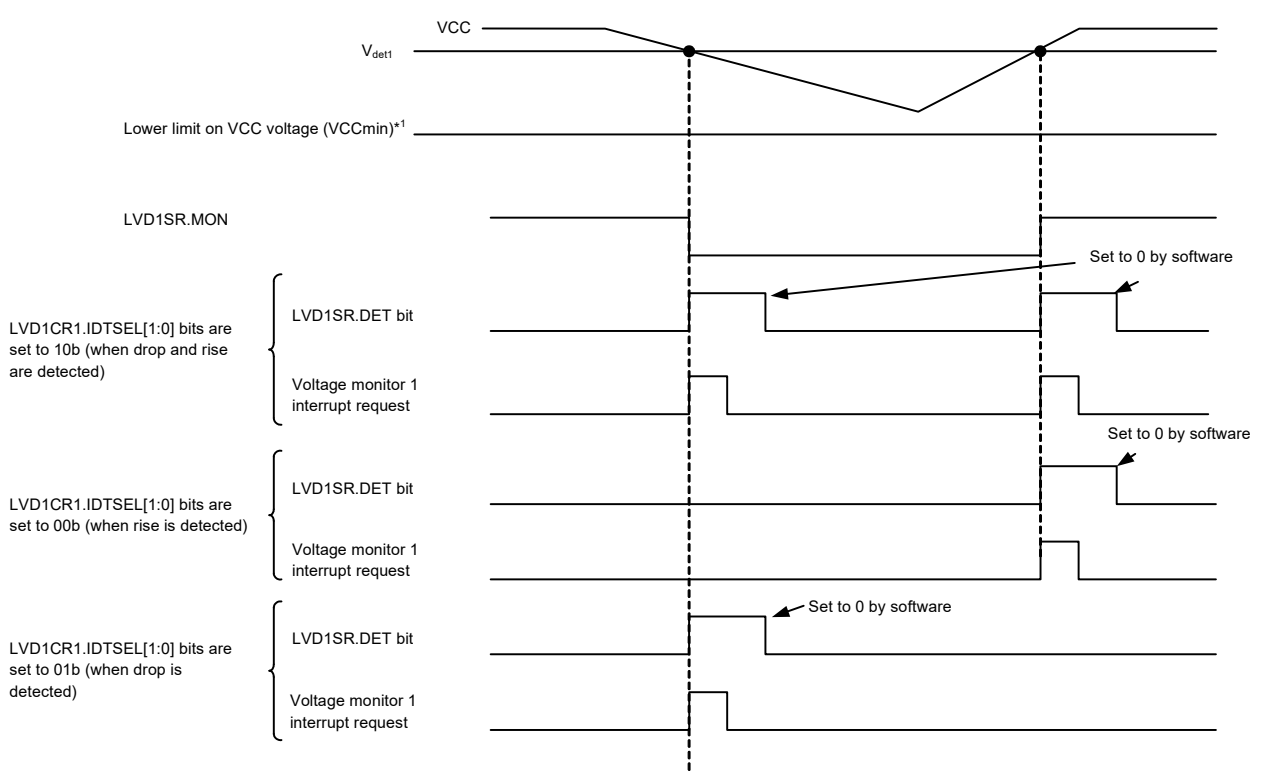
Step		Voltage monitor 1 interrupt (voltage monitor 1 ELC event output), voltage monitor 1 reset
Stopping the enabling of output	1	Set LVD1CR0.CMPE = 0 to disable output of the comparison results from voltage monitor 1.
	2	Set LVD1CR0.RIE = 0 to disable the voltage monitor 1 interrupt or reset*1.
Stopping 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 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 reset do not change.

Figure 8.5 shows an example of the voltage monitor 1 interrupt operation.



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

Figure 8.5 Voltage monitor 1 interrupt operation example

8.6 Interrupt and Reset from Voltage Monitor 2

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

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

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

(1) Setting in Software Standby mode

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

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

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

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

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

Note 1. Steps 5 to 8 can be performed during the wait time of step 4. For details of $t_{d(E-A)}$, see section 47, Electrical Characteristics.

Note 2. Step 8 is not required if only the ELC event signal is to be output.

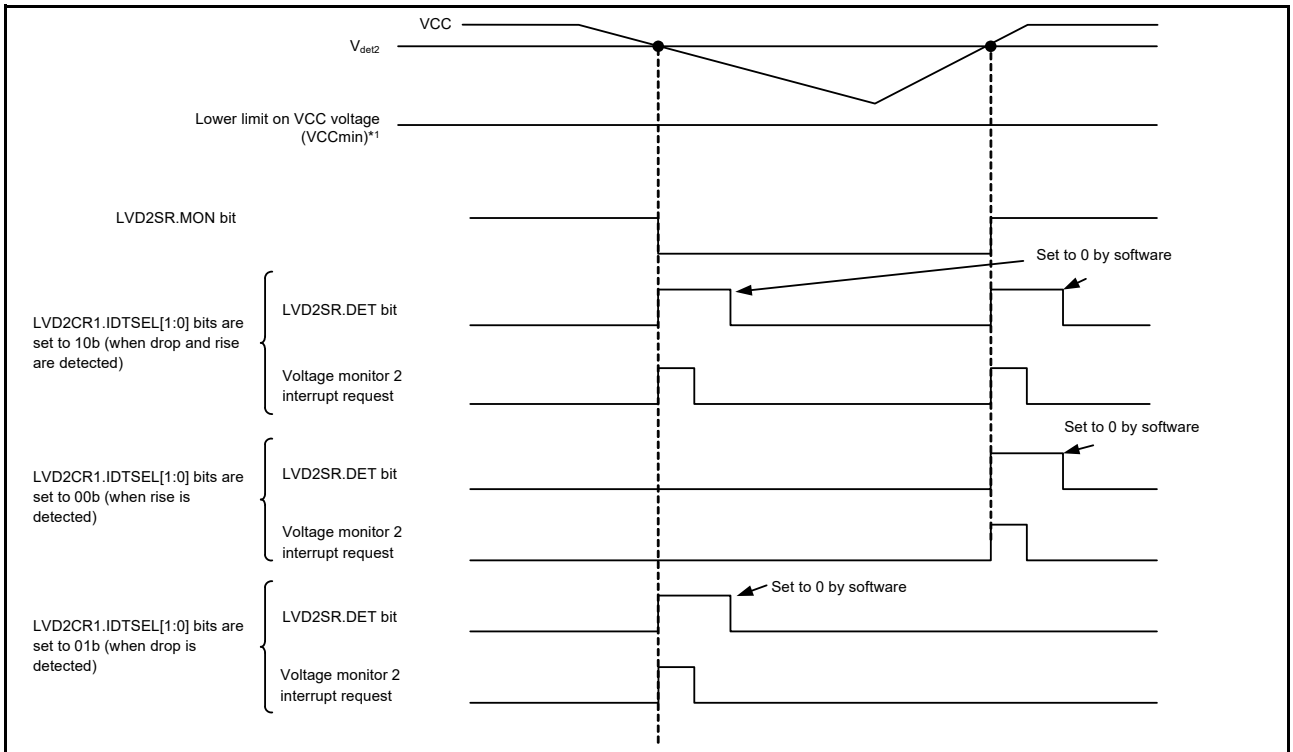
Table 8.7 Procedure for setting bits related to voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitor stops

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

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

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

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



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

Figure 8.6 Example of voltage monitor 2 interrupt operation

8.7 Event Link Output

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

(1) V_{det1} Crossing Detection Event

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

(2) V_{det2} Crossing Detection Event

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

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

8.7.1 Interrupt Handling and Event Linking

The LVD provides bits to individually enable or disable the voltage monitor 1 and 2 interrupts. When an interrupt source is generated and the interrupt is enabled by the interrupt enable bit, the interrupt signal (LVD1CR0.RIE and LVD2CR0.RIE) is output to the CPU.

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

It is possible to output voltage monitor 1 and 2 interrupts in Software Standby mode. The event signals for the ELC in Software Standby mode are output as follows:

- When a V_{det1} or V_{det2} passage event is detected in Software Standby mode, event signals are not generated for the ELC because the clock is not supplied in Software Standby mode. Because the V_{det1} and V_{det2} passage detection flags are saved, when the clock supply resumes after returning from Software Standby mode, the event signals for the ELC are output based on the state of the Vdet1 and Vdet2 detection flags.

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	Specifications
Main clock oscillator (MOSC)	Resonator frequency	1 MHz to 20 MHz*1
	External clock input frequency	Up to 20 MHz*1
	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	
High-speed on-chip oscillator (HOCO)	Oscillation frequency	24/32/48/64 MHz
	User trimming	Available
Middle-speed on-chip oscillator (MOCO)	Oscillation frequency	8 MHz
	User trimming	Available
Low-speed on-chip oscillator (LOCO)	Oscillation frequency	32.768 kHz
	User trimming	Available
IWDT-dedicated on-chip oscillator (IWDTLOCO)	Oscillation frequency	15 kHz
	User trimming	No
External clock input for SWD (SWCLK)	Input clock frequency	Up to 12.5 MHz

Note 1. The frequency depends on the supply voltage. See the table of Clock timing in the Electrical Characteristics chapter for more information.

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	CPU, DTC, FLASH, SRAM	Up to 48 MHz Division ratios: 1/2/4/8/16/32/64
Peripheral module clock B (PCLKB)	MOSC/SOSC/HOCO/ MOCO/LOCO	Peripheral module (CAC, ELC, I/O Ports, KINT, POEG, GPT, AGT, RTC, WDT, IWDT, USBFS, SCI, IIC, CAN, SPI, CRC, ADC16, SDADC24, DAC8, DAC12, OPAMP, ACMPHS, ACMPLP, CTSU, DOC, AES, and TRNG)	Up to 32 MHz Division ratios:1/2/4/8/16/32/64
Peripheral module clock D (PCLKD)	MOSC/SOSC/HOCO/ MOCO/LOCO	Peripheral module (GPT count clock, ADC16 conversion clock)	Up to 64 MHz (when ADC16 is not used) 1 MHz to 32 MHz (when ADC16 is used) Division ratios: 1/2/4/8/16/32/64
Flash interface clock (FCLK)	MOSC/SOSC/HOCO/ MOCO/LOCO	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	USBFS	48 MHz
CAN clock (CANMCLK)	MOSC	CAN	1 MHz to 20 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
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
24-bit Sigma-Delta A/D converter clock (SDADCCLK)	MOSC/HOCO	SDADC24	4 MHz to 20 MHz (MOSC) 24/32/48/64 MHz (HOCO)

Note: Restrictions on setting the clock frequency: ICLK ≥ PCLKB, PCLKD ≥ PCLKB
Restrictions on the clock frequency ratio: (N: integer, and up to 64)
ICLK:FCLK = N:1, ICLK:PCLKB = N:1, ICLK:PCLKD = N:1 or 1:N

Note: The minimum FCLK frequency is 1 MHz in Programming/Erasure (P/E) mode.

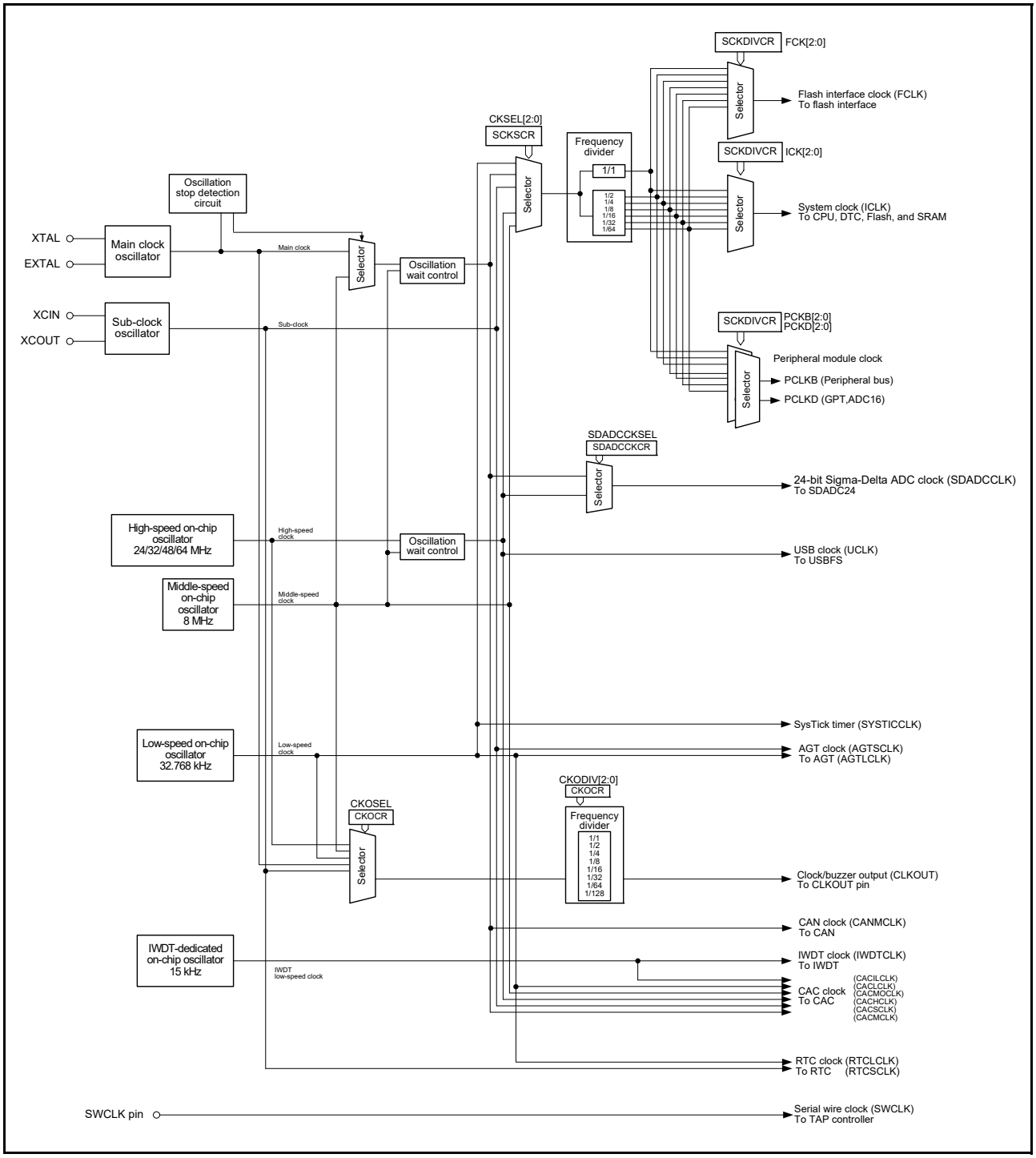


Figure 9.1 Clock generation circuit block diagram

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

Table 9.3 IClock generation circuit input/output pins

Pin name	I/O	Description
XTAL	Output	These pins are used to connect a crystal resonator. The EXTAL pin can also be used to input an external clock. For details, section 9.3.2, External Clock Input.
EXTAL	Input	
XCIN	Input	These pins are used to connect a 32.768-kHz crystal resonator
XCOUT	Output	
CLKOUT	Output	This pin is used to output the CLKOUT/BUZZER clock
SWCLK	Input	This pin is used to input from the SWD

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	0	0	0

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

Bit	Symbol	Bit name	Description	R/W
b2 to b0	PCKD[2:0]	Peripheral Module Clock D (PCLKD) Select*2	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
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	PCKB[2:0]	Peripheral Module Clock B (PCLKB) Select*1	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
b23 to b11	—	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

Bit	Symbol	Bit name	Description	R/W
b30 to b28	FCK[2:0]	FlashIF Clock (FCLK) Select*3	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 peripheral module clock (PCLKB) should be ICLK:PCLKB = N:1 (N: integer).
If a setting is made where ICLK < PCLKB, then that setting is ignored.

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

Note 3. 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 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), peripheral module clock (PCLKB, 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).

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

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

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

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

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

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). Other settings are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Selecting a system clock source that is faster than 32 MHz (system clock source > 32 MHz) is prohibited when the SCKDIVCR.ICK[2:0] bits select division by 1 and MEMWAIT.MEMWAIT = 0.

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

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

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

- System clock (ICLK)
- Peripheral module clocks (PCLKB 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).

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 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 bit selects division by 1 and SCKSCR.CKSEL[2:0] bits select the system clock source that is faster than 32 MHz (ICLK > 32 MHz).

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).

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

Table 9.4 MEMWAIT bit setting

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

✓: Setting is possible.
x : Setting is not possible.

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

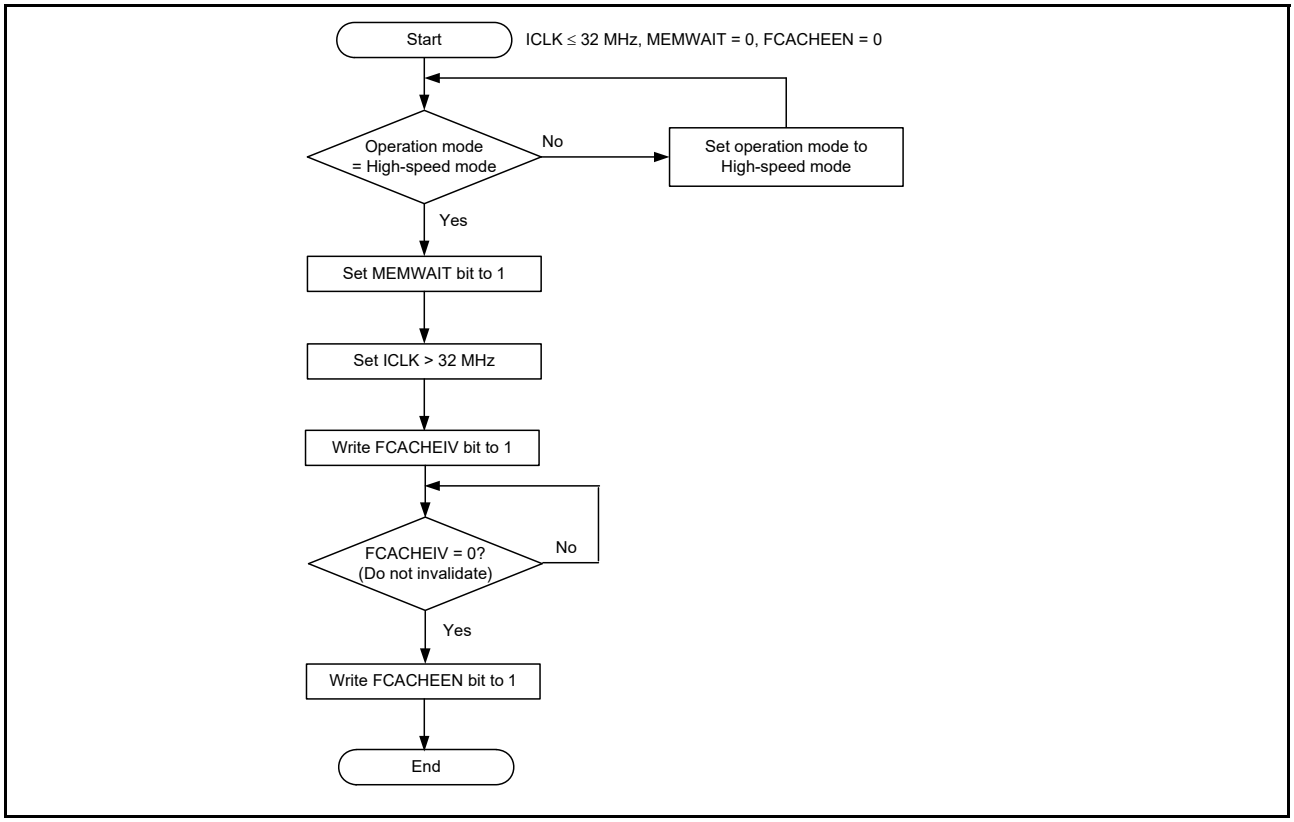


Figure 9.2 When setting the ICLK > 32 MHz

Figure 9.3 shows an example of setting the ICLK ≤ 32 MHz from ICLK > 32 MHz.

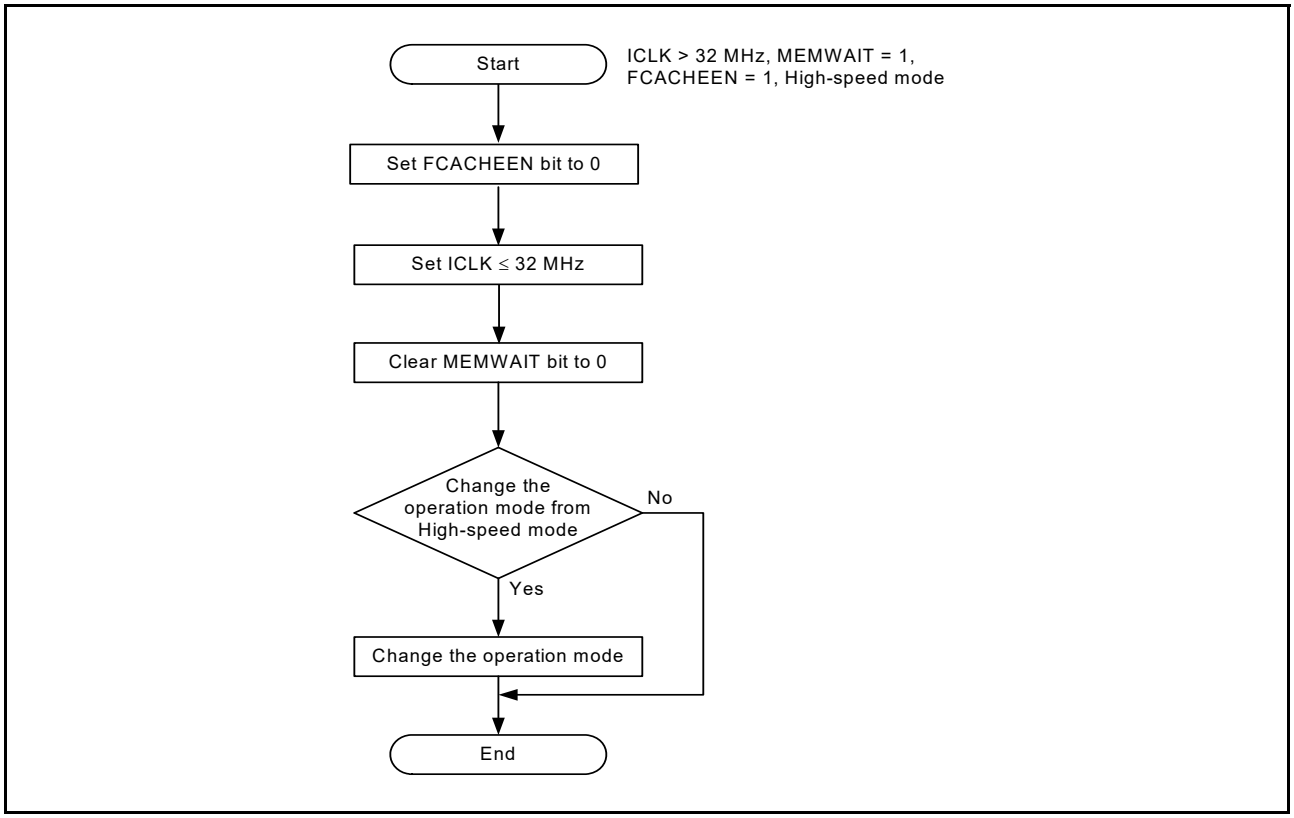


Figure 9.3 When setting the ICLK ≤ 32 MHz from ICLK > 32 MHz

9.2.4 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 the value is updated.

When using the main clock, the Main Clock Oscillator Mode Oscillation Control Register (MOMCR) and the Main Clock Oscillator Wait Control Register (MOSCWTCR) must be set before setting MOSTP to 0. When the MOSCCR.MOSTP bit 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 stabilization wait time is required after setting the main clock oscillator to start operation. A fixed wait time is also required for oscillation to stop after stopping the main clock oscillator.

The following restrictions apply when starting and stopping operation:

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

Writing 1 to MOSTP is prohibited under the following condition:

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

9.2.5 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 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.

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, only execute subsequent instructions after reading the bit to check that the value is updated. Use the SOSTP bit when using the sub-clock oscillator as the source for a peripheral module, for example the RTC. When using the sub-clock oscillator, set the Sub-Clock Oscillator Mode Control Register (SOMCR) before setting SOSTP to 0.

After setting the SOSTP bit to 0, only use the sub-clock oscillator after the sub-clock oscillation stabilization wait time (t_{SUBOSC}) elapses. A fixed stabilization wait time is required after selecting the sub-clock operation with the SOSTP bit. A fixed wait time is also required for oscillation to stop.

The following restrictions apply when starting and stopping the operation:

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

Writing 1 to SOSTP is prohibited under the following condition:

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

9.2.6 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: Stop the LOCO clock.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The LOCOCR register controls the LOCO clock.

LCSTP bit (LOCO Stop)

The LCSTP bit starts or stops the LOCO clock.

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

The following restrictions apply when starting and stopping operation:

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

Writing 1 to LCSTP is prohibited under the following condition:

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

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

Address(es): SYSTEM.HOCOCR 4001 E036h

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

Bit	Symbol	Bit name	Description	R/W
b0	HCSTP	HOCO Stop	0: Operate the HOCO clock*2, *4 1: Stop the HOCO clock.	R/W*3
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 1. The HCSTP bit value after a reset is 0 when the OFS1.HOCOEN bit is 0. It is 1 when the OFS1.HOCOEN bit is 1.

Note 2. If the operating frequency of HOCO is 48 MHz, VCC must be more than 1.8 V (VCC ≥ 1.8 V) when operating the HOCO. If the operating frequency of HOCO is 64 MHz, VCC must be more than 2.4 V (VCC ≥ 2.4 V) when operating the HOCO.

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

Note 4. When using the HOCO (HCSTP = 0), set the OFS1.HOCOFRQ1[2:0] bits to an optimum value. During low-voltage mode, HOCOCR.HCSTP bit must always be 0.

The HOCOCR register controls the HOCO clock.

HCSTP bit (HOCO Stop)

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

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

The following are restrictions when starting and stopping the 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 clock 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 while HOCOCR. HCSTP bit is 0
- 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).

9.2.8 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: MOCO is operating 1: MOCO is stopped.	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 clock.

MCSTP bit (MOCO Stop)

The MCSTP bit starts or stops the MOCO clock.

After setting the MCSTP bit to 0, use the MOCO clock only after the MOCO clock oscillation stabilization time (t_{MOCO}) elapses. A fixed stabilization wait time is required after setting the MOCO clock to start operation. A fixed wait time is also required for oscillation to stop after setting MCSTP to 1.

The following restrictions apply when starting and stopping the oscillator:

- After stopping the MOCO clock, allow a stop interval of at least 5 MOCO clock cycles before restarting it
- Confirm that MOCO oscillation is stable before stopping the MOCO clock
- Regardless of whether the MOCO clock is selected as the system clock, confirm that MOCO oscillation is stable

before executing a WFI instruction to place the MCU in Software Standby mode

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

Writing 1 to MCSTP is prohibited under the following condition:

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

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

Because the MOCO clock is used to measure the wait time for other oscillators, the MOCO clock oscillates while the wait time for other oscillators is being measured, regardless of the setting of MOCOCR.MCSTP. Therefore, the MOCO clock might be unintentionally supplied even if the MCSTP is set to stop.

9.2.9 Oscillation Stabilization Flag Register (OSCSF)

Address(es): SYSTEM.OSCSF 4001 E03Ch

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

Bit	Symbol	Bit name	Description	R/W
b0	HOCOSF	HOCO Clock Oscillation Stabilization Flag	0: The HOCO clock is stopped or is not yet stable 1: The HOCO clock is stable, so is available for use as the system clock.	R
b2, b1	—	Reserved	These bits are read as 0	R
b3	MOSCSF	Main Clock Oscillation Stabilization Flag	0: The main clock oscillator is stopped (MOSTP = 1) or is not yet stable*2 1: The main clock oscillator is stable, so is available for use as the system clock.	R
b7 to b4	—	Reserved	These bits are read as 0	R

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

Note 2. An appropriate value is set in the Wait Control register for the given oscillator. If the wait time is not sufficient, the oscillation stabilization flag is set to 1 and supply of the clock signal to the internal circuits starts before oscillation is stable.

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

HOCOSF flag (HOCO Clock Oscillation Stabilization Flag)

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

[Setting condition]

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

[Clearing condition]

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

MOSCSF flag (Main Clock Oscillation Stabilization Flag)

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

[Setting condition]

- 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 middle-speed clock cycles set in the MOSCWTCR.MSTS[3:0] bits elapse.

[Clearing condition]

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

9.2.10 Oscillation Stop Detection Control Register (OSTDCR)

Address(es): SYSTEM.OSTDCR 4001 E040h

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

Bit	Symbol	Bit name	Description	R/W
b0	OSTDIE	Oscillation Stop Detection Interrupt Enable	0: Disable oscillation stop detection interrupt (do not notify the POEG) 1: Enable oscillation stop detection interrupt (notify the POEG).	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OSTDE	Oscillation Stop Detection Function Enable	0: Disable oscillation stop detection function 1: Enable oscillation stop detection function.	R/W

The OSTDCR register controls the oscillation stop detection function.

OSTDIE bit (Oscillation Stop Detection Interrupt Enable)

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

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

OSTDE bit (Oscillation Stop Detection Function Enable)

The OSTDE bit enables the oscillation stop detection function.

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

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

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

The following restrictions apply when using the oscillation stop detection function:

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

9.2.11 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 oscillation is restarted. The OSTDF flag is set to 0 by writing 0 after reading it as 1.

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

OSTDSR.OSTDF cannot be set to 0 under the following conditions:

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

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

[Setting condition]

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

[Clearing condition]

- 1 is read and then 0 is written when the SCKSCR.CKSEL[2:0] bits are not 011b (system clock is MOSC).

9.2.12 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 clock automatically oscillates when necessary, regardless of the value of the MOCOCR.MCSTP bit. After the specified wait time elapses, supply of the main clock starts internally in the MCU, and the OSCSF.MOSCSF flag is set to 1. If the specified wait time is short, supply of the main clock starts before oscillation of the clock becomes stable.

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

9.2.13 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) When HOCO operating frequency is 48 MHz and the operation power control mode is other than low-voltage mode • Wait time = 679 cycles (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 this register under any other conditions.

HSTS[2:0] bits (HOCO wait time setting)

The oscillation stabilization wait circuit measures the wait time and controls the clock supply in the MCU 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 setting. Hardware automatically controls the running and stopping of the middle-speed oscillator for wait time measurement.

9.2.14 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 state, the pin is set as a port.
Note: The MOSCCR.MOSTP bit must be 1 (MOSC is stopped) before changing this register.

MODRV1 bit (Main Clock Oscillator Drive Capability 1 Switching)

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

MOSEL bit (Main Clock Oscillator Switching)

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

9.2.15 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.

9.2.16 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, or 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 47, Electrical Characteristics.

CKOEN bit (Clock Out Enable)

The CKOEN bit enables output from the CLKOUT pin.

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

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

9.2.17 LOCO User Trimming Control Register (LOCOUTCR)

Address(es): SYSTEM.LOCOUTCR 4001 E492h

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

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

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

9.2.18 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 frequency stabilization wait time corresponds to the time when it is stabilized at the start of the MCU operation. When the ratio of the MOCO frequency and the other oscillation frequency is an integer value, changing the MOCOUTCR value is prohibited.

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

When UCKSEL.UCKSELC = 1, writing any other value except 00h to HOCOUTCR is prohibited. For UCKSEL register, see section 26, USB 2.0 Full-Speed Module (USBFS).

9.2.20 24-bit Sigma-Delta A/D Converter Clock Control Register (SDADCCKCR)

Address(es): SYSTEM.SDADCCKCR 4001 E0D1h

	b7	b6	b5	b4	b3	b2	b1	b0
	SDADC CKEN	—	—	—	—	—	—	SDADC CKSEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SDADCCKSEL	24-bit Sigma-Delta A/D Converter Clock Select	0: MOSC is selected as the source clock of the 24-bit Sigma-Delta A/D Converter Clock 1: HOCO is selected as the source clock of the 24-bit Sigma-Delta A/D Converter Clock.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SDADCCKEN	24-bit Sigma-Delta A/D Converter Clock Enable	0: 24-bit Sigma-Delta A/D Converter Clock is disabled 1: 24-bit Sigma-Delta A/D Converter Clock is enabled.	R/W

SDADCCKSEL bit (24-bit Sigma-Delta A/D Converter Clock Select)

The SDADCCKSEL bit specifies the source clock of the 24-bit Sigma-Delta A/D Converter Clock.

SDADCCKEN bit (24-bit Sigma-Delta A/D Converter Clock Enable)

The SDADCCKEN bit enables the 24-bit Sigma-Delta A/D Converter Clock.

When SDADCCKEN bit is set to 1, the 24-bit Sigma-Delta A/D Converter Clock is output after 2 cycles of the source clock. When SDADCCKEN bit is set to 0, the 24-bit Sigma-Delta A/D Converter Clock is stopped after 2 cycles of the source clock. To modify the SDADCCKSEL bit, you must completely stop the 24-bit Sigma-Delta A/D Converter

Clock. To transition to Software Standby mode, set the SDADCCKEN bit to 0.

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 the use of an external feedback resistor (Rf), insert an Rf between EXTAL and XTAL by following the instructions.

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

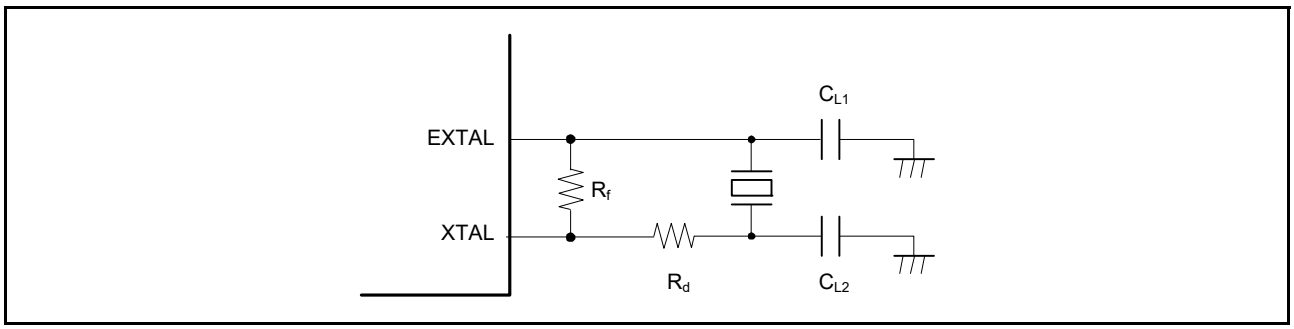


Figure 9.4 Example of crystal resonator connection

9.3.2 External Clock Input

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

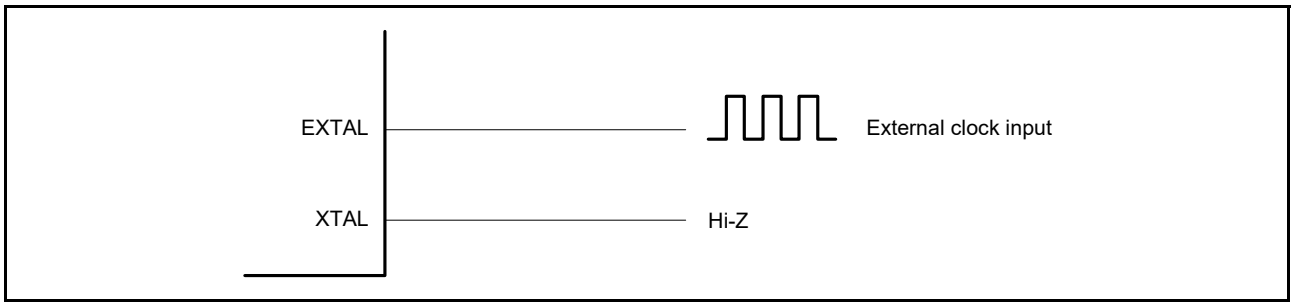


Figure 9.5 Equivalent circuit for external clock

9.3.3 Notes on External Clock Input

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

9.4 Sub-Clock Oscillator

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

9.4.1 Connecting a 32.768-kHz Crystal Resonator

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

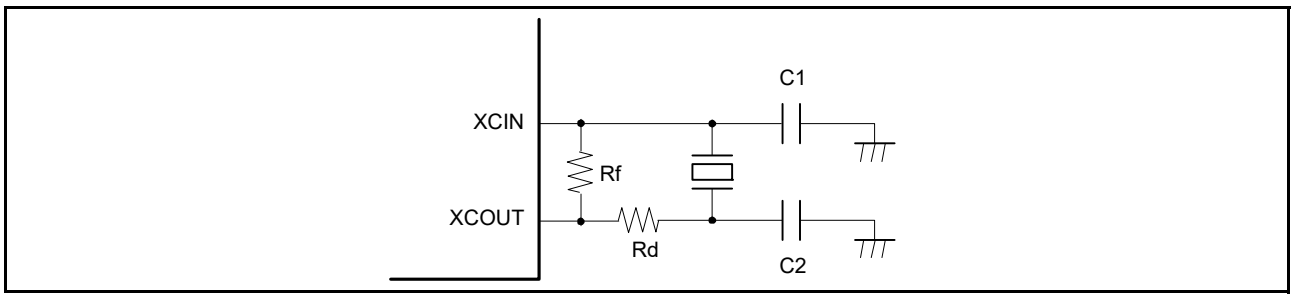


Figure 9.6 Connection example of 32.768-kHz crystal resonator

9.5 Oscillation Stop Detection Function

9.5.1 Oscillation Stop Detection and Operation after Detection

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

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

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

The main clock oscillation stop is detected when the input clock remains at 0 or 1 for a certain period, for example, when a malfunction occurs in the main clock oscillator. See section 47, Electrical Characteristics.

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

OSTDF controls the switched clock as follows:

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

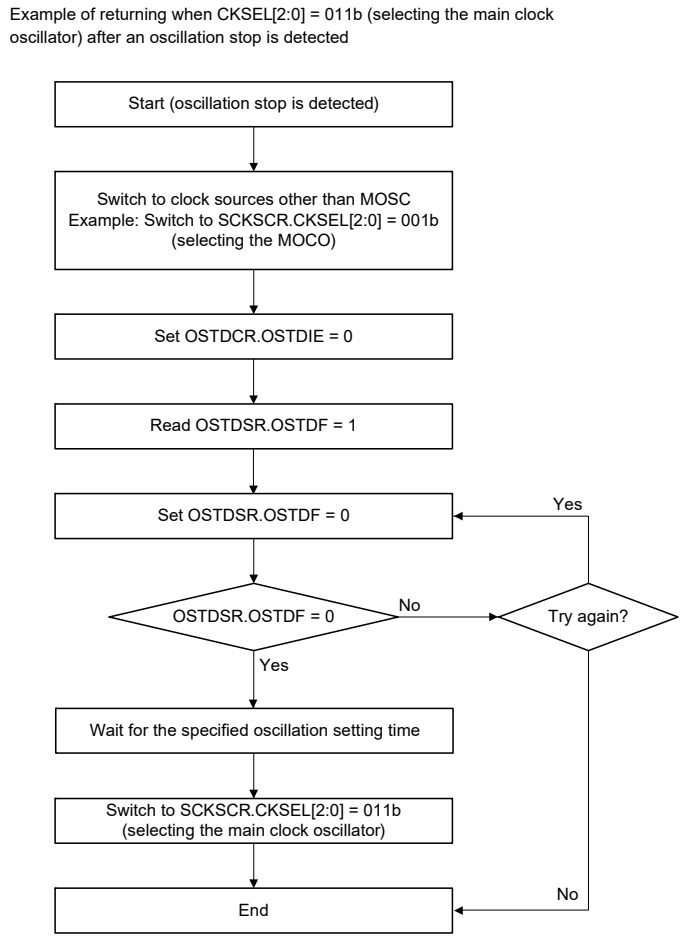
To switch the clock source to the main clock again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock and clear the OSTDF flag to 0. Also, check that the OSTDF flag is not 1, then set the CKSEL[2:0] bits to the main clock after the specified oscillation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

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

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

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

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



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 9.7 Flow of recovery on detection of oscillator stop

9.5.2 Oscillation Stop Detection Interrupts

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

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

The oscillation stop detection interrupt is a non-maskable interrupt. Because non-maskable interrupts are disabled in the initial state after a reset release, enable non-maskable interrupts through software before using oscillation stop detection interrupts. For details, see section 13, Interrupt Controller Unit (ICU).

9.6 Internal Clock

Clock sources for the internal clock signals include:

- Main clock oscillator
- Sub-clock oscillator

- HOCO clock
- MOCO clock
- LOCO clock
- Dedicated clock for the IWDT.

The following internal clocks are produced from these sources:

- Operating clock for the CPU, DTC, flash memory, and SRAM — System clock (ICLK)
- Operating clocks for peripheral modules — PCLKB 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
- Clock for external pin output — CLKOUT
- Operating clock for the 24-bit Sigma-Delta A/D Converter — SDADCCLK.

For details of the registers used to set the frequencies of the internal clocks, see section 9.6.1, System Clock (ICLK) to section 9.6.12, 24-bit Sigma-Delta A/D Converter Clock (SDADCCLK).

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

9.6.1 System Clock (ICLK)

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

The ICLK frequency is specified in the following bits:

- ICK[2:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- 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 9.8 and Figure 9.9.

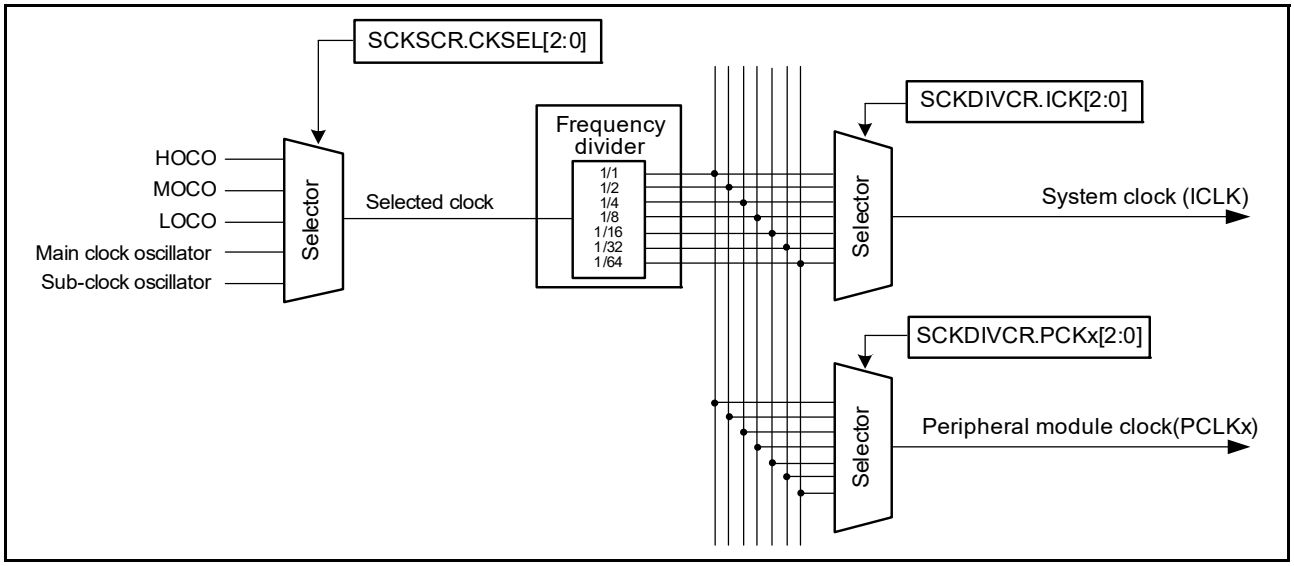


Figure 9.8 Clock source selector block diagram

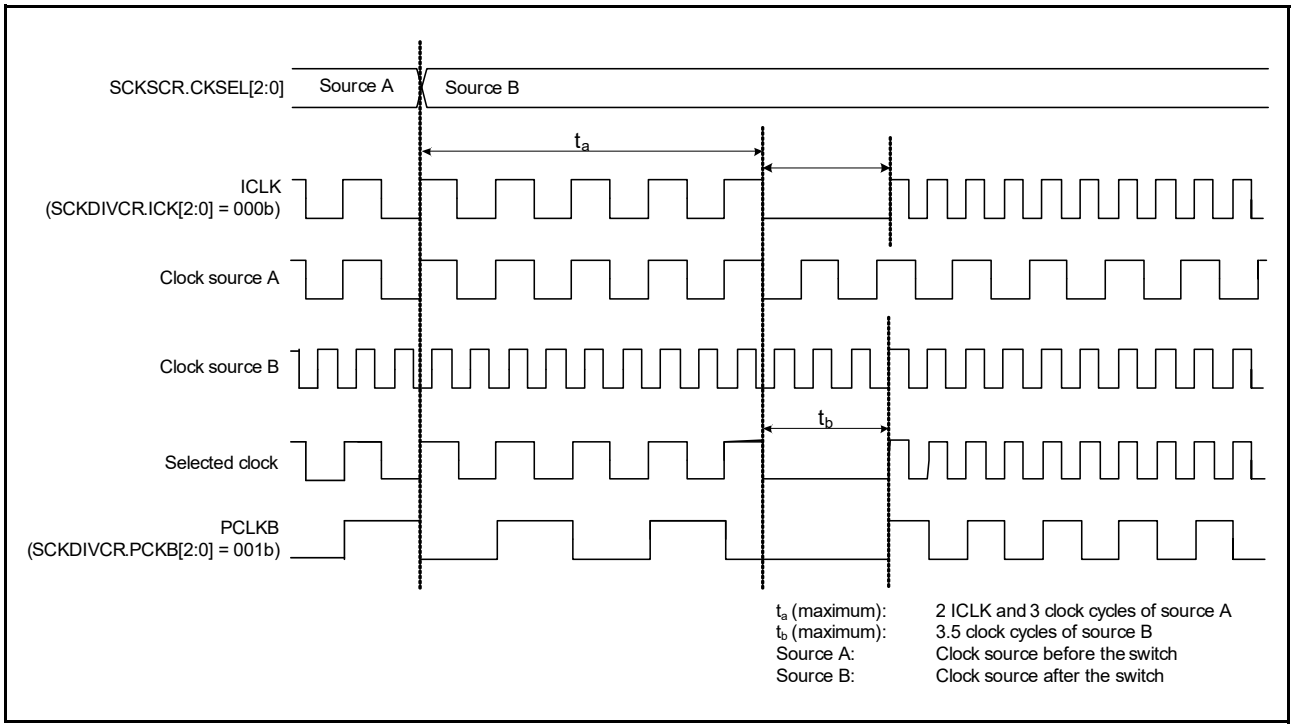


Figure 9.9 Clock source switching timing diagram

9.6.2 Peripheral Module Clock (PCLKB, PCLKD)

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

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

- PCKB[2:0] and PCKD[2:0] in SCKDIVCR
- CKSEL[2:0] in SCKSCR
- HOCOFRQ1[2:0] 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 9.8 and Figure 9.9.

9.6.3 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 register
- CKSEL[2:0] bits in SCKSCR register
- HOCOFRQ1[2:0] bits in OFS1 register.

9.6.4 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 by HOCOFRQ1[2:0] bits in OFS1.

9.6.5 CAN Clock (CANMCLK)

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

9.6.6 CAC Clock (CACCLK)

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

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

9.6.7 RTC-Dedicated Clock (RTCSCLK, RTCLCLK)

The RTC-dedicated clocks, RTCSCLK and RTCLCLK, are the operating clocks for the RTC. RTCSCLK is generated by the sub-clock oscillator, and RTCLCLK is generated by the LOCO clock.

9.6.8 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.

9.6.9 AGT-Dedicated Clock (AGTSCLK, AGTLCLK)

The AGT-dedicated clocks, AGTSCLK and AGTLCLK, are the operating clocks for the AGT. AGTSCLK is generated by the sub-clock oscillator, and AGTLCLK is generated by the LOCO clock.

9.6.10 SysTick Timer-Dedicated Clock (SYSTICCLK)

The SysTick timer-dedicated clock, SYSTICCLK, is the operating clock for the SysTick Timer. SYSTICCLK is generated by the LOCO clock.

9.6.11 Clock/Buzzer Output Clock (CLKOUT)

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

The CLKOUT clock frequency is specified in the following bits:

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

9.6.12 24-bit Sigma-Delta A/D Converter Clock (SDADCCLK)

The 24-bit Sigma-Delta A/D Converter Clock (SDADCCLK) is an operating clock for SDADC24.

SDADCCLK is output when SDADCCKCR.SDADCCKEN is set to 1. SDADCCLK is generated by the HOCO or main clock oscillator and is specified by the SDADCCKCR.SDADCCKSEL bit.

9.7 Usage Notes

9.7.1 Notes on Clock Generation Circuit

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

- Each frequency must be selected within the operation-guaranteed range of the clock cycle time (t_{cyc}) specified in the AC electrical characteristics, see section 47, Electrical Characteristics
- The frequencies must not exceed the ranges listed in Table 9.2
- The peripheral modules operate on PCLKB. 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 (PCLKB and PCLKD), and flash interface clock (FCLK) must be set according to Table 9.2.

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

9.7.2 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.

9.7.3 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. Route other signal lines away from the oscillation circuit as shown in Figure 9.10, to prevent electromagnetic induction from interfering with correct oscillation.

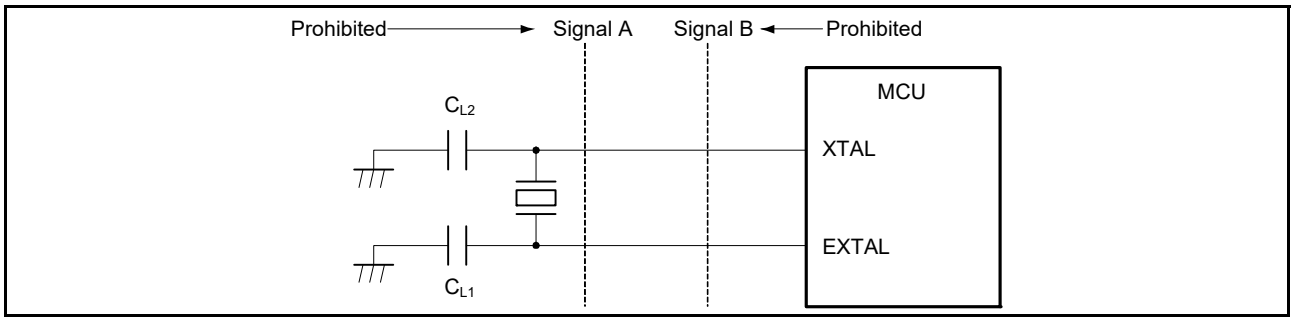


Figure 9.10 Signal routing in board design for oscillation circuit (applicable to the main clock oscillator as well as sub-clock oscillator)

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

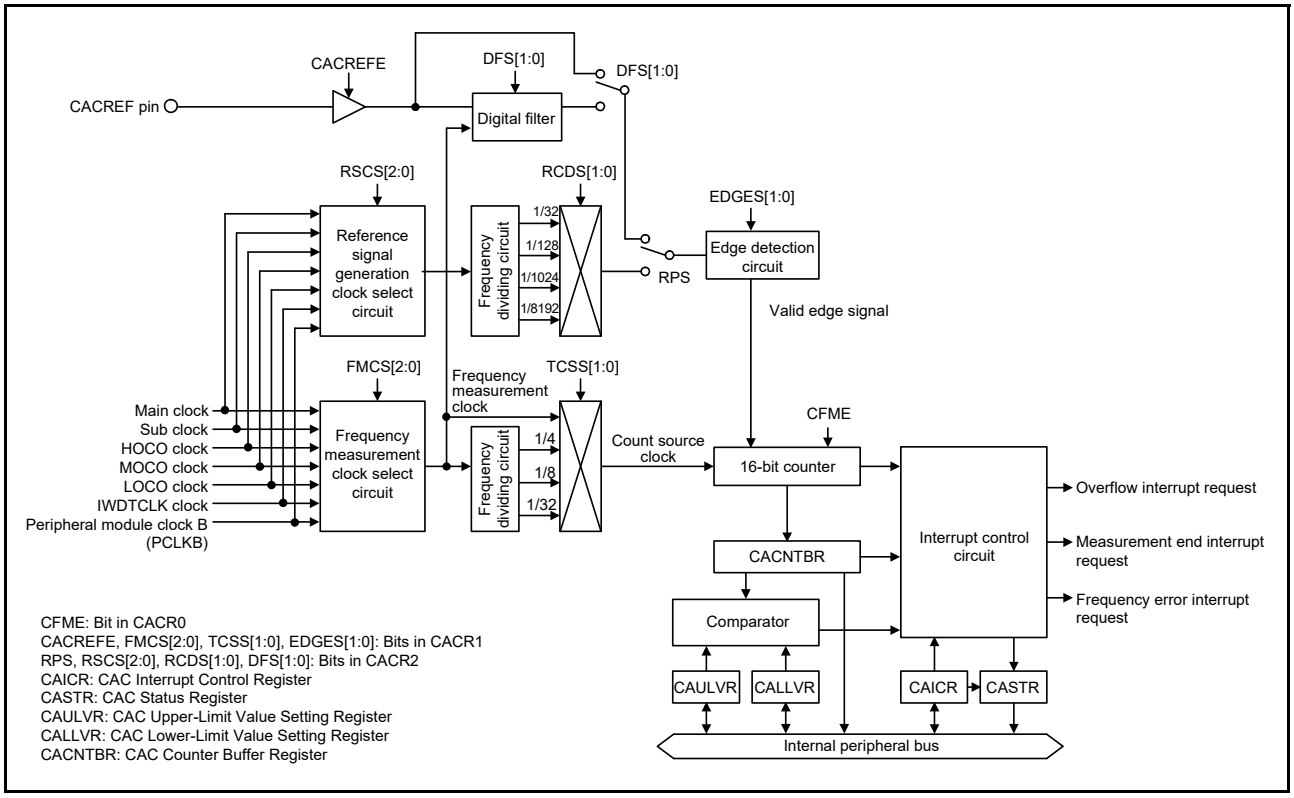


Figure 10.1 CAC block diagram

Table 10.2 CAC pin configuration

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

10.2 Register Descriptions

10.2.1 CAC Control Register 0 (CACR0)

Address(es): CAC.CACR0 4004 4600h

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

Bit	Symbol	Bit name	Description	R/W
b0	CFME	Clock Frequency Measurement Enable	0: Clock frequency measurement is disabled 1: Clock frequency measurement is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CFME bit (Clock Frequency Measurement Enable)

The CFME bit enables clock frequency measurement. Read the CFME bit to confirm that the bit value has changed. Additional write accesses are ignored before the change is complete.

10.2.2 CAC Control Register 1 (CACR1)

Address(es): CAC.CACR1 4004 4601h

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

Bit	Symbol	Bit name	Description	R/W
b0	CACREFE	CACREF Pin Input Enable	0: CACREF pin input is disabled 1: CACREF pin input is enabled.	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: Frequency error interrupt request is disabled 1: Frequency error interrupt request is enabled.	R/W
b1	MENDIE	Measurement End Interrupt Request Enable	0: Measurement end interrupt request is disabled 1: Measurement end interrupt request is enabled.	R/W
b2	OVFIE	Overflow Interrupt Request Enable	0: Overflow interrupt request is disabled 1: Overflow interrupt request is enabled.	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 overflows.

[Clearing condition]

- 1 is written to the OVFFCL bit.

10.2.6 CAC Upper-Limit Value Setting Register (CAULVR)

Address(es): CAC.CAULVR 4004 4606h

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

CAULVR is a 16-bit read/write register that specifies the upper value of the allowable range. When the counter value exceeds the value specified in this register, a frequency error is detected. Write to this register when the CACR0.CFME bit is 0.

The counter value stored in CACNTBR can vary depending on the difference between the phases of the digital filter and edge-detection circuit, and the signal on the CACREF pin. Ensure that this setting allows an adequate margin.

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 stores the measurement result.

10.3 Operation

10.3.1 Measuring Clock Frequency

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

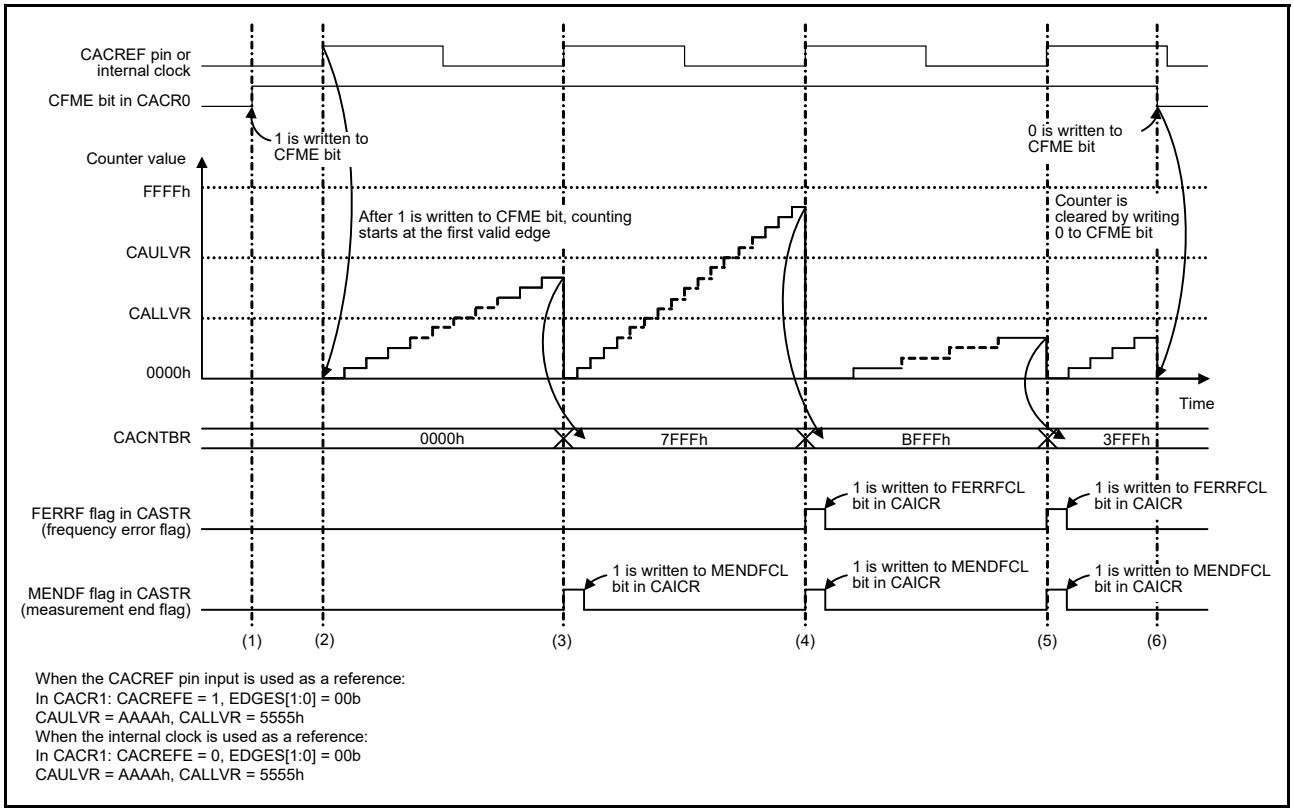


Figure 10.2 CAC operating example

In Figure 10.2:

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 in CACNTBR and compared with the values of CAULVR and CALLVR. If both CACNTBR ≤ CAULVR and CACNTBR ≥ CALLVR are true, only the MENDF flag in CASTR is set to 1 because the clock frequency is correct. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
4. When the next valid edge is input, the counter value is transferred in 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 in 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 in 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 the CACREF pin are transmitted to the internal circuitry after three consecutive matches in the selected sampling interval. The same level continues to be transmitted internally until the level on the pin has three consecutive matches again. Enabling or disabling of the digital filter and its sampling clock are selectable.

The counter value transferred 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 request:

- 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 Notes

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 DTC and SRAM operate.

Table 11.1 Specifications of the low power mode functions

Parameter	Specifications
Reducing power consumption by switching clock signals	The frequency division ratio can be selected independently for the system clock (ICLK), peripheral module clock (PCLKB and PCLKD), 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	Selectable*10,*13	Selectable*10,*13
Low-speed on-chip oscillator	Selectable	Selectable	Selectable
IWDT-dedicated on-chip oscillator	Selectable*4	Selectable*4	Selectable*4
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	Operating	Stop (Retained)	Selectable
Flash memory	Operating	Stop (Retained)	Stop (Retained)
Data Transfer Controller (DTC)	Selectable	Stop (Retained)	Selectable
USB 2.0 Full-Speed Module (USBFS)	Selectable	Stop (Retained)*5	Operation prohibited*5

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

Parameter	Sleep mode	Software Standby mode	Snooze mode*1
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
Low Power Asynchronous General Purpose Timer (AGTn, n = 0, 1)	Selectable	Selectable*6	Selectable*6
16-bit A/D Converter (ADC16)	Selectable	Stop (Retained)	Selectable*12
24-bit Sigma-Delta A/D Converter (SDADC24)	Selectable	Operation prohibited	Operation prohibited
Sensor Power Supply (SBIAS)	Selectable	Selectable	Selectable
8-bit D/A Converter (DAC8)	Selectable	Stop (Retained)	Selectable
12-bit D/A Converter (DAC12)	Selectable	Stop (Retained)	Selectable
Capacitive Touch Sensing Unit (CTSU)	Selectable	Stop (Retained)	Selectable
Data Operation Circuit (DOC)	Selectable	Stop (Retained)	Selectable
Serial Communications Interface (SCI0)	Selectable	Stop (Retained)	Selectable*9
Serial Communications Interface (SCIn, n = 1, 9)	Selectable	Stop (Retained)	Operation prohibited
I^2C Bus interface (IIC0)	Selectable	Selectable	Selectable*11
I^2C Bus interface (IIC1)	Selectable	Stop (Retained)	Operation prohibited
Event Link Controller (ELC)	Selectable	Stop (Retained)	Selectable*7
High-Speed Analog Comparator (ACMPHS)	Selectable	Selectable*8	Selectable*8
Low-Power Analog Comparator (ACMPLP0)	Selectable	Selectable*8	Selectable*8
Low-Power Analog Comparator (ACMPLP1)	Selectable	Selectable*8	Selectable*8
Operational Amplifier (OPAMP)	Selectable	Selectable	Selectable
NMI, IRQn (n = 0 to 7) pin interrupt	Selectable	Selectable	Selectable
Key Interrupt Function (KINT)	Selectable	Selectable	Selectable
Low Voltage Detection (LVD)	Selectable	Selectable	Selectable
Power-on reset circuit	Operating	Operating	Operating
Other peripheral modules	Selectable	Stop (Retained)	Operation prohibited
I/O ports	Operating	Retained	Operating

Note: Selectable means that operating or not operating can be selected 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.
Otherwise, proper operation is not guaranteed in Snooze mode.

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

Note 2. When using SCI0 in Snooze mode, MOSCCR.MOSTP bits must be 1.

Note 3. Stopped when the Clock Output Source Select bits (CKOCR.CKOSEL[2:0]) are set to a value other than 010b (LOCO) and 100b (SOSC).

Note 4. In IWDT-dedicated on-chip oscillator and IWDT, operating or stopping is selected by setting the IWDT Stop Control bit (IWDTSTPCTL) in Option Function Select Register 0 (OFS0) in IWDT auto start mode.
In WDT, operating or stopping is selected by setting the WDT Stop Control bit (WDTSTPCTL) in Option Function Select Register 0 (OFS0) in WDT auto start mode.

Note 5. Detection of USBFS resumption is possible.

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

Note 7. Event lists the restrictions described in section 11.9.13, ELC Event in Snooze Mode.

Note 8. Only VCOUT function is permitted. The VCOUT pin operates when ACMPHS and ACMPLP use no digital filter.

For details on digital filter, see section 38, High-Speed Analog Comparator (ACMPHS) and section 39, Low-Power Analog Comparator (ACMPLP).

Note 9. Serial communication modes of SCI0 is only in asynchronous mode.

Note 10. When DACPC.PUMPEN or DAPC.PUMPEN bit is 1, MOCO clock divided by 8 is supplied to switches which are used for DAC output.

Note 11. Only wakeup interrupt is available.

Note 12. When using the 16-bit A/D Converter (ADC160) in Snooze mode, the ADCMPCR.CMPAE or ADCMPCR.CMPBE bit must be 1.

Note 13. When AMPCPC.PUMP0EN, AMPCPC.PUMP1EN, or AMPCPC.PUMP2EN bit is 1, MOCO clock divided by 8 is supplied to switches which are used for OPAMP.

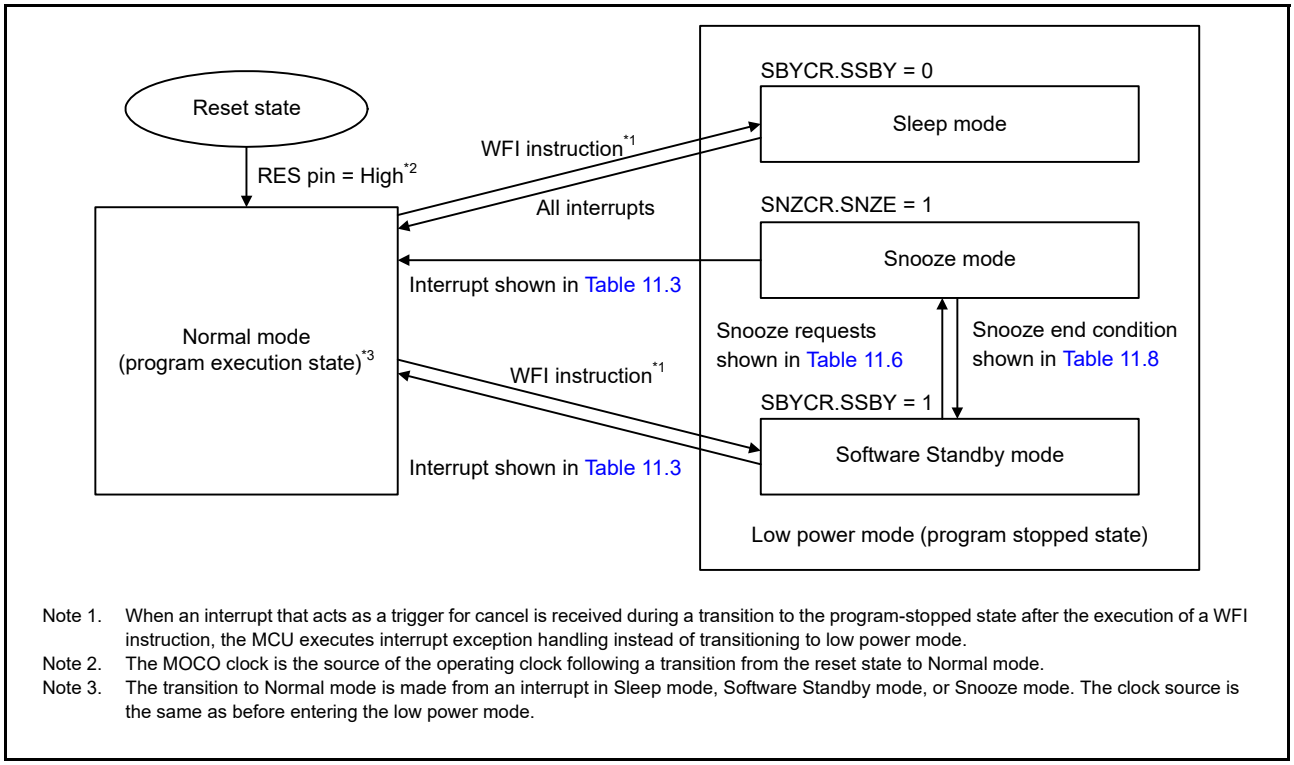
Table 11.3 Available interrupt sources to transition to Normal mode from Snooze mode and Software Standby mode

Interrupt source	Name	Software Standby mode	Snooze mode
NMI		Yes	Yes
Port	PORT_IRQn (n = 0 to 7)	Yes	Yes
LVD	LVD_LVD1	Yes	Yes
	LVD_LVD2	Yes	Yes
IWDT	IWDT_NMIUNDF	Yes	Yes
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
ADC160	ADC160_WCMPM	No	Yes with SELSR0*1, *3
	ADC160_WCMPUM	No	Yes with SELSR0*1, *3
SCI0	SCI0_AM	No	Yes with SELSR0*1, *2
	SCI0_RXI_OR_ERI	No	Yes with SELSR0*1, *2
DTC	DTC_COMPLETE	No	Yes with SELSR0*1
DOC	DOC_DOPCI	No	Yes with SELSR0*1
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 by SELSR0 register. See section 13, Interrupt Controller Unit (ICU). When a trigger selected by SELSR0 register 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. Event that is enabled by the SNZEDCR register must not be used.



Note 1. When an interrupt that acts as a trigger for cancel is received during a transition to the program-stopped state after the execution of a WFI instruction, the MCU executes interrupt exception handling instead of transitioning to low power mode.
Note 2. The MOCO clock is the source of the operating clock following a transition from the reset state to Normal mode.
Note 3. The transition to Normal mode is made 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	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after 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	SSBY	Software Standby	0: Sleep mode 1: Software Standby mode.	R/W

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

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

When the FENTRYR.FENTRY0 bit is 1 or the FENTRYR.FENTRYD bit is 1, the setting of SSBY bit is ignored. Even if 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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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
b21 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b22	MSTPA22	Data Transfer Controller Module Stop*1	Target module: 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. When rewriting the MSTPA22 bit from 0 to 1, disable the 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 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
b2	MSTPB2	Controller Area Network 0 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 Full-Speed 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

Bit	Symbol	Bit name	Description	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
b29 to b23	—	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 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 the MSTPB11 bit, wait for 2 USB clock (UCLK) cycles after writing, then execute a WFI instruction.

11.2.4 Module Stop Control Register C (MSTPCRC)

Address(es): MSTP.MSTPCRC 4004 7004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPC 31	—	—	MSTPC 28	—	—	—	—	—	—	—	—	—	—	—	—
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 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
b12 to b4	—	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

Bit	Symbol	Bit name	Description	R/W
b27 to b15	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b28	MSTPC28*2	Random Number Generator Module Stop	Target module: TRNG 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b30, b29	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b31	MSTPC31	AES Module Stop	Target module: AES 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, then execute a WFI instruction.

Note 2. Set the MSTPC28 bit once to 0 at the beginning of the program to initialize the unused circuit even if the TRNG is not used in this MCU. See section 11.9.15, Module-Stop Function for an Unused Circuit in section 11.9, Usage Notes.

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 28	—	—	—	—	—	—	—	MSTPD 20	MSTPD 19	—	MSTPD 17	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	Low Power 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	Low Power 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 320 Module Stop	Target module: GPT320 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b6	MSTPD6	General PWM Timer GPT161 to GPT166 Module Stop	Target modules: GPT161 to GPT166 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	16-bit A/D Converter Module Stop	Target module: ADC160 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W

Bit	Symbol	Bit name	Description	R/W
b17	MSTPD17	24-bit Sigma-Delta A/D Converter Module Stop	Target module: SDADC24 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b19	MSTPD19	8-bit D/A Converter Module Stop	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
b27 to b21	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b28	MSTPD28	High-Speed Analog Comparator 0 Module Stop	Target module: ACMPHS0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b29	MSTPD29	Low-Power Analog Comparator Module Stop	Target module: ACMPLP 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b30	—	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 bit 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 bit is set to 1. If the count source is the sub-clock oscillator or LOCO, this bit must be set to 1 except when accessing the AGT0 registers.

11.2.6 Operating Power Control Register (OPCCR)

Address(es): SYSTEM.OPCCR 4001 E0A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	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 register 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 bit settings.

Writing to OPCCR.OPCM[1:0] bits is prohibited while HOCOCR.HCSTP and OSCSF.HOCOSF bits 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 completes. 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
	—	—	—	SOPCMTSF	—	—	—	SOPCM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SOPCM	Sub Operating Power Control Mode Select	0: Not Subosc-speed mode 1: Subosc-speed mode.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SOPCMTSF	Sub Operating Power Control Mode Transition Status Flag	0: Transition completed 1: Transition in progress.	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SOPCCR register is used to reduce power consumption in Normal mode, Sleep mode, and Snooze mode by initiating entry to and exit from Subosc-speed mode. Subosc-speed mode is only available when using the sub-clock oscillator or LOCO without dividing the frequency.

For the procedure to change operating power control modes, see section 11.5, Function for Lower Operating Power Consumption.

SOPCM bit (Sub Operating Power Control Mode Select)

The SOPCM bit selects the operating power control mode in Normal mode, Sleep mode, and Snooze mode. Setting this bit to 1 allows transition to Subosc-speed mode. Setting this bit to 0 allows a return to the operating mode (set in OPCCR.OPCM[1:0]) before the transition to Subosc-speed mode.

Table 11.4 shows the relationship between the operating power control modes, the OPCM[1:0], and SOPCM bit 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 completes. Read this flag and confirm that it is 0 before proceeding.

Table 11.4 shows the operating power control modes.

Table 11.4 Relationship between the operating power control modes, and the OPCM[1:0] and SOPCM bits

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	0: Disable DTC operation in Snooze mode 1: Enable DTC operation in Snooze mode.	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SNZE	Snooze Mode Enable	0: Disable Snooze mode 1: Enable Snooze mode.	R/W

RXDREQEN bit (RXD0 Snooze Request Enable)

The RXDREQEN bit specifies whether to detect a falling edge of the RXD0 pin in Software Standby mode. This bit is only available when SCI0 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 transfers 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	ADC160 Compare Match Snooze End Enable	0: Disable the snooze end request 1: Enable the snooze end request.	R/W
b4	AD0UMTED	ADC160 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 in the SNZEDCR register.

AGTUNFED bit (AGT1 Underflow Snooze End Enable)

The AGTUNFED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on an AGT1 underflow. For details of the condition of the trigger, see section 22, Low Power 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 on completion of the last DTC transmission, that is, when CRA or CRB register in the DTC is 0. For details on the trigger conditions, see section 16, Data Transfer Controller (DTC).

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

The DTCNZRED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on completion of each DTC transmission, that is, when CRA or CRB register in the DTC is not 0. For details on the trigger conditions, see section 16, Data Transfer Controller (DTC).

AD0MATED bit (ADC160 Compare Match Snooze End Enable)

The AD0MATED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on an ADC160 event when the conversion result matches the expected data. For details on the trigger conditions, see section 32, 16-Bit A/D Converter (ADC16).

AD0UMTED bit (ADC160 Compare Mismatch Snooze End Enable)

The AD0UMTED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on an ADC160 event when the conversion result does not match the expected data. For details on the trigger conditions, see section 32, 16-Bit A/D Converter (ADC16).

SCI0UMTED bit (SCI0 Address Mismatch Snooze End Enable)

The SCI0UMTED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on an SCI0 event when an address received in Software Standby mode does not match the expected data. For details on the trigger conditions, see section 27, 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
	—	—	—	—	—	—	—	—	SNZREQEN7	SNZREQEN6	SNZREQEN5	SNZREQEN4	SNZREQEN3	SNZREQEN2	SNZREQEN1	SNZREQEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SNZREQEN0	Snooze Request Enable 0	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	SNZREQEN5	Snooze Request Enable 5	Enable IRQ5 pin snooze request: 0: Disable the snooze request 1: Enable the snooze request.	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
b7	SNZREQEN7	Snooze Request Enable 7	Enable IRQ7 pin snooze request: 0: Disable the snooze request 1: Enable the snooze request.	R/W
b16 to b8	—	Reserved	These bits are 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

Bit	Symbol	Bit name	Description	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 13, 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 13, 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*1,*2	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*1,*2)

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

Note 1. When changing the value of the FLSTOP bit from 1 to 0 to start flash memory operation, 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 2. Writing to FLSTOP.FLSTOP is prohibited while HOCOCR.HCSTP and OSCSF.HOCOSF are 0 (HOCO is in stabilization wait counting).

FLSTPF flag (Flash Memory Operation Status Flag)

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

11.2.12 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	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)

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

The clock frequency changes when the following bits are set:

- SCKDIVCR.FCK[2:0]
- ICK[2:0]
- PCKB[2:0]
- PCKD[2:0].

The module and clock associations are as follows:

- The CPU, DTC, flash, and SRAM use the operating clock specified in the ICK[2:0] bits.
- Peripheral modules use the operating clock specified in the PCKB[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 DTC are placed in the module-stop state. Do not access the module while the associated MSTPmi bit is 1, otherwise the read/write data or the operation of the module is not guaranteed. Also, do not set the MSTPmi bit to 1 while the associated 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 conditions such as the voltage range and the frequency range are 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 shows the oscillators that can be used in each mode.

Table 11.5 Available oscillators in each mode

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

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

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

Operation begins in High-speed mode.

1. Disable the flash cache by resetting FCACHEE.FCACHEEN bit when the flash cache is cacheable in High-speed mode.
2. Change the oscillator to that used in Low-speed mode. Set the frequency of each clock lower than or equal to the maximum operating frequency in Low-speed mode.
3. Turn off the oscillator that is not required in Low-speed mode.
4. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
5. Set the OPCCR.OPCM bit to 11b (Low-speed mode).
6. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
7. Perform the following steps when the flash cache is cacheable in Low-speed mode:
   a. Invalidate the flash cache by setting FCACHEIV.FCACHEIV bit.
   b. Check that FCACHEIV.FCACHEIV bit is 0.
   c. Enable the flash cache by setting 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 bit when the flash cache is cacheable in High-speed mode.
2. Switch the clock source to sub-clock oscillator.
3. Turn off HOCO, MOCO, and main oscillator.
4. Confirm that all clock sources other than the sub-clock oscillator are stopped.
5. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).
6. Set the SOPCCR.SOPCM bit to 1 (Subosc-speed mode).

7. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).
8. 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 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 SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).
5. Turn on the required oscillator in High-speed mode.
6. Set the frequency of each clock to lower than or equal to 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.

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 bit 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 required oscillator in High-speed mode.
6. Set the frequency of each clock to lower than or equal to 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

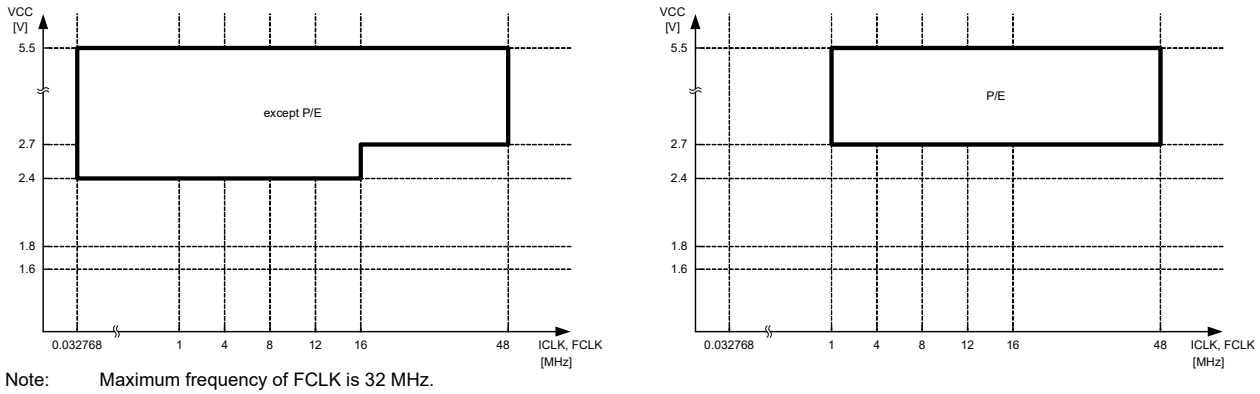
11.5.2 Operating Range

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 5.5 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 5.5 V.

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



Note: 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 5.5 V during flash read. However, for ICLK and FCLK, the maximum operating frequency during flash read is 8 MHz when the operating voltage is 1.8 V or larger and smaller than 2.4 V.

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

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

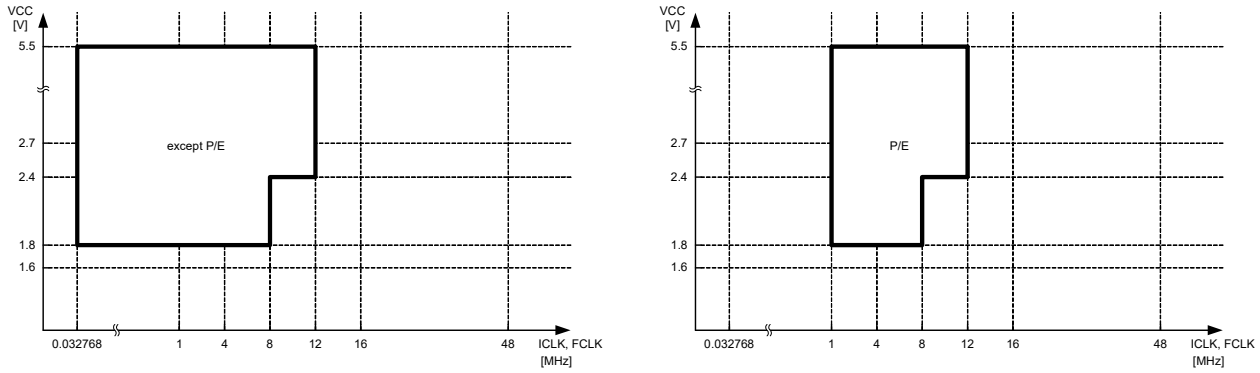


Figure 11.3 Operating voltages and frequencies in Middle-speed mode

Low-voltage mode

After a reset is canceled, operation is started from this mode.

The maximum operating frequency during flash read is 4 MHz for ICLK and FCLK. The operating voltage range is 1.6 to 5.5 V during flash read.

During flash programming and erasure, the operating frequency range is 1 to 4 MHz and the operating voltage range is 1.8 to 5.5 V.

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

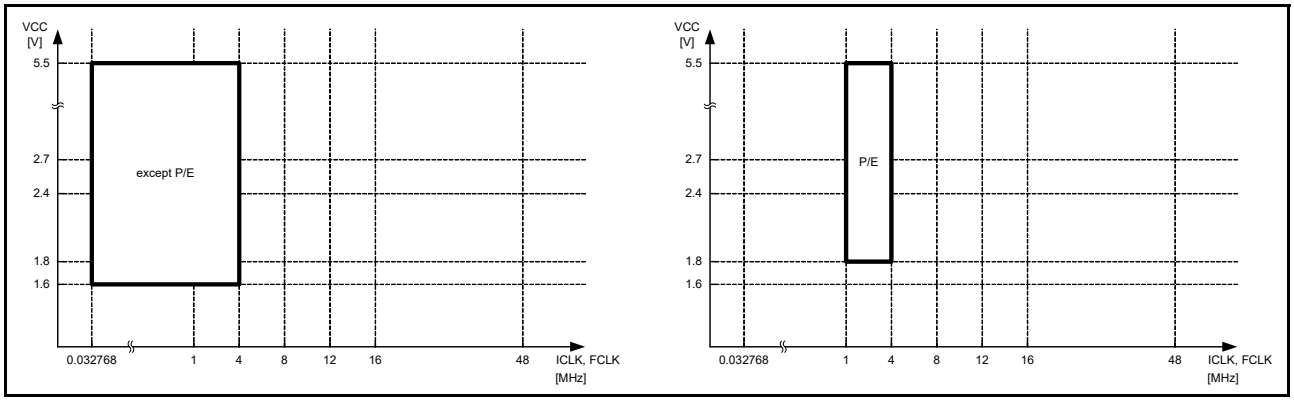


Figure 11.4 Operating voltages and frequencies in Low-voltage mode

Low-speed mode

The maximum operating frequency during flash read is 1 MHz for ICLK and FCLK. The operating voltage range is 1.8 to 5.5 V during flash read.

P/E operations for flash memory are prohibited.

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

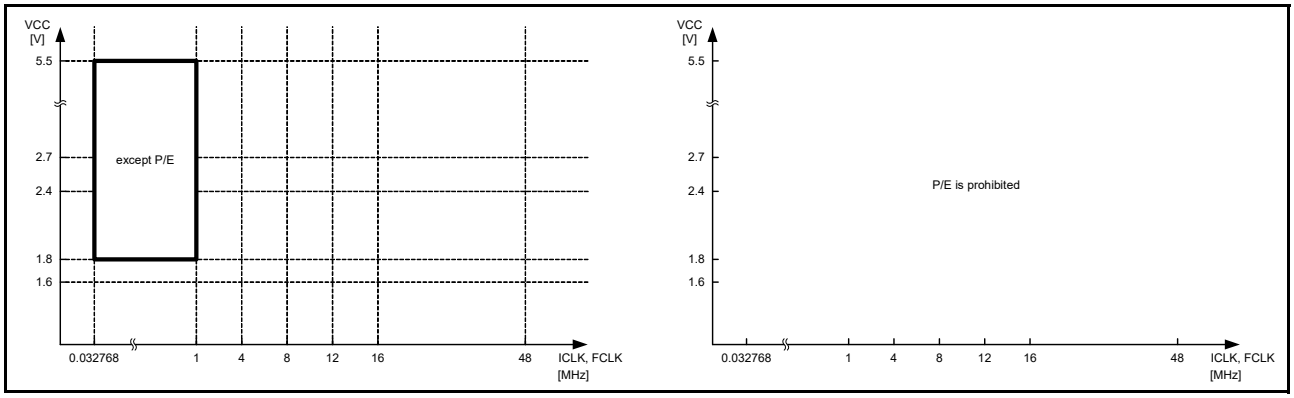


Figure 11.5 Operating voltages and frequencies in Low-speed mode

Subosc-speed mode

The maximum operating frequency during flash read is 37.6832 kHz for ICLK and FCLK. The operating voltage range is 1.8 to 5.5 V during flash read. P/E operations for flash memory are prohibited.

Using the oscillators other than the sub-clock oscillator or low-speed on-chip oscillator is prohibited.

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

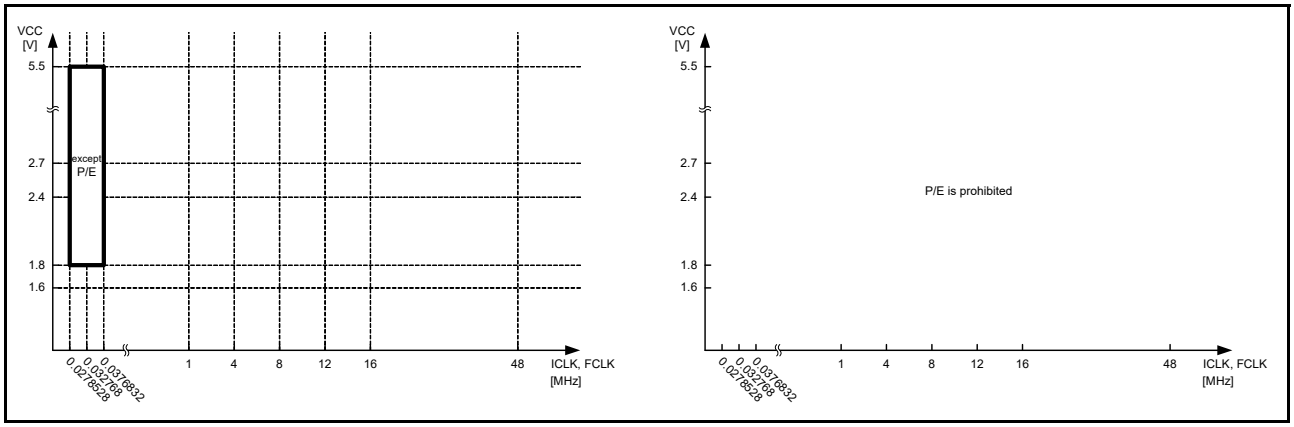


Figure 11.6 Operating voltages and frequencies in Subosc-speed mode

11.6 Sleep Mode

11.6.1 Transitioning to Sleep Mode

When a WFI instruction is executed while SBYCR.SSBY bit is 0, the MCU enters Sleep mode. In this mode, the CPU stops operating, but the contents of its internal registers are retained. Other peripheral functions do not stop. Available resets or interrupts in Sleep mode cause the MCU to cancel Sleep mode. All interrupt sources are available. If using an interrupt to cancel Sleep mode, you must set the associated IELSRn register before executing a WFI instruction. For details, see section 13, 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
- A RES pin reset
- A power-on reset
- A voltage monitor reset
- An SRAM parity error reset
- An SRAM ECC error reset
- A bus master MPU error reset
- A bus slave MPU error reset
- A reset caused by an IWDT or WDT underflow.

The operations are as follows:

1. Canceling by an interrupt
   When an interrupt request occurs, 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 47, 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 13, Interrupt Controller Unit (ICU).

11.7 Software Standby Mode

11.7.1 Transitioning 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 the 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 13.2.8 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 13, Interrupt Controller Unit (ICU).

Clear the DTCST.DTCST bit to 0 before executing WFI instruction, except when using DTC in Snooze mode. If DTC is required in Snooze mode, set the DTCST.DTCST bit to 1 before executing a WFI instruction.

Counting by IWDT stops when the MCU enters Software Standby mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 1 (IWDT stops in Sleep mode, Software Standby mode, 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
- A RES pin reset
- A power-on reset
- A voltage monitor reset
- A reset caused by an IWDT underflow.

You can cancel Software Standby mode in any of the following ways:

1. Canceling by an interrupt
   When an available interrupt request (for available interrupts, see Table 11.3) is generated, an oscillator that 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. See section 13.2.8 Wake Up Interrupt Enable Register (WUPEN) for information on waking up the MCU from Software Standby mode.
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 47, 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 an 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 00b (falling edge) in Normal mode, and the IRQCRi.IRQMD[1:0] bits set to 01b (rising edge). Next, 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 13, Interrupt Controller Unit (ICU). The oscillation stabilization time in Figure 11.7 is specified in section 47, Electrical Characteristics.

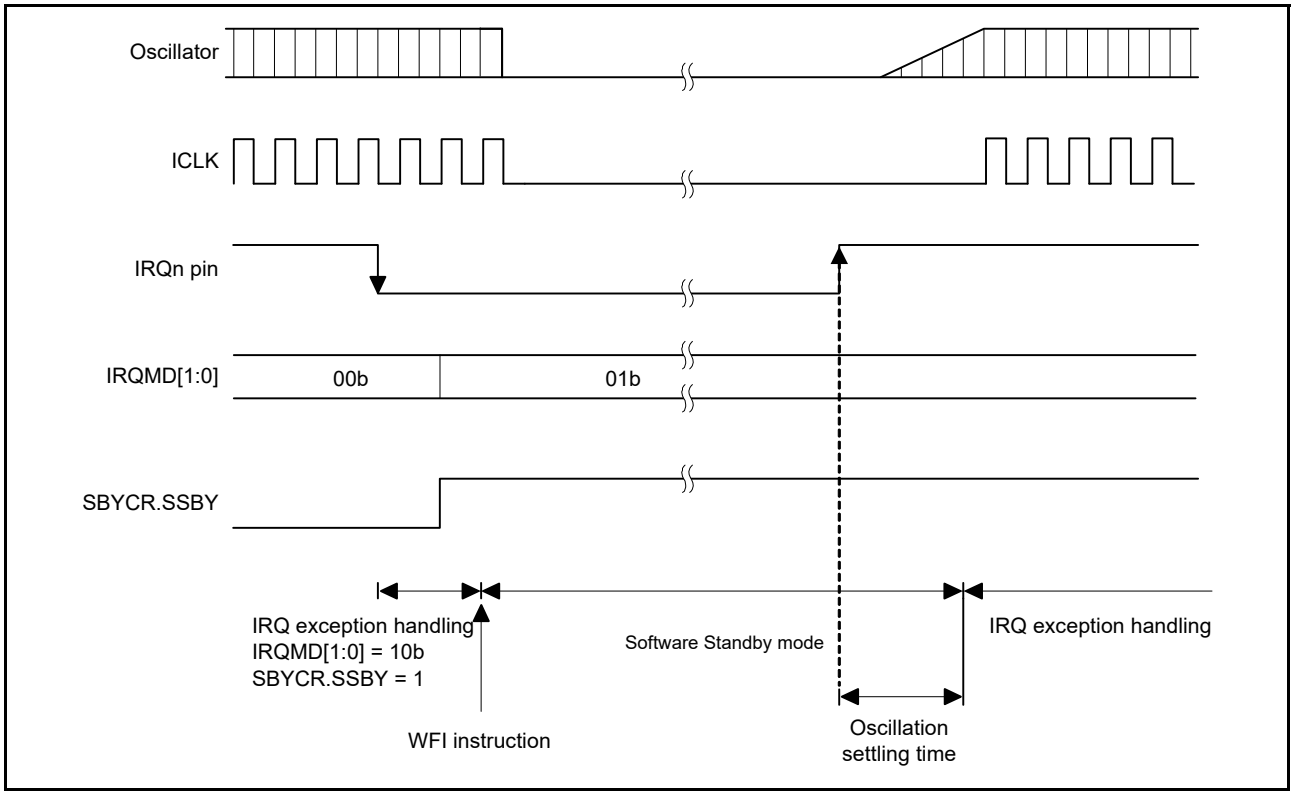


Figure 11.7 Example of Software Standby mode application

11.8 Snooze Mode

11.8.1 Transitioning 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. Table 11.2 shows the peripheral modules that can operate in Snooze mode. Also, DTC operation can be selected in Snooze mode by setting the SNZCR.SNZDTCEN bit.

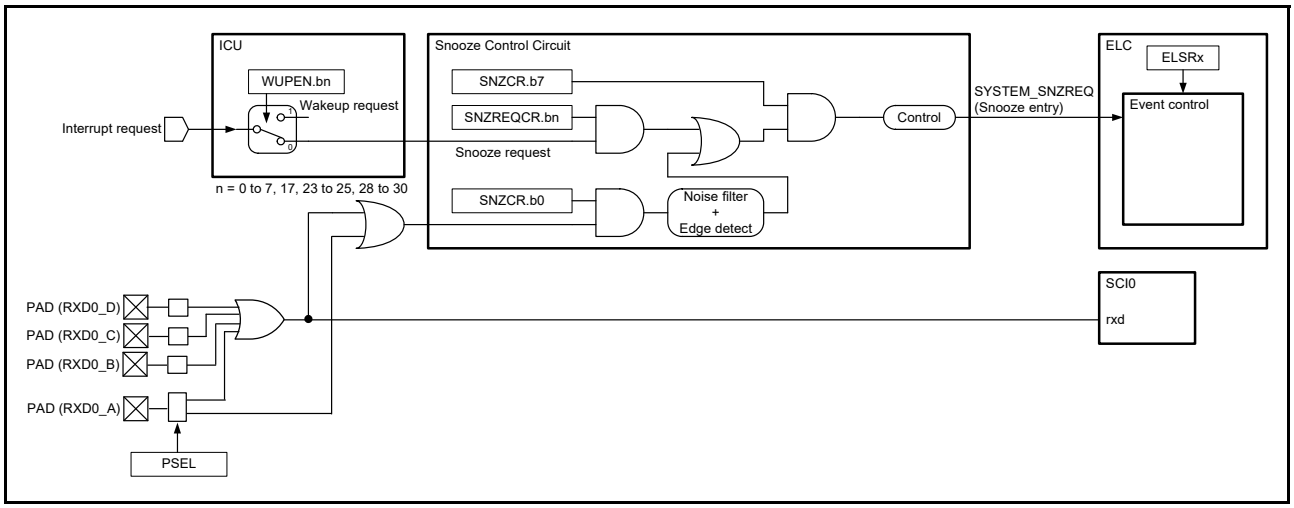


Figure 11.8 Snooze mode entry configuration

Table 11.6 shows the snooze requests to switch the MCU from Software Standby mode to Snooze mode. To use the listed snooze requests as a trigger to switch to Snooze mode, you must set the associated SNZREQENn bit of the SNZREQCR register or RXDREQEN bit of the SNZCR register before entering Software Standby mode.

Note: Do not enable multiple snooze requests at the same time.

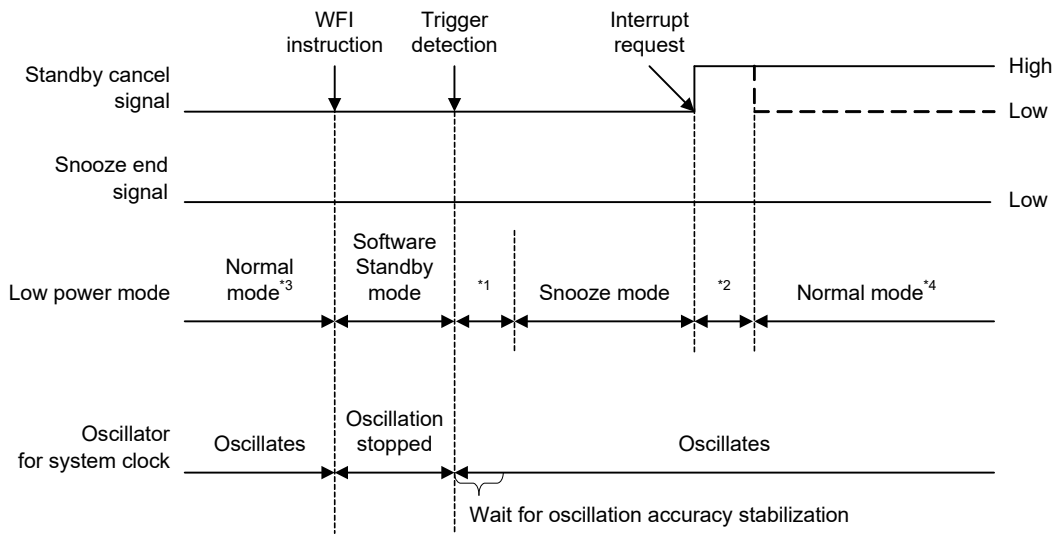
Table 11.6 Available snooze requests to switch to Snooze mode

Snooze request	Control Register	
	Register	**Bit**
PORT_IRQn (n = 0 to 7)	SNZREQCR	SNZREQENn (n = 0 to 7)
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. An action triggered by the interrupt requests selected in SELSR0 register, cancels Snooze mode. The interrupt that cancels Snooze mode must be selected in IELSRn register (n = 0 to 31) to link to the NVIC for the corresponding interrupt handling. See section 13, Interrupt Controller Unit (ICU) for the setting of SELSR0 and IELSRn registers.



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

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

11.8.3 Returning to Software Standby Mode

Table 11.7 shows the snooze end requests that can be used as triggers to return to Software Standby mode. The snooze end requests are available only in Snooze mode. If the requests are generated when the MCU is not in Snooze mode, they are ignored. When multiple requests are selected, each of the requests invokes transition to Software Standby mode from Snooze mode.

Table 11.8 shows the snooze end conditions that consist of the snooze end requests and the conditions of the peripheral modules. CTSU, SCI0, ADC160, and DTC can keep the MCU in Snooze mode until they complete the 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 depending on which snooze end requests are set in the SNZEDCR register. A snooze request is cleared automatically after returning to Software Standby mode.

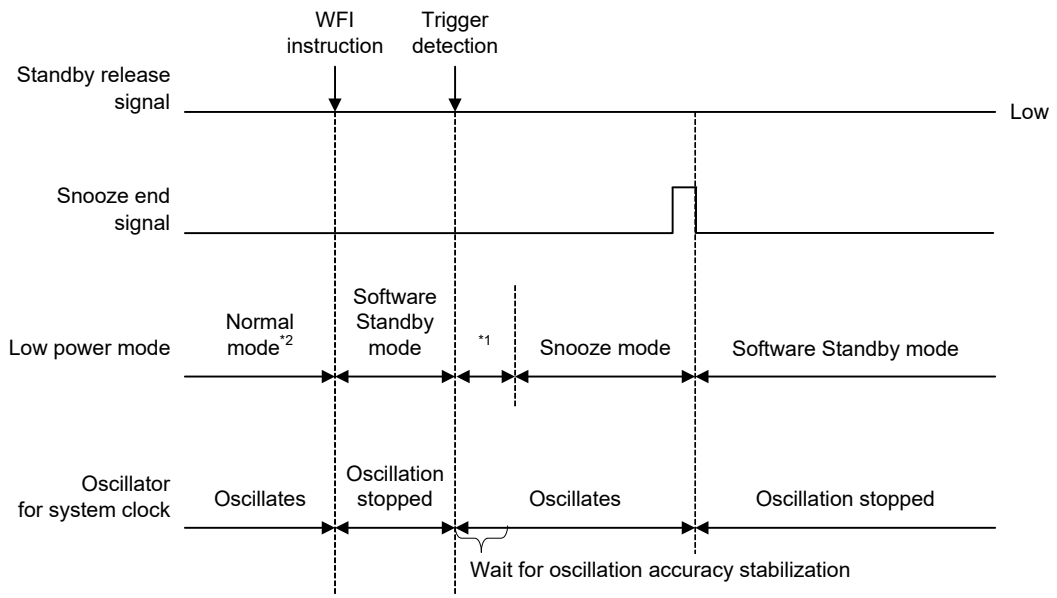
Table 11.7 Available snooze end requests (triggers to return to Software Standby mode)

Snooze end request	Enable/disable control	
	Register	Bit
AGT1 underflow or measurement complete (AGT1_AGTI)	SNZEDCR	b0
DTC transfer completion (DTC_COMPLETE)	SNZEDCR	b1
Not DTC transfer completion (DTC_TRANSFER)	SNZEDCR	b2
ADC160 window A/B compare match (ADC160_WCMPM)	SNZEDCR	b3
ADC160 window A/B compare mismatch (ADC160_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
ADC160		
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 ADC160, CTSU, or SCI0, the MCU transitions to software standby mode after a snooze end request is generated.



Note 1. Transition time from Software Standby mode to Snooze mode.
Note 2. Enable Snooze mode (SNZCR.SNZE = 1) immediately before transitioning to Software Standby mode.

Figure 11.10 Canceling of Snooze mode when an interrupt request signal is not generated

11.8.4 Snooze Operation Example

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

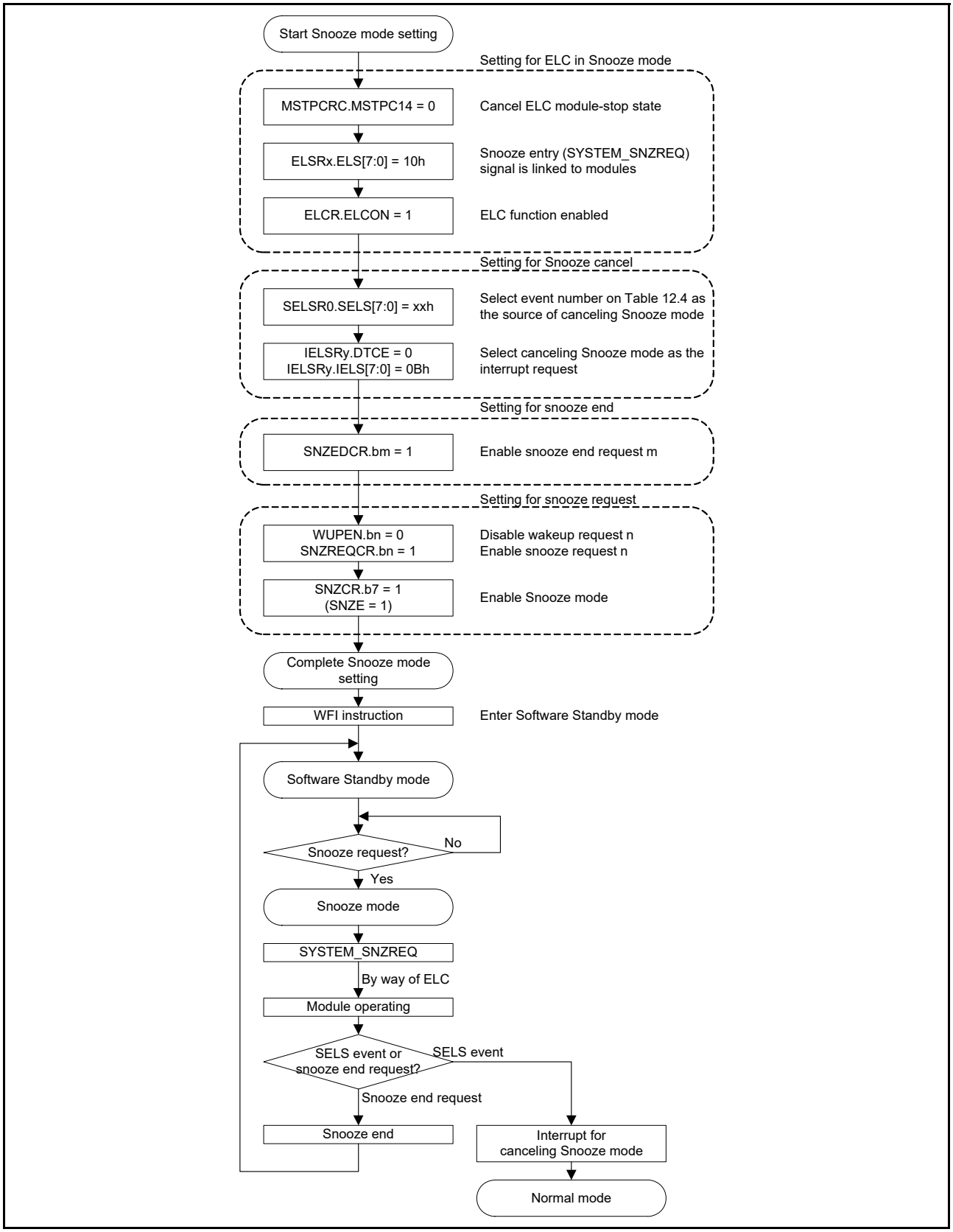


Figure 11.11 Setting example of using ELC in Snooze mode

The MCU can transmit and receive data in SCI0 asynchronous mode without CPU intervention. When using the SCI0 in Snooze mode, use one of the following operating modes:

- High-speed mode
- Middle-speed mode
- Low-speed mode.

Do not use 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 using the SCI0 in Snooze mode, set the following bits:

- BGDM = 0
- ABCS = 0
- ABCSE = 0.

See section 27, Serial Communications Interface (SCI) for information on these bits.

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

Table 11.9 HOCO: ± 1.0% (Ta = -20 to 85°C) **(Unit: bps)**

Maximum division ratio of ICLK, PCLKB, PCLKD, and FCLK	HOCO frequency			
	24 MHz	32 MHz	48 MHz	64 MHz
1	9600*1	9600*4	—	
2	9600*2	9600*5	4800	2400
4	9600*3	9600*6	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 = 3Eh, SCI0.MDDR = 9Dh must be used for 9600 bps.
Note 5. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 32h, SCI0.MDDR = FEh must be used for 9600 bps.
Note 6. 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, 85 to 105°C) **(Unit: bps)**

Maximum division ratio of ICLK, PCLKB, PCLKD, and FCLK	HOCO frequency			
	24 MHz	32 MHz	48 MHz	64 MHz
1	2400	2400	-	-
2	2400	2400	2400	1200
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 SCI0 in Snooze mode entry.

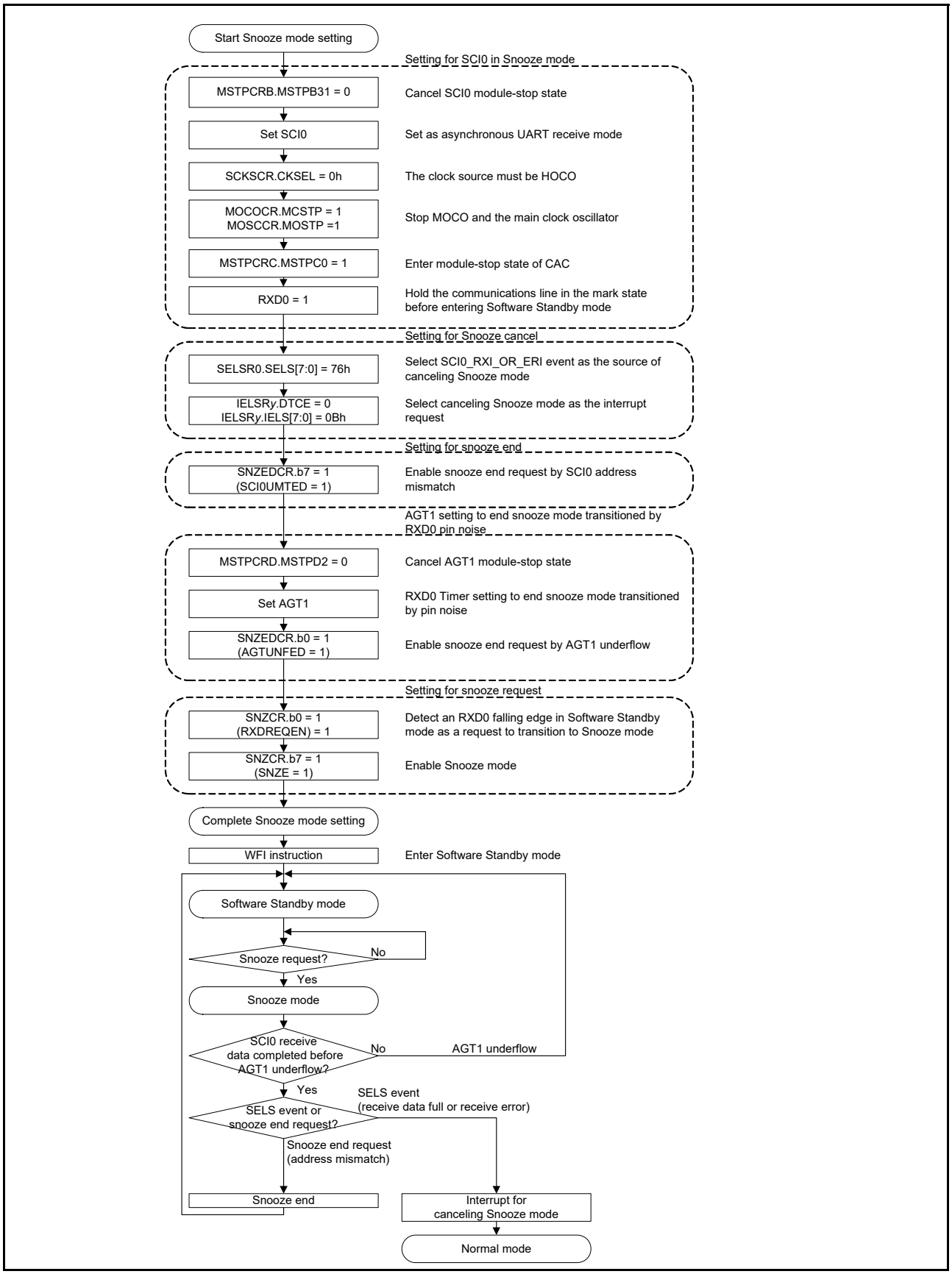


Figure 11.12 Setting example of using SCI0 in Snooze mode entry

11.9 Usage Notes

11.9.1 Register Access

(1) Invalid register write accesses during specific modes or transitions

Do not write to registers under any of the conditions listed in this section:

[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. Each register has certain prohibited settings under conditions other than those related to the operating power control modes. See section 9, Clock Generation Circuit for these other conditions of each register.

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

Mode	Valid setting						
	SCKSCR.CKSEL [2:0], CKOCR. CKOSEL[2:0]	SCKDIVCR.FCK [2:0], ICK[2:0]	HOCOCR. HCSTP	MOCOCR. MCSTP	LOCOCR. LCSTP	MOSCCR. MOSTP	SOSCCR. SOSTP
High-speed Middle-speed Low-voltage Low-speed	000b (HOCO) 001b (MOCO) 010b (LOCO) 011b (MOSC) 100b (SOSC)	000b (1/1) 001b (1/2) 010b (1/4) 011b (1/8) 100b (1/16) 101b (1/32) 110b (1/64)	0 (operating) 1 (stop)	0 (operating) 1 (stop)	0 (operating) 1 (stop)	0 (operating) 1 (stop)	0 (operating) 1 stop
Subosc-speed	010b (LOCO) 100b (SOSC)	000b (1/1)	1 (stop)	1 (stop)	0 (operating) 1 (stop)	1 (stop)	0 (operating) 1 (stop)

Table 11.12 Valid setting of clock-related registers (2)

Operating oscillator	Valid setting	
	SOPCCR SOPCM	OPCCR OPCM[1:0]
High-speed on-chip oscillator	0	00b, 01b, 10b, 11b
Middle-speed on-chip oscillator		
Main clock oscillator		
Low-speed on-chip oscillator	0, 1	00b, 01b, 10b, 11b
Sub-clock oscillator		
IWDT-dedicated on-chip oscillator		

(3) Invalid registers write accesses in subosc-speed mode

Do not write to registers under the condition listed in this section.

[Registers]

- SCKSCR, OPCCR.

[Condition]

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

(4) Invalid register write accesses by DTC

Do not write to registers listed in this section using the DTC.

[Registers]

- MSTPCRA.

(5) Invalid register write accesses in Snooze mode

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

[Registers]

- SNZCR, SNZEDCR, SNZREQCR.

(6) Invalid write access to set FLSTOP.FLSTOP bit to 1

Do not set the FLSTOP.FLSTOP bit to 1 under the conditions listed in this section.

[Conditions]

- SOPCCR.SOPCM = 0, OPCCR.OPCM[1:0] = 00b (High-speed mode)
- SOPCCR.SOPCM = 0, OPCCR.OPCM[1:0] = 01b (Middle-speed mode)
- SOPCCR.SOPCM = 0, OPCCR.OPCM[1:0] = 11b (Low-speed mode)
- SOPCCR.SOPCM = 1 (Subosc-speed mode).

(7) Invalid write access to set MEMWAIT.MEMWAIT bit to 1

Do not set the MEMWAIT.MEMWAIT bit to 1 under the conditions listed in this section.

[Conditions]

- SOPCCR.SOPCM = 0, OPCCR.OPCM[1:0] = 01 (Middle-speed mode)
- SOPCCR.SOPCM = 0, OPCCR.OPCM[1:0] = 10 (Low-voltage mode)
- SOPCCR.SOPCM = 0, OPCCR.OPCM[1:0] = 11 (Low-speed mode)
- SOPCCR.SOPCM = 1 (Subosc-speed mode).

(8) Invalid write access when PRCR.PRC1 bit is 0

Do not write to registers listed in this section when PRCR.PRC1 bit is 0.

[Registers]

- SBYCR, SNZCR, SNZEDCR, SNZREQCR, FLSTOP, 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 DTC

Before writing 1 to MSTPCRA.MSTPA22, clear the DTCST.DTCST bit of the DTC to 0. For details, see section 16, 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 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®-M23 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 writes are complete, in which case operation might not be as intended. This can happen if the WFI is placed immediately after a write to an I/O register. To avoid this problem, it is recommended that you read back the register that was written to confirm that the write has completed.

11.9.7 Writing WDT/IWDT Registers by DTC in Sleep Mode or Snooze Mode

Do not write registers in WDT or IWDT by 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 the transition from Software Standby mode to Snooze mode takes longer.

11.9.9 Snooze Mode Entry by RXD0 Falling Edge

When the SNZCR.RXDREQEN bit is 1, noise on the RXD0 pin might cause the MCU to transition from Software Standby mode to Snooze mode. Any subsequent RXD0 data can be received in Snooze mode by a noise on the RXD0 pin. If the MCU does not receive RXD0 data after the noise, interrupts such as SCI0_ERI, or SCI0_RXI, and address mismatch events are not generated, and the MCU stays in Snooze mode. To avoid this, an AGT1 underflow interrupt must be used to return to Software Standby mode or Normal mode when using SCI0 in Snooze mode. However, do not use the AGT1 underflow as a source to return to Software Standby mode during an SCI communication. This causes 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 and the main clock oscillator must stop before entering Software Standby mode
- The RXD0 pin must be kept high before entering Software Standby mode
- A transition to Software Standby mode must not occur during an 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. Do not use a software trigger or ADTRG0 pin.

11.9.12 Conditions of CTSU in Snooze Mode

The CTSU can only be started by the ELC in Snooze mode.

11.9.13 ELC Event in Snooze Mode

The ELC events available in Snooze mode are listed in this section. Do not use any other events. If starting peripheral modules for the first time after entering Snooze mode, the Event Link Setting Register (ELSRn) must set a Snooze mode entry event (SYSTEM_SNZREQ) as the trigger.

- Snooze mode entry (SYSTEM_SNZREQ)
- DTC transfer end (DTC_DTCEND)

- ADC160 window A/B compare match (ADC160_WCMPM)
- ADC160 window A/B compare mismatch (ADC160_WCMPUM)
- Data operation circuit interrupt (DOC_DOPCI).

11.9.14 Module-Stop Function for ADC160

When entering the Software Standby mode, it is recommended that you set the ADC160 module-stop state to reduce power consumption. In this case, the ADC160 can be available in Snooze mode by releasing the ADC160 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 can be increased to a value greater than that stated in this User's Manual by up to 600 µA. Initialize the unused circuit as shown in Figure 11.13.

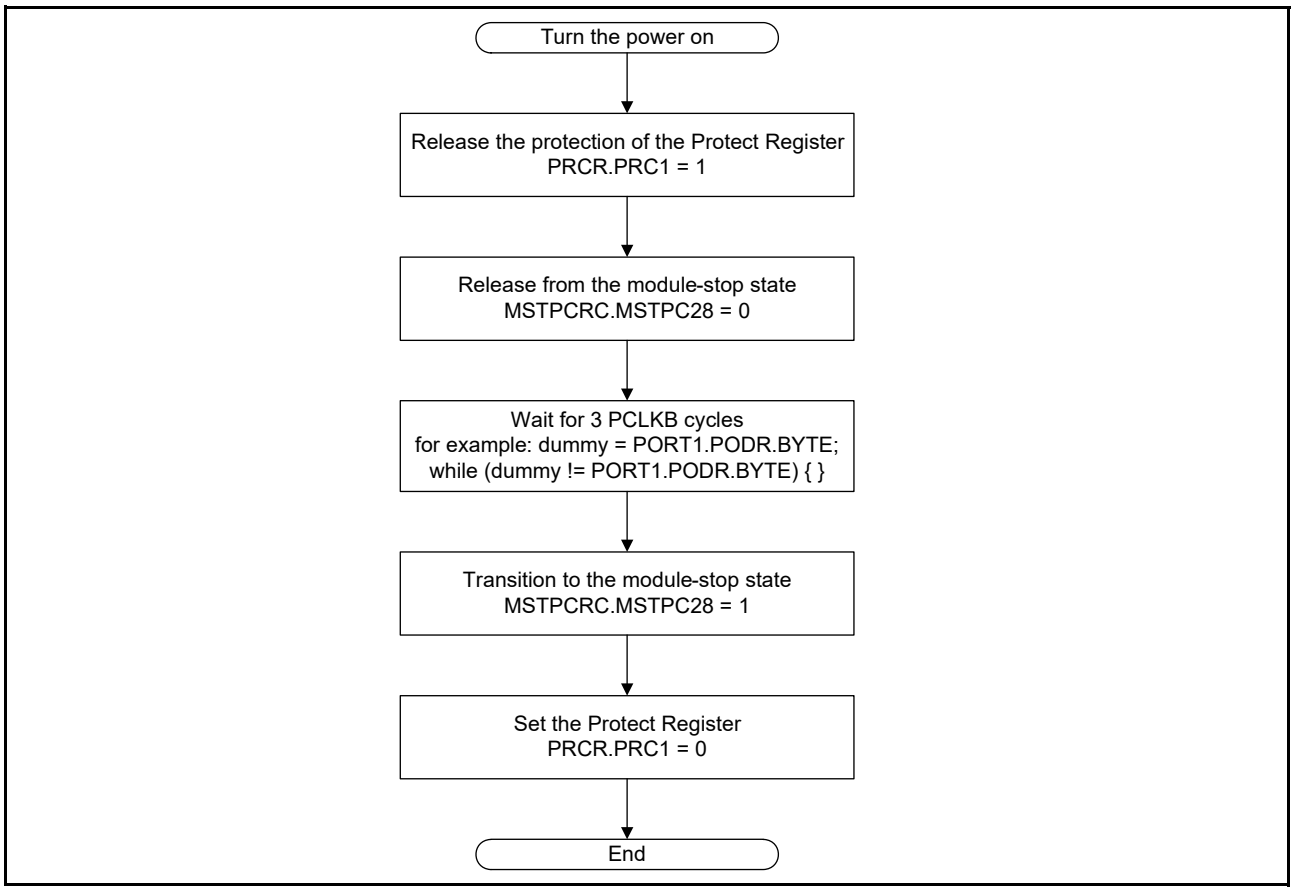


Figure 11.13 Example of initial setting flow for an unused circuit

12. Register Write Protection

12.1 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 12.1 lists the association between the PRCR bits and the registers to be protected.

Table 12.1 Association between PRCR bits and registers to be protected

PRCR bit	Register to be protected
PRC0	• Registers related to the clock generation circuit: SCKDIVCR, SCKSCR, MEMWAIT, MOSCCR, HOCOCR, MOCOCR, CKOCR, OSTDCR, OSTDSR, MOCOUTCR, HOCOUTCR, MOSCWTCR, MOMCR, SOSCCR, SOMCR, LOCOCR, LOCOUTCR, HOCOWTCR, SDADCCKCR
PRC1	• Registers related to the low power modes: SBYCR, SNZCR, SNZEDCR, SNZREQCR, FLSTOP, OPCCR, SOPCCR, SYOCDCR
PRC3	• Registers related to the LVD: LVD1CR1, LVD1SR, LVD2CR1, LVD2SR, LVCMPCR, LVDLVLR, LVD1CR0, LVD2CR0

12.2 Register Descriptions

12.2.1 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: Write disabled 1: Write enabled.	R/W
b1	PRC1	Protect Bit 1	Enables writing to the registers related to the low power modes: 0: Write disabled 1: Write enabled.	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: Write disabled 1: Write enabled.	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 the 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 12.1. Setting PRCn to 1 or 0 enables or disables writing, respectively.

13. Interrupt Controller Unit (ICU)

13.1 Overview

The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC) and Data Transfer Control (DTC) modules. The ICU also controls non-maskable interrupts.

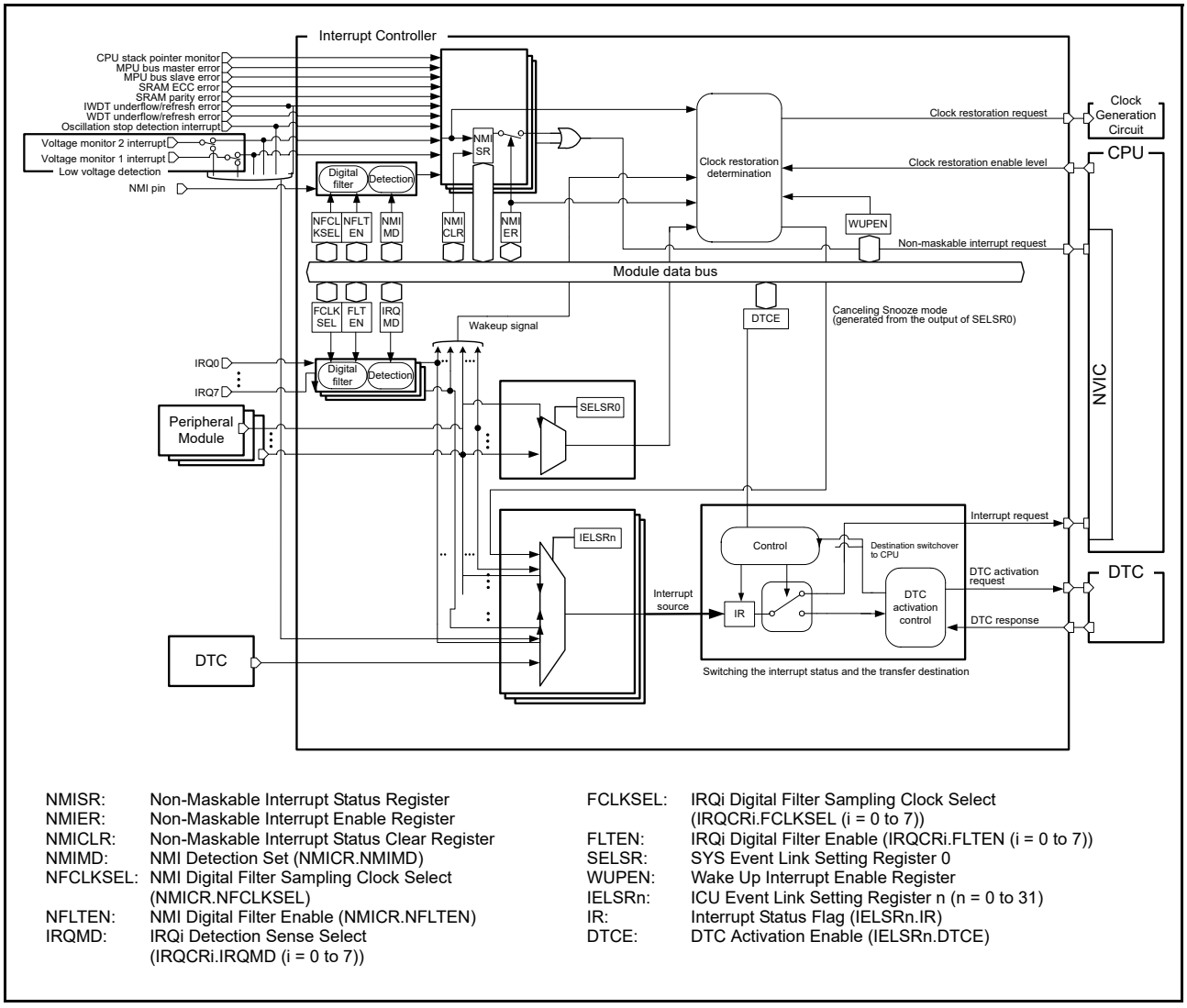
Table 13.1 lists the ICU specifications, Figure 13.1 shows a block diagram, and Table 13.2 lists the I/O pins.

Table 13.1 ICU specifications

Parameter		Specifications
Interrupts	Peripheral function interrupts	• Interrupts from peripheral modules Number of sources: 132 (select factor within event list numbers 9 to 141)
	External pin interrupts	• Interrupt detection on low level*4, falling edge, rising edge, rising and falling edges. One of these detection methods can be set for each source. • Digital filter function supported • 8 sources, with interrupts from IRQ0 to IRQ7 pins.
	DTC control	The DTC 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 1 (LVD_LVD1)
	Voltage monitor 2 interrupt*3	Voltage monitor interrupt of low voltage detection 2 (LVD_LVD2)
	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*5. • Snooze mode: return is initiated by non-maskable interrupts. Interrupt can be selected with the SELSR0 and WUPEN registers*5.

Note 1. For the DTC activation sources, see Table 13.4, Event table.
Note 2. Non-maskable interrupts can be enabled only once after a reset release.
Note 3. These non-maskable interrupts can also be used as event signals. When used as interrupts, do not change the value of the NMIER register from the reset state.To enable voltage monitor 1 and voltage monitor 2 interrupts, set the LVD1CR1.IRQSEL and LVD2CR1.IRQSEL bits to 1.
Note 4. Low level: interrupt detection is not canceled if you do not clear it after a detection.
Note 5. See section 13.2.7, SYS Event Link Setting Register (SELSR0) and section 13.2.8, Wake Up Interrupt Enable Register (WUPEN).

Figure 13.1 shows a block diagram of the ICU.



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 7))
FCLKSEL: IRQi Digital Filter Sampling Clock Select (IRQCRi.FCLKSEL (i = 0 to 7))
FLTEN: IRQi Digital Filter Enable (IRQCRi.FLTEN (i = 0 to 7))
SELSR: SYS Event Link Setting Register 0
WUPEN: Wake Up Interrupt Enable Register
IELSRn: ICU Event Link Setting Register n (n = 0 to 31)
IR: Interrupt Status Flag (IELSRn.IR)
DTCE: DTC Activation Enable (IELSRn.DTCE)

Figure 13.1 ICU block diagram

Table 13.2 lists the ICU input/output pins.

Table 13.2 ICU I/O pins

Pin name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ0 to IRQ7	Input	External interrupt request pins

13.2 Register Descriptions

This chapter does not describe the Arm® NVIC internal registers. For information about these registers, see the *ARM® Cortex®-M23 Processor Technical Reference Manual (ARM DDI 0550C).*

13.2.1 IRQ Control Register i (IRQCRi) (i = 0 to 7)

Address(es): ICU.IRQCR0 4000 6000h, ICU.IRQCR1 4000 6001h, ICU.IRQCR2 4000 6002h, ICU.IRQCR3 4000 6003h, ICU.IRQCR4 4000 6004h, ICU.IRQCR5 4000 6005h, ICU.IRQCR6 4000 6006h, ICU.IRQCR7 4000 6007h

	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 is disabled 1: Digital filter is enabled.	R/W

IRQCRi register changes must satisfy the following conditions:

- For a CPU interrupt or DTC trigger:
  Change the IRQCRi register 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 wakeup enable signal:
  Change the IRQCRi register setting before setting the target WUPEN.IRQWUPEN[n] (n = 0 to 7).
  You can only change the register values when the target WUPEN.IRQWUPEN[n] is 0.

IRQMD[1:0] bits (IRQi Detection Sense Select)

The IRQMD[1:0] bits set the detection sensing method for the IRQi external pin interrupt sources. For setting method when using external pin interrupt, see section 13.4.4, 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 details of the digital filter, see section 13.4.3, Digital Filter.

FLTEN bit (IRQi Digital Filter Enable)

The FLTEN bit enables the digital filter used for the IRQi external pin interrupt sources. The digital filter is enabled when the IRQCRi.FLTEN bit is 1 and disabled when the IRQCRi.FLTEN bit is 0. The IRQi pin level is sampled at the clock cycle specified in the IRQCRi.FCLKSEL[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes. For details of the digital filter, see section 13.4.3, Digital Filter.

13.2.2 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	—	—	LVD2ST	LVD1ST	WDTST	IWDTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	IWDTST	IWDT Underflow/Refresh Error Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b1	WDTST	WDT Underflow/Refresh Error Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b2	LVD1ST	Voltage Monitor 1 Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b3	LVD2ST	Voltage Monitor 2 Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b5, b4	—	Reserved	These bits are read as 0.	R
b6	OSTST	Main Clock Oscillation Stop Detection Interrupt Status Flag	0: Interrupt not requested for main clock oscillation stop 1: Interrupt requested for main clock oscillation stop.	R
b7	NMIST	NMI Pin Interrupt Status Flag	0: Interrupt not requested 1: 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 are generated during handler processing.

IWDTST flag (IWDT Underflow/Refresh Error Status Flag)

The IWDTST flag indicates an IWDT underflow/refresh error interrupt request. It is read-only and cleared by the NMICLR.IWDTCLR bit.

[Setting condition]

- When the IWDT underflow/refresh error interrupt is generated and this interrupt source is enabled.

[Clearing condition]

- When 1 is written to the NMICLR.IWDTCLR bit.

WDTST flag (WDT Underflow/Refresh Error Status Flag)

The WDTST flag indicates a WDT underflow/refresh error interrupt request. It is read-only and cleared by the NMICLR.WDTCLR bit.

[Setting condition]

- When the WDT underflow/refresh error interrupt is generated.

[Clearing condition]

- When 1 is written to the NMICLR.WDTCLR bit.

LVD1ST flag (Voltage Monitor 1 Interrupt Status Flag)

The LVD1ST flag indicates a request for voltage monitor 1 interrupt. It is read-only and cleared by the NMICLR.LVD1CLR bit.

[Setting condition]

- When the voltage monitor 1 interrupt is generated and this interrupt source is enabled.

[Clearing condition]

- When 1 is written to the NMICLR.LVD1CLR bit.

LVD2ST flag (Voltage Monitor 2 Interrupt Status Flag)

The LVD2ST flag indicates a request for voltage monitor 2 interrupt. It is read-only and cleared by the NMICLR.LVD2CLR bit.

[Setting condition]

- When the voltage monitor 2 interrupt is generated and this interrupt source is enabled.

[Clearing condition]

- When 1 is written to the NMICLR.LVD2CLR bit.

OSTST flag (Main Clock Oscillation Stop Detection Interrupt Status Flag)

The OSTST flag indicates a main clock oscillation stop detection interrupt request. It is read-only and cleared by the NMICLR.OSTCLR bit.

[Setting condition]

- When the main clock oscillation stop detection interrupt is generated.

[Clearing condition]

- When 1 is written to the NMICLR.OSTCLR bit.

NMIST flag (NMI Pin Interrupt Status Flag)

The NMIST flag indicates an NMI pin interrupt request. It is read-only and cleared by the NMICLR.NMICLR bit.

[Setting condition]

- When an edge specified by the NMICR.NMIMD bit is input to the NMI pin.

[Clearing condition]

- When 1 is written to the NMICLR.NMICLR bit.

RPEST flag (SRAM Parity Error Interrupt Status Flag)

The RPEST flag indicates an SRAM parity error interrupt request.

[Setting condition]

- When an interrupt is generated in response to an SRAM parity error.

[Clearing condition]

- When 1 is written to the NMICLR.RPECLR bit.

RECCST flag (SRAM ECC Error Interrupt Status Flag)

The RECCST flag indicates an SRAM ECC error interrupt request.

[Setting condition]

- When an interrupt is generated in response to an SRAM ECC error.

[Clearing condition]

- When 1 is written to the NMICLR.RECCCLR bit.

BUSSST flag (MPU Bus Slave Error Interrupt Status Flag)

The BUSSST flag indicates a bus slave error interrupt request.

[Setting condition]

- When an interrupt is generated in response to a bus slave error.

[Clearing condition]

- When 1 is written to the NMICLR.BUSSCLR bit.

BUSMST flag (MPU Bus Master Error Interrupt Status Flag)

The BUSMST flag indicates a bus master error interrupt request.

[Setting condition]

- When an interrupt is generated in response to a bus master error.

[Clearing condition]

- When 1 is written to the NMICLR.BUSMCLR bit.

SPEST flag (CPU Stack Pointer Monitor Interrupt Status Flag)

The SPEST flag indicates a CPU stack pointer monitor interrupt request.

[Setting condition]

- When an interrupt is generated in response to a CPU stack pointer monitor error.

[Clearing condition]

- When 1 is written to the NMICLR.SPECLR bit.

13.2.3 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	—	—	LVD2EN	LVD1EN	WDTEN	IWDTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	IWDTEN	IWDT Underflow/Refresh Error Interrupt Enable	0: Disabled 1: Enabled.	R/(W) *1, *2
b1	WDTEN	WDT Underflow/Refresh Error Interrupt Enable	0: Disabled 1: Enabled.	R/(W) *1, *2
b2	LVD1EN	Voltage monitor 1 Interrupt Enable	0: Disabled 1: Enabled.	R/(W) *1, *2
b3	LVD2EN	Voltage Monitor 2 Interrupt Enable	0: Disabled 1: Enabled.	R/(W) *1, *2
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	OSTEN	Main Clock Oscillation Stop Detection Interrupt Enable	0: Disabled 1: Enabled.	R/(W) *1, *2

Bit	Symbol	Bit name	Description	R/W
b7	NMIEN	NMI Pin Interrupt Enable	0: Disabled 1: Enabled.	R/(W) *1
b8	RPEEN	SRAM Parity Error Interrupt Enable	0: Disabled 1: Enabled.	R/(W) *1
b9	RECCEN	SRAM ECC Error Interrupt Enable	0: Disabled 1: Enabled.	R/(W) *1
b10	BUSSEN	MPU Bus Slave Error Interrupt Enable	0: Disabled 1: Enabled.	R/(W) *1
b11	BUSMEN	MPU Bus Master Error Interrupt Enable	0: Disabled 1: Enabled.	R/(W) *1
b12	SPEEN	CPU Stack Pointer Monitor Interrupt Enable	0: Disabled 1: Enabled.	R/(W) *1
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.

LVD2EN bit (Voltage Monitor 2 Interrupt Enable)

The LVD2EN bit enables voltage monitor 2 interrupt as an NMI trigger.

OSTEN bit (Main Clock Oscillation Stop Detection Interrupt Enable)

The OSTEN bit enables main clock oscillation stop detection interrupt as an NMI trigger.

NMIEN bit (NMI Pin Interrupt Enable)

The NMIEN bit enables NMI pin interrupt as an NMI trigger.

RPEEN bit (SRAM Parity Error Interrupt Enable)

The RPEEN bit enables SRAM parity error interrupt as an NMI trigger.

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.

13.2.4 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	—	—	LVD2CLR	LVD1CLR	WDTCLR	IWDTCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	IWDTCLR	IWDT Clear	0: No effect 1: Clear the NMISR.IWDTST flag.	R/(W)*1
b1	WDTCLR	WDT Clear	0: No effect 1: Clear the NMISR.WDTST flag.	R/(W)*1
b2	LVD1CLR	LVD1 Clear	0: No effect 1: Clear the NMISR.LVD1ST flag.	R/(W)*1
b3	LVD2CLR	LVD2 Clear	0: No effect 1: Clear the NMISR.LVD2ST flag.	R/(W)*1
b5, b4	—	Reserved	These bits are 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. This bit is read as 0.

WDTCLR bit (WDT Clear)

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

LVD1CLR bit (LVD1 Clear)

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

LVD2CLR bit (LVD2 Clear)

Writing 1 to the LVD2CLR bit clears the NMISR.LVD2ST flag. This bit is read as 0.

OSTCLR bit (OST Clear)

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

NMICLR bit (NMI Clear)

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

RPECLR bit (SRAM Parity Error Clear)

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

RECCCLR bit (SRAM ECC Error Clear)

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

BUSSCLR bit (Bus Slave Error Clear)

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

BUSMCLR bit (Bus Master Error Clear)

Writing 1 to the BUSMCLR bit clears the NMISR.BUSMST flag. This bit is read as 0.

SPECLR bit (CPU Stack Pointer Monitor Interrupt Clear)

Writing 1 to the SPECLR bit clears the NMISR.SPEST flag. This bit is read as 0.

13.2.5 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: Disabled 1: Enabled.	R/W

Change the NMICR register settings before enabling NMI pin interrupts, that is, before setting NMIER.NMIEN to 1.

NMIMD bit (NMI Detection Set)

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

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

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

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

For details of the digital filter, see section 13.4.3, 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 NMICR.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 13.4.3, Digital Filter.

13.2.6 ICU Event Link Setting Register n (IELSRn) (n = 0 to 31)

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 or DTC module 00000001 to 10001101:Event signal number to be linked. For details, see Table 13.4. Other settings are prohibited.	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 generated 1: An interrupt request generated.	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: Disabled 1: 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 1. Writing 1 to the IR flag is prohibited.

The IELSRn register selects the IRQ source used by the NVIC. For details, see Table 13.4. 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 or 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 IR flag. DTCE must be set to 0 before writing 0 to the IR flag.

To clear the IR flag:

1. Negate the input interrupt signal.
2. Read access the peripheral once and wait for 2 clock cycles of the target module clock (PCLKB or PCLKD).

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 DTCE bit.

13.2.7 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 10001101:Event signal number to be linked. For details, see Table 13.4, Event table. Other settings are prohibited.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register requires halfword access.

The SELSR0 register selects the events that wake up the CPU from Snooze mode. You can use only the events listed in Table 13.4 checked as “Canceling Snooze mode”. When 0Bh is set in IELSRn.IELS[7:0], an Interrupt is generated that cancels snooze mode.

13.2.8 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	—	—	—	LVD2WUPEN	LVD1WUPEN	KEYWUPEN	IWDTWUPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	IRQWUPEN[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	IRQWUPEN[7:0]	IRQ Interrupt Software Standby Returns Enable	0: Software Standby returns by IRQn interrupt disabled 1: Software Standby returns by IRQn interrupt enabled.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	IWDTWUPEN	IWDT Interrupt Software Standby Returns Enable	0: Software Standby returns by IWDT interrupt disabled 1: Software Standby returns by IWDT interrupt enabled.	R/W
b17	KEYWUPEN	Key Interrupt Software Standby Returns Enable	0: Software Standby returns by KEY interrupt disabled 1: Software Standby returns by KEY interrupt enabled.	R/W
b18	LVD1WUPEN	LVD1 Interrupt Software Standby Returns Enable	0: Software Standby returns by LVD1 interrupt disabled 1: Software Standby returns by LVD1 interrupt enabled.	R/W
b19	LVD2WUPEN	LVD2 Interrupt Software Standby Returns Enable	0: Software Standby returns by LVD2 interrupt disabled 1: Software Standby returns by LVD2 interrupt enabled.	R/W
b22 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23	ACMPLP0WUPEN	ACMPLP0 Interrupt Software Standby Returns Enable	0: Software Standby returns by ACMPLP0 interrupt disabled 1: Software Standby returns by ACMPLP0 interrupt enabled.	R/W
b24	RTCALMWUPEN	RTC Alarm Interrupt Software Standby Returns Enable	0: Software Standby returns by RTC alarm interrupt disabled 1: Software Standby returns by RTC alarm interrupt enabled.	R/W
b25	RTCPRDWUPEN	RTC Period Interrupt Software Standby Returns Enable	0: Software Standby returns by RTC period interrupt disabled 1: Software Standby returns by RTC period interrupt enabled.	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: Software Standby returns by USBFS interrupt disabled 1: Software Standby returns by USBFS interrupt enabled.	R/W
b28	AGT1UDWUPEN	AGT1 Underflow Interrupt Software Standby Returns Enable	0: Software Standby returns by AGT1 underflow interrupt disabled 1: Software Standby returns by AGT1 underflow interrupt enabled.	R/W
b29	AGT1CAWUPEN	AGT1 Compare Match A Interrupt Software Standby Returns Enable	0: Software Standby returns by AGT1 compare match A interrupt disabled 1: Software Standby returns by AGT1 compare match A interrupt enabled.	R/W
b30	AGT1CBWUPEN	AGT1 Compare Match B Interrupt Software Standby Returns Enable	0: Software Standby returns by AGT1 compare match B interrupt disabled 1: Software Standby returns by AGT1 compare match B interrupt enabled.	R/W
b31	IIC0WUPEN	IIC0 Address Match Interrupt Software Standby Returns Enable	0: Software Standby returns by IIC0 address match interrupt disabled 1: Software Standby returns by IIC0 address match interrupt enabled.	R/W

The bits in this register control whether the associated interrupt can wake up the CPU from Software Standby mode.

IRQWUPEN[7:0] bits (IRQ Interrupt Software Standby Returns Enable)

The IRQWUPEN[7:0] bits enable the use of IRQn interrupts to cancel Software Standby mode.

IWDTWUPEN bit (IWDT Interrupt Software Standby Returns Enable)

The IWDTWUPEN bit enables the use of IWDT interrupts to cancel Software Standby mode.

KEYWUPEN bit (Key Interrupt Software Standby Returns Enable)

The KEYWUPEN bit enables the use of Key interrupts to cancel Software Standby mode.

LVD1WUPEN bit (LVD1 Interrupt Software Standby Returns Enable)

The LVD1WUPEN bit enables the use of LVD1 interrupts to cancel Software Standby mode.

LVD2WUPEN bit (LVD2 Interrupt Software Standby Returns Enable)

The LVD2WUPEN bit enables the use of LVD2 interrupts to cancel Software Standby mode.

ACMPLP0WUPEN bit (ACMPLP0 Interrupt Software Standby Returns Enable)

The ACMPLP0WUPEN bit enables the use of ACMPLP0 interrupts 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 interrupts to cancel Software Standby mode.

USBFSWUPEN bit (USBFS Interrupt Software Standby Returns Enable)

The USBFSWUPEN bit enables the use of USBFS interrupts to cancel Software Standby mode.

AGT1UDWUPEN bit (AGT1 Underflow Interrupt Software Standby Returns Enable)

The AGT1UDWUPEN bit enables the use of the AGT1 underflow interrupts 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 interrupts 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 interrupts to cancel Software Standby mode.

IIC0WUPEN bit (IIC0 Address Match Interrupt Software Standby Returns Enable)

The IIC0WUPEN bit enables the use of IIC0 interrupts to cancel Software Standby mode.

13.3 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®-M23 Processor Technical Reference Manual (ARM DDI 0550C).*

13.3.1 Interrupt Vector Table

Table 13.3 describes the interrupt vectors. The addresses conform to the NVIC specifications.

Table 13.3 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	Reserved
5	-	014h	Arm	Reserved
6	-	018h	Arm	Reserved
7	-	01Ch	Arm	Reserved
8	-	020h	Arm	Reserved
9	-	024h	Arm	Reserved

Table 13.3 Interrupt vector table (2 of 2)

Exception number	IRQ number	Vector offset	Source	Description
10	-	028h	Arm	Reserved
11	-	02Ch	Arm	Supervisor Call (SVCall)
12	-	030h	Arm	Reserved
13	-	034h	Arm	Reserved
14	-	038h	Arm	Pendable request for system service (PendableSrvReq)
15	-	03Ch	Arm	System Tick timer (SysTick)
16	0	040h	ICU.IELSR0	Event selected in the ICU.IELSR0 register
17	1	044h	ICU.IELSR1	Event selected in the ICU.IELSR1 register
18	2	048h	ICU.IELSR2	Event selected in the ICU.IELSR2 register
19	3	04Ch	ICU.IELSR3	Event selected in the ICU.IELSR3 register
20	4	050h	ICU.IELSR4	Event selected in the ICU.IELSR4 register
21	5	054h	ICU.IELSR5	Event selected in the ICU.IELSR5 register
22	6	058h	ICU.IELSR6	Event selected in the ICU.IELSR6 register
23	7	05Ch	ICU.IELSR7	Event selected in the ICU.IELSR7 register
24	8	060h	ICU.IELSR8	Event selected in the ICU.IELSR8 register
25	9	064h	ICU.IELSR9	Event selected in the ICU.IELSR9 register
26	10	068h	ICU.IELSR10	Event selected in the ICU.IELSR10 register
27	11	06Ch	ICU.IELSR11	Event selected in the ICU.IELSR11 register
28	12	070h	ICU.IELSR12	Event selected in the ICU.IELSR12 register
29	13	074h	ICU.IELSR13	Event selected in the ICU.IELSR13 register
30	14	078h	ICU.IELSR14	Event selected in the ICU.IELSR14 register
31	15	07Ch	ICU.IELSR15	Event selected in the ICU.IELSR15 register
32	16	080h	ICU.IELSR16	Event selected in the ICU.IELSR16 register
33	17	084h	ICU.IELSR17	Event selected in the ICU.IELSR17 register
34	18	088h	ICU.IELSR18	Event selected in the ICU.IELSR18 register
35	19	08Ch	ICU.IELSR19	Event selected in the ICU.IELSR19 register
36	20	090h	ICU.IELSR20	Event selected in the ICU.IELSR20 register
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

13.3.2 Event Number

The following table lists heading details for Table 13.4, which describes each event number.

Heading	Description
Interrupt request source	Name of the source generating the interrupt request
Name	Name of the interrupt
Connect to NVIC	"✓" indicates the interrupt can be used as a CPU interrupt
Invoke DTC	"✓" indicates the interrupt can be used to request DTC activation
Canceling Snooze mode	"✓" indicates the interrupt can be used to request a return from Snooze mode
Canceling Software Standby mode	"✓" indicates the interrupt can be used to request a return from Software Standby mode

Table 13.4 Event table (1 of 4)

Event number	Interrupt request source	Name	IELSRn		Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC		
01h	Port	PORT_IRQ0	✓	✓	✓	✓
02h		PORT_IRQ1	✓	✓	✓	✓
03h		PORT_IRQ2	✓	✓	✓	✓
04h		PORT_IRQ3	✓	✓	✓	✓
05h		PORT_IRQ4	✓	✓	✓	✓
06h		PORT_IRQ5	✓	✓	✓	✓
07h		PORT_IRQ6	✓	✓	✓	✓
08h		PORT_IRQ7	✓	✓	✓	✓
09h	DTC	DTC_COMPLETE	✓	-	✓*4	-
0Bh	ICU	ICU_SNZCANCEL	✓	-	✓	-
0Ch	FCU	FCU_FRDYI	✓	-	-	-
0Dh	LVD	LVD_LVD1	✓	-	✓	✓
0Eh		LVD_LVD2	✓	-	✓	✓
0Fh	MOSC	MOSC_STOP	✓	-	-	-
10h	Low power mode	SYSTEM_SNZREQ	-	✓	-	-
11h	AGT0	AGT0_AGTI	✓	✓	-	-
12h		AGT0_AGTCMAI	✓	✓	-	-
13h		AGT0_AGTCMBI	✓	✓	-	-
14h	AGT1	AGT1_AGTI	✓	✓	✓	✓
15h		AGT1_AGTCMAI	✓	✓	✓	✓
16h		AGT1_AGTCMBI	✓	✓	✓	✓
17h	IWDT	IWDT_NMIUNDF	✓	-	✓	✓
18h	WDT	WDT_NMIUNDF	✓	-		
19h	RTC	RTC_ALM	✓	-	✓	✓
1Ah		RTC_PRD	✓	-	✓	✓
1Bh		RTC_CUP	✓	-	-	-

Table 13.4 Event table (2 of 4)

Event number	Interrupt request source	Name	IELSRn		Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC		
1Ch	ADC16	ADC160_ADI	✓	✓	-	-
1Dh		ADC160_GBADI	✓	✓	-	-
1Eh		ADC160_CMPAI	✓	-	-	-
1Fh		ADC160_CMPBI	✓	-	-	-
20h		ADC160_WCMPM	-	✓	✓*4	-
21h		ADC160_WCMPUM	-	✓	✓*4	-
22h	ACMPHS	ACMP_HS0	✓	-	-	-
23h	ACMPLP	ACMP_LP0	✓	-	✓	✓
24h		ACMP_LP1	✓	-	-	-
25h	USBFS	USBFS_USBI	✓	-	-	-
26h		USBFS_USBR	✓	-	✓	✓
27h	IIC0	IIC0_RXI	✓	✓	-	-
28h		IIC0_TXI	✓	✓	-	-
29h		IIC0_TEI	✓	-	-	-
2Ah		IIC0_EEI	✓	-	-	-
2Bh		IIC0_WUI	✓	-	✓	✓
2Ch	IIC1	IIC1_RXI	✓	✓	-	-
2Dh		IIC1_TXI	✓	✓	-	-
2Eh		IIC1_TEI	✓	-	-	-
2Fh		IIC1_EEI	✓	-	-	-
30h	CTSU	CTSU_CTSUWR	✓	✓	-	-
31h		CTSU_CTSURD	✓	✓	-	-
32h		CTSU_CTSUFN	✓	-	✓*4	
33h	KINT	KEY_INTKR	✓	-	✓*1	✓*1
34h	DOC	DOC_DOPCI	✓	-	✓*4	-
35h	CAC	CAC_FERRI	✓	-	-	-
36h		CAC_MENDI	✓	-	-	-
37h		CAC_OVFI	✓	-	-	-
38h	CAN0	CAN0_ERS	✓	-	-	-
39h		CAN0_RXF	✓	-	-	-
3Ah		CAN0_TXF	✓	-	-	-
3Bh		CAN0_RXM	✓	-	-	-
3Ch		CAN0_TXM	✓	-	-	-
3Dh	I/O Ports	IOPORT_GROUP1	✓	✓*2	-	-
3Eh		IOPORT_GROUP2	✓	✓*2	-	-
3Fh	ELC	ELC_SWEVT0	✓*3	✓	-	-
40h		ELC_SWEVT1	✓*3	✓	-	-
41h	POEG	POEG_GROUP0	✓	-	-	-
42h		POEG_GROUP1	✓	-	-	-
43h	SDADC24	SDADC_ADI	✓	✓	-	-
44h		SDADC_SCANEND	✓	✓	-	-
45h		SDADC_CALIEND	✓	-	-	-

Table 13.4 Event table (3 of 4)

Event number	Interrupt request source	Name	IELSRn		Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC		
46h	GPT320	GPT0_CCMPA	✓	✓	-	-
47h		GPT0_CCMPB	✓	✓	-	-
48h		GPT0_CMPC	✓	✓	-	-
49h		GPT0_CMPD	✓	✓	-	-
4Ah		GPT0_OVF	✓	✓	-	-
4Bh		GPT0_UDF	✓	✓	-	-
4Ch	GPT161	GPT1_CCMPA	✓	✓	-	-
4Dh		GPT1_CCMPB	✓	✓	-	-
4Eh		GPT1_CMPC	✓	✓	-	-
4Fh		GPT1_CMPD	✓	✓	-	-
50h		GPT1_OVF	✓	✓	-	-
51h		GPT1_UDF	✓	✓	-	-
52h	GPT162	GPT2_CCMPA	✓	✓	-	-
53h		GPT2_CCMPB	✓	✓	-	-
54h		GPT2_CMPC	✓	✓	-	-
55h		GPT2_CMPD	✓	✓	-	-
56h		GPT2_OVF	✓	✓	-	-
57h		GPT2_UDF	✓	✓	-	-
58h	GPT163	GPT3_CCMPA	✓	✓	-	-
59h		GPT3_CCMPB	✓	✓	-	-
5Ah		GPT3_CMPC	✓	✓	-	-
5Bh		GPT3_CMPD	✓	✓	-	-
5Ch		GPT3_OVF	✓	✓	-	-
5Dh		GPT3_UDF	✓	✓	-	-
5Eh	GPT164	GPT4_CCMPA	✓	✓	-	-
5Fh		GPT4_CCMPB	✓	✓	-	-
60h		GPT4_CMPC	✓	✓	-	-
61h		GPT4_CMPD	✓	✓	-	-
62h		GPT4_OVF	✓	✓	-	-
63h		GPT4_UDF	✓	✓	-	-
64h	GPT165	GPT5_CCMPA	✓	✓	-	-
65h		GPT5_CCMPB	✓	✓	-	-
66h		GPT5_CMPC	✓	✓	-	-
67h		GPT5_CMPD	✓	✓	-	-
68h		GPT5_OVF	✓	✓	-	-
69h		GPT5_UDF	✓	✓	-	-
6Ah	GPT166	GPT6_CCMPA	✓	✓	-	-
6Bh		GPT6_CCMPB	✓	✓	-	-
6Ch		GPT6_CMPC	✓	✓	-	-
6Dh		GPT6_CMPD	✓	✓	-	-
6Eh		GPT6_OVF	✓	✓	-	-
6Fh		GPT6_UDF	✓	✓	-	-
70h	GPT	GPT_UVWEDGE	✓	-	-	-

Table 13.4 Event table (4 of 4)

Event number	Interrupt request source	Name	IELSRn		Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC		
71h	SCI0	SCI0_RXI	✓	✓		
72h		SCI0_TXI	✓	✓		
73h		SCI0_TEI	✓	-	-	-
74h		SCI0_ERI	✓	-	-	-
75h		SCI0_AM	✓	-	✓*4	
76h		SCI0_RXI_OR_ERI	-	-	✓*4	
77h	SCI1	SCI1_RXI	✓	✓	-	-
78h		SCI1_TXI	✓	✓	-	-
79h		SCI1_TEI	✓	-	-	-
7Ah		SCI1_ERI	✓	-	-	-
7Bh		SCI1_AM	✓	-	-	-
7Ch	SCI9	SCI9_RXI	✓	✓	-	-
7Dh		SCI9_TXI	✓	✓	-	-
7Eh		SCI9_TEI	✓	-	-	-
7Fh		SCI9_ERI	✓	-	-	-
80h		SCI9_AM	✓	-	-	-
81h	SPI0	SPI0_SPRI	✓	✓	-	-
82h		SPI0_SPTI	✓	✓	-	-
83h		SPI0_SPII	✓	-	-	-
84h		SPI0_SPEI	✓	-	-	-
85h		SPI0_SPTEND	✓	-	-	-
86h	SPI1	SPI1_SPRI	✓	✓	-	-
87h		SPI1_SPTI	✓	✓	-	-
88h		SPI1_SPII	✓	-	-	-
89h		SPI1_SPEI	✓	-	-	-
8Ah		SPI1_SPTEND	✓	-	-	-
8Bh	AES	AES_WRREQ	✓	✓	-	-
8Ch		AES_RDREQ	✓	✓	-	-
8Dh	TRNG	TRNG_RDREQ	✓	-	-	-

Note 1. Only supported when KRCTL.KRMD is 1.
Note 2. Only the first edge detection is valid.
Note 3. Only interrupts after DTC transfer are supported.
Note 4. Using SELSR0.

13.4 Interrupt Operation

The ICU performs the following functions:

- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations such as CPU interrupt or DTC activation.

13.4.1 Detecting Interrupts

External pin interrupt requests are detected by 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 module, see section 13.3.2, Event Number. Events must be accepted by the NVIC before an interrupt occurs and is accepted by the CPU.

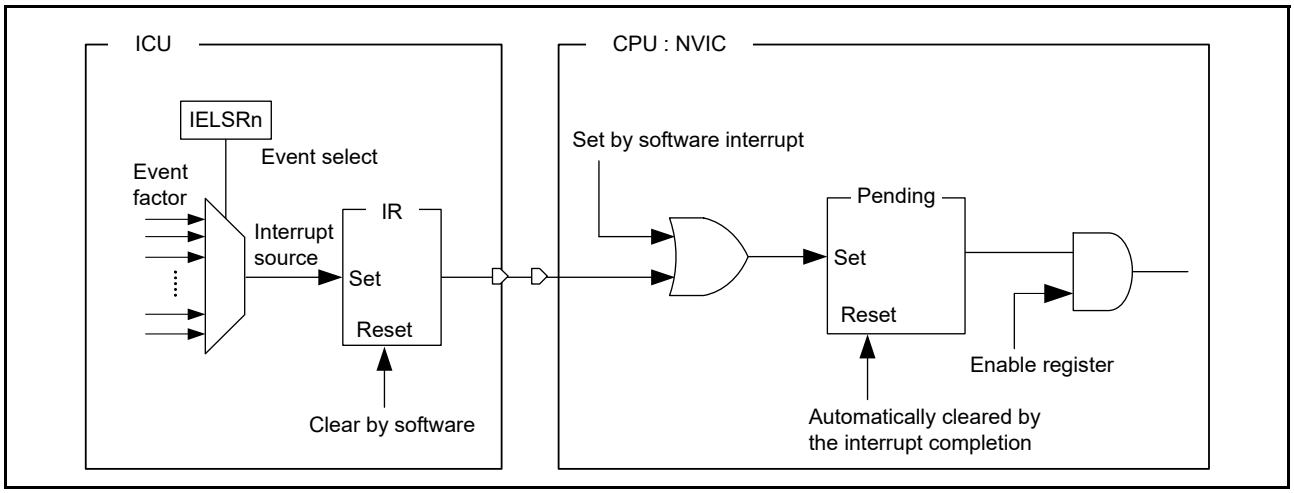


Figure 13.2 Interrupt path of the ICU and CPU: NVIC

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_ISPR/NVIC_ICPR) are set.
- When a software interrupt occurs:
  Set the Interrupt Set-Pending register (NVIC_ISPR).
- When an interrupt is complete:
  Clear the IELSRn.IR flag with software.
  The Interrupt Set/Clear-Pending register (NVIC_ISPR/NVIC_ICPR) clears automatically.

When interrupts are enabled:

1. Set the Interrupt Set-Enable register (NVIC_ISER).
2. Set the IELSRn.IELS[7:0] bits as the interrupt source.
3. Specify the operation settings for the event source.

When interrupts are disabled:

1. Disable the settings for the event source.
2. Clear the IELSRn.IELS[7:0] bits (IELSRn.IELS[7:0] = 00h). Clear the IELSRn.IR flag as required.
3. Clear the Interrupt Clear-Enable register (NVIC_ICER). Clear the Interrupt Clear-Pending register (NVIC_ICPR) as required.

When polling for interrupts:

1. Set the Interrupt Clear-Enable register (NVIC_ICER) (disabling interrupts).
2. Set the IELSRn.IELS[7:0] bits (selecting the source).
3. Specify the operation settings for the event source.
4. Poll the Interrupt Set-Pending register (NVIC_ISPR).
5. When polling is no longer required, follow the procedure for clearing an interrupt when it is complete.

13.4.2 Selecting Interrupt Request Destinations

The interrupt output destination, CPU or DTC, can be independently selected for each interrupt source. The available destinations are fixed for each interrupt, as described in Table 13.4, Event table.

Note: Do not use an interrupt request destination setting that is not indicated by a check, ✓, in the event list (Table 13.4, Event table).

If you select the CPU or DTC in the IELSRn register, setting the same interrupt factor in any other IELSRn register is prohibited.

If the DTC is selected as the destination for requests from an IRQi pin, you must set the IRQMD[1:0] bits in IRQCRi for that interrupt to select edge detection.

13.4.2.1 CPU interrupt request

When IELSRn.DTCE = 0, the event specified in the IELSRn register is output to the NVIC. Set the IELSRn.IELS[7:0] bits to the target event and set the IELSRn.DTCE bit to 0.

13.4.2.2 DTC activation

When IELSRn.DTCE = 1, the event specified in the IELSRn register is output to the DTC. Use the following procedure:

1. Set the IELSRn.IELS[7:0] bits to the target event and set the IELSRn.DTCE bit to 1.
2. Set the DTC module activation bit (DTCST.DTCST) to 1.

Table 13.5 shows operation when the DTC is the request destination.

Table 13.5 Operations when DTC is activated

Interrupt request destination	DISEL*1	Remaining transfer operations	Operation per request	IR*2	Interrupt request destination after transfer
DTC*3	1	≠ 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	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 16.3, Chain transfer conditions in section 16, Data Transfer Controller (DTC).

13.4.3 Digital Filter

A digital filter function is provided for the external interrupt request pins (IRQi, i = 0 to 7) and the NMI pin interrupt. It samples input signals on the filter sampling clock (PCLKB) and removes any signal with a pulse width less than 3 sampling cycles.

- To use the digital filter for an IRQi pin:

1)	Set the sampling clock cycle to PCLKB, PCLKB/8, PCLKB/32, or PCLKB/64 in the IRQCRi.FCLKSEL[1:0] bits (i = 0 to 7).
2)	Set the IRQCRi.FLTEN bit (i = 0 to 7) to 1 (digital filter enabled).

- To use the digital filter for 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 13.3 shows an example of digital filter operation.

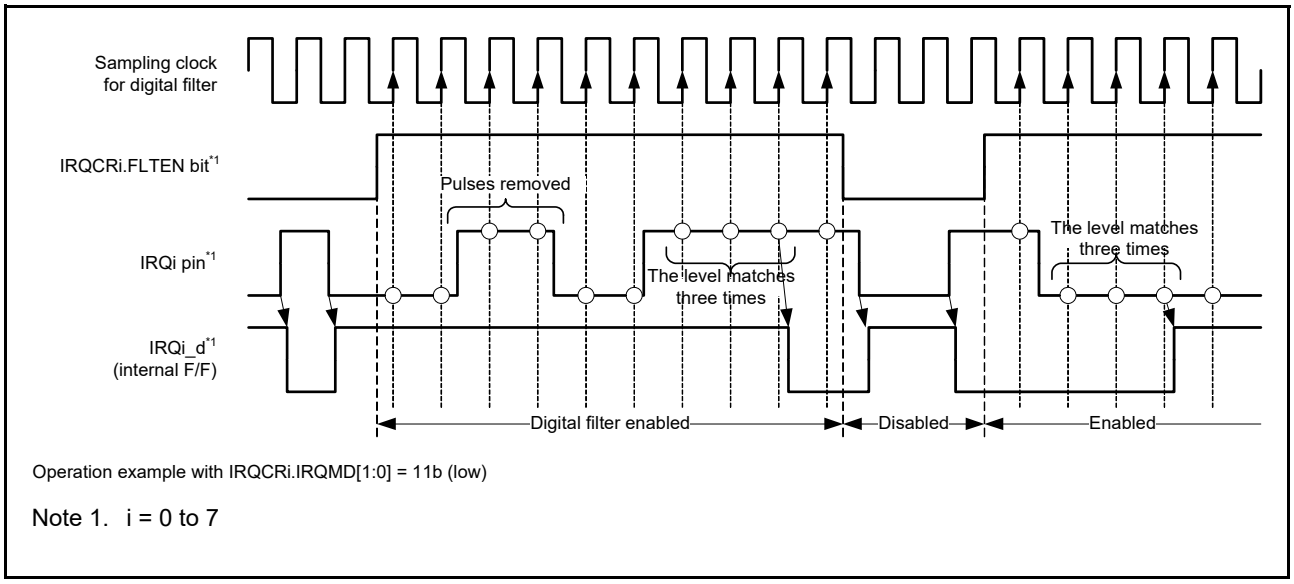


Figure 13.3 Digital filter operation example

Before entering Software Standby mode, disable the digital filters by clearing the IRQCRi.FLTEN and NMICR.NFLTEN bits. The ICU clock stops in Software Standby mode. You can enable the digital filters again after exiting Software Standby mode.

The circuit detects the edge by comparing the state before standby to the state after standby release. If the input changes during Software Standby mode, an incorrect edge might be detected.

13.4.4 External Pin Interrupts

To use external pin interrupts:

1. Clear the IRQCRi.FLTEN bit (i = 0 to 7) to 0 (digital filter disabled).
2. Set or confirm the I/O port settings.
3. Set the IRQMD[1:0] bits, the FCLKSEL[1:0] bits, and the FLTEN bit of the IRQCRi register.
4. Select the IRQ pin as follows:
   - If the IRQ pin is to be used for CPU interrupt requests, set the IELSRn.IELS[7:0] bits and the IELSRn.DTCE bit to 0
   - If the IRQ pin is to be used for DTC activation, set the IELSRn.IELS[7:0] bits and the IELSRn.DTCE bit to 1.

13.5 Non-Maskable Interrupt Operation

The following sources can trigger a non-maskable interrupt:

- NMI pin interrupt
- Oscillation stop detection interrupt
- WDT underflow/refresh error interrupt
- IWDT underflow/refresh error interrupt
- Voltage monitor 1 interrupt
- Voltage monitor 2 interrupt
- 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. 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.

13.6 Return from Low Power Modes

Table 13.4, Event table lists the interrupt sources you can use to exit Sleep or Software Standby mode. For more information, see section 11, Low Power Modes. Sections 13.6.1 to 13.6.3 describe how to use interrupts to return from Sleep, Software Standby, and Snooze modes.

13.6.1 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.

13.6.2 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 13.2.8, 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 IRQn pins that do not satisfy these conditions are not detected while the clock is stopped in Software Standby mode.

13.6.3 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.SELS[7:0] and set the value 0Bh (ICU_SNZCANCEL) in IELSRn.IELS[7:0]
   b. Set the event that you want to trigger a return to Normal mode from Snooze mode in IELSRn.IELS[7:0].
2. Select the CPU as the interrupt 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.

13.7 Using the WFI Instruction with Non-Maskable Interrupt

Whenever a WFI instruction is executed, confirm that all status flags in the NMISR register are 0.

13.8 Reference

ARM® Cortex®-M23 Processor Technical Reference Manual (ARM DDI 0550C).

14. Buses

14.1 Overview

Table 14.1 lists the bus specifications, Figure 14.1 shows the bus configuration, and Table 14.2 lists the addresses assigned for each bus.

Table 14.1 Bus specifications

Bus type		Specifications
Main bus	System bus (CPU)	• Connected to CPU • Connected to on-chip memory and internal peripheral bus.
	DMA bus	• Connected to DTC • Connected to on-chip memory and internal peripheral bus.
Slave Interface	Memory bus 1	• Connected to code flash memory
	Memory bus 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, ADC16, DAC12, DOC, GPT, SCI, SPI, and CRC)
	Internal peripheral bus 5	• Connected to peripheral modules (KINT, AGT, USBFS, DAC8, OPAMP, ACMPHS, ACMPLP, SDADC24, and CTSU)
	Internal peripheral bus 7	• Connected to Secure IPs
	Internal peripheral bus 9	• Connected to flash memory (in P/E) and data flash memory

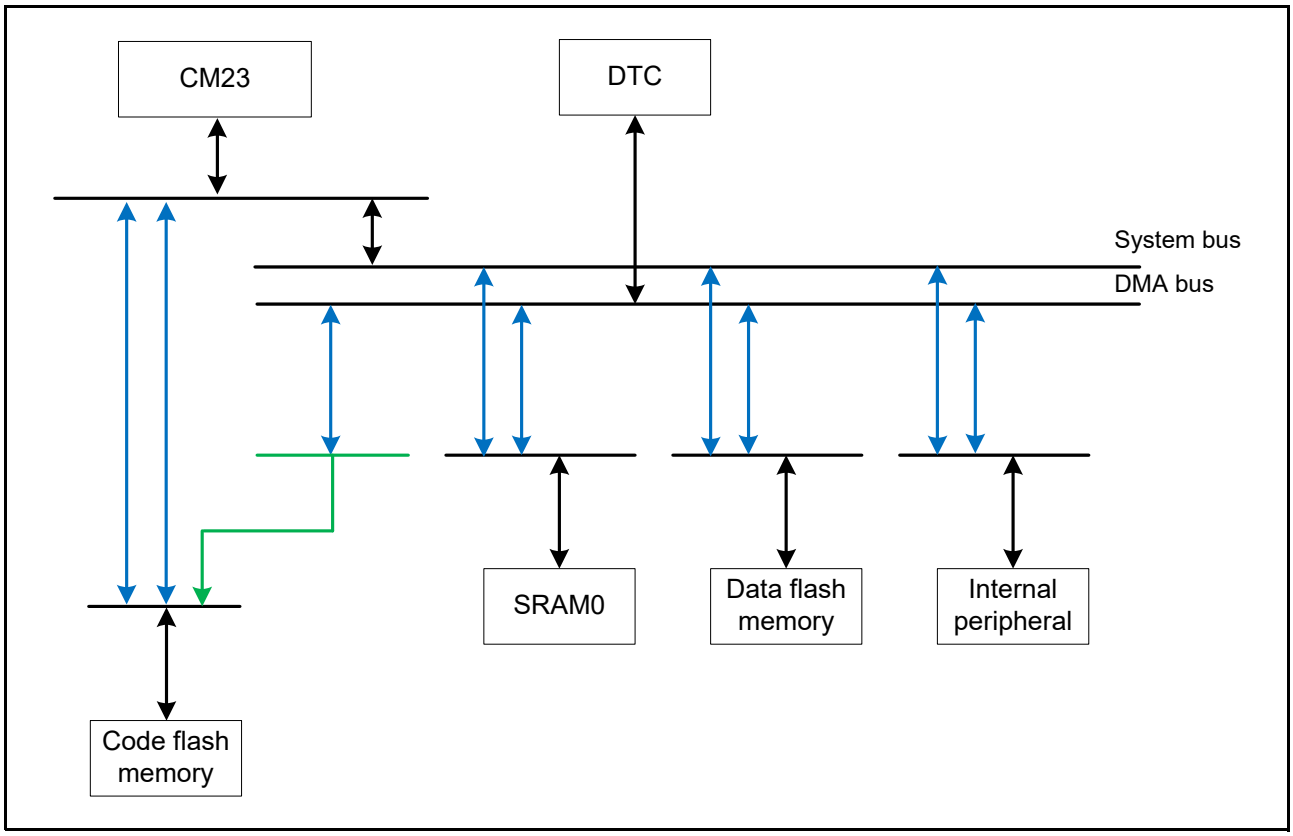


Figure 14.1 Bus configuration

Table 14.2 Addresses assigned for each bus

Address	Bus	Area
0000 0000h to 01FF FFFFh	Memory bus 1, 3	Code flash memory
2000 0000h to 2000 7FFFh	Memory bus 4	SRAM0
4000 0000h to 4001 8FFFh	Internal peripheral bus 1	Peripheral I/O registers
4001 9000h to 4001 9FFFh	Memory bus 4	MTB I/O registers
4001 A000h to 4001 FFFFh	Internal peripheral bus 1	Peripheral I/O registers
4004 0000h to 4007 FFFFh	Internal peripheral bus 3	
4008 0000h to 400B 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

14.2 Description of Buses

14.2.1 Main Buses

The main buses for the CPU consist of the system bus and DMA bus. System bus and DMA bus are connected to the following:

- Code flash memory
- SRAM0
- Data flash memory
- Internal peripheral bus.

The system bus is used for instruction code and data code access to the CPU.

Different master and slave transfer combinations can proceed simultaneously. In addition, requests for bus access from masters other than the DTC are not accepted during reads of transfer control information for the DTC.

14.2.2 Slave Interface

For connections from the main bus to the slave interface, see the slave interface in Table 14.1, Bus specifications.

Bus access from the system bus and DMA bus are arbitrated. The arbitration method is selectable as either fixed priority or round-robin. For more information, see section 14.3.2.

Different master and slave transfer combinations can proceed simultaneously.

14.2.3 Parallel Operation

Parallel operation is possible when different bus masters request access to different slave modules. Figure 14.2 shows an example of parallel operations. In this example, the CPU uses the instruction and operand buses for simultaneous access to the flash and SRAM, respectively. Additionally, the DTC simultaneously uses the DMA bus for access to a peripheral bus during access to the flash and SRAM by the CPU.

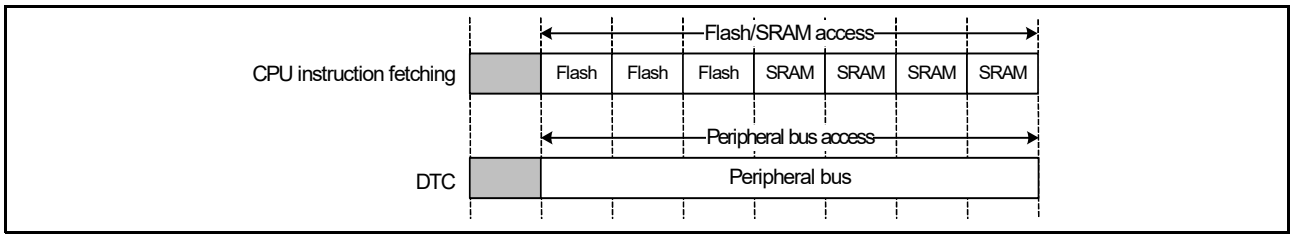


Figure 14.2 Example of parallel operations

14.2.4 Restriction on Endianness

Memory space must be little-endian to execute code on the Cortex®-M23 core.

14.3 Register Descriptions

14.3.1 Master Bus Control Register (BUSMCNT<master>)

Address(es): 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 14.3 lists the registers associated with each bus type.

Table 14.3 Associations between bus types and registers

Bus type	Master Bus Control Register	Slave Bus Control Register	Bus Error Address Register	Bus Error Status Register
System bus (CPU)	BUSMCNTSYS	-	BUS3ERRADD	BUS3ERRSTAT
DMA bus	BUSMCNTDMA	-	BUS4ERRADD	BUS4ERRSTAT
Memory bus 1	—	BUSSCNTFLI	-	-
Memory bus 4	—	BUSSCNTRAM0	-	-
Internal peripheral bus 1, 3, 5, 7	—	BUSSCNTPnB [n = 0, 2, 4, 6]	-	-
Internal peripheral bus 9	—	BUSSCNTFBU	-	-

14.3.2 Slave Bus Control Register (BUSSCNT<slave>)

Address(es): BUS.BUSSCNTFLI 4000 4100h, BUS.BUSSCNTRAM0 4000 410Ch, BUS.BUSSCNTP0B 4000 4114h, BUS.BUSSCNTP2B 4000 4118h, 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	Specifies the group priorities: b5 b4 0 0: Fixed priority 0 1: Round-robin 1 0: Setting prohibited 1 1: Setting prohibited.	R/W
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Changing 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 14.4. For round-robin, see Table 14.5. For the associations between bus types and registers, see Table 14.3.

Table 14.4 Fixed priority (ARBMET[1:0] = 00b)

Slave Bus Control Register	Slave interface	Priority
BUSSCNTFLI	Memory bus 1	DMA bus > System bus (CPU)
BUSSCNTRAM0	Memory bus 4	DMA bus > System bus (CPU)
BUSSCNTPnB [n = 0, 2, 4, 6]	Internal peripheral bus 1, 3, 5, 7	DMA bus > System bus (CPU)
BUSSCNTFBU	Internal peripheral bus 9	DMA bus > System bus (CPU)

Table 14.5 Round-robin priority (ARBMET[1:0] = 01b)

Slave Bus Control Register	Slave interface	Priority*1
BUSSCNTFLI	Memory bus 1	DMA bus ↔ System bus (CPU)
BUSSCNTRAM0	Memory bus 4	DMA bus ↔ System bus (CPU)
BUSSCNTPnB [n = 0, 2, 4, 6]	Internal peripheral bus 1, 3, 5, 7	DMA bus ↔ System bus (CPU)
BUSSCNTFBU	Internal peripheral bus 9	DMA bus ↔ System bus (CPU)

Note 1. Round-robin priority is denoted by '↔'.

14.3.3 Bus Error Address Register (BUSnERRADD) (n = 3, 4)

Address(es): BUS.BUS3ERRADD 4000 4820h, BUS.BUS4ERRADD, 4000 4830h

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

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

Bit	Symbol	Bit name	Description	R/W
b31 to b0	BERAD[31:0]	Bus Error Address	When a bus error occurs, these bits store the error address	R

Note: This register is only cleared by resets other than MPU-related resets. For more information, see section 6, Resets, and section 15, Memory Protection Unit (MPU).

Table 14.3 lists the registers associated with each bus type.

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

The BERAD[31:0] bits store the accessed address when a bus error occurred. For more information, see the description of the BUSnERRSTAT.ERRSTAT bit and section 14.4, Bus Error Monitoring Section.

The value of the BUSnERRADD.BERAD[31:0] bits (n = 3, 4) is valid only when the BUSnERRSTAT.ERRSTAT bit (n = 3, 4) is set to 1.

14.3.4 BUS Error Status Register (BUSnERRSTAT) (n = 3, 4)

Address(es): 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
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 15, Memory Protection Unit (MPU).

Table 14.3 lists the registers associated with each bus type.

ACCSTAT bit (Error Access Status)

The ACCSTAT bit indicates the access status, write or read access, when a bus error occurs. For more information, see the description of the BUSnERRSTAT.ERRSTAT bit and section 14.4, Bus Error Monitoring Section.

The value is valid only when the BUSnERRSTAT.ERRSTAT bit (n = 3, 4) is set to 1.

ERRSTAT bit (Bus Error Address)

The ERRSTAT bit indicates whether a bus error occurred. When a bus error occurs, the access address and status of write or read access are stored. The BUSnERRSTATn.ERRSTAT bit (n = 3, 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 (this can occur with the wait setting), BUSnERRSTAT.ERRSTAT (n = 3, 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 = 3, 4) is set to 1 after the bus access that caused the MPU error completes.

For more information on bus errors, see section 14.4, Bus Error Monitoring Section and section 15, Memory Protection Unit (MPU).

14.4 Bus Error Monitoring Section

The monitoring system monitors each individual area, and whenever it detects an error, it returns the error to the requesting master IP using the AHB-Lite error response protocol.

14.4.1 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 14.6 lists the address ranges where access leads to illegal address access errors. The reserved area in the slave does not trigger an illegal address access error.

For more information on bus master MPU and bus slave MPU, see section 15, Memory Protection Unit (MPU).

14.4.2 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 14.3.3 and 14.3.4.

Note: DTC does not receive bus errors. If the DTC accesses the bus, the transfer continues.

14.4.3 Conditions Leading to Illegal Address Access Errors

Table 14.6 lists the address spaces for each bus that trigger illegal address access errors.

Table 14.6 Conditions leading to illegal address access errors

Address	Slave bus name	Master bus	
		CPU	
		System	DMA
0000 0000h to 01FF FFFFh	Memory bus 1, 3	—	—
0200 0000h to 1FFF FFFFh	Reserved	E	E
2000 0000h to 2000 7FFFh	Memory bus 4	—	—
2000 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 4007 FFFFh	Peripheral bus 3	—	—
4008 0000h to 400B FFFFh	Peripheral bus 5	—	—
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 DFFF FFFFh	Reserved	E	E
E000 0000h to FFFF FFFFh	System for Cortex-M23	—	E

E indicates the path where an illegal address access error occurs.

— indicates the path where an illegal address access error does not occur.

Note: The bus module detects an access error resulting from access to reserved area, for example if no area is assigned for the slave.
0200 0000h to 1FFF FFFFh: Access error detection.
0000 0000h to 01FF FFFFh: Memory bus 1 no access error detection.

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.

14.4.4 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.

14.5 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.

14.6 References

1. *ARM®v8-M Architecture Reference Manual* (ARM DDI 0553B.a)
2. *ARM® Cortex®-M23 Processor User Guide* (ARM DUI 0963B)
3. *ARM® AMBA® 5 AHB-Lite Protocol Specification* (ARM IHI 0033B.b).

15. Memory Protection Unit (MPU)

15.1 Overview

The MCU has four Memory Protection Units (MPUs) and a CPU stack pointer monitor function. Table 15.1 lists the supported MPU specifications, and Table 15.2 shows the behavior on detection of each MPU error.

Table 15.1 MPU specifications

Classification	Module/Function	Specifications
Illegal memory access	Arm® Cortex®-M23 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 sub regions and background region.
	Bus master MPU	Memory protection function for each bus master except for the CPU: • Bus master MPU Group A: 4 regions.
	Bus slave MPU	Memory protection function for each bus slave
Security	Security MPU	Protect accesses from non-secure programs to the following secure regions: • 2 regions (PC) • 4 regions (code flash, SRAM, two secure functions).

Table 15.2 Behavior on MPU error detection

MPU type	Notice method	Bus access on error detection	Storing of error access information
CPU stack pointer monitor	Reset or non-maskable interrupt	Don't care	Not stored
Arm MPU	Hard fault	• Does not correctly have write access • Does not correctly have read access.	Not stored
Bus master MPU	Reset or non-maskable interrupt	• Write access to the protection region • Read access to the protection region.	Stored
Bus slave MPU	Reset or non-maskable interrupt 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 15.7. For information on error access for other MPUs, see section 14.3.3, Bus Error Address Register (BUSnERRADD) (n = 3, 4) and section 14.3.4, BUS Error Status Register (BUSnERRSTAT) (n = 3, 4) in section 14, Buses.

15.2 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 15.3 lists the specifications of the CPU stack pointer monitor. Figure 15.1 shows a block diagram, and Figure 15.2 shows the register setting flow.

Table 15.3 CPU stack pointer monitor specifications

Parameter	Description
SRAM region	Region to be covered by memory protection
Number of regions	2 regions: • Main Stack Pointer (MSP) • Process Stack Pointer (PSP).
Address specification for individual regions	Region start and end addresses configurable
Stack pointer monitor enable or disable setting for individual regions	Stack pointer monitor for individual regions can be enabled or disabled
Operation on error detection	Reset or non-maskable interrupts can be generated
Register protection	Registers can be protected from illegal writes

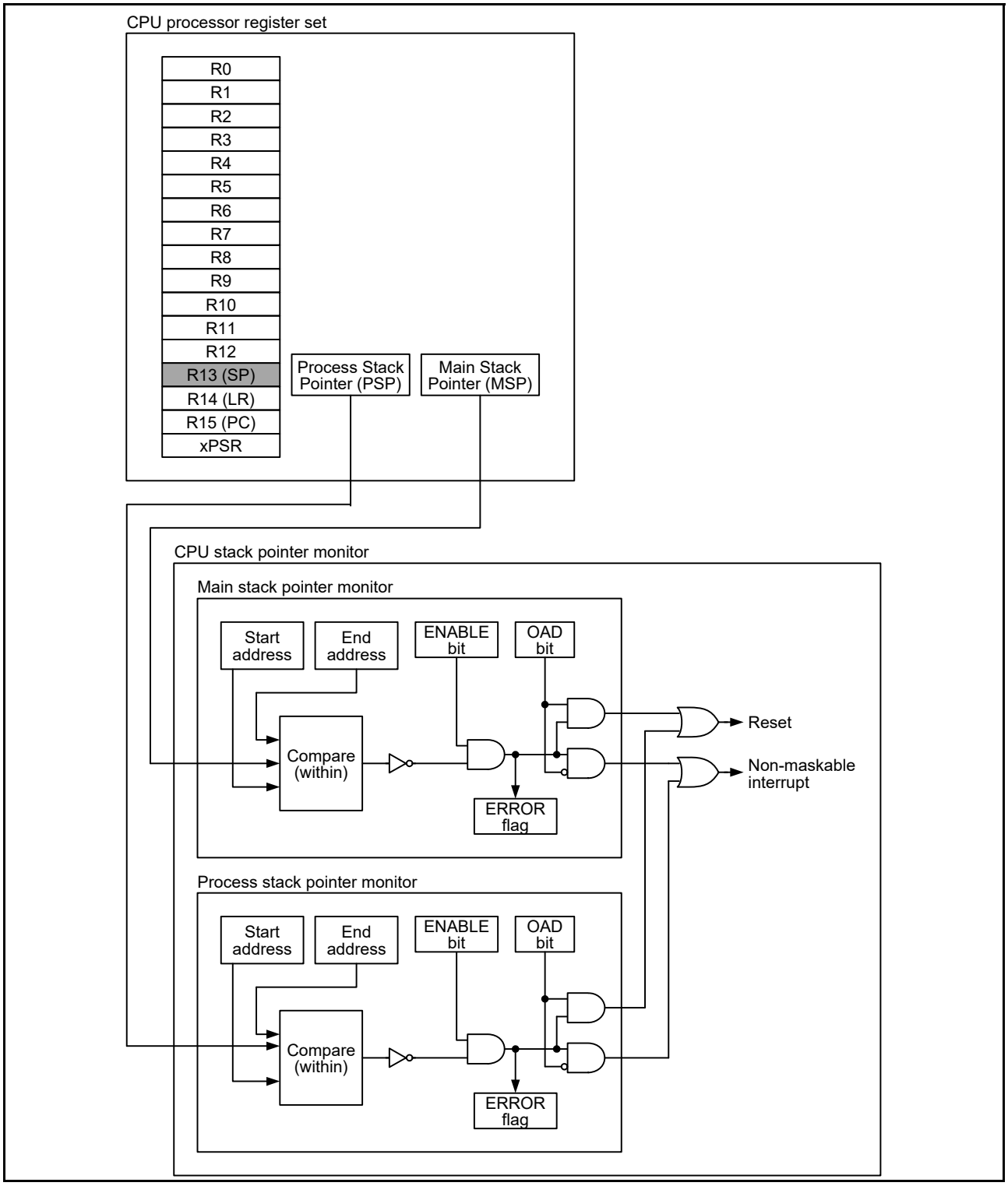


Figure 15.1 CPU stack pointer monitor block diagram

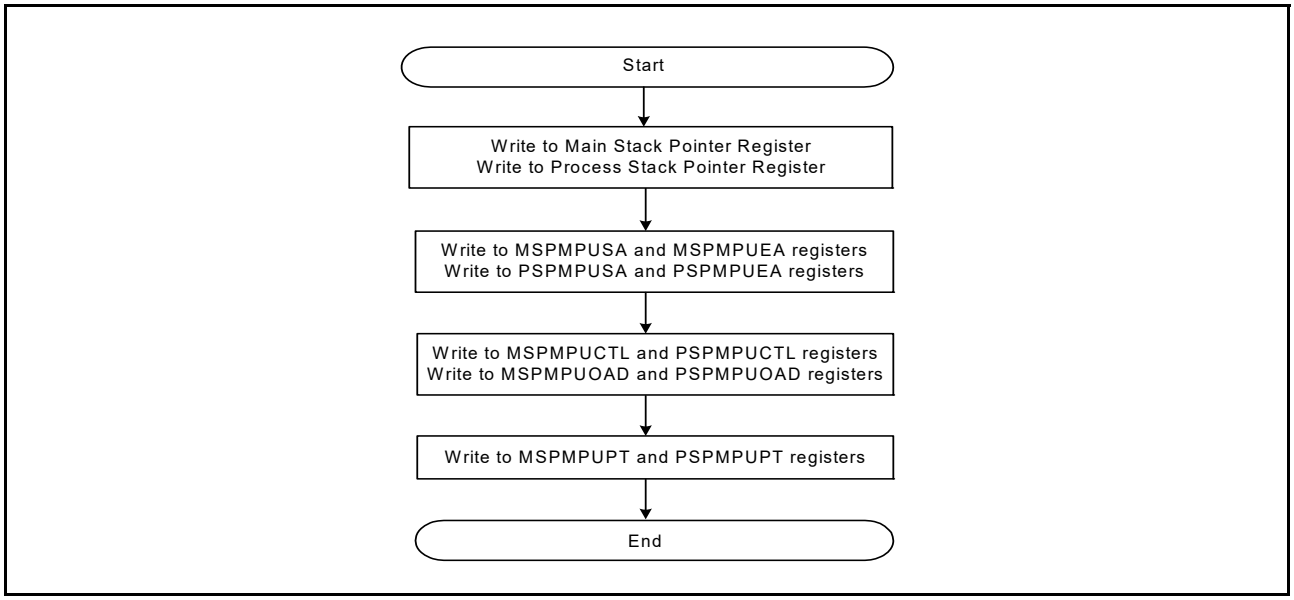


Figure 15.2 Register setting flow

15.2.1 Protection of Registers

Registers related to the CPU stack pointer monitor can be protected with the PROTECT bit.

15.2.2 Overflow/Underflow Error

If 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 13, 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 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.

15.2.3 Register Descriptions

Note: Bus access must be stopped before writing to MPU registers.

15.2.3.1 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 1FF0 0000h to 200F FFFCh, excluding reserved areas.	R/W

The MSPMPUSA and MSPMPUEA registers specify the CPU stack region in the SRAM (1FF0 0000h to 200F FFFFh, excluding reserved areas). For SRAM area to be covered, see Figure 4.1, Memory map.

15.2.3.2 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 is from 1FF0 0003h to 200F FFFFh, excluding reserved areas.	R/W

15.2.3.3 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 is from 1FF0 0000h to 200F FFFCh, excluding reserved areas.	R/W

The PSPMPUSA and PSPMPUEA registers specify the CPU stack region in the SRAM (1FF0 0000h to 200F FFFFh, excluding reserved areas). For SRAM area to be covered, see Figure 4.1, Memory map.

15.2.3.4 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 1FF0 0003h to 200F FFFFh, excluding reserved areas.	R/W

15.2.3.5 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 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. When writing to the OAD bit, write A5h simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the OAD bit. When writing to the OAD bit, simultaneously write A5h to KEY[7:0] bits. When values other than A5h are written to the KEY[7:0] bits, the OAD bit is not updated. The KEY[7:0] bits always read as 00h.

15.2.3.6 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 of 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 pointer monitor has an independent ERROR bit.

[Setting condition]

- Overflow or underflow of the stack pointer.

[Clearing condition]

- 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.

15.2.3.7 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 permitted 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 the PROTECT bit, simultaneously write A5h to KEY[7:0] bits. 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.

15.3 Arm MPU

The Arm MPU provides full support for:

- 8 protection regions
- 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 15.7, References.

15.4 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 4 regions. The bus master MPU monitors access to each region based on these settings. If access to a protected region is detected, the bus master MPU generates an internal reset or a non-maskable interrupt. For information on error access, see 14.3.3 and 14.3.4 in section 14, Buses.

Table 15.4 lists the specifications of the bus master MPU, and Figure 15.3 shows a block diagram.

Table 15.4 Bus master MPU specifications

Parameter	Description
Protected master groups	Bus master MPU group A: DMA bus
Protected regions	0000 0000h to FFFF FFFFh
Number of regions	Bus master MPU group A: 4 regions
Address specification for individual regions	Region start and end addresses configurable
Enable or disable setting for memory protection in individual regions	Setting enabled or disabled for the associated region
Access-control settings for individual regions	Permission to read and to write
Operation on error detection	Reset or non-maskable interrupts
Register protection	Register can be protected from illegal writes

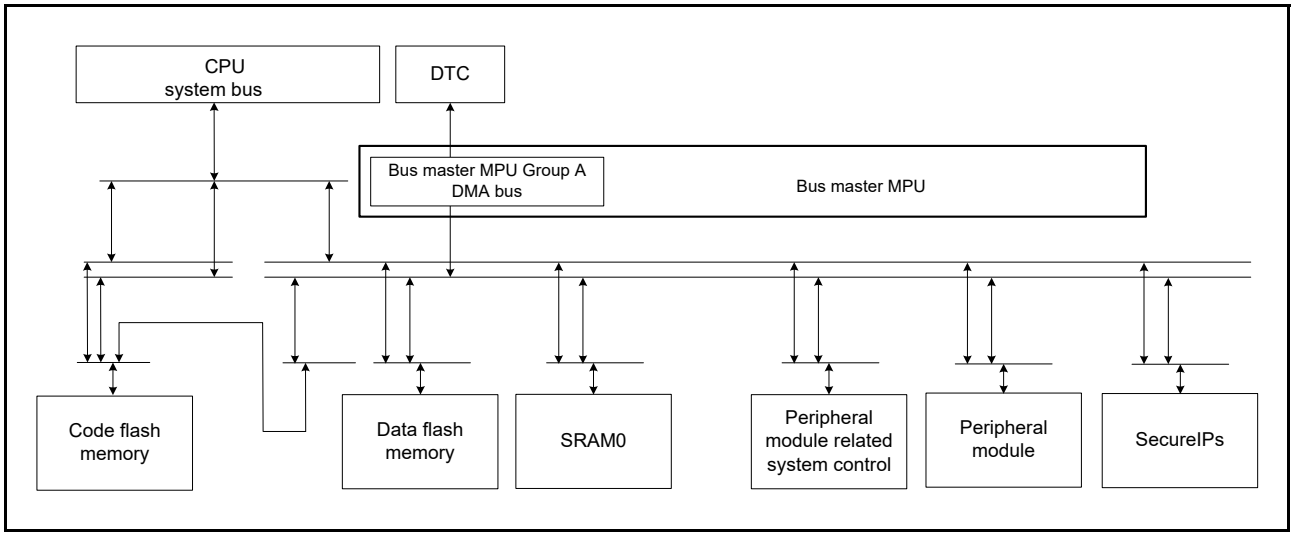


Figure 15.3 MPU bus master block diagram

Figure 15.4 shows the MPU bus master group A.

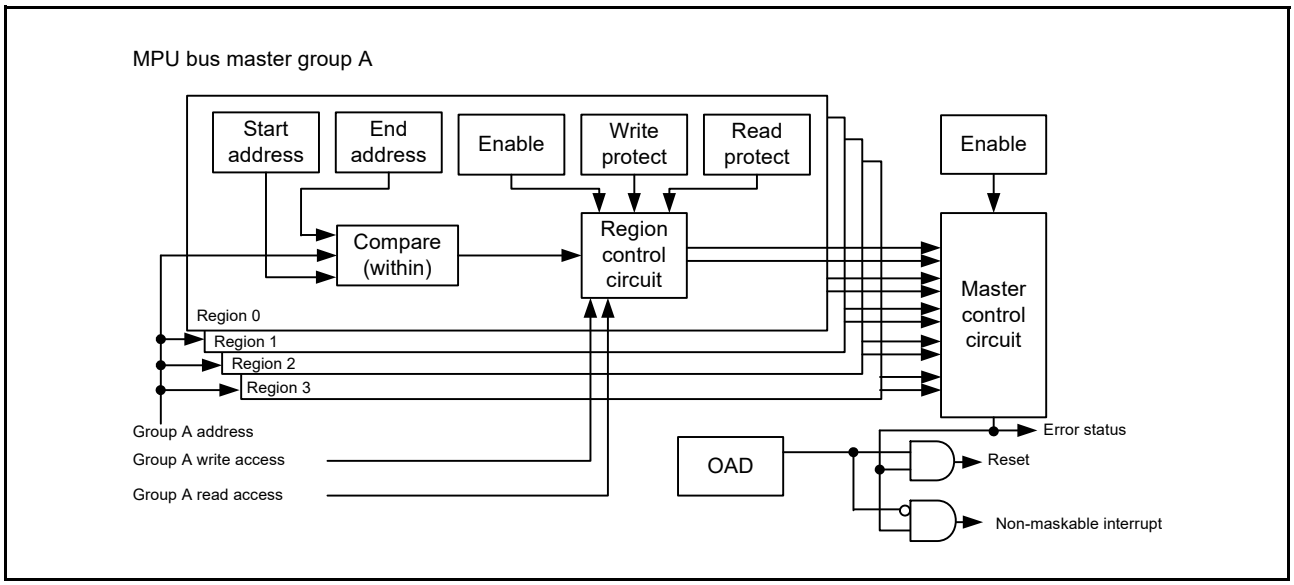


Figure 15.4 MPU bus master group A

15.4.1 Register Descriptions

Note: Bus access must be stopped before writing to MPU registers.

15.4.1.1 Group A Region n Start Address Register (MMPUSAn) (n = 0 to 3)

Address(es): MMPU.MMPUSA0 4000 0204h, MMPU.MMPUSA1 4000 0214h, MMPU.MMPUSA2 4000 0224h, MMPU.MMPUSA3 4000 0234h

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

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MMPUSA[31:0]	Region Start Address	Address where the region starts, for use in region determination. The lower 2 bits should be 0.	R/W

15.4.1.2 Group A Region n End Address Register (MMPUEAn) (n = 0 to 3)

Address(es): MMPU.MMPUEA0 4000 0208h, MMPU.MMPUEA1 4000 0218h, MMPU.MMPUEA2 4000 0228h, MMPU.MMPUEA3 4000 0238h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MMPUEA[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	MMPUEA[31:0]	Region End Address	Address where the region ends, for use in region determination. The lower 2 bits should be 1.	R/W

15.4.1.3 Group A Region n Access Control Register (MMPUACAn) (n = 0 to 3)

Address(es): MMPU.MMPUACA0 4000 0200h, MMPU.MMPUACA1 4000 0210h, MMPU.MMPUACA2 4000 0220h, MMPU.MMPUACA3 4000 0230h

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

Bit	Symbol	Bit name	Description	R/W
b0	ENABLE	Region Enable	0: Group A region n unit disabled 1: Group A region n unit enabled.	R/W
b1	RP	Read Protection	0: Read 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 unit.

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 and WP bits 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 15.5 shows the correspondence of the output information from the group A area n unit when the area set by the MMPUACAn register is accessed.

Table 15.5 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	Protected region
				Outside	Outside of region
	1	0	Read	Inside	Protected region
				Outside	Outside of region
			Write	Inside	Permitted region
				Outside	Outside of region
	1	1	Read	Inside	Protected region
				Outside	Outside of region
			Write	Inside	Protected region
				Outside	Outside of region

n = 0 to 3

Table 15.6 Function of master control circuit

MMPUCTLA.ENABLE	Output of group A region 0 unit	Output of group A region 1 unit	Output of group A region 2 to 3 unit	Function of group A
1	Protected region	Don’t care	Don’t care	Generate error
1	Don’t care	Protected region	Don’t care	Generate error
1	Don’t care	Don’t care	Protected region	Generate error
1	Outside of region	Outside of region	Outside of region	Generate error
Other case				No error

A master MPU error occurs on the following conditions:

- MMPUCTLA.ENABLE = 1, and output of one or more region n units is to a protected region
- MMPUCTLA.ENABLE = 1, and output of all region n units is outside of region.

Other cases are handled as permitted regions.

15.4.1.4 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 1, MMPUACAn is available. When this bit is set to 0, MMPUACAn is unavailable, including permission for all regions. When writing to the ENABLE bit, simultaneously write A5h to the KEY[7:0] bits using halfword access.

OAD bit (Operation After Detection)

The OAD bit generates either a reset or non-maskable interrupt when access to the protected region is detected by the bus master MPU. When the OAD bit is set, 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 ENABLE and OAD bits. When writing to the ENABLE and OAD bits, simultaneously write A5h to the KEY[7:0] bits. 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.

15.4.1.5 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

Bit	Symbol	Bit name	Description	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 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, simultaneously write A5h to the KEY[7:0] bits. 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.

15.4.2 Operation

15.4.2.1 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 configured for up to four protected regions. Protected regions include those with overlapping permitted and protected regions, and those with two overlapping permitted regions.

The bus master MPU provides group A. The memory protection function checks the address of the bus for a unified 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. A permitted region is set up within the protected region for each region. If access to the protected region is detected, the bus master MPU generates an error.

Figure 15.5 shows the use case of a bus master MPU.

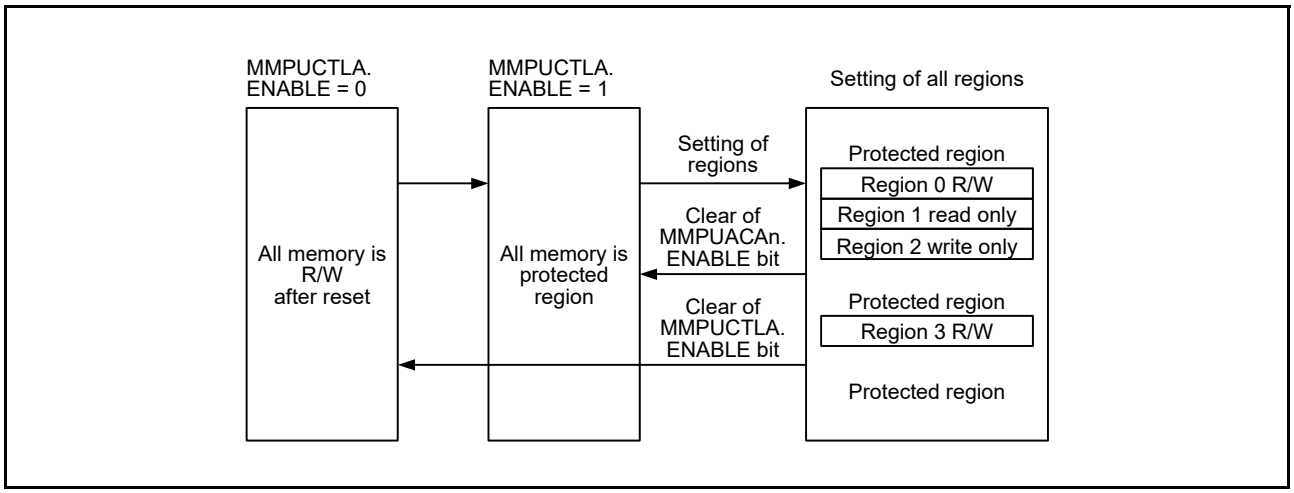


Figure 15.5 Use case of bus master MPU

Figure 15.6 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.

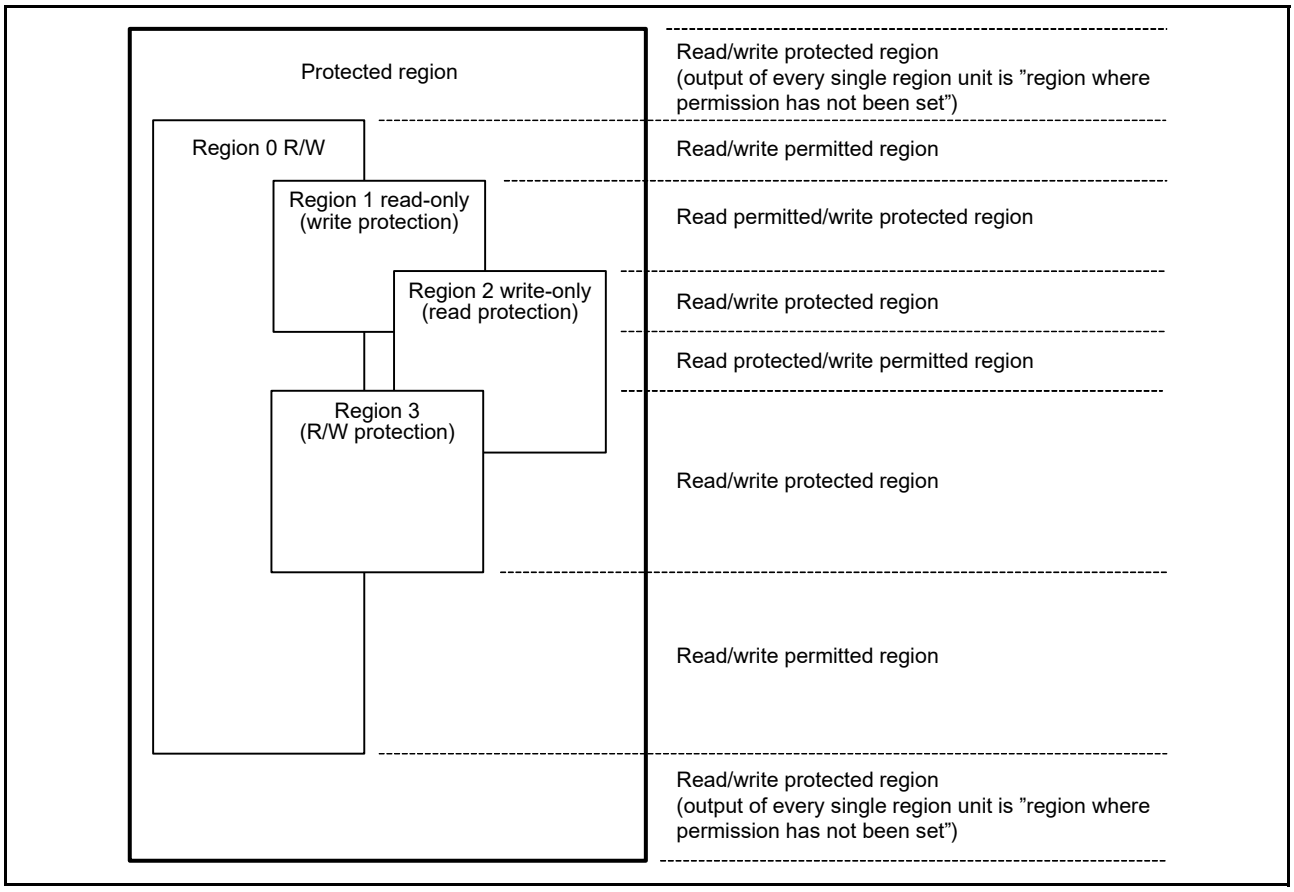


Figure 15.6 Access permission or protection by overlap of the bus master MPU regions

Figure 15.7 shows the register setting flow after reset. During this register setting, stop the bus master except the CPU.

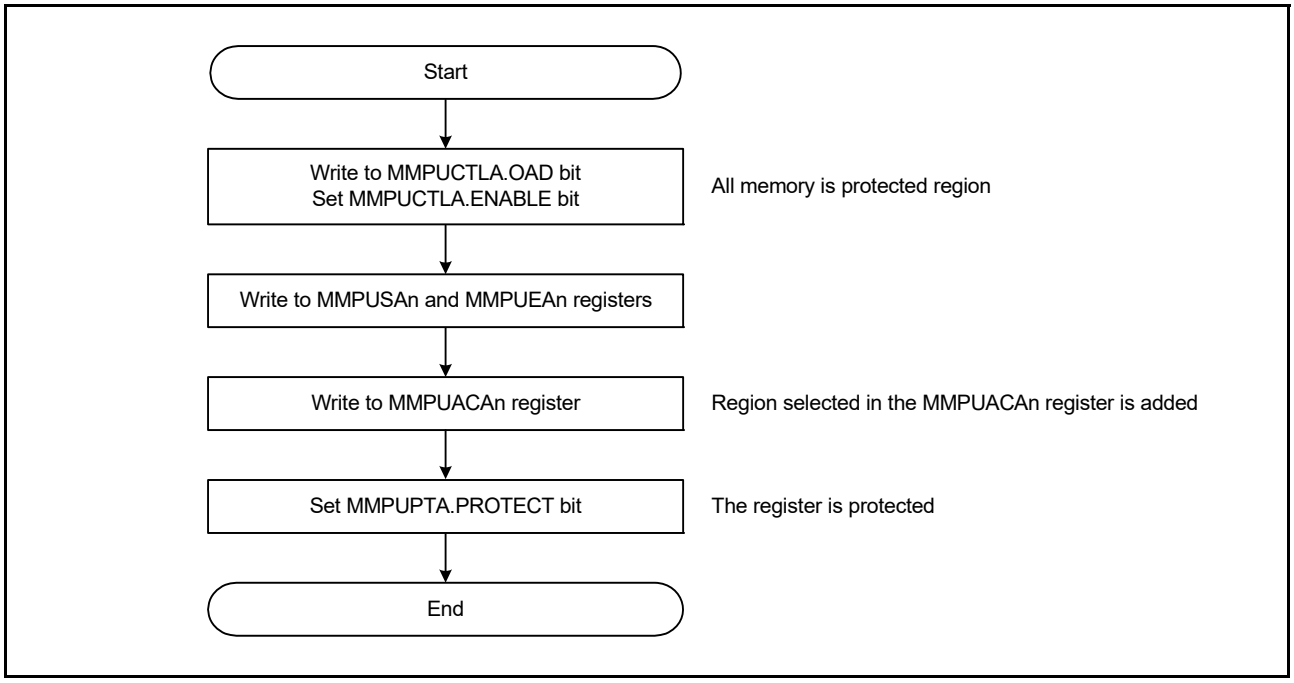


Figure 15.7 Register setting flow after reset

Figure 15.8 shows the register setting flow for adding regions. During this register setting, stop all masters except the CPU.

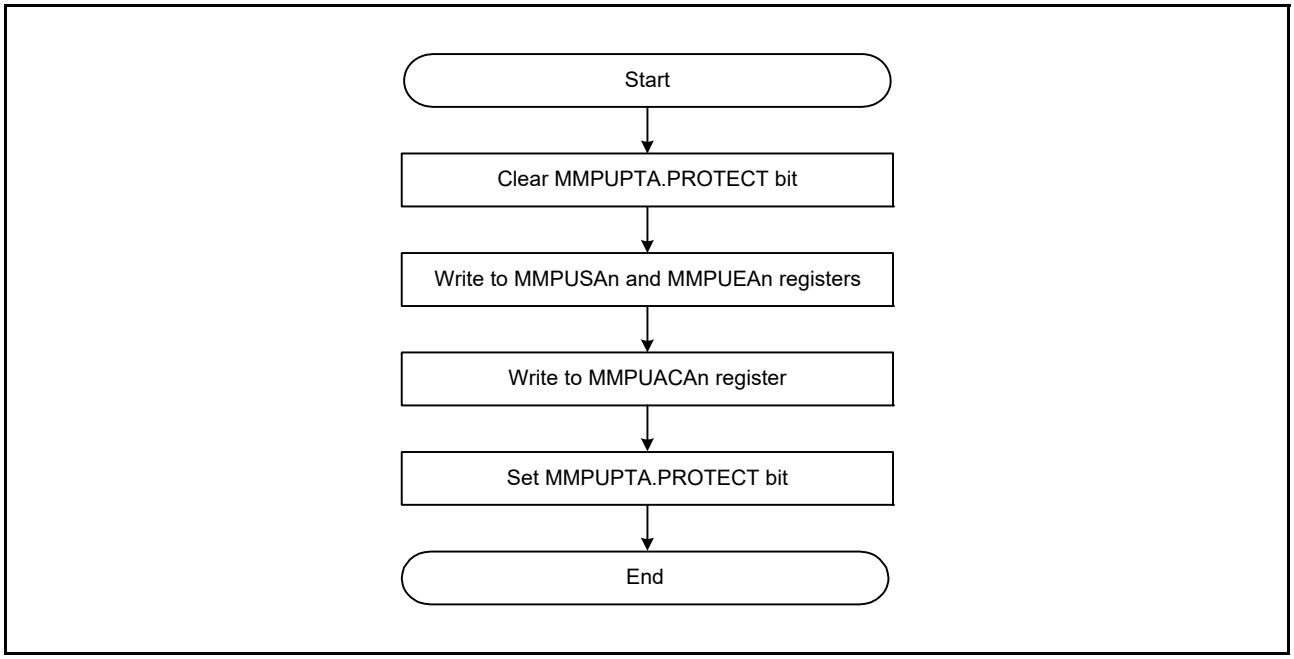


Figure 15.8 Register setting flow for region addition

15.4.2.2 Protecting the registers

To protect the registers related to the bus master MPU, set the PROTECT bit in the MMPUPTA register.

15.4.2.3 Memory protection error

If access to a protected region is detected, the bus master MPU generates an error. Set the OAD bit to select whether the error is reported as a non-maskable interrupt or a reset. The non-maskable interrupt status is indicated in ICU.NMISR.BUSMST. For details, see section 13, Interrupt Controller Unit (ICU). The reset status is indicated in

SYSTEM.RSTSR1.BUSMRF. For details, see section 6, Resets.

15.5 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 store the bus error status, error access status, and bus error address in the I/O Registers. For details, see 14.3.3 and 14.3.4 in section 14, Buses. The supported access control information for the individual regions consists of permission to read and to write.

Table 15.7 lists the specifications of the bus slave MPU and Figure 15.9 shows a block diagram.

Table 15.7 Specifications of bus slave MPU

Specifications	Description
Protected bus master	Bus master MPU group A: DMA bus
Protected slave functions	• Memory bus 1: Code flash memory • Memory bus 4: SRAM0 • Internal peripheral bus 1: Connected to peripheral modules related system control • Internal peripheral bus 3: Connected to peripheral modules (CAC, ELC, I/O Ports, POEG, RTC, WDT, IWDT, IIC, CAN, ADC16, SDADC24, DOC, GPT, SCI, SPI, and CRC) • Internal peripheral bus 5: Connected to peripheral modules (KINT, AGT, USBFS, DAC12, DAC8, OPAMP, ACMPHS, ACMPLP, and CTSU) • Internal peripheral bus 7: Connected to Secure IPs (AES and TRNG) • Internal peripheral bus 9: Flash memory (in P/E) and data flash memory.
Access-control settings for individual regions	Permission to read and write
Operation on error detection	Reset or non-maskable interrupt
Protection of register	Register can be protected from illegal writes

The bus slave MPU is located on each bus slave side and controls the permission or protection of access from each bus master to each bus slave.

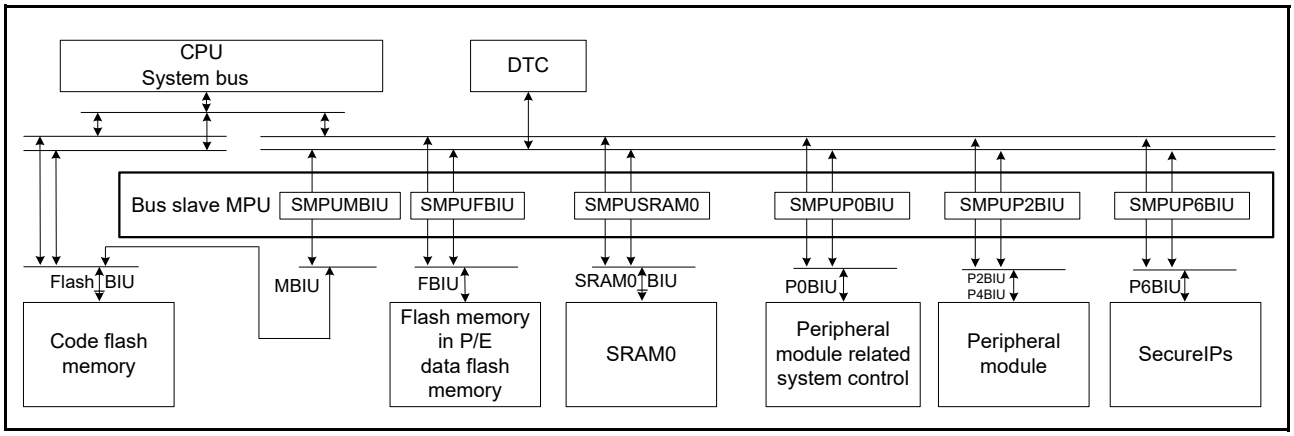


Figure 15.9 Bus slave MPU block diagram

15.5.1 Register Descriptions

Note: Bus access must be stopped before writing to MPU registers.

15.5.1.1 Access Control Register for Memory Bus 1 (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 MPU Group A Read Protection	0: Memory protection read for master MPU group A disabled 1: Memory protection read for master MPU group A enabled.	R/W
b3	WPGRPA	Master MPU Group A Write Protection	0: Memory protection write for master MPU group A disabled 1: Memory protection write for master MPU group A enabled.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RPGRPA bit (Master MPU Group A Read Protection)

The RPGRPA bit enables or disables memory protection for reads by master MPU group A on memory bus 1.

WPGRPA bit (Master MPU Group A Write Protection)

The WPGRPA bit enables or disables memory protection for writes by master MPU group A on memory bus 1.

15.5.1.2 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 MPU Group A Read protection	0: Memory protection for master MPU group A read disabled 1: Memory protection for master MPU group A read enabled.	R/W
b3	WPGRPA	Master MPU Group A Write protection	0: Memory protection for master MPU group A write disabled 1: Memory protection for master MPU group A write enabled.	R/W
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 reads by CPU on internal peripheral bus 9.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for writes by CPU on internal peripheral bus 9.

RPGRPA bit (Master MPU Group A Read protection)

The RPGRPA bit enables or disables memory protection for reads by master MPU group A on internal peripheral bus 9.

WPGRPA bit (Master MPU Group A Write protection)

The WPGRPA bit enables or disables memory protection for writes by master MPU group A on internal peripheral bus 9.

15.5.1.3 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 MPU Group A Read protection	0: Memory protection for master MPU group A read disabled 1: Memory protection for master MPU group A read enabled.	R/W
b3	WPGRPA	Master MPU Group A Write protection	0: Memory protection for master MPU group A write disabled 1: Memory protection for master MPU group A write enabled.	R/W
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 reads by CPU on memory bus 4.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for writes by CPU on memory bus 4.

RPGRPA bit (Master MPU Group A Read protection)

The RPGRPA bit enables or disables memory protection for reads by master MPU group A on memory bus 4.

WPGRPA bit (Master MPU Group A Write protection)

The WPGRPA bit enables or disables memory protection for writes by master MPU group A on memory bus 4.

15.5.1.4 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
b2	RPGRPA	Master MPU Group A Read protection	0: Memory protection for master MPU group A read disabled 1: Memory protection for master MPU group A read enabled.	R/W
b3	WPGRPA	Master MPU Group A Write protection	0: Memory protection for master MPU group A write disabled 1: Memory protection for master MPU group A write enabled.	R/W

Bit	Symbol	Bit name	Description	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 reads by CPU on internal peripheral bus 1.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for writes by CPU on internal peripheral bus 1.

RPGRPA bit (Master MPU Group A Read protection)

The RPGRPA bit enables or disables memory protection for reads by master MPU group A on internal peripheral bus 1.

WPGRPA bit (Master MPU Group A Write protection)

The WPGRPA bit enables or disables memory protection for writes by master MPU group A on internal peripheral bus 1.

Note: The read/write protection by SMPUP0BIU register is not controlled in the MTB I/O register area (4001 9000h to 4001 9FFFh).

15.5.1.5 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 MPU Group A Read protection	0: Memory protection for master MPU group A read disabled 1: Memory protection for master MPU group A read enabled.	R/W
b3	WPGRPA	Master MPU Group A Write protection	0: Memory protection for master MPU group A write disabled 1: Memory protection for master MPU group A write enabled.	R/W
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 reads by CPU on internal peripheral buses 3 and 5.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for writes by CPU on internal peripheral buses 3 and 5.

RPGRPA bit (Master MPU Group A Read protection)

The RPGRPA bit enables or disables memory protection for reads by master MPU group A on internal peripheral buses 3 and 5.

WPGRPA bit (Master MPU Group A Write protection)

The WPGRPA bit enables or disables memory protection for writes by master MPU group A on internal peripheral buses 3 and 5.

15.5.1.6 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: CPU read of memory protection disabled 1: CPU read of memory protection enabled.	R/W
b1	WPCPU	CPU Write protection	0: CPU write of memory protection disabled 1: CPU write of memory protection enabled.	R/W
b2	RPGRPA	Master MPU Group A Read protection	0: Master MPU Group A read of memory protection disabled 1: Master MPU Group A read of memory protection enabled.	R/W
b3	WPGRPA	Master MPU Group A Write protection	0: Master MPU Group A write of memory protection disabled 1: Master MPU Group A write of memory protection enabled.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read protection)

The RPCPU bit enables or disables memory protection for reads by CPU internal peripheral bus 7.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for writes by CPU internal peripheral bus 7.

RPGRPA bit (Master MPU Group A Read protection)

The RPGRPA bit enables or disables memory protection for reads by master MPU group A internal peripheral bus 7.

WPGRPA bit (Master MPU Group A Write protection)

The WPGRPA bit enables or disables memory protection for writes by master MPU group A internal peripheral bus 7.

15.5.1.7 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 MPU register writes are permitted 1: All bus slave MPU 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 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 writing to the OAD bit, simultaneously write A5h to the KEY[7:0] bits using halfword access.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected. SMPUCTL.PROTECT controls the following registers:

- SMPUMBIU
- SMPUFBIU
- SMPUSRAM0
- SMPUP0BIU
- SMPUP2BIU
- SMPUP6BIU.

When 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 OAD and PROTECT bits. When writing to the OAD and PROTECT bits, simultaneously write A5h to the KEY[7:0] bits. When other values are written, the OAD and the PROTECT bits are not updated. The KEY[7:0] bits are always read as 00h.

15.5.2 Operation

15.5.2.1 Memory protection

The bus slave MPU monitoring uses access control information that is set for the individual access control registers. If access to a 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).

15.5.2.2 Protecting the registers

Registers related to the bus slave MPU can be protected with the PROTECT bit in the SMPUCTL register.

15.5.2.3 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 a reset.

The non-maskable interrupt status is indicated in ICU.NMISR.BUSSST. For details, see section 13, Interrupt Controller Unit (ICU). The reset status is indicated in SYSTEM.RSTSR1.BUSSRF. For details, see section 6, Resets.

15.6 Security MPU

The MCU incorporates a security MPU with four secure regions that include the code flash, SRAM, and two security functions. The secure regions can be protected from non-secure program accesses. A non-secure program cannot access a protected region.

Table 15.8 lists the specifications of the security MPU and Figure 15.10 shows a block diagram.

Table 15.8 Security MPU specifications (1 of 2)

Specifications	Description
Secure regions	Code flash, SRAM, two security functions
Protected regions	0000 0000h to FFFF FFFFh

Table 15.8 Security MPU specifications (2 of 2)

Specifications	Description
Number of regions	Program Counter: 2 regions Data Access: 4 regions
Address specification for individual regions	Region start and end addresses configurable
Enable/disable setting for memory protection in individual regions	Settings enabled or disabled for the associated region

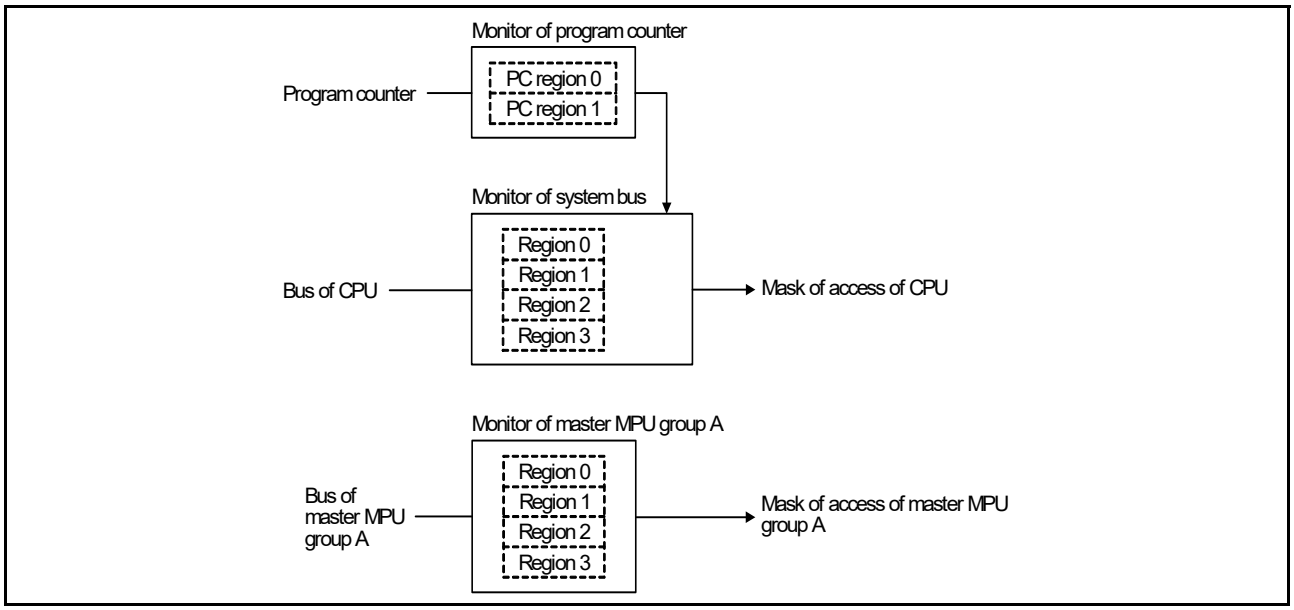


Figure 15.10 Security MPU block diagram

15.6.1 Register Descriptions (Option-Setting Memory)

All security MPU registers are option-setting memory. Option-setting memory refers to a set of registers that are available for selecting the state of the microcontroller after a reset. The option-setting memory is allocated in the code flash.

15.6.1.1 Security MPU Program Counter Start Address Register n (SECMPUPCSn) (n = 0, 1)

Address(es): SECMPUPCS0 0000 0408h/0000 2408h*1, SECMPUPCS1 0000 0410h/0000 2410h*1

	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															

Note 1. The address of these registers will be changed when the boot swap is set.

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 is from 0000 0000h to 000F FFFCh or 1FF0 0000h to 200F FFFCh, excluding reserved areas. When programming to the code flash, the write value should be 0.	R

The SECMPUPCSn and SECMPUPCEn registers specify the security fetch region of the code flash (0000 0000h to 000F FFFFh, excluding reserved areas) or SRAM (1FF0 0000h to 200F FFFFh, excluding 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 SECMPUPCSn register specifies the address where the region starts. Setting of the memory mirror space (0200 0000h to 027F FFFFh) for MMF is prohibited.

15.6.1.2 Security MPU Program Counter End Address Register n (SECMPUPCEn) (n = 0, 1)

Address(es): SECMPUPCE0 0000 040Ch/0000 240Ch*1, SECMPUPCE1 0000 0414h/0000 2414h*1

	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															

Note 1. The address of these registers will be changed when the boot swap is set.

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 is from 0000 0003h to 000F FFFFh or 1FF0 0003h to 200F FFFFh, excluding reserved areas. When programming to the code flash, the write value should be 1.	R

The SECMPUPCEn register specifies the address where the region ends.

15.6.1.3 Security MPU Region 0 Start Address Register (SECMPUS0)

Address(es): SECMPUS0 0000 0418h/0000 2418h*1

	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															

Note 1. The address of these registers will be changed when the boot swap is set.

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 is from 0000 0000h to 000F FFFCh, excluding reserved areas. When programming to the code flash, the write value should be 0.	R
b31 to b24	—	Reserved	These bits are read as 0. When programming to the code flash, the write value should be 0.	R

The SECMPUS0 and SECMPUE0 registers specify the secure region of the code flash (0000 0000h to 000F FFFFh, excluding 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.

The SECMPUS0 register specifies the address where the region starts. Setting of the vector table area is prohibited.

15.6.1.4 Security MPU Region 0 End Address Register (SECMPUE0)

Address(es): SECMPUE0 0000 041Ch/0000 241Ch*1

	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															

Note 1. The address of these registers will be changed when the boot swap is set.

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. When programming to the code flash, the write value should be 1. The value range is from 0000 0003h to 000F FFFFh, excluding reserved areas.	R
b31 to b24	—	Reserved	These bits are read as 0. When programming to the code flash, the write value should be 0.	R

The SECMPUE0 register specifies the address where the region ends.

15.6.1.5 Security MPU Region 1 Start Address Register (SECMPUS1)

Address(es): SECMPUS1 0000 0420h/0000 2420h*1

	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															

Note 1. The address of these registers will be changed when the boot swap is set.

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 is from 1FF0 0000h to 200F FFFCh, excluding reserved areas. When programming to the code flash, the write value should be 0.	R

The SECMPUS1 and SECMPUE1 registers specify the secure region of the SRAM (1FF0 0000h to 200F FFFFh, excluding reserved areas). The memory space defined in the SECMPUS1 and SECMPUE1 registers can only be accessed from the secure program set up in the SECMPUPCSn and SECMPUPCEn registers.

The SECMPUS1 register specifies the address where the region starts. Setting of the stack area and the vector table are prohibited.

15.6.1.6 Security MPU Region 1 End Address Register (SECMPUE1)

Address(es): SECMPUE1 0000 0424h/0000 2424h*1

	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															

Note 1. The address of these registers will be changed when the boot swap is set.

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 is from 1FF0 0003h to 200F FFFCh, excluding reserved areas. When programming to the code flash, the write value should be 1.	R

The SECMPUE1 register specifies the address where the region ends.

15.6.1.7 Security MPU Region 2 Start Address Register (SECMPUS2)

Address(es): SECMPUS2 0000 0428h/0000 2428h*1

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SECMPUS2[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
	SECMPUS2[15:0]															
Value after reset:	The value set by user															

Note 1. The address of these registers will be changed when the boot swap is set.

Bit	Symbol	Bit name	Description	R/W
b31 to b0	SECMPUS2[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 is from 400C 0000h to 400D FFFCh and 4010 0000h to 407F FFFCh. When programming to the code flash, the write value should be 0.	R

The SECMPUS2 and SECMPUE2 registers specify the secure region of the security functions (400C 0000h to 400D FFFFh and 4010 0000h 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.

The SECMPUS2 register specifies the address where the region starts.

15.6.1.8 Security MPU Region 2 End Address Register (SECMPUE2)

Address(es): SECMPUE2 0000 042Ch/0000 242Ch*1

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SECMPUE2[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
	SECMPUE2[15:0]															
Value after reset:	The value set by user															

Note 1. The address of these registers will be changed when the boot swap is set.

Bit	Symbol	Bit name	Description	R/W
b31 to b0	SECMPUE2[31:0]	Region End Address	Address that determines where the region ends. The lower 2 bits are read as 1. When programming to the code flash, the write value should be 1. The value range is from 400C 0003h to 400D FFFFh and 4010 0003h to 407F FFFFh.	R

The SECMPUE2 register specifies the address where the region ends.

15.6.1.9 Security MPU Region 3 Start Address Register (SECMPUS3)

Address(es): SECMPUS3 0000 0430h/0000 2430h*1

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SECMPUS3[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
SECMPUS3[15:0]															
Value after reset:	The value set by user														

Note 1. The address of these registers will be changed when the boot swap is set.

Bit	Symbol	Bit name	Description	R/W
b31 to b0	SECMPUS3[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 is from 400C 0000h to 400D FFFCh and 4010 0000h to 407F FFFCh. When programming to the code flash, the write value should be 0.	R

The SECMPUS3 and SECMPUE3 registers specify the secure region 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.

The SECMPUS3 register specifies the address where the region starts.

15.6.1.10 Security MPU Region 3 End Address Register (SECMPUE3)

Address(es): SECMPUE3 0000 0434h/0000 2434h*1

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SECMPUE3[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
SECMPUE3[15:0]															
Value after reset:	The value set by user														

Note 1. The address of these registers will be changed when the boot swap is set.

Bit	Symbol	Bit name	Description	R/W
b31 to b0	SECMPUE3[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 is from 400C 0003h to 400D FFFFh and 4010 0003h to 407F FFFFh. When programming to the code flash, the write value should be 1.	R

The SECMPUE3 register specifies the address where the region ends.

15.6.1.11 Security MPU Access Control Register (SECMPUAC)

Address(es): SECMPUAC 0000 0438h/0000 2438h*1

	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			

Note 1. The address of these registers will be changed when the boot swap is set.

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 programming to the code flash, the write value 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 programming to the code flash, the write value 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 15.6.2, 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 SECMPUS0 and SECMPUE0 is secure data.

DIS1 bit (Region 1 Disable)

The DIS1 bit enables or disables the security MPU region 1. If security MPU region 1 is enabled, the SRAM region within the limits set up by SECMPUS1 and SECMPUE1 is secure data.

DIS2 bit (Region 2 Disable)

The DIS2 bit enables or disables the security MPU region 2. If security MPU region 2 is enabled, the region within the limits set up by SECMPUS2 and SECMPUE2 is secure data.

DIS3 bit (Region 3 Disable)

The DIS3 bit enables or disables the security MPU region 3. If security MPU region 3 is enabled, the region within the limits set up by SECMPUS3 and SECMPUE3 is secure data.

DISPC0 bit (PC Region 0 Disable)

The DISPC0 bit enables or disables the security MPU PC region 0. If security MPU PC region 0 is enabled, the code flash 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.

15.6.2 Memory Protection

The security MPU protects the regions (the code flash, the SRAM, and the security functions) from being accessed by non-secure programs. If access to a protected region is detected, the access becomes invalid.

When the security MPU is enabled, DISPC0 or DISPC1 in the Security MPU Access Control Register (SECMPUAC), and DIS0, DIS1, DIS2, or DIS3 in the Security MPU Access Control Register (SECMPUAC) must be set to 0.

When the security MPU is disabled, all bits in DISPC0, DISPC1, DIS0, DIS1, DIS2, and DIS3 in the Security MPU Access Control Register (SECMPUAC) must be set to 1. Other settings in the Security MPU Access Control Register (SECMPUAC) are prohibited.

The security MPU provides access protection in the following conditions:

- Secure data is accessed from a non-secure program
- Secure data is accessed from other than the CPU (DTC)
- Secure data is accessed from the debugger.

Secure data is accessible only from a secure program.

Note:	Secure program:	Code flash or SRAM region within the limits set up by SECMPUPCS0 and SECMPUPCE0 Code flash or SRAM region within the limits set up by SECMPUPCS1 and SECMPUPCE1
	Non-secure program:	All regions 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.

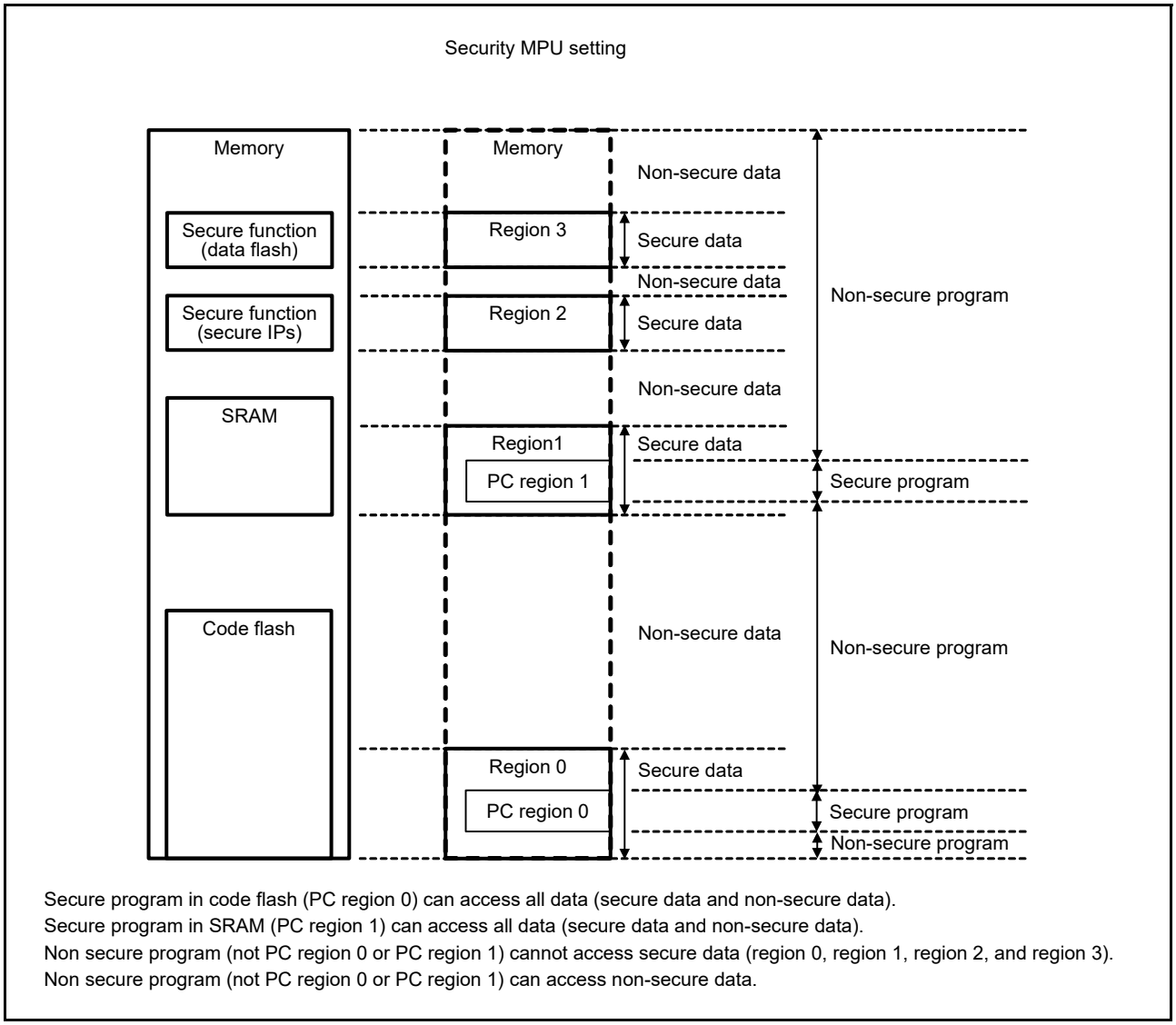


Figure 15.11 Use case of Security MPU

15.6.3 Usage Notes

The protected memory cannot be debugged if the security MPU is enabled. Disable the security MPU when debugging a secure program.

15.7 References

1. *ARM®v8-M Architecture Reference Manual* (ARM DDI 0553B.a)
2. *ARM® Cortex®-M23 Processor Technical Reference Manual* (ARM DDI 0550C)
3. *ARM® Cortex®-M23 Processor User Guide* (ARM DUI 0963B).

16. Data Transfer Controller (DTC)

16.1 Overview

The MCU includes a Data Transfer Controller (DTC) that performs data transfers when activated by an interrupt request. Table 16.1 lists the DTC specifications and Figure 16.1 shows a block diagram.

Table 16.1 DTC specifications

Parameter	Specifications
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 single block transfer. 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 can be set to either execute when the counter is 0, or always execute.
Transfer space	• 4 GB area from 0000 0000h to FFFF FFFFh, excluding reserved areas
Data transfer units	• Single data unit: 1 byte (8 bits), 1 halfword (16 bits), 1 word (32 bits) • Single block size: 1 to 256 data units.
CPU interrupt source	• An interrupt request can be generated to the CPU on a DTC activation interrupt • An interrupt request can be generated to the CPU after a single data transfer • An interrupt request can be generated to the CPU after a data transfer of a specified volume.
Event link function	An event link request is generated after one data transfer (for block, after one block transfer)
Read skip	Read of transfer information can be skipped
Write-back skip	When the transfer source or destination address is specified as fixed, a write-back of transfer information can be skipped
Module-stop function	Module-stop state can be set to reduce power consumption

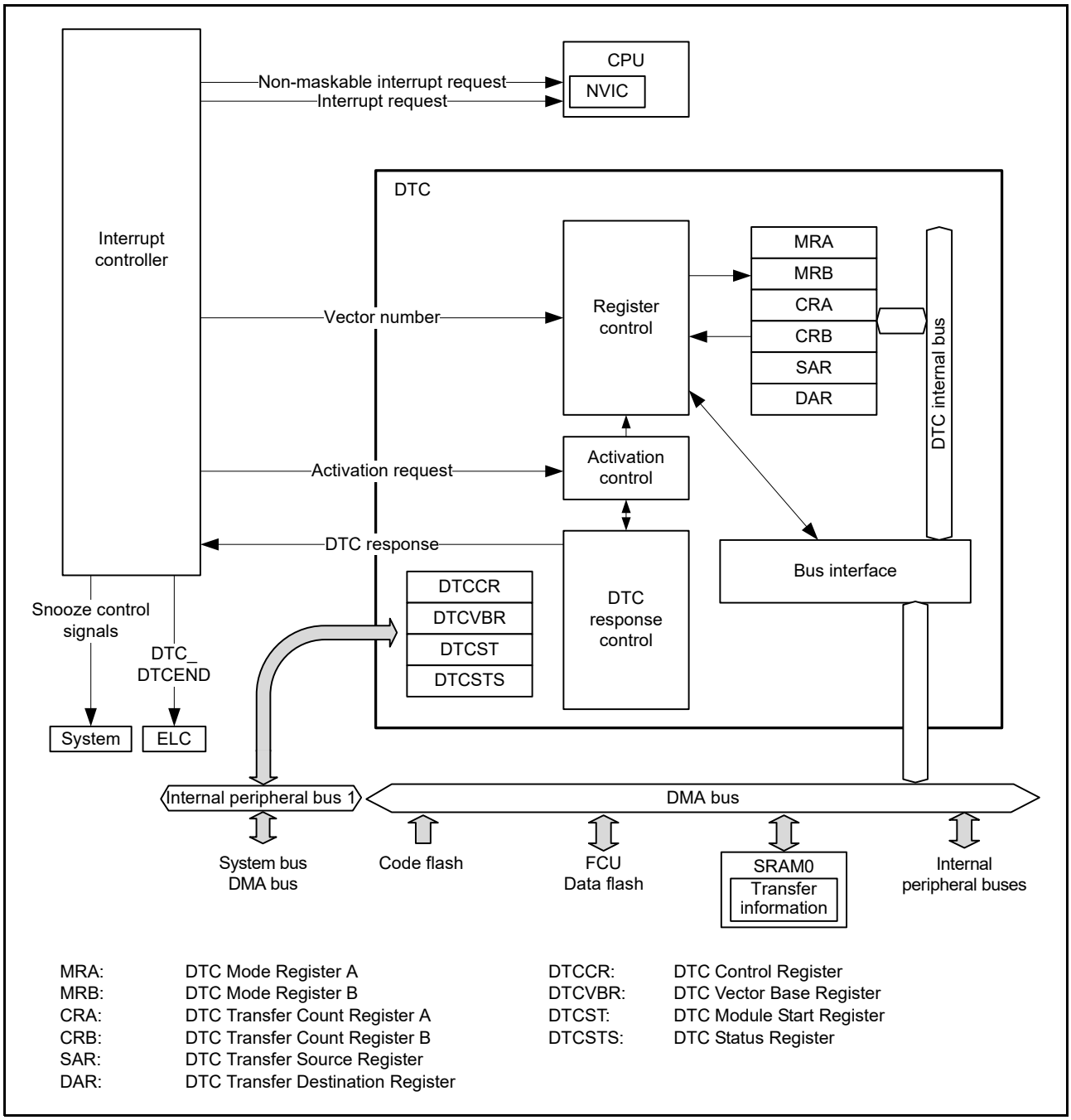


Figure 16.1 DTC block diagram

See Overview in section 13, Interrupt Controller Unit (ICU) for the connections between the DTC and NVIC in the CPU.

16.2 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.

16.2.1 DTC Mode Register A (MRA)

Address(es): (inaccessible directly from the CPU. See section 16.3.1)

	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] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b. 1 1: SAR value is decremented after data transfer: -1 when SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b.	—
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 register cannot be accessed directly from the CPU. However, the CPU can access SRAM area (transfer information (n) start address + 03h) and DTC transfer it automatically to and from the MRA register. See section 16.3.1, Allocating Transfer Information and DTC Vector Table.

16.2.2 DTC Mode Register B (MRB)

Address(es): (inaccessible directly from the CPU. See section 16.3.1)

	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] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b. 1 1: DAR value is decremented after data transfer: -1 when MRA.SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b.	—
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: Chain transfer is continuous 1: Chain transfer occurs only when the transfer counter changes from 1 to 0 or 1 to CRAH.	—
b7	CHNE	DTC Chain Transfer Enable	0: Chain transfer is disabled 1: Chain transfer is enabled.	—

The MRB register cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 02h) and DTC transfer it automatically from and to the MRB register. See section 16.3.1, Allocating Transfer Information and DTC Vector Table.

DTS bit (DTC Transfer Mode Select)

The DTS bit specifies whether the transfer source or destination is the repeat or block area in repeat or block transfer mode.

CHNS bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition. When CHNE is 0, the CHNS setting is ignored. For details on the conditions for chain transfer, see Table 16.3, Chain transfer conditions.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the activation source flag is not cleared, and an interrupt request to the CPU is not generated.

CHNE bit (DTC Chain Transfer Enable)

The CHNE bit enables chain transfer. The chain transfer condition is selected by the CHNS bit. For details on chain transfer, see section 16.4.6, Chain Transfer.

16.2.3 DTC Transfer Source Register (SAR)

Address(es): (inaccessible directly from the CPU. See section 16.3.1)

	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 DTC transfer it automatically from and to the SAR register. See section 16.3.1, Allocating Transfer Information and DTC Vector Table.

Note: Misalignment is prohibited for DTC transfers.

16.2.4 DTC Transfer Destination Register (DAR)

Address(es): (inaccessible directly from the CPU. See section 16.3.1)

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

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

x: Undefined

The DAR sets the transfer destination start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 08h) and DTC transfer it automatically from and to the DAR register. See section 16.3.1, Allocating Transfer Information and DTC Vector Table. Misalignment is prohibited for DTC transfers. Bit 0 must be 0 when MRA.SZ[1:0] = 01b, and bit 1 and bit 0 must be 0 when MRA.SZ[1:0] = 10b.

Note: Misalignment is prohibited for DTC transfers.

16.2.5 DTC Transfer Count Register A (CRA)

Address(es): (inaccessible directly from the CPU. See section 16.3.1)

- 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 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 DTC transfer it automatically to and from the CRA register. See section 16.3.1, Allocating Transfer Information and DTC Vector Table.

(1) Normal transfer mode (MRA.MD[1:0] = 00b)

In normal transfer mode, CRA functions as a 16-bit transfer counter. The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively. The CRA value is decremented (-1) on each data transfer.

(2) Repeat transfer mode (MRA.MD[1:0] = 01b)

In repeat transfer mode, the CRAH register holds the transfer count and the CRAL register functions as an 8-bit transfer counter. The transfer count is 1, 255, and 256 when the set value is 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] = 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.

16.2.6 DTC Transfer Count Register B (CRB)

Address(es): (inaccessible directly from the CPU. See section 16.3.1)

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

x: Undefined

The CRB sets the block transfer count for block transfer mode. The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively. The CRB value is decremented (-1) when the final data of a single block size is transferred. When normal transfer mode or repeat transfer mode is selected, this register is not used and the set value is ignored.

The CRB cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0Ch) and DTC transfer it automatically to and from the CRB register. See section 16.3.1, Allocating Transfer Information and DTC Vector Table.

16.2.7 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, 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.

16.2.8 DTC Vector Base Register (DTCVBR)

Address(es): DTC.DTCVBR 4000 5404h

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

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

Bit	Bit name	Description	R/W
b31 to b0	DTC Vector Base Address	Set the DTC vector base address (lower 10 bits should be 0)	R/W

The DTCVBR sets the base address for calculating the DTC vector table address, which can be set in the range of 0000 0000h to FFFF FFFFh (4 GB) in 1-KB units.

16.2.9 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 this bit is set to 0, transfer requests are no longer accepted. If this bit is set to 0 during a data transfer, the accepted transfer request is active until processing completes.

DTCST must be set to 0 before transitioning to one of the following state or mode:

- Module-stop state
- Software Standby mode without Snooze mode transition.

For details on these transitions, see section 16.10, Module-Stop Function, and section 11, Low Power Modes.

16.2.10 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, these bits indicate the vector number associated with the activation source for the transfer. The value read from the VECN[7:0] bits is valid if the ACT flag is 1, indicating a DTC transfer in progress, and invalid if the ACT flag is 0, indicating no DTC transfer is in progress.

ACT flag (DTC Active Flag)

The ACT flag indicates the state of the DTC transfer operation.

[Setting condition]

- When the DTC is activated by a transfer request.

[Clearing condition]

- When transfer by the DTC, in response to a transfer request, is complete.

16.3 Activation Sources

The DTC is activated by an interrupt request. Setting the ICU.IELSRn.DTCE bit to 1 enables activation of the DTC by the associated interrupt. The selector output *n* number set in ICU.IELSRn is defined as the interrupt vector number, where n = 0 to 31. For an enabled interrupt, the specific DTC interrupt source associated with each interrupt vector number *n* is selected in ICU.IELSRn.IELS[7:0] where n = 0 to 31, as listed in Table 13.4, Event table in section 13, Interrupt Controller Unit (ICU). For activation by software, see section 17.2.2, 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 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.

16.3.1 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 16.2 shows the relationship between the DTC vector table and transfer information. Figure 16.3 shows the allocation of transfer information in the SRAM area.

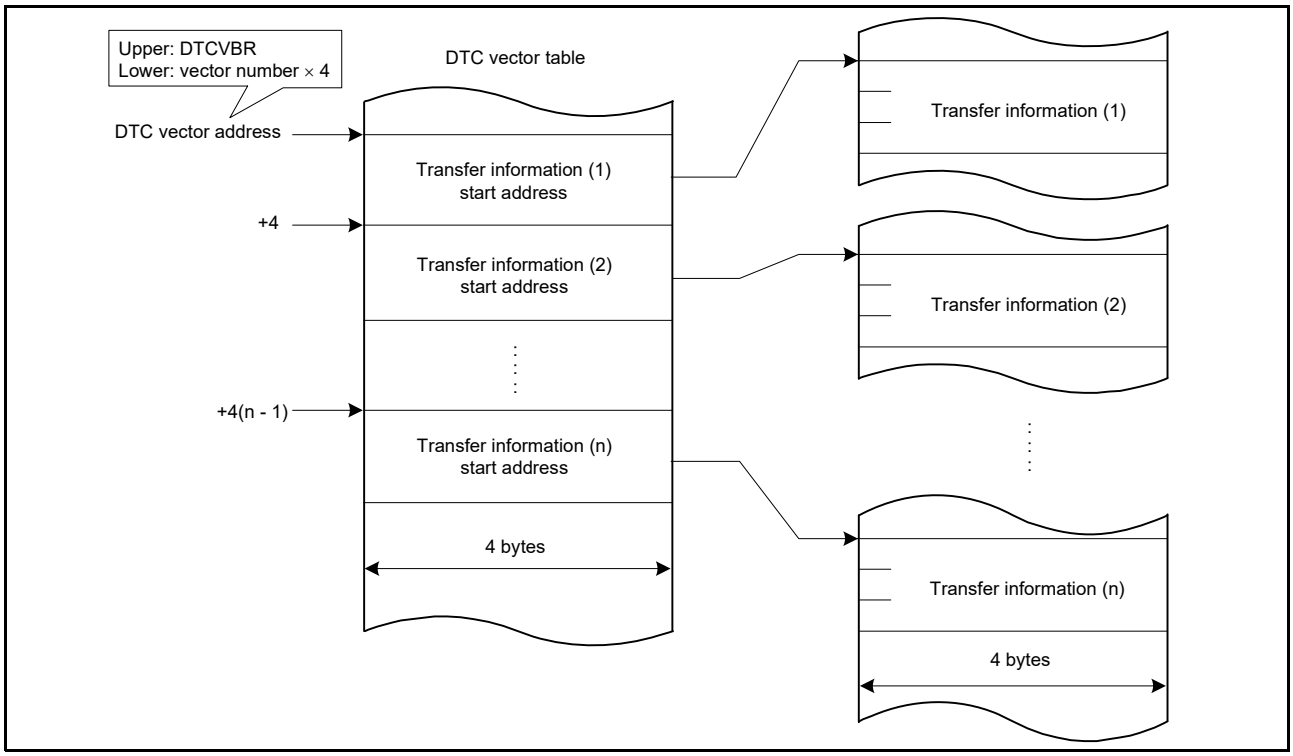


Figure 16.2 DTC vector table and transfer information

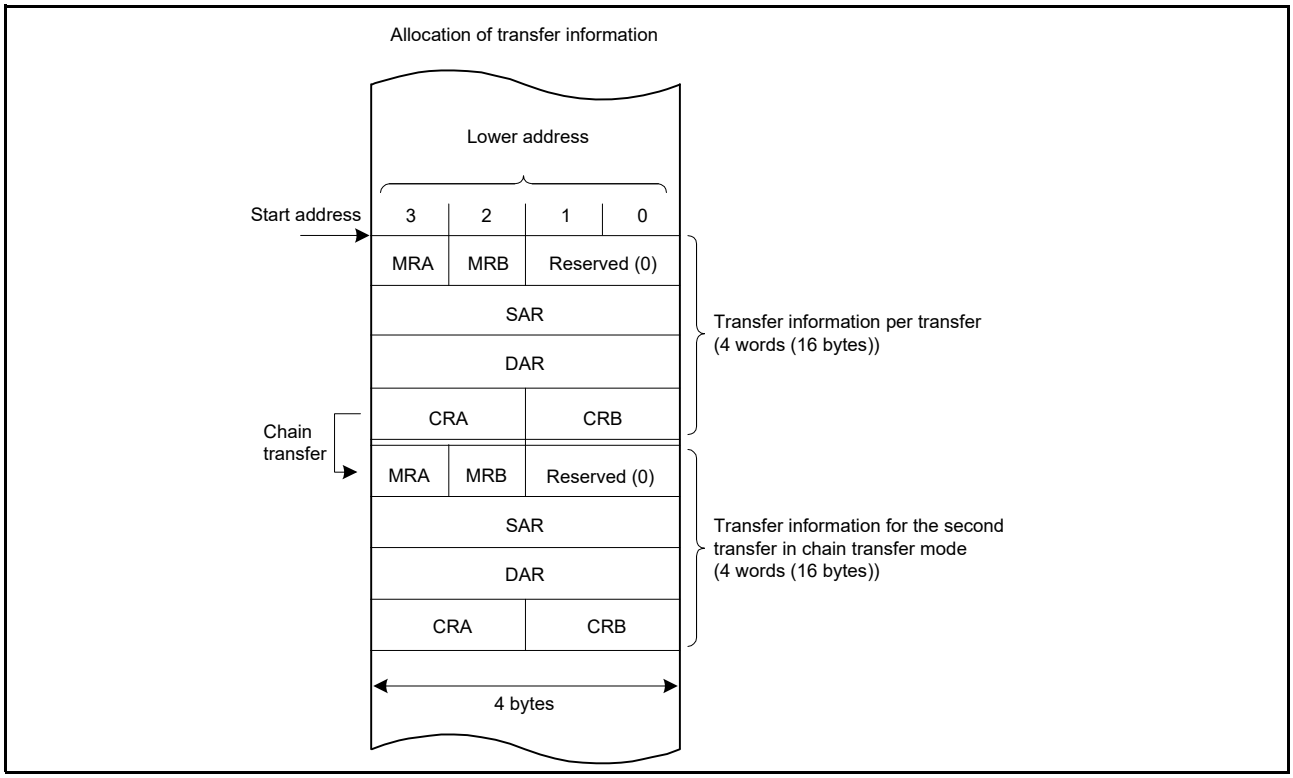


Figure 16.3 Allocation of transfer information in the SRAM area

16.4 Operation

The DTC transfers data according to the transfer information. Storage of the transfer information in the SRAM area is required before a DTC operation. When the DTC is activated, it reads the DTC vector associated with the vector number. The DTC reads the transfer information from the transfer information store address referenced by the DTC vector and transfers the data. After the data transfer, the DTC writes back the transfer information. Storing the transfer information in the SRAM area allows data transfer of any number of channels.

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 16.2 describes the DTC transfer modes.

Table 16.2 DTC transfer modes

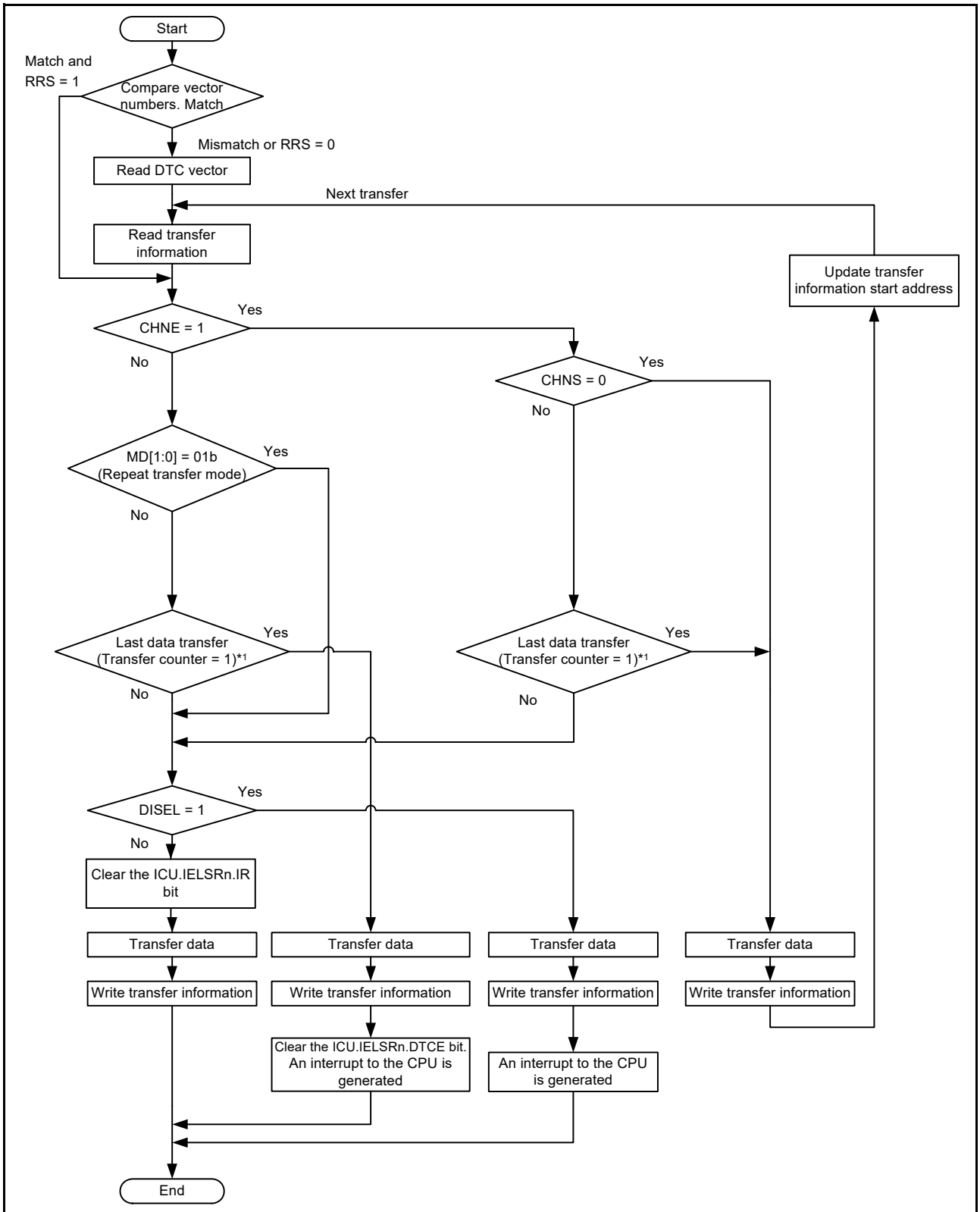
Transfer mode	Data size transferred on single transfer request	Increment or decrement of memory address	Settable transfer count
Normal transfer mode	1 byte (8 bit), 1 halfword (16 bit), 1 word (32 bit)	Incremented or decremented by 1, 2, or 4 or address fixed	1 to 65536
Repeat transfer mode*1	1 byte (8 bit), 1 halfword (16 bit), 1 word (32 bit)	Incremented or decremented by 1, 2, or 4 or address fixed	1 to 256*3
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), or 1 to 256 words (4 to 1024 bytes))	Incremented or decremented by 1, 2, or 4 or address fixed	1 to 65536

Note 1. Set the transfer source or transfer destination as the repeat area.
Note 2. Set the transfer source or transfer destination as the block area.

Note 3. After a data transfer of the specified count, the initial state is restored and operation restarts.

Setting the MRB.CHNE bit to 1 allows multiple transfers or chain transfer on a single activation source. It also enables a chain transfer when the specified data transfer is complete.

Figure 16.4 shows the operation flow of the DTC. Table 16.3 lists the chain transfer conditions. The combination of control information for the second and subsequent transfers are omitted in this table.



Note 1. Counter value before starting data transfer.

Figure 16.4 DTC operation flow

Table 16.3 Chain transfer conditions

First transfer				Second transfer*3				
CHNE bit	**CHNS bit**	**DISEL bit**	**Transfer counter*1,*2**	**CHNE bit**	**CHNS bit**	**DISEL bit**	**Transfer counter*1,*2**	**DTC transfer**
0	—	0	Other than (1 → 0)	—	—	—	—	Ends after the first transfer
0	—	0	(1 → 0)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	0	Other than (1 → *)	—	—	—	—	Ends after the first transfer
1	1	—	(1 → *)	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	1	Other than (1 → *)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU

Note 1. The transfer counters used depend on the transfer modes as follows:
Normal transfer mode — CRA register
Repeat transfer mode — CRAL register
Block transfer mode — CRB register

Note 2. On completion of a data transfer, the counters operate as follows:
1 → 0 in normal and block transfer modes
1 → CRAH in repeat transfer mode
(1 → *) in the table indicates both of these two operations, depending on the mode.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The conditions for the combination of the second transfer and CHNE = 1 is omitted.

16.4.1 Transfer Information Read Skip Function

Reading of vector addresses and transfer information can be skipped by setting the DTCCR.RRS bit. When a DTC activation request is generated, the current DTC vector number is compared with the DTC vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, the DTC data transfer is performed without reading the vector address and transfer information. However, when the previous transfer is a chain transfer, the vector address and transfer information are read. Additionally, when the transfer counter (CRA register) becomes 0 during the previous normal transfer, and when the transfer counter (CRB register) becomes 0 during the previous block transfer, transfer information is read regardless of the RRS bit. Figure 16.12 shows an example of a transfer information read skip.

To update the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, then set the RRS bit to 1. The stored vector number is discarded by setting the RRS bit to 0. The updated DTC vector table and transfer information are read in the next activation process.

16.4.2 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 16.4 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 16.4 Transfer information write-back skip conditions and applicable registers

MRA.SM[1:0] bits		MRB.DM[1:0] bits		SAR register	DAR register
b3	b2	b3	b2		
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		

16.4.3 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 to 1 to 65536. Transfer source addresses and transfer destination addresses can be independently set to increment, decrement, or fixed. This mode enables an interrupt request to the CPU to be generated at the end of a specified-count transfer.

Table 16.5 lists register functions in normal transfer mode, and Figure 16.5 shows the memory map of normal transfer mode.

Table 16.5 Register functions in normal transfer mode

Register	Description	Value written back by writing transfer information
SAR	Transfer source address	Increment, decrement, or fixed*1
DAR	Transfer destination address	Increment, decrement, fixed*1
CRA	Transfer counter A	CRA - 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped in address-fixed mode.

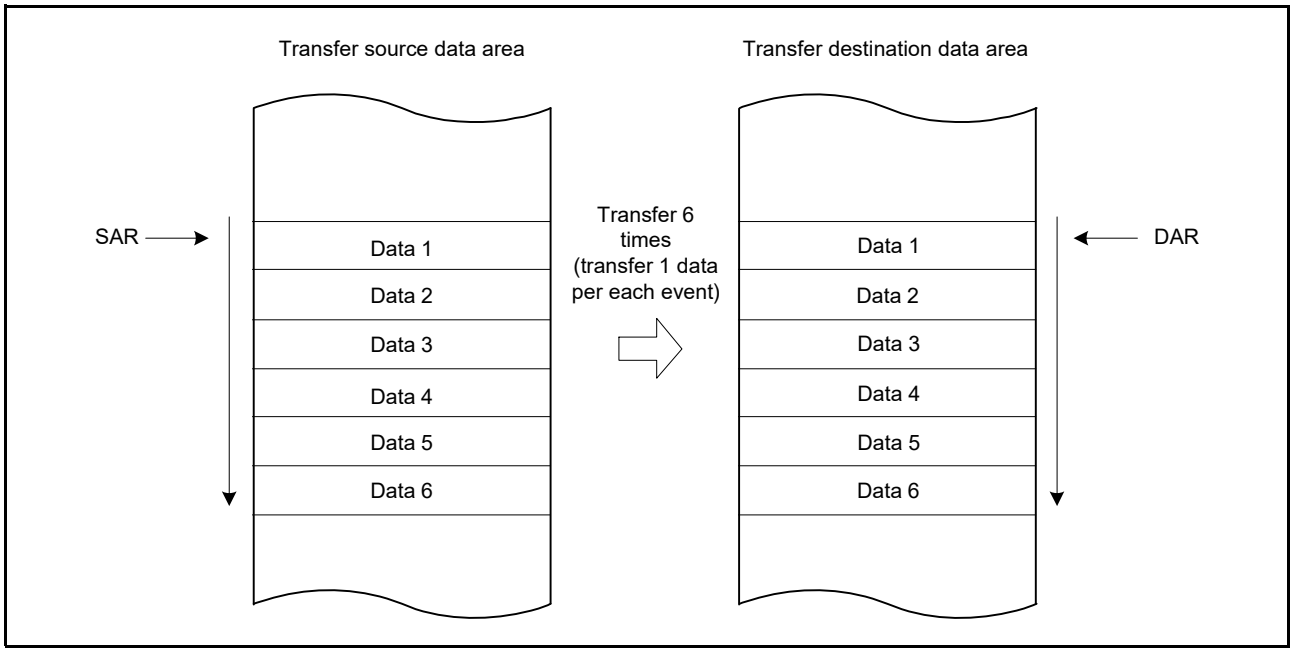


Figure 16.5 Memory map of normal transfer mode (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRA=0006h)

16.4.4 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 completes.

Table 16.6 lists the register functions in repeat transfer mode, and Figure 16.6 shows the memory map of repeat transfer mode.

Table 16.6 Register functions in repeat transfer mode

Register	Description	Value written back by writing transfer information	
		When CRAL is not 1	When CRAL is 1
SAR	Transfer source address	Increment, decrement, fixed*1	• When the MRB.DTS bit is 0 Increment, decrement, or fixed*1 • When the MRB.DTS bit is 1 SAR register initial value
DAR	Transfer destination address	Increment, decrement, or fixed*1	• When the MRB.DTS bit is 0 DAR register initial value • When the MRB.DTS bit is 1 Increment, decrement, or fixed*1
CRAH	Retains transfer counter	CRAH	CRAH
CRAL	Transfer counter A	CRAL - 1	CRAH
CRB	Transfer counter B	Not updated	Not updated

Note 1. Write-back is skipped in address-fixed mode.

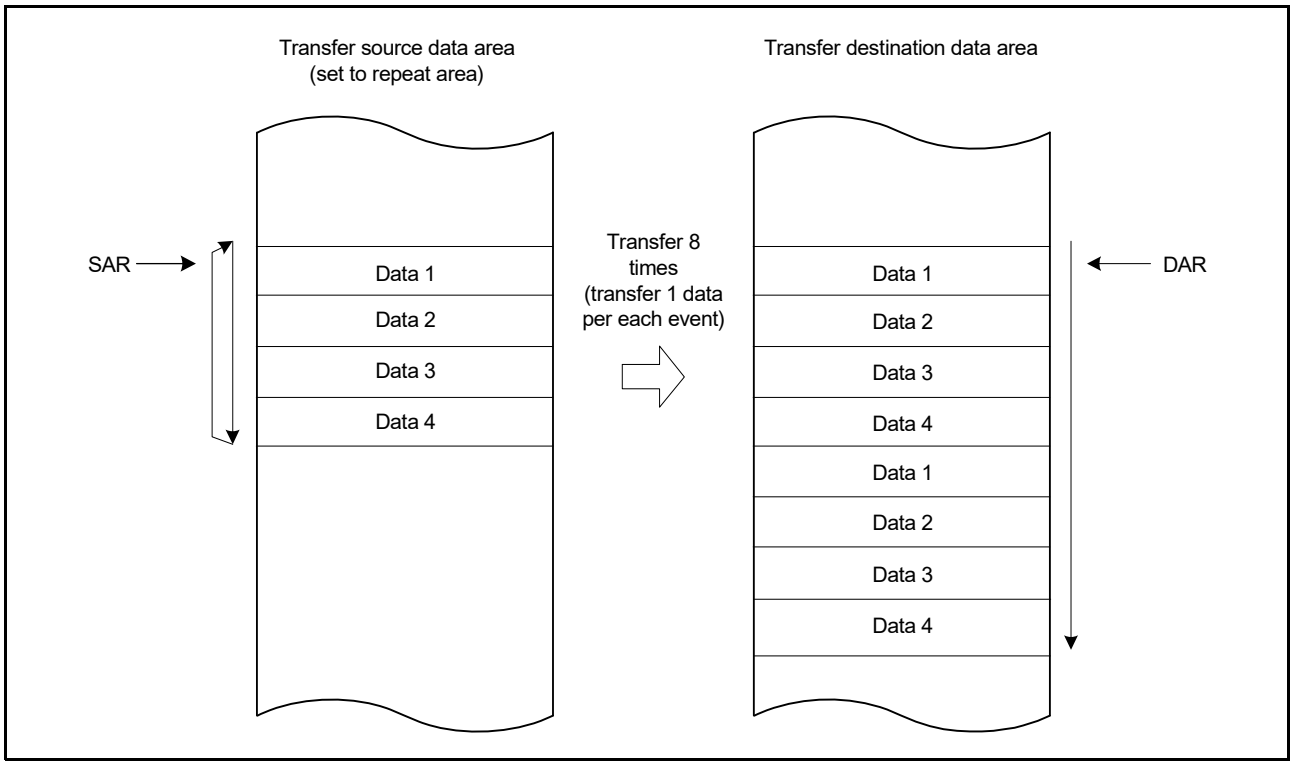


Figure 16.6 Memory map of repeat transfer mode when transfer source is a repeat area (MRA.SM[1:0] = 10b, MRB.DM[1:0] =10b, CRAH=04h)

16.4.5 Block Transfer Mode

The block transfer mode allows single-block data transfer on a single activation source. Transfer source or transfer destination for the block area must be specified in the MRB.DTS bit. The block size can be set from 1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), or 1 to 256 words (4 to 1024 bytes). When transfer of the specified block completes, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS = 1 or the DAR register when the DTS = 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.

The transfer count (block count) can be set from 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of the specified-count block transfer.

Table 16.7 lists the register functions in block transfer mode, and Figure 16.7 shows the memory map for block transfer mode.

Table 16.7 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.

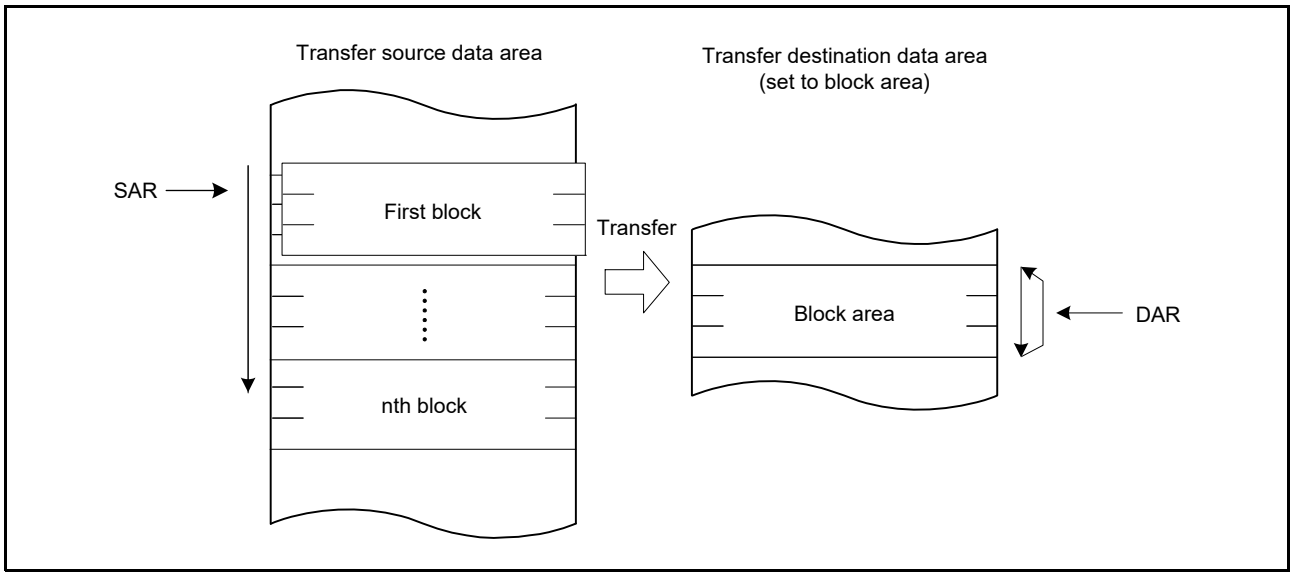


Figure 16.7 Memory map of block transfer mode

16.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single activation source. If the MRB.CHNE is set to 1 and CHNS to 0, an interrupt request to the CPU is not generated on completion of the specified number of rounds of transfer or by setting the MRB.DISEL bit to 1. An interrupt request is sent to the CPU each time DTC data transfer is performed. Data transfer has no effect on the ICU.IELSRn.IR bit of the activation source.

The SAR, DAR, CRA, CRB, MRA, and MRB registers can be set independently of each other to define the data transfer. Figure 16.8 shows a chain transfer operation.

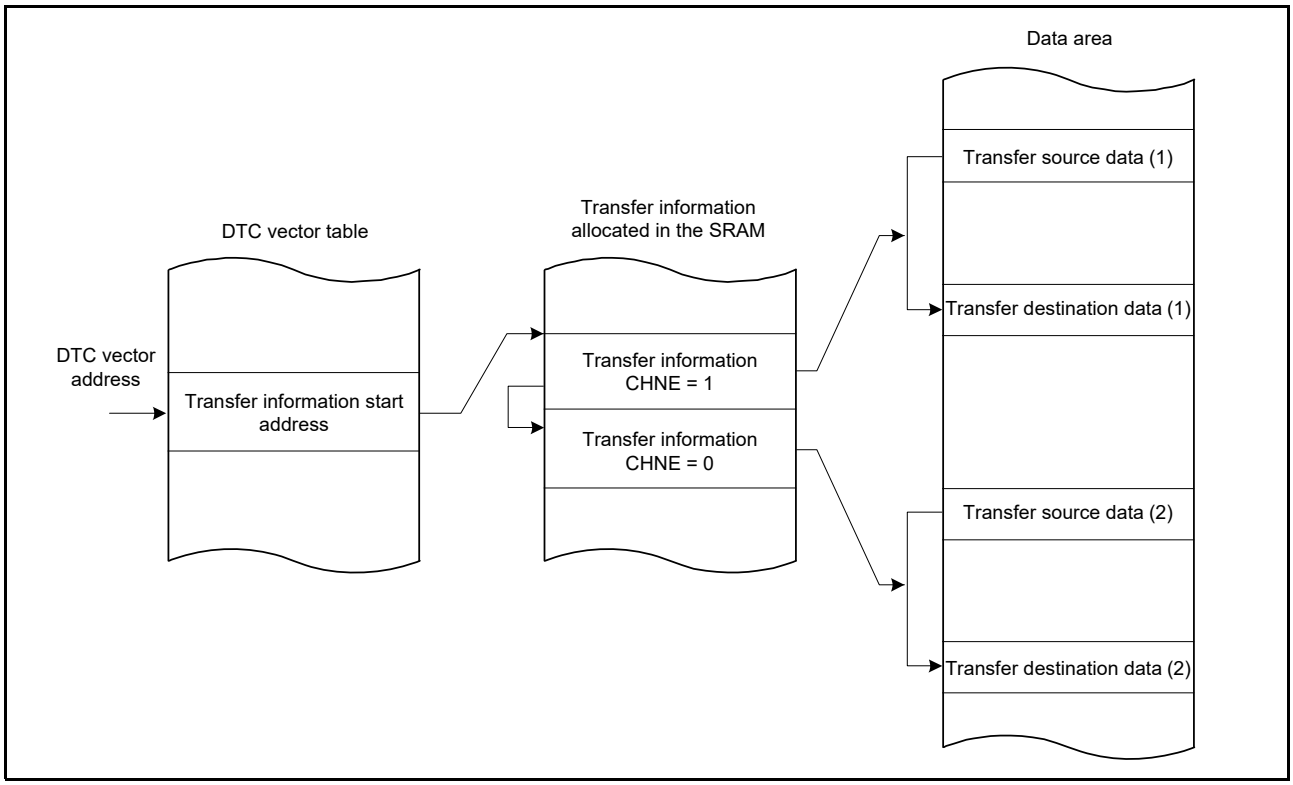


Figure 16.8 Chain transfer operation

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 16.3, Chain transfer conditions.

16.4.7 Operation Timing

Figure 16.9 to Figure 16.12 are timing diagrams that show the minimum number of execution cycles.

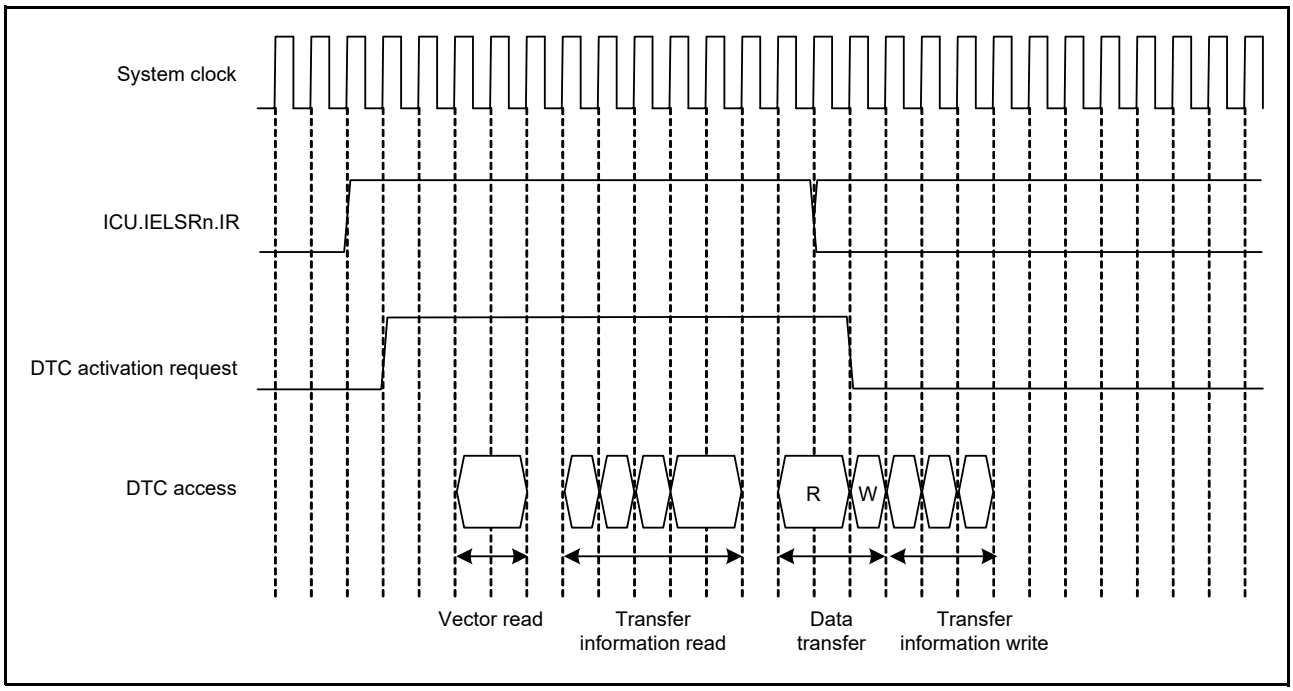


Figure 16.9 Example 1 of DTC operation timing in normal transfer and repeat transfer modes

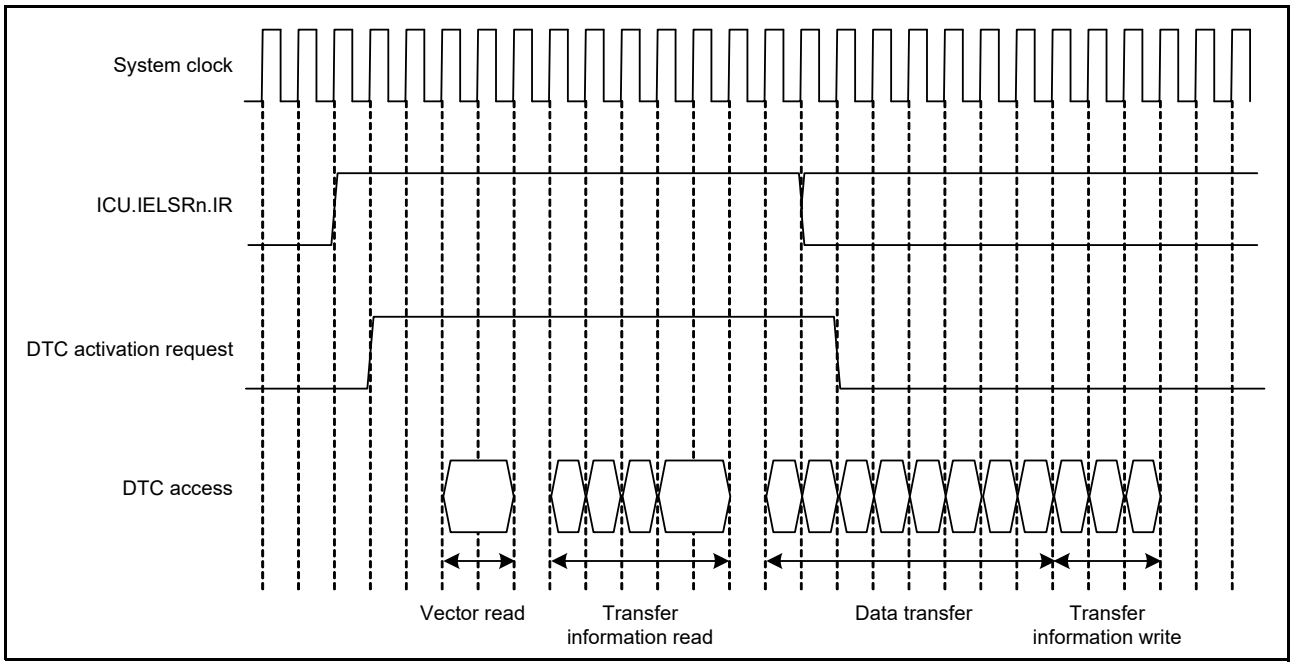


Figure 16.10 Example 2 of DTC operation timing in block transfer mode when the block size = 4

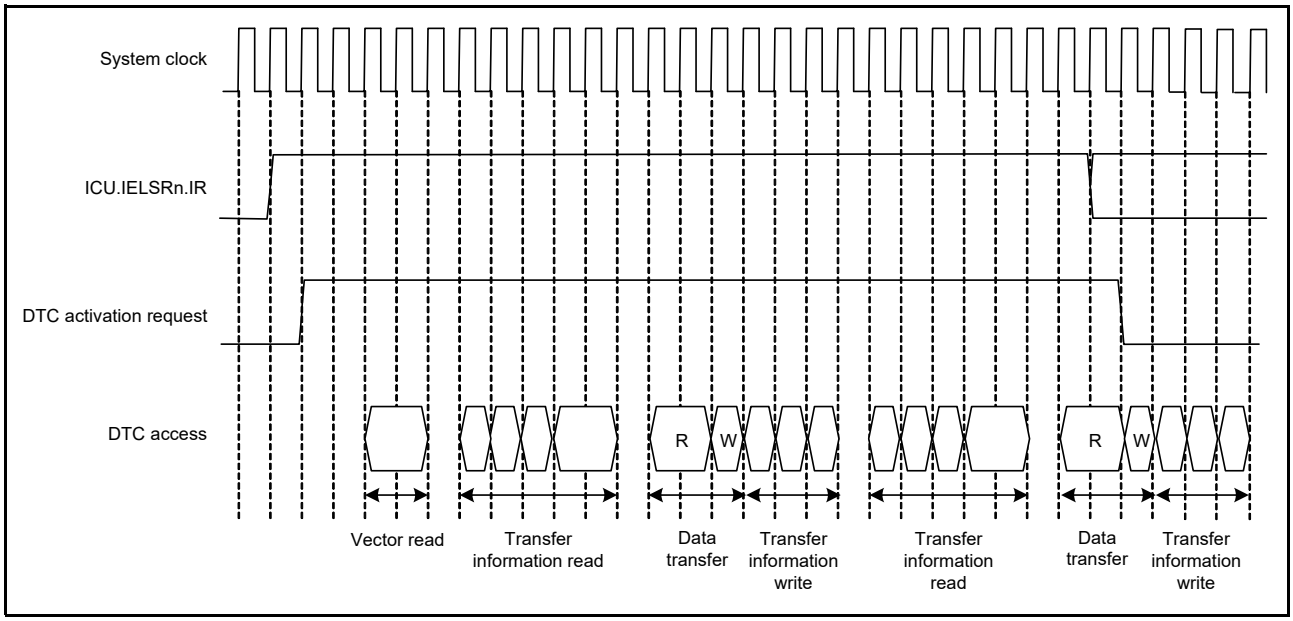
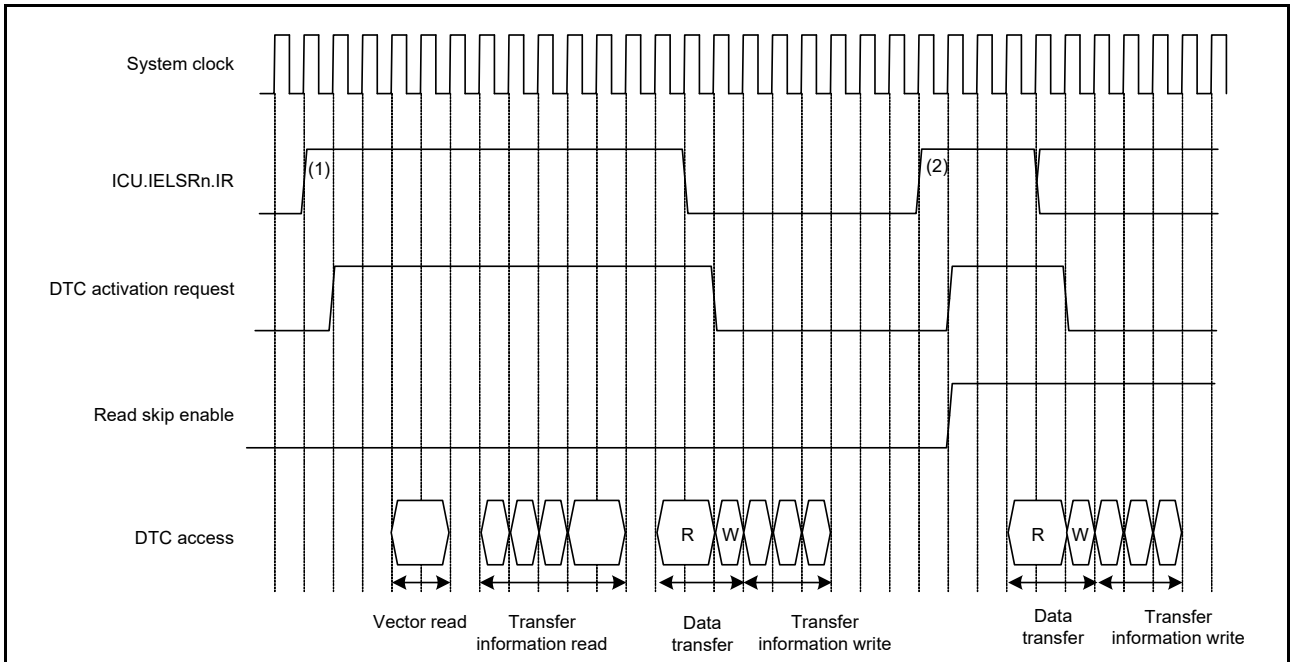


Figure 16.11 Example 3 of DTC operation timing for chain transfer



Note: When activation sources (vector numbers) of (1) and (2) are the same and the RRS = 1, the transfer information read for request (2) is skipped.

Figure 16.12 Example of operation when a transfer information read is skipped with the vector, transfer information, and transfer destination data on the SRAM, and the transfer source data on the peripheral module

16.4.8 Execution Cycles of DTC

Table 16.8 lists the execution cycles of single data transfer of the DTC. For the order of the execution states, see section 16.4.7, Operation Timing.

Table 16.8 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 x Ci + 1	0*1	3 x Ci + 1*2	2 x Ci + 1*3	Ci*4	Cr + 1	Cw + 1	2	0*1
Repeat								Cr + 1	Cw + 1		
Block*5								P x Cr	P x Cw		

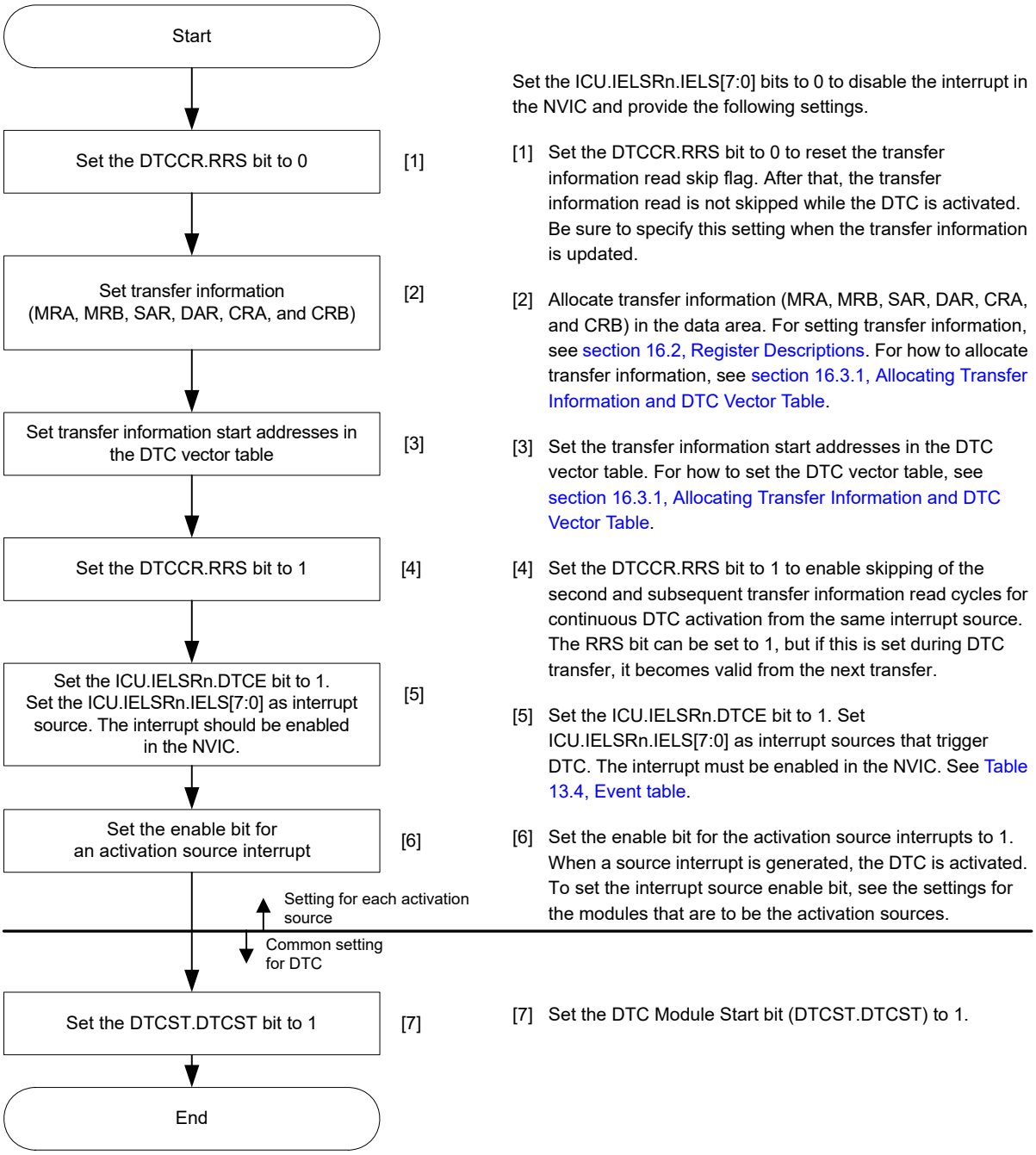
Note 1. When transfer information read is skipped.
Note 2. When neither SAR nor DAR is set to address-fixed mode.
Note 3. When SAR or DAR is set to address-fixed mode.
Note 4. When SAR and DAR are set to address-fixed mode.
Note 5. When the block size is 2 or more. If the block size is 1, the cycle number for normal transfer applies.
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 and section 43, Flash Memory.
The frequency ratio of the system clock and peripheral clock is also taken into consideration.
The DTC response time is the time from when the DTC activation source is detected until DTC transfer starts.
Table 16.8 does not include the time until DTC data transfer starts after the DTC activation source becomes active.

16.4.9 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 14, Buses.

16.5 DTC Setting Procedure

Before using the DTC, set the DTC Vector Base Register (DTCVBR). Figure 16.13 shows the procedure for setting the DTC.



Set the ICU.IELSRn.IELS[7:0] bits 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, the transfer information read is not skipped while the DTC is activated. Be sure to specify this setting when the transfer information is updated.

[2] Allocate transfer information (MRA, MRB, SAR, DAR, CRA, and CRB) in the data area. For setting transfer information, see section 16.2, Register Descriptions. For how to allocate transfer information, see section 16.3.1, Allocating Transfer Information and DTC Vector Table.

[3] Set the transfer information start addresses in the DTC vector table. For how to set the DTC vector table, see section 16.3.1, Allocating Transfer Information and DTC Vector Table.

[4] Set the DTCCR.RRS bit to 1 to enable skipping of the second and subsequent transfer information read cycles for continuous DTC activation from the same interrupt source. The RRS bit can be set to 1, but if this is set during DTC transfer, it becomes valid from the next transfer.

[5] Set the ICU.IELSRn.DTCE bit to 1. Set ICU.IELSRn.IELS[7:0] as interrupt sources that trigger DTC. The interrupt must be enabled in the NVIC. See Table 13.4, Event table.

[6] Set the enable bit for the activation source interrupts to 1. When a source interrupt is generated, the DTC is activated. To set the interrupt source enable bit, see the settings for the modules that are to be the activation sources.

[7] Set the 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 16.13 DTC setting procedure

16.6 Examples of DTC Usage

16.6.1 Normal Transfer

This section provides an example of DTC usage and its application when receiving 128 bytes of data from an SCI.

(1) Transfer information settings

In the MRA register, select a fixed source address (MRA.SM[1:0] = 00b), normal transfer mode (MRA.MD[1:0] = 00b), and byte-sized transfer (MRA.SZ[1:0] = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] = 10b) and single data transfer by a single interrupt (MRB.CHNE = 0 and MRB.DISEL = 0). The MRB.DTS bit can be set to any value. Set the RDR register address of the SCI in the SAR register, the start address of the SRAM area for data storage in the DAR register, and 128 (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[7:0] as the SCI interrupt. The interrupt must be enabled in the NVIC. Set the DTCST.DTCST bit to 1.

(4) SCI settings

Enable the 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 allow the CPU to accept receive error interrupts.

(5) DTC transfer

Each time a reception of 1 byte by the SCI is complete, 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.

16.6.2 Chain transfer

This section provides an example of chain transfer by the DTC and describes its use in the output of pulses by the General PWM Timer (GPT). You can use chain transfer to transfer PWM timer compare data and change the period of the PWM timer for the GPT.

For the first of the chain transfers, normal transfer mode is specified for transfer to the GPTm.GTCCRC register (m = 320, 16H1 to 16H3, 164 to 166). For the second transfer, normal transfer mode is specified for transfer to the GPTm.GTCCRE registers. For the third transfer of the chained transfer, normal transfer mode for transfer to the GPTm.GTPBR registers is specified. This is because clearing of the activation source and generation of an interrupt on completion of the specified number of transfers are restricted to the third of the chain transfers, that is, transfer while MRB.CHNE = 0.

The following example shows how to use the counter overflow interrupt with the 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] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up chain transfer (MRB.CHNE = 1 and MRB.CHNS = 0).

4. Set the SAR 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 for transfer to the GPT320.GTCCRE register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up chain transfer (MRB.CHNE = 1, MRB.CHNS = 0).
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the 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] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up single data transfer per interrupt (MRB.CHNE = 0, MRB.DISEL = 0). The MRB.DTS bit can be set to any value.
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT320.GTPBR registers.
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] bits to 74 (4Ah) 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.

16.6.3 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 information in the first data transfer 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 16.14 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 = 0000h (65536) times.
   c. MRB.CHNE = 1 (chain transfer is enabled).
   d. MRB.CHNS = 1 (chain transfer is performed only when the transfer counter is 0).
   e. MRB.DISEL = 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 65536 times of the transfer destination address for the first data transfer in different area such as the flash. For example, when setting the input buffer to 20 0000h to 21 FFFFh, prepare 21h and 20h.
3. For the second data transfer:
   a. Set the repeat transfer mode (with the source as the repeat area) to reset the transfer destination address of the first data transfer.
   b. Specify the upper 8 bits of the DAR register in the first transfer information area for the transfer destination.
   c. Set the MRB.CHNE = 0 (chain transfer is disabled).
   d. Set the MRB.DISEL = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
   e. 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 65536 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 become 0000h.
5. In succession, the first data transfer is performed by an interrupt 65536 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 become 0000h.
6. Steps 4 and 5 are repeated indefinitely. Because the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

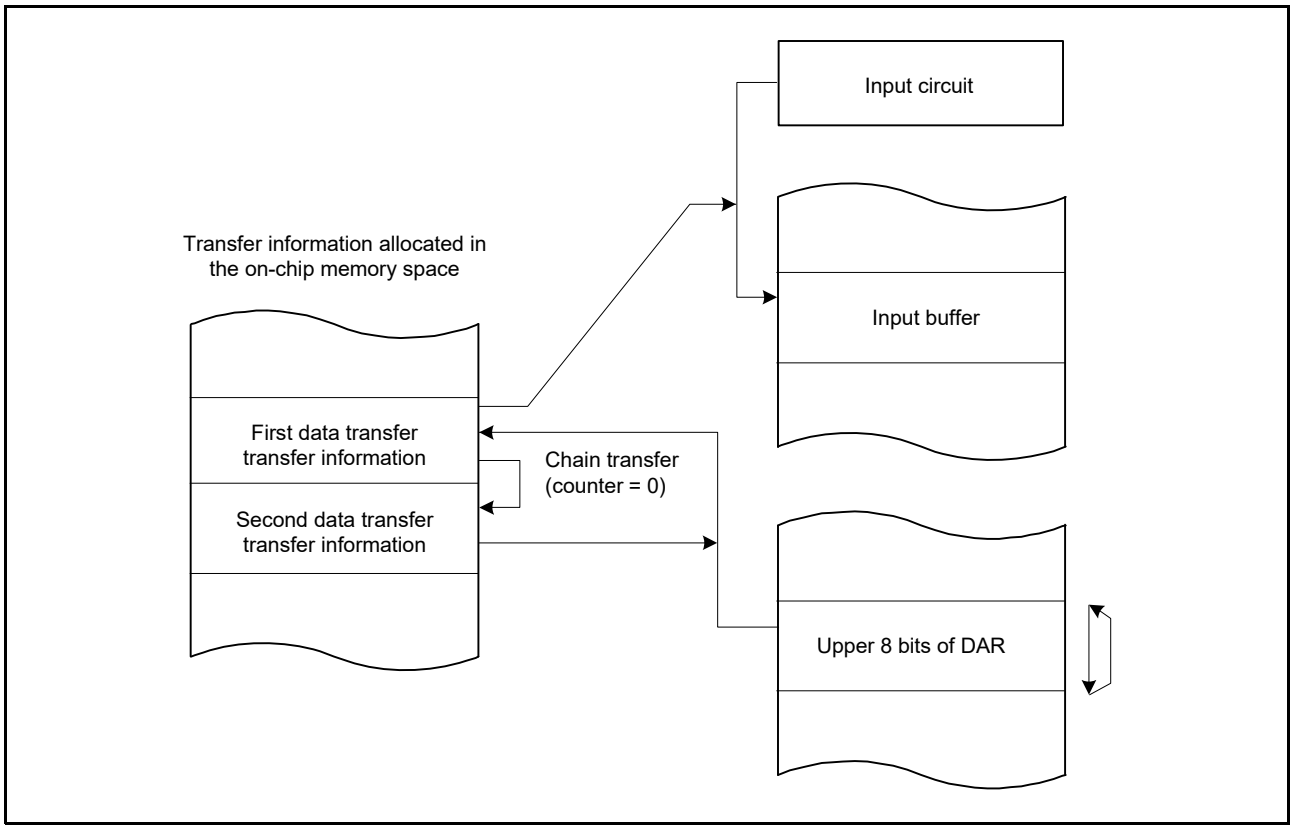


Figure 16.14 Chain transfer when counter = 0

16.7 Interrupt Sources

When the DTC completes data transfer of the specified count or when data transfer with MRB.DISEL set to 1 is complete, a DTC activation source generates an interrupt to the CPU. Interrupts to the CPU are controlled according to the settings in the NVIC and ICU.IELSRn.IELS[7:0] bits. See section 13, 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.

16.8 Event Link

The DTC can produce an event link request on completion of one transfer request.

16.9 Snooze Control Interface

To return to Software Standby mode from Snooze mode through the DTC, set the SYSTEM.SNZEDCR.DTCZRED or SYSTEM.SNZEDCR.DTCNZRED bit to 1. See section 11.8.3, Returning 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.

16.10 Module-Stop Function

Before transitioning to the module-stop state or Software Standby mode without a Snooze mode transition, set the DTCST.DTCST bit to 0, then perform the operations described in the following sections. The DTC is available in Snooze mode by setting the SYSTEM.SNZCR.SNZDTCEN bit to 1. See section 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. When 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, Transitioning 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, Transitioning to Snooze Mode. DTC operation in Snooze mode can be selected in the SYSTEM.SNZCR.SNZDTCEN bit. If DTC operation is enabled in Snooze mode, before transitioning to Software Standby mode, set the DTCST.DTCST bit to 1. To return to Software Standby mode through DTC, set SYSTEM.SNZEDCR.DTCZRED or SYSTEM.SNZEDCR.DTCNZRED to 1. See section 11.8.3, Returning 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 a Snooze mode transition, set the DTCST.DTCST bit to 1 again.

To use a request that is generated in Software Standby mode as an interrupt request to the CPU but not as a DTC activation request, specify the CPU as the interrupt request destination as described in section 13.4.2, Selecting Interrupt Request Destinations, then execute the WFI instruction. If DTC operation is enabled in Snooze mode, do not use the module-stop function of the DTC.

16.11 Usage Notes

16.11.1 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.

17. Event Link Controller (ELC)

17.1 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 17.1 lists the ELC specifications and Figure 17.1 shows a block diagram.

Table 17.1 ELC specifications

Parameter	Specifications
Event link function	111 types of event signals can be directly connected to modules. The ELC generates the ELC event signal, and events that activate the DTC.
Module-stop function	Module-stop state can be set

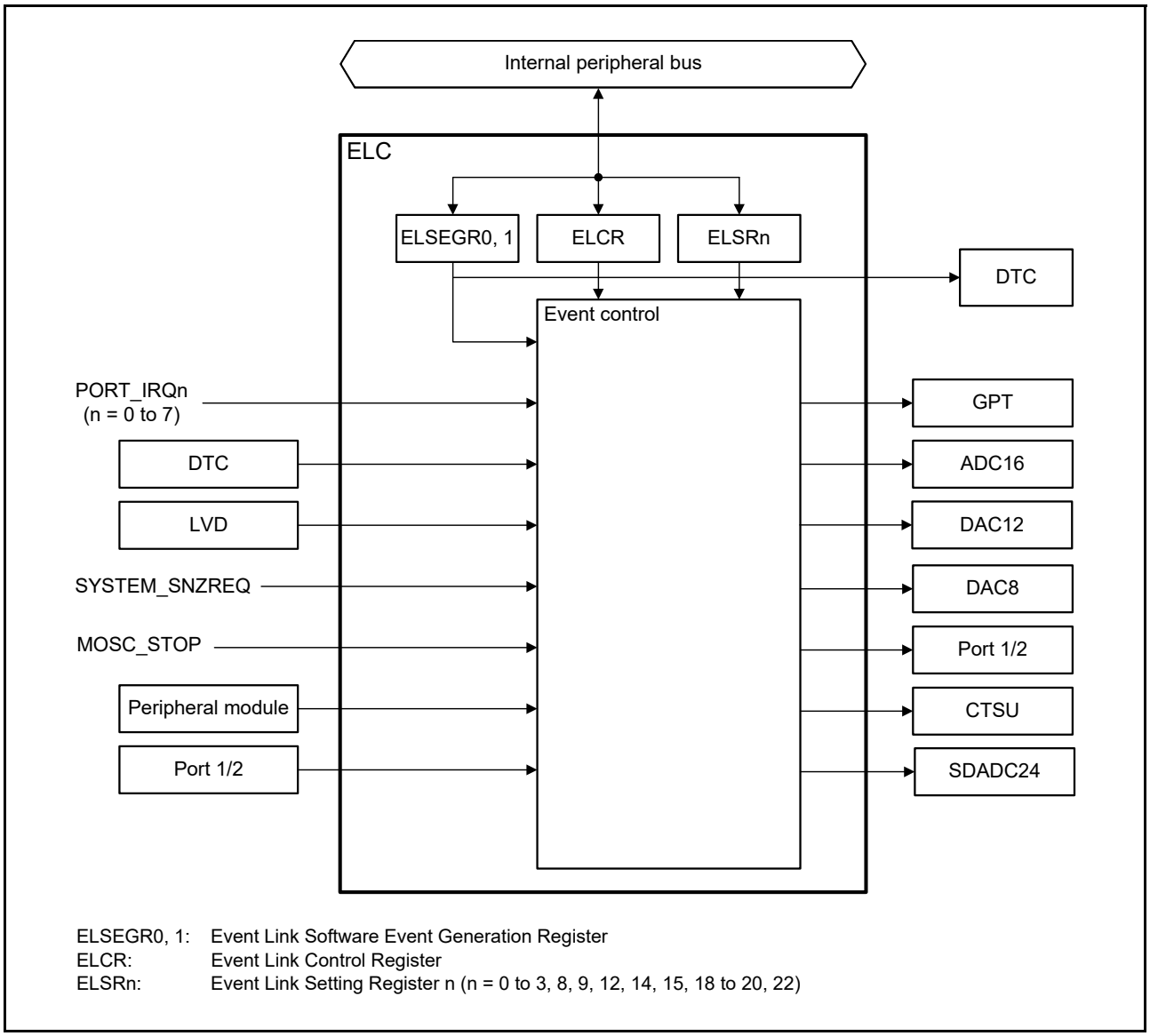


Figure 17.1 ELC block diagram

17.2 Register Descriptions

17.2.1 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.

17.2.2 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.

17.2.3 Event Link Setting Register n (ELSRn) (n = 0 to 3, 8, 9, 12, 14, 15, 18 to 20, 22)

Address(es): ELC.ELSR0 4004 1010h, ELC.ELSR1 4004 1014h, ELC.ELSR2 4004 1018h, ELC.ELSR3 4004 101Ch, ELC.ELSR8 4004 1030h, ELC.ELSR9 4004 1034h, ELC.ELSR12 4004 1040h, ELC.ELSR14 4004 1048h, ELC.ELSR15 4004 104Ch, ELC.ELSR18 4004 1058h, ELC.ELSR19 4004 105Ch, ELC.ELSR20 4004 1060h, ELC.ELSR22 4004 1068h

	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 10001010: 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 17.2 shows the associations between the ELSRn registers and the peripheral modules. Table 17.3 shows the association between the event signal names set in the ELSRn registers and the signal numbers.

Table 17.2 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
ELSR8	ADC16A	ELC_AD00
ELSR9	ADC16B	ELC_AD01
ELSR12	DAC12	ELC_DAC12
ELSR14	PORT 1	ELC_PORT1
ELSR15	PORT 2	ELC_PORT2
ELSR18	CTSU	ELC_CTSU
ELSR19	DA80	ELC_DA80
ELSR20	DA81	ELC_DA81
ELSR22	SDADC24	ELC_SDADC

Table 17.3 Association between event signal names set in ELSRn.ELS[7:0] bits and signal numbers (1 of 4)

Event number	Interrupt request source	Name	Description
01h	Port	PORT_IRQ0*1	External pin interrupt 0
02h		PORT_IRQ1*1	External pin interrupt 1
03h		PORT_IRQ2*1	External pin interrupt 2
04h		PORT_IRQ3*1	External pin interrupt 3
05h		PORT_IRQ4*1	External pin interrupt 4
06h		PORT_IRQ5*1	External pin interrupt 5
07h		PORT_IRQ6*1	External pin interrupt 6
08h		PORT_IRQ7*1	External pin interrupt 7
0Ah	DTC	DTC_DTCEND*3	DTC transfer end

Table 17.3 Association between event signal names set in ELSRn.ELS[7:0] bits and signal numbers (2 of 4)

Event number	Interrupt request source	Name	Description
0Dh	LVD	LVD_LVD1	Voltage monitor 1 interrupt
0Eh		LVD_LVD2	Voltage monitor 2 interrupt
0Fh	MOSC	MOSC_STOP	Main clock oscillation stop
10h	Low Power mode	SYSTEM_SNZREQ*2, *3	Snooze entry
11h	AGT0	AGT0_AGTI	AGT interrupt
12h		AGT0_AGTCMAI	Compare match A
13h		AGT0_AGTCMBI	Compare match B
14h	AGT1	AGT1_AGTI	AGT interrupt
15h		AGT1_AGTCMAI	Compare match A
16h		AGT1_AGTCMBI	Compare match B
17h	IWDT	IWDT_NMIUNDF	IWDT underflow
18h	WDT	WDT_NMIUNDF	WDT underflow
1Ah	RTC	RTC_PRD	Periodic interrupt
1Ch	ADC16	ADC160_ADI	A/D scan end interrupt
20h		ADC160_WCMPM*3	Compare match
21h		ADC160_WCMPUM*3	Compare mismatch
22h	ACMPHS	ACMP_HS0*1	High-speed analog comparator interrupt 0
23h	ACMPLP	ACMP_LP0*1	Low-power analog comparator interrupt 0
24h		ACMP_LP1*1	Low-power analog comparator interrupt 1
27h	IIC0	IIC0_RXI	Receive data full
28h		IIC0_TXI	Transmit data empty
29h		IIC0_TEI	Transmit end
2Ah		IIC0_EEI	Transfer error
2Ch	IIC1	IIC1_RXI	Receive data full
2Dh		IIC1_TXI	Transmit data empty
2Eh		IIC1_TEI	Transmit end
2Fh		IIC1_EEI	Transfer error
34h	DOC	DOC_DOPCI*3	Data operation circuit interrupt
3Dh	I/O Ports	IOPORT_GROUP1	Port 1 event
3Eh		IOPORT_GROUP2	Port 2 event
3Fh	ELC	ELC_SWEVT0	Software event 0
40h		ELC_SWEVT1	Software event 1
43h	SDADC24	SDADC_ADI	A/D conversion end interrupt
44h		SDADC_SCANEND	A/D automatic scan completion interrupt
46h	GPT320	GPT0_CCMPA	Compare match A
47h		GPT0_CCMPB	Compare match B
48h		GPT0_CMPC	Compare match C
49h		GPT0_CMPD	Compare match D
4Ah		GPT0_OVF	Overflow
4Bh		GPT0_UDF	Underflow

Table 17.3 Association between event signal names set in ELSRn.ELS[7:0] bits and signal numbers (3 of 4)

Event number	Interrupt request source	Name	Description
4Ch	GPT161	GPT1_CCMPA	Compare match A
4Dh		GPT1_CCMPB	Compare match B
4Eh		GPT1_CMPC	Compare match C
4Fh		GPT1_CMPD	Compare match D
50h		GPT1_OVF	Overflow
51h		GPT1_UDF	Underflow
52h	GPT162	GPT2_CCMPA	Compare match A
53h		GPT2_CCMPB	Compare match B
54h		GPT2_CMPC	Compare match C
55h		GPT2_CMPD	Compare match D
56h		GPT2_OVF	Overflow
57h		GPT2_UDF	Underflow
58h	GPT163	GPT3_CCMPA	Compare match A
59h		GPT3_CCMPB	Compare match B
5Ah		GPT3_CMPC	Compare match C
5Bh		GPT3_CMPD	Compare match D
5Ch		GPT3_OVF	Overflow
5Dh		GPT3_UDF	Underflow
5Eh	GPT164	GPT4_CCMPA	Compare match A
5Fh		GPT4_CCMPB	Compare match B
60h		GPT4_CMPC	Compare match C
61h		GPT4_CMPD	Compare match D
62h		GPT4_OVF	Overflow
63h		GPT4_UDF	Underflow
64h	GPT165	GPT5_CCMPA	Compare match A
65h		GPT5_CCMPB	Compare match B
66h		GPT5_CMPC	Compare match C
67h		GPT5_CMPD	Compare match D
68h		GPT5_OVF	Overflow
69h		GPT5_UDF	Underflow
6Ah	GPT166	GPT6_CCMPA	Compare match A
6Bh		GPT6_CCMPB	Compare match B
6Ch		GPT6_CMPC	Compare match C
6Dh		GPT6_CMPD	Compare match D
6Eh		GPT6_OVF	Overflow
6Fh		GPT6_UDF	Underflow
70h	GPT	GPT_UVWEDGE	UVW edge event
71h	SCI0	SCI0_RXI*4	Receive data full
72h		SCI0_TXI*4	Transmit data empty
73h		SCI0_TEI	Transmit end
74h		SCI0_ERI*4	Receive error
75h		SCI0_AM	Address match event

Table 17.3 Association between event signal names set in ELSRn.ELS[7:0] bits and signal numbers (4 of 4)

Event number	Interrupt request source	Name	Description
77h	SCI1	SCI1_RXI	Receive data full
78h		SCI1_TXI	Transmit data empty
79h		SCI1_TEI	Transmit end
7Ah		SCI1_ERI	Receive error
7Bh		SCI1_AM	Address match event
7Ch	SCI9	SCI9_RXI	Receive data full
7Dh		SCI9_TXI	Transmit data empty
7Eh		SCI9_TEI	Transmit end
7Fh		SCI9_ERI	Receive error
80h		SCI9_AM	Address match event
81h	SPI0	SPI0_SPRI	Receive buffer full
82h		SPI0_SPTI	Transmit buffer empty
83h		SPI0_SPII	Idle
84h		SPI0_SPEI	Error
85h		SPI0_SPTEND	Transmission completed event
86h	SPI1	SPI1_SPRI	Receive buffer full
87h		SPI1_SPTI	Transmit buffer empty
88h		SPI1_SPII	Idle
89h		SPI1_SPEI	Error
8Ah		SPI1_SPTEND	Transmission completed event

Note 1. Only pulse (edge detection) is supported.
Note 2. ELSR8, ELSR9, ELSR14, ELSR15, and ELSR18 can select this event.
Note 3. This event can occur in Snooze mode.
Note 4. This event is not supported in FIFO mode.

17.3 Operation

17.3.1 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.

17.3.2 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 17.4 lists the operations of modules when an event occurs.

Table 17.4 Module operations when event occurs (1 of 2)

Module	Operations when event occurs
GPT	• Start counting • Stop counting • Clear counting • Up counting • Down counting • Input capture.
ADC16	Start A/D conversion
DAC12	Start D/A conversion
DAC8	Start D/A conversion

Table 17.4 Module operations when event occurs (2 of 2)

Module	Operations when event occurs
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.
CTSU	Start measurement operation
SDADC24	Start A/D conversion
DTC	Start DTC data transfer

17.3.3 Example 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 00000000b 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.

17.4 Usage Notes

17.4.1 Linking DTC Transfer End Signals as Events

When linking the DTC transfer end signals as events, do not set the same peripheral module as the DTC transfer destination and event link destination. If set, the peripheral module might be started before DTC transfer to the peripheral module is complete.

17.4.2 Setting the 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 the specific low power mode in which the module is stopped (Software Standby mode). Some modules can perform in Snooze mode. For more information, see Table 17.3 and section 11, Low Power Modes.

17.4.3 Setting the Module Stop Function

The Module Stop Control Register C (MSTPCRC) can enable or disable ELC operation. The ELC is initially stopped after reset. Releasing the module-stop state enables access to the registers. The ELCON bit must be set to 0 before disabling ELC operation using the Module Stop Control Register. For more information, see Table 17.3 and section 11, Low Power Modes.

17.4.4 ELC Delay Time

In Figure 17.2, 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 17.5.

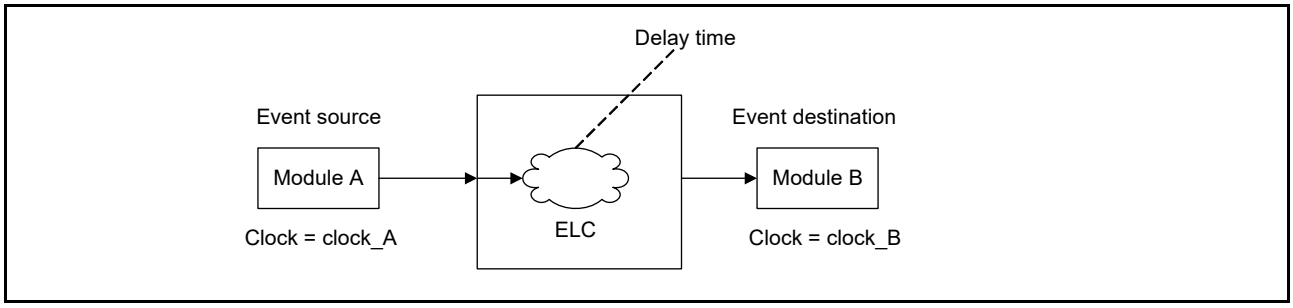


Figure 17.2 ELC delay time

Table 17.5 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

18. I/O Ports

18.1 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, or port group function for the ELC. All pins operate as input pins immediately after a reset, and pin functions are switched by register settings. The I/O ports and peripheral modules for each pin are specified in the associated registers.

Figure 18.1 shows a connection diagram for the I/O port registers. The configuration of the I/O ports differs depending on the package. Table 18.1 lists the I/O port specifications by package, and Table 18.2 lists the port functions.

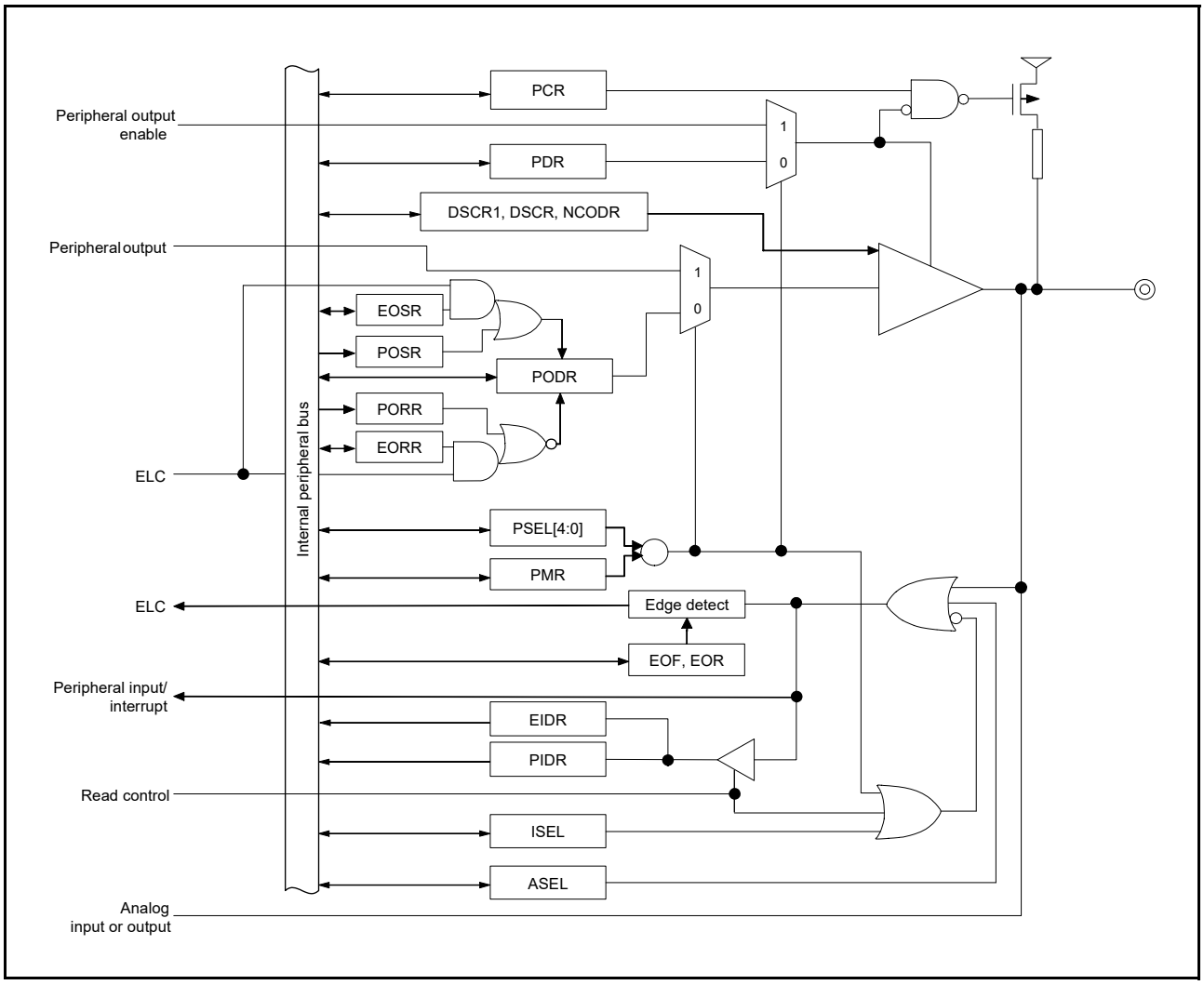


Figure 18.1 Connection diagram for I/O port registers

Note: Figure 18.1 shows a basic port configuration. The configuration differs depending on the ports.

Table 18.1 I/O port specifications (1 of 2)

Port	Package 64 pins	Number of pins	Package 48 pins	Number of pins	Package 40 pins	Number of pins	Package 36 pins	Number of pins	Package 32 pins	Number of pins
PORT0	P000 to P003, P012 to P015	8	P000, P013 to P015	4	P000, P013	2	P000	1	N/A	0
PORT1	P100 to P112	13	P100 to P105, P108 to P110	9	P100 to P103, P108 to P110	7	P100, P101, P108 to P110	5	P100, P101, P108 to P110	5

Table 18.1 I/O port specifications (2 of 2)

Port	Package		Package		Package		Package		Package	
	64 pins	Number of pins	48 pins	Number of pins	40 pins	Number of pins	36 pins	Number of pins	32 pins	Number of pins
PORT2	P200, P201, P204 to P206, P212 to P215	9	P200, P201, P206, P212 to P215	7	P200, P201, P212 to P215	6	P200, P201, P212 to P215	6	P200, P201, P204 to P206, P212, P213	7
PORT3	P300 to P304	5	P300 to P302	3	P300, P301	2	P300, P301	2	P300, P301	2
PORT4	P400 to P403, P407 to P411	9	P400, P401, P407 to P409	5	P400, P407, P408	3	P400, P407, P408	3	P400, P407, P408	3
PORT5	P500 to P502	3	P500 to P502	3	P500 to P502	3	P500 to P502	3	P500 to P502	3
PORT9	P914, P915	2	P914, P915	2	P914, P915	2	P914, P915	2	N/A	0
	Total pins	49	Total pins	33	Total pins	25	Total pins	22	Total pins	20

Table 18.2 I/O port functions

Port	Port name	Input pull-up	Input mode switching	Open drain output	Drive capacity switching	5 V tolerant
PORT0	P000	✓	CMOS/TTL	✓	Low, middle	✓
	P001 to P003, P012 to P015	✓	—	—	Low, middle	—
PORT1	P100, P101	✓	—	✓	Low, middle	—
	P102 to P108	✓	—	—	Low, middle	—
	P109, P110	✓	CMOS/TTL	✓	Low, middle	—
	P111, P112	✓	CMOS/TTL	✓	Low, middle	✓
PORT2	P200, P214, P215	—	—	—	—	—
	P201	✓	—	—	Low, middle	—
	P204	✓	—	✓	Low, middle	—
	P205, P206	✓	CMOS/TTL	✓	Low, middle	✓
	P212, P213	✓	—	✓	—	—
PORT3	P300, P303	✓	—	—	Low, middle	—
	P301	✓	CMOS/TTL	✓	Low, middle	✓
	P302, P304	✓	—	✓	Low, middle	—
PORT4	P400	✓	CMOS/TTL	✓	Low, middle	—
	P401	✓	CMOS/TTL	✓	Low, middle	✓
	P402, P403	✓	—	—	Low, middle	—
	P410, P411	✓	—	✓	Low, middle	—
	P407, P409	✓	CMOS/TTL	✓	Low, middle, middle (IIC, SPI)	✓
	P408	✓	CMOS/TTL	✓	Low, middle, middle (IIC, SPI)	—
PORT5	P500, P501	✓	—	✓	Low, middle	—
	P502	✓	—	—	Low, middle	—
PORT9	P914, P915	—	—	—	—	—

✓: available
—: Setting prohibited

18.2 Register Descriptions

18.2.1 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 port output data, and is accessed in 32-bit units. The PDRn (bits [15:0] in PCNTR1) and PODRn (bits [31:16] in PCNTR1) respectively, are accessed in 16-bit units.

PDRn bits (Pmn Direction)

The PDRn bits select the input or output direction for individual pins on the associated port when the pins are configured as general I/O pins. Each pin on port m is associated with a PORTm.PCNTR1.PDRn bit. The I/O direction can be specified in 1-bit units. Bits associated with non-existent pins are reserved. The write value should be 0. P200, P214, P215 are input only, so PORT2.PCNTR1.PDR00, PDR14, and PDR15 bits are reserved. The PDRn bit in the PORTm.PCNTR1 register serves the same function as the PDR bit in the PFS.PmnPFS register.

PODRn bits (Pmn Output Data)

The PODRn bits hold data to be output from the general I/O pins. Bits of non-existent port m are reserved. Write 0 to these bits. Bits associated with non-existent pins are reserved. The write value should be 0. P200, P214, and P215 are input only, so PORT2.PCNTR1.PODR00, PODR14, and PODR15 bits are reserved. The PODRn bit in the PORTm.PCNTR1 register serves the same function as the PODR bit in the PFS.PmnPFS register.

18.2.2 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

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 an ELC_PORTx occurs: 0: Low input 1: High input.	R

m = 0 to 5, 9
n = 00 to 15
x = 1, 2

Note 1. Supported for PORT1 and PORT2.

The Port Control Register 2 (PCNTR2/EIDR/PIDR) allows read access to the Pmn state and the port event input data using 32-bit or 16-bit access.

The PCNTR2 represents the Pmn state and the port event input data, and is accessed in 32-bit units. The PIDRn (bits [15:0] in PCNTR2) and EIDRn (bits [31:16] in PCNTR2) respectively, are accessed in 16-bit units. Bits associated with non-existent pins are reserved. Reserved bits are read as undefined.

PIDRn bits (Pmn State)

The PIDRn bits reflect individual pin states of the port, regardless of the values set in PmnPFS.PMR and PORTm.PCNTR1.PDRn. The PIDRn bit in the PORTm.PCNTR2 register serves the same function as the PIDR bit in the PFS.PmnPFS register.

A pin state cannot be reflected in PIDRn when one of the following functions is enabled:

- Main clock oscillator (MOSC)
- Sub-clock oscillator (SOSC)
- Analog function (ASEL = 1)
- Capacitive Touch Sensing Unit (CTSU)
- USB 2.0 Full-Speed Module (USBFS).

EIDRn bits (Port Event Input Data)

The EIDRn bits latch a pin state when an ELC_PORTx signal occurs. Pin states can only be input to EIDRn when PmnPFS.PMR and PORTm.PCNTR1.PDRn are 0. When the PmnPFS.ASEL bit is set to 1, the associated pin state is not reflected in EIDRn.

18.2.3 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

The Port Control Register 3 (PCNTR3/PORR/POSR) is a 32-bit 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.

POSRn bits (Pmn Output Set)

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, POSR14, and POSR15 bits are reserved.

PORRn bits (Pmn Output Reset)

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, PORR14, and PORR15 bits are reserved.

Note: When EORRn or EOSRn is set, writing is prohibited to PODRn, PORRn, and POSRn.
Note: PORRn and POSRn should not be set at the same time.

18.2.4 Port Control Register 4 (PCNTR4/EORR/EOSR)

Address(es): PORT1.PCNTR4 4004 002Ch, PORT2.PCNTR4 4004 004Ch

PORT1.EORR 4004 002Ch, PORT2.EORR 4004 004Ch

PORT1.EOSR 4004 002Eh, PORT2.EOSR 4004 004Eh

	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, 2
n = 00 to 15
x = 1, 2

The Port Control Register 4 (PCNTR4/EORR/EOSR) 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.

EOSRn bits (Pmn Event Output Set)

EOSR changes PODR when set because an ELC_PORTx signal occurs. For example, for P100 if PORT1.PCNTR4.EOSR00 is set to 1 when the ELC_PORTx occurs, PORT1.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. P200, P214, and P215 are input only, so PORT2.PCNTR4.EOSR00, EOSR14, and EOSR15 bits are reserved.

EORRn bits (Pmn Event Output Reset)

EORR changes PODR when reset because an ELC_PORTx signal occurs. For example, for P100 if PORT1.PCNTR4.EORR00 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, EORR14, and EORR15 bits are reserved.

Note: When EORRn or EOSRn is set, writing is prohibited to PODRn, PORRn, and POSRn.
Note: EORRn and EOSRn should not be set at the same time.

18.2.5 Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) (m = 0 to 5, 9; n = 00 to 15)

Address(es): PFS.P000PFS 4004 0800h to PFS.P003PFS 4004 080Ch, PFS.P012PFS 4004 0830h to PFS.P015PFS 4004 083Ch,
PFS.P100PFS 4004 0840h to PFS.P112PFS 4004 0870h,
PFS.P200PFS 4004 0880h to 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 to PFS.P304PFS 4004 08D0h,
PFS.P400PFS 4004 0900h to PFS.P403PFS 4004 090Ch, PFS.P407PFS 4004 091Ch to PFS.P411PFS 4004 092Ch,
PFS.P500PFS 4004 0940h to PFS.P502PFS 4004 0948h,
PFS.P914PFS 4004 0A78h to PFS.P915PFS 4004 0A7Ch

PFS.P000PFS_HA 4004 0802h to PFS.P003PFS_HA 4004 080Eh, PFS.P012PFS_HA 4004 0832h to PFS.P015PFS_HA 4004 083Eh,
PFS.P100PFS_HA 4004 0842h to PFS.P112PFS_HA 4004 0872h, PFS.P200PFS_HA 4004 0882h to PFS.P201PFS_HA 4004 0886h,
PFS.P204PFS_HA 4004 0892h to PFS.P206PFS_HA 4004 0896h, PFS.P212PFS_HA 4004 08B2h to PFS.P215PFS_HA 4004 08BEh,
PFS.P300PFS_HA 4004 08C2h to PFS.P304PFS_HA 4004 08D2h, PFS.P400PFS_HA 4004 0902h to PFS.P403PFS_HA 4004 090Eh,
PFS.P407PFS_HA 4004 091Eh to PFS.P411PFS_HA 4004 092Eh, PFS.P500PFS_HA 4004 0942h to PFS.P502PFS_HA 4004 094Ah,
PFS.P914PFS_HA 4004 0A7Ah to PFS.P915PFS_HA 4004 0A7Eh

PFS.P000PFS_BY 4004 0803h to PFS.P003PFS_BY 4004 080Fh, PFS.P012PFS_BY 4004 0833h to PFS.P015PFS_BY 4004 083Fh,
PFS.P100PFS_BY 4004 0843h to PFS.P112PFS_BY 4004 0873h, PFS.P200PFS_BY 4004 0883h to PFS.P201PFS_BY 4004 0887h,
PFS.P204PFS_BY 4004 0893h to PFS.P206PFS_BY 4004 0897h, PFS.P212PFS_BY 4004 08B3h to PFS.P215PFS_BY 4004 08BFh,
PFS.P300PFS_BY 4004 08C3h to PFS.P304PFS_BY 4004 08D3h, PFS.P400PFS_BY 4004 0903h to PFS.P403PFS_BY 4004 090Fh,
PFS.P407PFS_BY 4004 091Fh to PFS.P411PFS_BY 4004 092Fh, PFS.P500PFS_BY 4004 0943h to PFS.P502PFS_BY 4004 094Bh,
PFS.P914PFS_BY 4004 0A7Bh to PFS.P915PFS_BY 4004 0A7Fh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	PSEL[4:0]					—	—	—	—	—	—	—	PMR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*2

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	EOF	EOR	DSCR1 *3	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 input pull-up 1: Enable 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
b11, b10	DSCR1/DSCR	Port Drive Capability	<P407 to P409> b11 b10 0 0: Low drive 0 1: Middle drive 1 0: Middle drive for IIC Fast mode and SPI 1 1: Setting prohibited. <Other than P407 to P409> b10 0: Low drive 1: Middle drive. b11 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 edge.	R/W

Bit	Symbol	Bit name	Description	R/W
b14	ISEL	IRQ Input Enable	0: Not used as an IRQn input pin 1: Used as an IRQn input pin.	R/W
b15	ASEL	Analog Input Enable	0: Not used as an analog pin 1: Used as an analog pin.	R/W
b16	PMR	Port Mode Control	0: Used as a general I/O pin. 1: Used as an I/O port for peripheral functions.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b24	PSEL[4:0]	Peripheral Select	These bits select the peripheral function. For individual pin functions, see the associated tables in this chapter.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Supported for PORT1 and PORT2.
Note 2. The initial value of P108, P201, P300, P914, and P915 is not 0000_0000h.
The initial value of P108 is 0001_0010h, P201 is 0000_0010h, P300 is 0001_0010h, and P914 and P915 is 0001_0000h.
Note 3. Only P407 to P409 have this bit. For other ports, this bit is reserved.

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. PmnPFS 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.

PODR bit (Port Output Data), PIDR bit (Pmn State), PDR bit (Port Direction)

The PDR, PIDR, and PODR bits serve the same function as the PCNTR. When these bits are read, the PCNTR value is read.

PCR bit (Pull-up Control)

The PCR bit enables or disables an input pull-up resistor on the individual port pins. When a pin is in the input state with the associated bit in PmnPFS.PCR set to 1, the pull-up resistor connected to the pin is enabled. When a pin is set as a general port output pin, or a peripheral function output pin, the pull-up resistor for the pin is disabled regardless of the PCR setting. The pull-up resistor is also disabled in the reset state. Bits associated with non-existent pins are reserved. The write value should be 0.

NCODR bit (N-Channel Open-Drain Control)

The NCODR bit specifies the output type for the port pins. Bits associated with non-existent pins are reserved. The write value should be 0.

DSCR1/DSCR bits (Port Drive Capability)

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. The write value should be 0.

EOF/EOR bits (Event on Falling/Event on Rising)

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. The write value should be 0.

ISEL bit (IRQ Input Enable)

The ISEL bit specifies IRQ input pins. This setting can be used in combination with the peripheral functions, although IRQn (external pin interrupt) of the same number must only be enabled for one pin.

ASEL bit (Analog Input Enable)

The ASEL bit specifies analog pins. When a pin is set as an analog pin by this bit:

1. Specify it as a general I/O port with the Port Mode Control bit (PmnPFS.PMR)*1.

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.

Note 1. When the D/A converter output level is output to a port, select the I/O port for peripheral functions using the Port Mode Control bit to set the D/A output with the PmnPFS.PSEL bit.

The ISEL bit for an unspecified IRQn is reserved. The ASEL bit for an unspecified analog input/output is reserved.

PMR bit (Port Mode Control)

The PMR bit specifies the port pin function. Bits associated with non-existent pins are reserved. The write value should be 0.

PSEL[4:0] bits (Peripheral Select)

The PSEL[4:0] bits assign the peripheral function.

For details of the peripheral select settings for each product, see section 18.6, Peripheral Select Settings for each Product.

18.2.6 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.

18.3 Operation

18.3.1 General I/O Ports

All pins except P108 and P300 operate as general I/O ports after reset. General I/O ports are organized as 16 bits per port and can be accessed by port with the Port Control Registers (PCNTRn, where n = 1 to 4), or by individual pins with the Port mn Pin Function Select register. For details on these registers, see section 18.2, 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 or 2 signal occurs

- Port Output Set bit (POSRn), which indicates the output value when a software write occurs
- Port Output Reset bit (PORRn), which indicates the output value when a software write occurs
- Event Output Set bit (EOSRn), which indicates the output value when an ELC_PORT1 or 2 signal occurs
- Event Output Reset bit (EORRn), which indicates the output value when an ELC_PORT1 or 2 signal occurs.

18.3.2 Port Function Select

The following port functions are available for configuring each pin:

- I/O configuration: CMOS output or NMOS open-drain output, pull-up control, and drive strength
- General I/O port: Port direction, output data setting, and read input data
- Alternate function: Configured function mapping to the pin.

Each pin is associated with a Port mn Pin Function Select register (PmnPFS), which includes the associated PODR, PIDR, and PDR bits. In addition, the PmnPFS register includes:

- 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 bits that select the drive capacity
- EOR: Event on rising bit used to detect rising edges on the port input
- EOF: Event on falling bit used to detect falling edges on the port input
- ISEL: IRQ input enable bit to specify an IRQ input pin
- ASEL: Analog input enable bit to specify an analog pin
- PMR: Port mode 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 Port mn Pin Function Select (PmnPFS) register. For details, see section 18.2.5, Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) (m = 0 to 5, 9; n = 00 to 15).

18.3.3 Port Group Function for the ELC

In the MCU, PORT1 and PORT2 are assigned for the port group function.

18.3.3.1 Behavior when ELC_PORT1 or 2 is input from the ELC

The MCU supports the two functions described in this section when the ELC_PORT1 or 2 signal comes from the ELC.

(1) Input to EIDR

For the GPI function (PDR = 0 and PMR = 0 in the PmnPFS register), when an ELC_PORT1 or 2 signal comes from the ELC, the input enable of the I/O cell is asserted, and data from the external pins are read into the EIDR bit. See Figure 18.2.

For the GPO function (PDR = 1) or the peripheral mode (PMR = 1), 0 is input into the EIDR bit from the external pins.

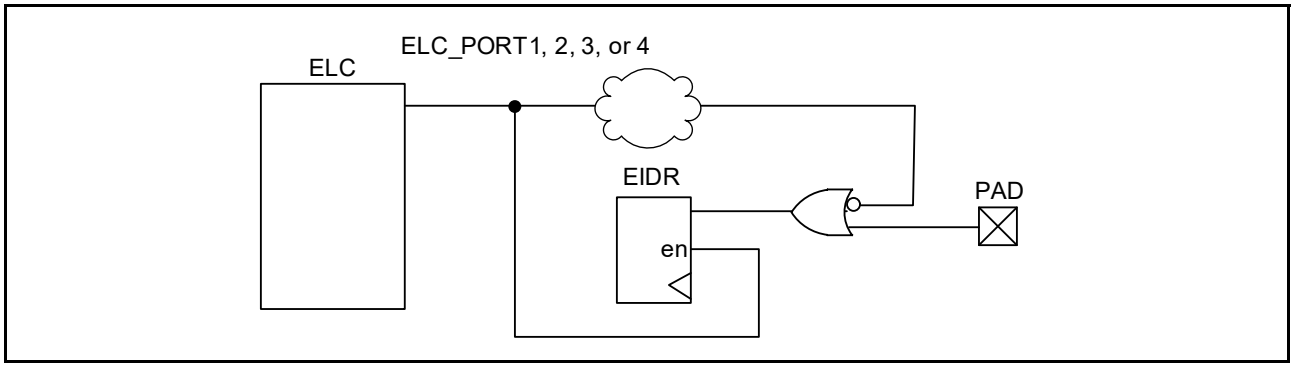


Figure 18.2 Event ports input data

(2) Output from PODR by EOSR/EORR

When an ELC_PORT1 or 2 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 or 2 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 or 2 signal occurs, the PODR register outputs 0 to the external pin. Otherwise, when EORR = 0, the PODR value is kept.

See Figure 18.3.

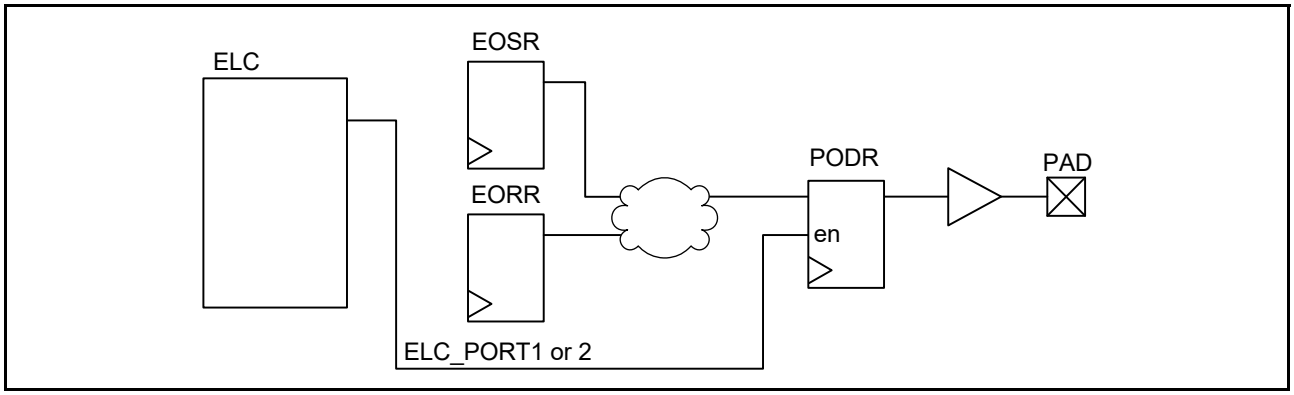


Figure 18.3 Ports output data

18.3.3.2 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 18.2.5, 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 P113, the data of those 14 pins is organized by OR logic. This data is formed into a one-shot pulse that goes to the ELC. The operation of PORT2 is also the same as PORT1. See Figure 18.4.

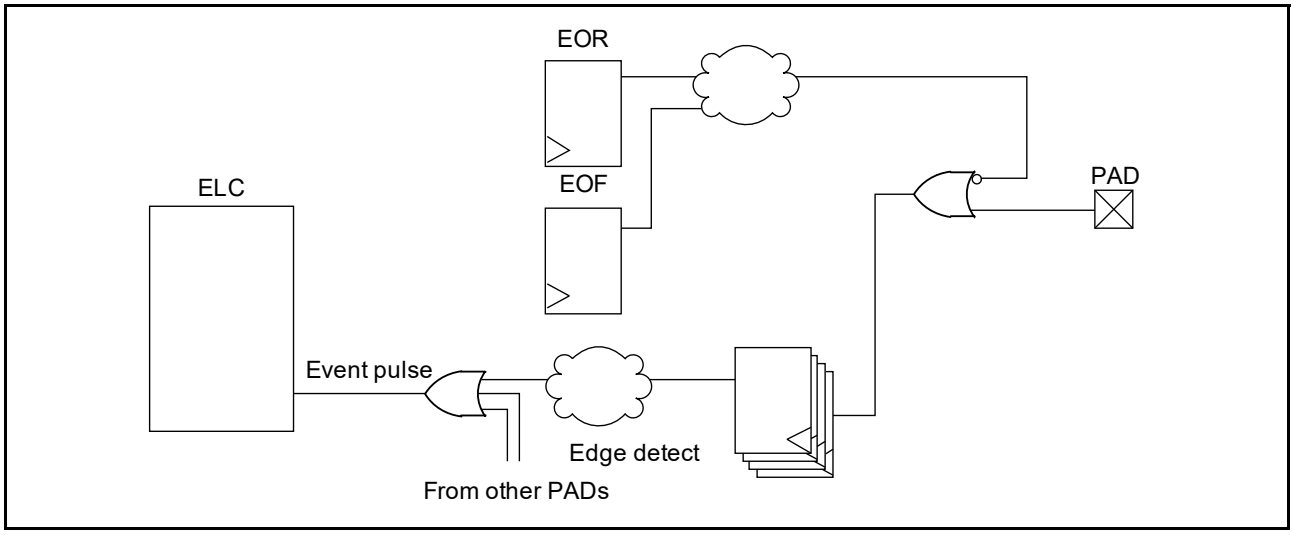


Figure 18.4 Generation of event pulse

18.4 Handling of Unused Pins

Table 18.3 shows how to handle unused pins.

Table 18.3 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 of P914PFS.PMR and P915PFS.PMR bits are set to 1, keep these pins open
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, do the same for P1x to P5x, and P9x.
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, do the same for P1x to P5x, and P9x. When the external clock is input to the EXTAL pin, keep 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, do the same for P1x to P5x, and P9x.
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, do the same for P1x to P5x, and P9x.
P000, P001, P108 to P112	• 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. • If the direction setting is for output (PCNTR1.PDRn = 1), release the pin*1, *2.
P002, P003, P012 to P015, P500 to P502	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
P100 to P107	If the direction setting is for input (PCNTR1.PDRn = 0), connect the associated pin to AVCC1 (pulled up) through a resistor or to AVSS1 (pulled down) through a resistor*1
P2x, to P4x, P9x	• If the direction setting is for input (PCNTR1.PDRn = 0), connect the associated pin to VCC (pulled up) through a resistor or to VSS (pulled down) through a resistor.*1 • If the direction setting is for output (PCNTR1.PDRn = 1), release the pin.*1, *2
VREFH0	Connect to AVCC0
VREFL0	Connect to AVSS0
ADREG	Keep these pins open
SBIAS/VREFI	Keep these pins open

Note 1. Clear the PmnPFS.PMR, PmnPFS.ISEL, PmnPFS.PCR, and PmnPFS.ASEL bits to 0.
Note 2. P108 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).

18.5 Usage Notes

18.5.1 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 bit 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.

18.5.2 Procedure for Using Port Group Input

To use the port group input (PORT1 and PORT2):

1. Set the ELSRx.ELS[7:0] bits to 0000 0000b to ignore unexpected pulses. For more information, see section 17, 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[7:0] bits to enable the event signals.

18.5.3 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 or 2 signal occurs.
2. Output 1 if PCNTR4.EOSRn is set to 1 when an ELC_PORT1 or 2 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 same time, the higher priority 1. is executed.

18.5.4 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.

18.5.5 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 18.4 shows the setting values of bits PFS.P914PFS.PMR and PFS.P915PFS.PMR with each selected pin.

Table 18.4 Selecting the USB/PORT pins

PMR bits 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.

18.6 Peripheral Select Settings for each Product

This section describes the pin function select configuration by the PmnPFS register. Some pin names have a _A, _B, _C, _D, _E, or _F suffix. The suffix can be ignored when assigning functionality, but assigning the same function to two or more pins simultaneously is prohibited.

Table 18.5 Register settings for I/O pin functions (PORT0)

PSEL[4:0] settings	Function	Pin							
		P000	**P001**	**P002**	**P003**	**P012**	**P013**	**P014**	**P015**
00000b (initial)	Hi-Z/SWD	Hi-Z							
00001b	AGT	AGTIO1_A	—	—	—	—	—	—	—
00011b	GPT	GTIOC4B_B	—	—	—	—	GTIOC6B_C	GTIOC6A_C	—
00101b	SCI	RXD9_A/ SCL9_A/ MISO9_A	CTS9_RTS9_A/ SS9_A	—	—	—	—	—	—
00110b	SPI	MISOB_A	RSPCKB_D	—	—	—	—	—	—
00111b	IIC	SCL0_C	—	—	—	—	—	—	—
01000b	KINT	KR00	—	—	—	—	—	—	—
01001b	CLKOUT/ ACMPHS/ ACMPLP/RTC	—	RTCOUT_D	—	—	—	—	—	—
01010b	CAC/ADC16/ SDADC24/ DAC12/DAC8	—	—	DA8_1	—	—	DA8_0	—	—
01100b	CTSU	TS16	TS15	—	—	—	—	—	—
ASEL bit		—	—	AN007/ DA8_1/ AMP2+	AN006/ AMP2-	AN008/ AMP2O	AN005/ DA8_0/ AMP1+/ IVCMP1/ VREFH	AN004/ AMP1-/ IVREF1/ VREFL	AN003/ AMP1O
ISEL bit		IRQ4_A	IRQ0_B	—	—	—	—	—	—
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M	L/M	L/M
NCODR bit		✓	—	—	—	—	—	—	—
PCR bit		✓	✓	✓	✓	✓	✓	✓	✓
64-pin product		✓	✓	✓	✓	✓	✓	✓	✓
48-pin product		✓	—	—*1	—	—	✓	✓	✓
40-pin product		✓	—	—*1	—	—	✓	—	—
36-pin product		✓	—	—*1	—	—	—*1	—	—
32-pin product		—	—	—*1	—	—	—*1	—	—

✓: Available
—: Setting prohibited
L/M: Low drive/Middle drive

Note 1. DAC8 channel n (n = 0, 1) output cannot be directly output from the pin, but the register can be set to output from a pin through OPAMP.

Table 18.6 Register settings for input/output pin function (PORT1) (1)

PSEL[4:0] settings	Function	Pin							
		P100	P101	P102	P103	P104	P105	P106	P107
00000b (initial)	Hi-Z/SWD	Hi-Z							
00001b	AGT	—	—	—	—	—	—	—	—
00010b	GPT	—	—	—	—	—	—	—	—
00011b	GPT	GTIOC5B_B	GTIOC5A_B	GTIOC6B_B	GTIOC6A_B	—	—	—	—
00100b	SCI	—	—	—	—	—	—	—	—
00101b	SCI	TXD9_D/ SDA9_D/ MOSI9_D	RXD9_C/ SCL9_C/ MISO9_C	CTS9_RTS9_D/ SS9_D	—	—	—	—	—
00110b	SPI	—	—	SSLB0_C	RSPCKB_C	MISOB_C	MOSIB_C	—	—
00111b	IIC	—	—	—	—	—	—	—	—
01000b	KINT	—	—	—	—	—	—	—	—
01001b	CLKOUT/ ACMPHS/ ACMPLP/RTC	—	—	—	—	—	—	—	—
01010b	CAC/ADC16/ SDADC24/ DAC12/DAC8	—	—	—	—	—	—	—	—
01100b	CTSU	TS23	TS22	TS21	TS20	TS19	TS18	—	—
10000b	CAN	—	—	—	—	—	—	—	—
ASEL bit		AN016/ ANSD0P/ IVCMP2	AN017/ ANSD0N/ IVREF2	AN018/ ANSD1P	AN019/ ANSD1N	AN020/ ANSD2P	AN021/ ANSD2N	AN022/ ANSD3P	AN023/ ANSD3N
ISEL bit		IRQ4_C	IRQ5_C	—	—	IRQ6_C	IRQ7_C	—	—
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M	L/M	L/M
NCODR bit		✓	✓	—	—	—	—	—	—
PCR bit		✓	✓	✓	✓	✓	✓	✓	✓
64-pin product		✓	✓	✓	✓	✓	✓	✓	✓
48-pin product		✓	✓	✓	✓	✓	✓	—	—
40-pin product		✓	✓	✓	✓	—	—	—	—
36-pin product		✓	✓	—	—	—	—	—	—
32-pin product		✓	✓	—	—	—	—	—	—

✓: Available
—: Setting prohibited
L/M: Low drive/Middle drive

Table 18.7 Register settings for input/output pin function (PORT1) (2)

PSEL[4:0] settings	Function	Pin				
		P108	P109	P110	P111	P112
00000b (initial)	Hi-Z/SWD	SWDIO	Hi-Z			
00001b	AGT	—	AGTOA0_A	AGTOB0_A	—	—
00010b	GPT	—	GTETRGB_A	GTOWLO_A	—	—
00011b	GPT	—	GTIOC1B_B	GTIOC2B_B	—	—
00100b	SCI	—	SCK0_C	TXD0_D/ SDA0_D/ MOSI0_D	—	—
00101b	SCI	—	TXD9_B/ SDA9_B/ MOSI9_B	RXD9_B/ SCL9_B/ MOSI9_B	—	—
00110b	SPI	—	MISOA_A	RSPCKA_A	RSPCKA_B	SSLA0_B
00111b	IIC	—	SCL1_A	SDA1_D	SCL1_C	SDA1_C
01000b	KINT	—	KR01	—	—	—
01001b	CLKOUT/ ACMPHS/ ACMPLP/RTC	—	VCOUT_A	CLKOUT_A	RTCOUT_B	CLKOUT_B
01010b	CAC/ADC16/ SDADC24/ DAC12/DAC8	—	ADTRG0_B	ADTRG0_A	—	—
01100b	CTSU	—	TS17	TSCAP_A	TS14	TSCAP_B
10000b	CAN	—	—	CTX0_C	—	—
ASEL bit		—	CMPREF0	CMPREF1	—	—
ISEL bit		—	IRQ3_A	IRQ2_A	IRQ6_B	IRQ7_B
DSCR bit		L/M	L/M	L/M	L/M	L/M
NCODR bit		—	✓	✓	✓	✓
PCR bit		✓	✓	✓	✓	✓
64-pin product		✓	✓	✓	✓	✓
48-pin product		✓	✓	✓	—	—
40-pin product		✓	✓	✓	—	—
36-pin product		✓	✓	✓	—	—
32-pin product		✓	✓	✓	—	—

✓: Available
—: Setting prohibited
L/M: Low drive/Middle drive

Table 18.8 Register settings for input/output pin function (PORT2)

PSEL[4:0] settings	Function	Pin								
		P200*1	P201	P204	P205	P206	P212	P213	P214	P215
00000b (initial)	Hi-Z/SWD	Hi-Z								
00001b	AGT	—	—	—	—	AGTIO0_B	AGTIO0_A	AGTEE1_B	—	—
00010b	GPT	—	—	—	GTOVLO_A	GTOVUP_A	GTETRGB_B	GTETRGA_B	—	—
00011b	GPT	—	—	—	GTIOC3B_A	GTIOC3A_A	GTIOC0B_B	GTIOC0A_B	—	—
00100b	SCI	—	—	RXD0_C/ SCL0_C/ MISO0_C	TXD0_C/ SDA0_C/ MOSI0_C	CTS0_RTS0_C/ SS0_C	—	—	—	—
00101b	SCI	—	—	SCK9_B	CTS1_RTS1_A/ SS1_A	TXD1_A/ SDA1_A/ MOSI1_A	TXD1_D/ SDA1_D/ MOSI1_D	RXD1_D/ SCL1_D/ MISO1_D	—	—
00110b	SPI	—	—	MOSIB_B	MISOB_B	SSLB0_A	—	—	—	—
00111b	IIC	—	—	—	SDA1_B	SCL1_B	—	—	—	—
01000b	KINT	—	—	—	—	—	—	—	—	—
01001b	CLKOUT/ ACMPHS/ ACMPLP/RTC	—	—	—	—	—	—	—	—	—
01010b	CAC/ADC16/ SDADC24/ DAC12/DAC8	—	—	—	—	—	—	—	—	—
01100b	CTSU	—	—	TS09	TS08	TS07	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—
ISEL bit		—	—	—	IRQ0_C	IRQ6_A	IRQ3_B	IRQ2_B	—	—
DSCR bit		—	L/M	L/M	L/M	L/M	—	—	—	—
NCODR bit		—	—	✓	✓	✓	✓	✓	—	—
PCR bit		—	✓	✓	✓	✓	✓	✓	—	—
64-pin product		✓	✓	✓	✓	✓	✓	✓	✓	✓
48-pin product		✓	✓	—	—	✓	✓	✓	✓	✓
40-pin product		✓	✓	—	—	—	✓	✓	✓	✓
36-pin product		✓	✓	—	—	—	✓	✓	✓	✓
32-pin product		✓	✓	✓	✓	✓	✓	✓	—	—

✓: Available
—: Setting prohibited
L/M: Low drive/Middle drive

Note 1. When using NMI pin interrupt, Port related resisters setting are not required.

Table 18.9 Register settings for input/output pin function (PORT3)

PSEL[4:0] settings	Function	Pin				
		P300	P301	P302	P303	P304
00000b (initial)	Hi-Z/SWD	SWCLK	Hi-Z			
00001b	AGT	—	AGTOB1_A	AGTOA1_A	—	—
00010b	GPT	—	GTOWUP_A	GTOVLO_B	—	—
00011b	GPT	—	GTIOC2A_B	GTIOC3B_B	GTIOC6B_A	GTIOC6A_A
00100b	SCI	—	RXD0_B/ SCL0_B/ MISO0_B	TXD0_B/ SDA0_B/ MOSI0_B	CTS0_RTS0_B/ SS0_B	SCK0_B
00101b	SCI	—	CTS9_RTS9_B/ SS9_B	RXD1_A/ SCL1_A/ MISO1_A	SCK1_A	TXD9_C/ SDA9_C/ MOSI9_C
00110b	SPI	—	MOSIB_A	RSPCKB_A	MOSIA_B	MISOA_B
00111b	IIC	—	SDA0_A	—	—	—
01000b	KINT	—	KR04	KR05	KR06	KR07
01001b	CLKOUT/ ACMPHS/ ACMPLP/RTC	—	RTCOUT_A	—	—	—
01010b	CAC/ADC16/ SDADC24/ DAC12/DAC8	—	—	CACREF_A	—	—
01100b	CTSU	—	TS13	TS12	TS11	TS10
10000b	CAN	—	—	—	CRX0_A	CTX0_A
ASEL bit		—	—	—	—	—
ISEL bit		—	IRQ5_A	IRQ4_B	—	—
DSCR bit		L/M	L/M	L/M	L/M	L/M
NCODR bit		—	✓	✓	—	✓
PCR bit		✓	✓	✓	✓	✓
64-pin product		✓	✓	✓	✓	✓
48-pin product		✓	✓	✓	—	—
40-pin product		✓	✓	—	—	—
36-pin product		✓	✓	—	—	—
32-pin product		✓	✓	—	—	—

✓: Available
—: Setting prohibited
L/M: Low drive/Middle drive

Table 18.10 Register settings for input/output pin function (PORT4)

PSEL[4:0] settings	Function	Pin								
		P400	P401	P402	P403	P407	P408	P409	P410	P411
00000b (initial)	Hi-Z/SWD	Hi-Z								
00001b	AGT	AGTEE0_A	AGTEE1_A	—	—	AGTIO0_C	AGTO0_A	AGTO1_A	—	—
00010b	GPT	GTETRGA_A	GTIU_A	GTIV_A	GTIW_A	GTOULO_A	GTOUUP_A	—	—	—
00011b	GPT	GTIOC1A_A	GTIOC4A_A	GTIOC0A_D	GTIOC0B_C	GTIOC0B_A	GTIOC0A_A	GTIOC0A_C	GTIOC5B_A	GTIOC5A_A
00100b	SCI	CTS0_RTS0_D/SS0_D	SCK0_D	—	—	TXD0_A/SDA0_A/MOSI0_A	RXD0_A/SCL0_A/MISO0_A	SCK0_A	CTS0_RTS0_A/SS0_A	TXD0_F/SDA0_F/MOSI0_F
00101b	SCI	RXD1_C/SCL1_C/MISO1_C	SCK9_A	CTS9_RTS9_C/SS9_C	SCK1_B	TXD9_A/SDA9_A/MOSI9_A	TXD1_C/SDA1_C/MOSI1_C	CTS1_RTS1_B/SS1_B	TXD1_B/SDA1_B/MOSI1_B	RXD1_B/SCL1_B/MISO1_B
00110b	SPI	MOSIA_A	SSLB1_A	SSLB2_A	SSLB3_A	RSPCKB_B	SSLA0_A	SSLA1_A	SSLA2_A	SSLA3_A
00111b	IIC	SDA1_A	SDA0_C	—	—	SCL0_A	SDA0_B	SCL0_B	—	—
01000b	KINT	KR02	KR03	—	—	—	—	—	—	—
01001b	CLKOUT/ACMPHS/ACMPLP/RTC	RTCOUT_C	VCOUT_B	—	—	—	—	—	—	—
01010b	CAC/ADC16/SDADC24/DAC12/DAC8	—	—	—	—	CACREF_B	—	—	—	—
01100b	CTSU	TS00	TS01	TS02	TS03	TSCAP_D	TS06	TSCAP_E	TS05	TS04
10000b	CAN	—	—	—	—	CTX0_D	CRX0_B	CTX0_B	—	—
10011b	USBFS	—	—	—	—	USB_VBUS	—	—	—	—
ASEL bit		CMPIN0	—	—	—	—	CMPIN1	—	—	—
ISEL bit		IRQ0_A	IRQ5_B	—	—	IRQ1_B	IRQ1_A	IRQ7_A	—	—
DSCR1, DSCR bit		L/M	L/M	L/M	L/M	L/M/M (IIC, SPI)	L/M/M (IIC, SPI)	L/M/M (IIC, SPI)	L/M	L/M
NCODR bit		✓	✓	—	—	✓	✓	✓	✓	✓
PCR bit		✓	✓	✓	✓	✓	✓	✓	✓	✓
64-pin product		✓	✓	✓	✓	✓	✓	✓	✓	✓
48-pin product		✓	✓	—	—	✓	✓	✓	—	—
40-pin product		✓	—	—	—	✓	✓	—	—	—
36-pin product		✓	—	—	—	✓	✓	—	—	—
32-pin product		✓	—	—	—	✓	✓	—	—	—

✓: Available
—: Setting prohibited
L/M: Low drive/Middle drive
L/M: Low drive/Middle drive/Middle drive for IIC Fast mode (IIC)

Table 18.11 Register settings for input/output pin function (PORT5)

PSEL[4:0] settings	Function	Pin		
		P500	P501	P502
00000b (initial)	Hi-Z/SWD	Hi-Z		
00011b	GPT	GTIOC5A_C	GTIOC5B_C	—
00100b	SCI	RXD0_D/ SCL0_D/ MISO0_D	TXD0_E/ SDA0_E/ MOSI0_E	CTS0_RTS0_E/ SS0_E
01010b	CAC/ADC16/ SDADC24/ DAC12/DAC8	DA12_0	—	—
01100b	CTSU	TS24	TS25	—
ASEL bit		AN000/ DA12_0/ AMP0+/ IVCMP0	AN001/ AMP0-/ IVREF0	AN002/ AMP0O
ISEL bit		IRQ3_C	IRQ2_C	IRQ1_C
DSCR bit		L/M	L/M	L/M
NCODR bit		✓	✓	—
PCR bit		✓	✓	✓
64-pin product		✓	✓	✓
48-pin product		✓	✓	✓
40-pin product		✓	✓	✓
36-pin product		✓	✓	✓
32-pin product		✓	✓	✓

✓: Available
—: Setting prohibited
L/M: Low drive/Middle drive

Table 18.12 Register settings for input/output pin function (PORT9)

PSEL[4:0] settings	Function	Pin	
		P914	P915
00000b (initial)	Hi-Z/SWD	Hi-Z	
ASEL bit		—	—
ISEL bit		—	—
DSCR bit		—	—
NCODR bit		—	—
PCR bit		—	—
64-pin product		✓	✓
48-pin product		✓	✓
40-pin product		✓	✓
36-pin product		✓	✓
32-pin product		—	—

✓: Available
—: Setting prohibited

19. Key Interrupt Function (KINT)

19.1 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 19.1 shows the pin assignment for key interrupt detection, Table 19.2 shows the function configuration, and Figure 19.1 shows a block diagram.

Table 19.1 Assignment of key interrupt detection pins

Key interrupt mode control n (n = 0-7)	Description
KRM0	Controls KR00 signal in 1-bit units
KRM1	Controls KR01 signal in 1-bit units
KRM2	Controls KR02 signal in 1-bit units
KRM3	Controls KR03 signal in 1-bit units
KRM4	Controls KR04 signal in 1-bit units
KRM5	Controls KR05 signal in 1-bit units
KRM6	Controls KR06 signal in 1-bit units
KRM7	Controls KR07 signal in 1-bit units

Table 19.2 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)

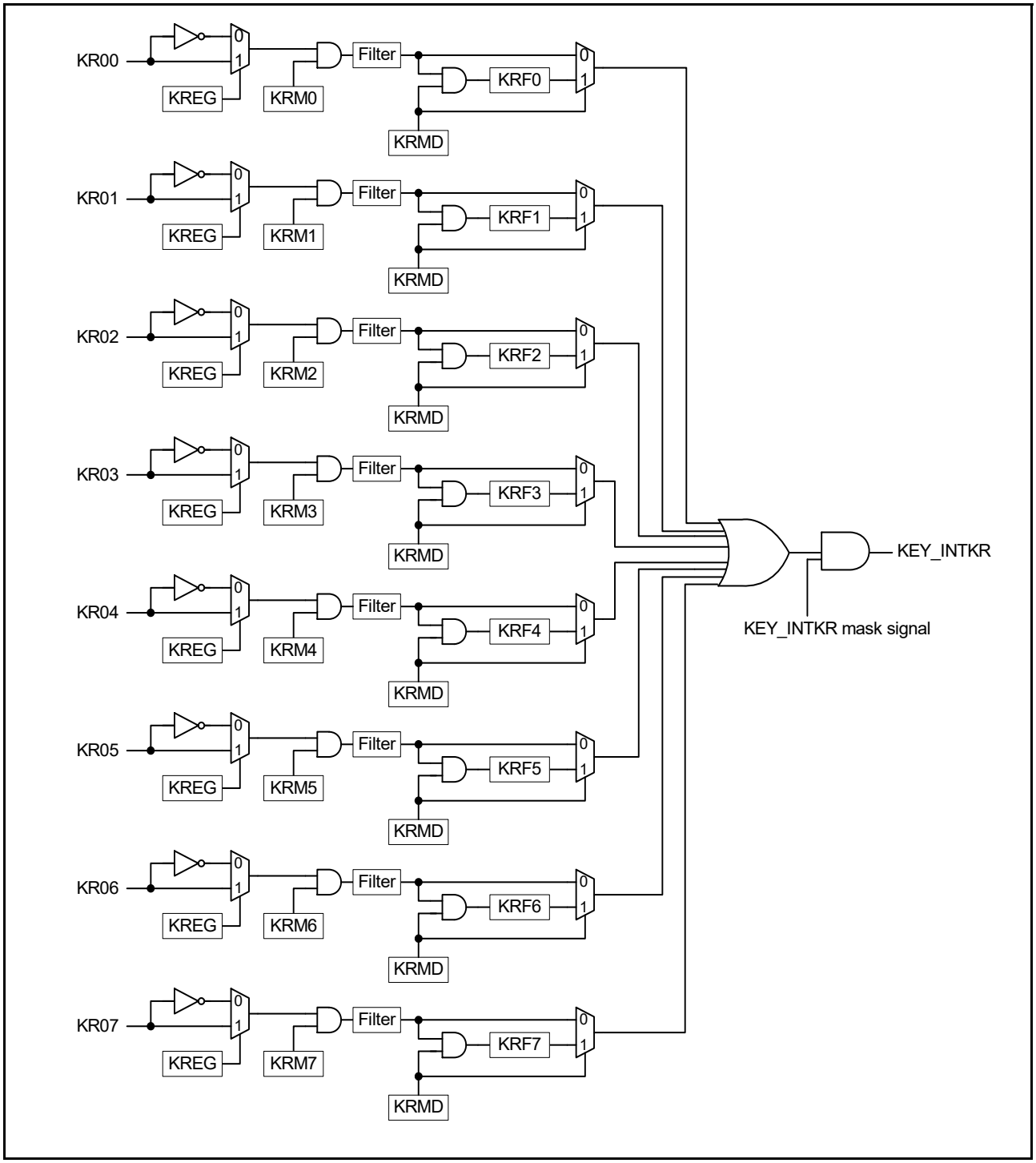


Figure 19.1 Key interrupt function block diagram

In Figure 19.1, 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.

19.2 Register Descriptions

19.2.1 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	Detection Edge Selection (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.

19.2.2 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 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.

19.2.3 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 function. For details, see section 18, I/O Ports.
Key interrupts can be assigned in the PmnPFS.PSEL[4:0] bits. For details, see section 18, 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 (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.

19.3 Operation

19.3.1 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.

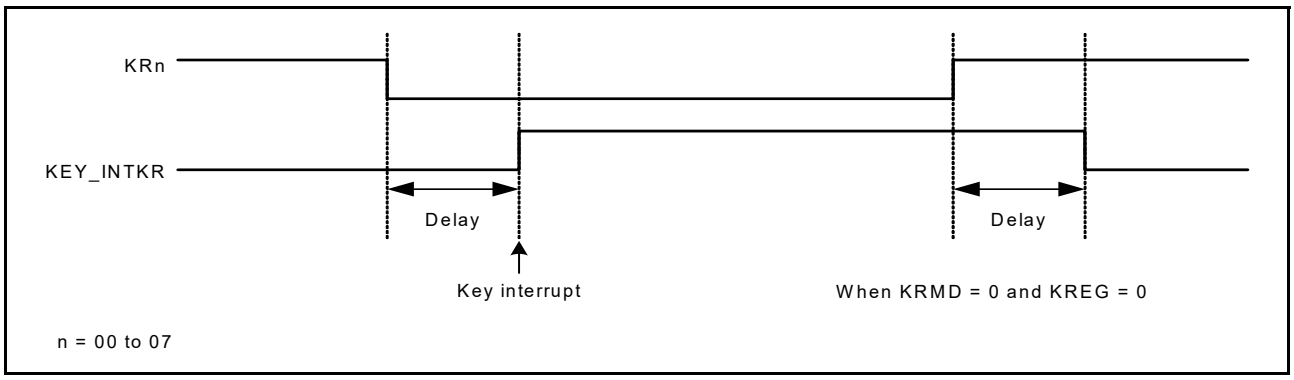


Figure 19.2 Operation of KEY_INTKR signal when key interrupt is input to a single channel

Figure 19.3 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 (when KREG is 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 19.3.

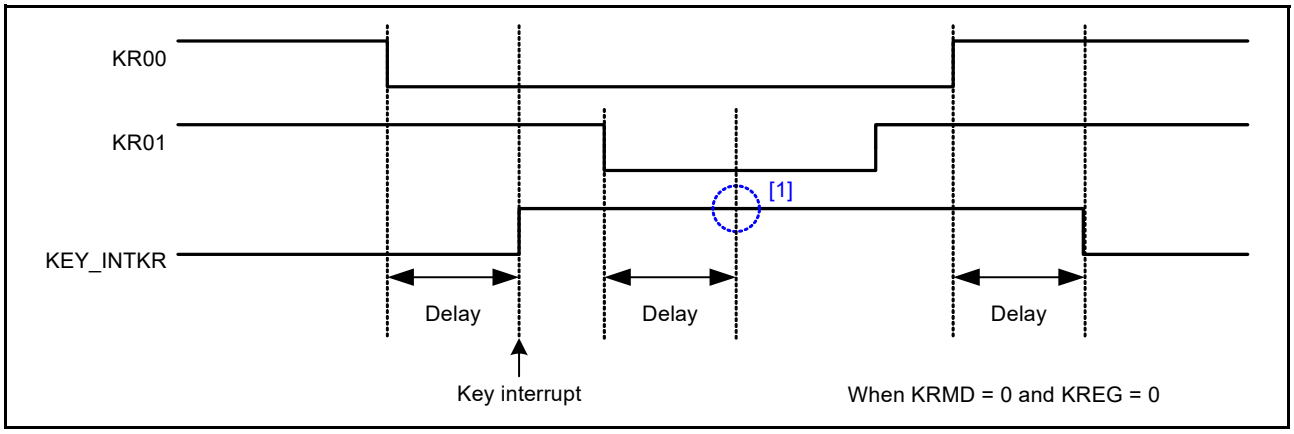


Figure 19.3 Operation of KEY_INTKR signal when key interrupts are input to multiple channels

19.3.2 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 19.4 shows, only one interrupt is generated each time a falling edge is input to one channel, that is, when KREG = 0, regardless of whether the KRFn bit is cleared before or after a rising edge is input.

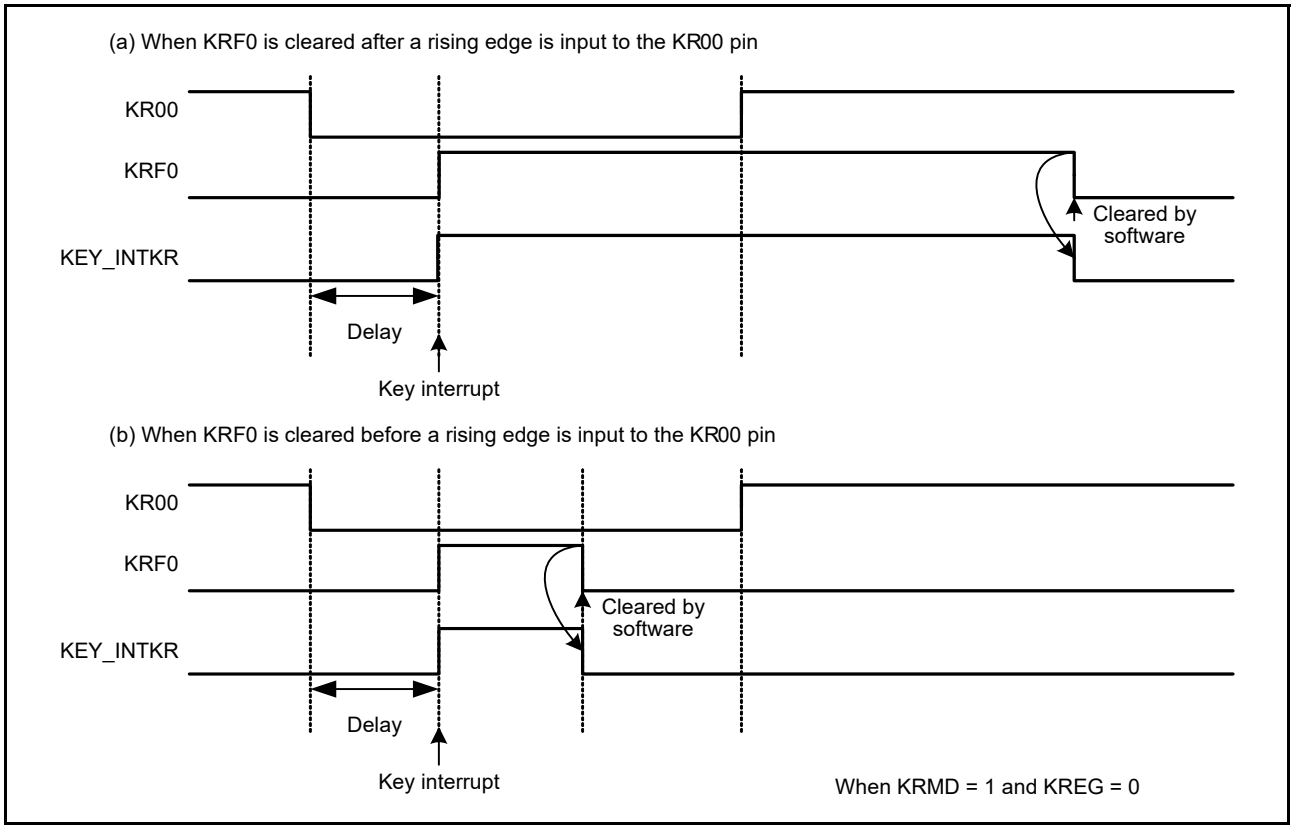


Figure 19.4 Basic operation of KEY_INTKR signal when key interrupt flag is used

Figure 19.5 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 (when KREG = 0). The KRF1 bit is set when the KRF0 bit is cleared. The KEY_INTKR signal is negated 1 PCLKB clock cycle, after the KRF0 bit is cleared. See [1] in Figure 19.5. Also, after a falling edge is input to the KR05 pin, the KRF5 bit is set. The KRF1 bit is cleared at time [2] in the figure. The KEY_INTKR signal is negated 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.

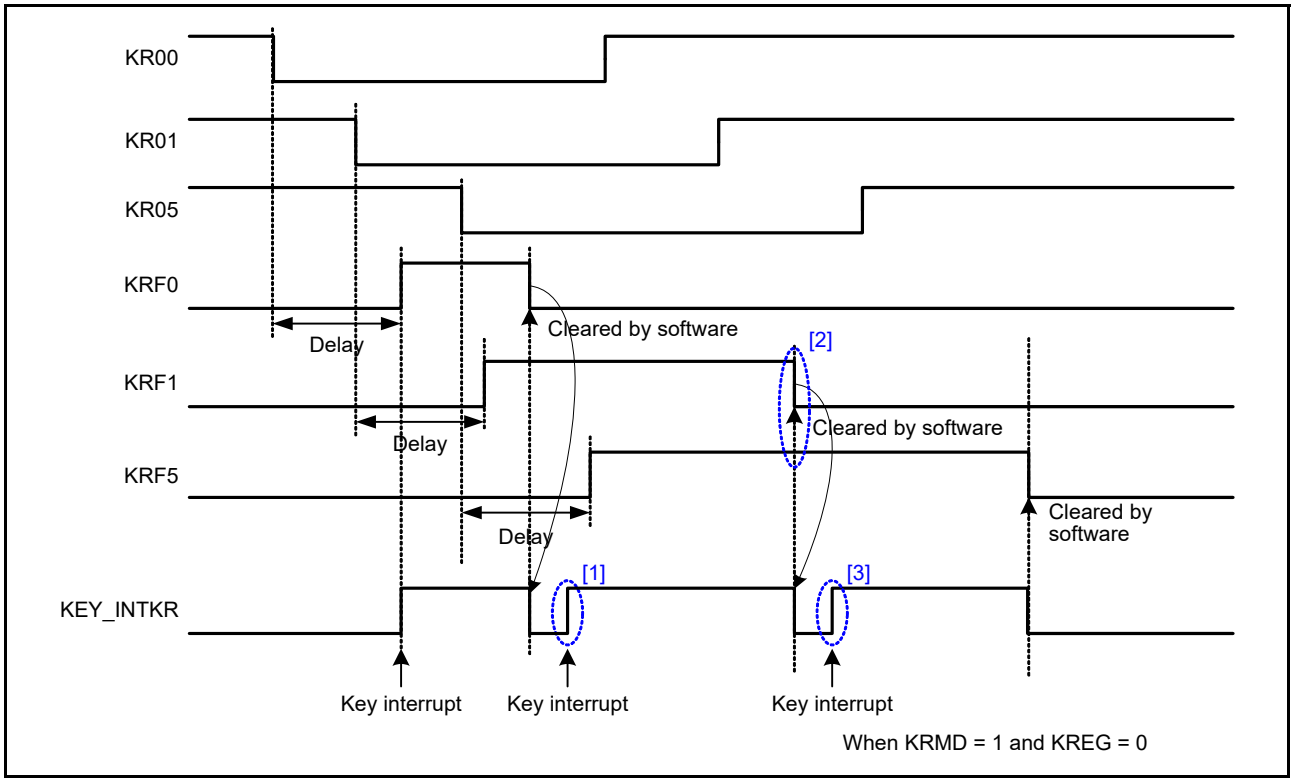


Figure 19.5 Operation of KEY_INTKR signal when key interrupts are input to multiple channels

19.4 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 mode and Software Standby mode, the KRMD bit must be set to 1
- When the key interrupt function (KINT) is assigned to a pin, this pin input is always enabled in the Software Standby mode, and if this pin level changes, the associated KRFn can be set. Therefore, a key interrupt might occur on canceling 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 can be set.

20. Port Output Enable for GPT (POEG)

20.1 Overview

The Port Output Enable for GPT (POEG) function can place the General PWM Timer (GPT) output pins in the output-disable state in one of the following ways:

- Input level detection of the GTETRGn (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 20.1 lists the POEG specifications, Figure 20.1 shows a block diagram, and Table 20.2 lists the input pins.

Table 20.1 POEG specifications

Parameter	Specifications
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 GTIOCmA and GTIOCmB (m = 0 to 6) pins 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 GTIOCmA and GTIOCmB pins are output-disabled.
Output-disable control through the comparator (ACMPHS) interrupt request detection	The GPT output pins can be disabled when an interrupt request is generated by a change in the output results of any of the comparators
Output-disable control through oscillation stop detection	The GPT output pins can be disabled when oscillation of the clock generation circuit stops
Output-disable control through 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 or ACMPHS.
External trigger output to the 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 • Signal state after polarity and filter selection can be monitored.

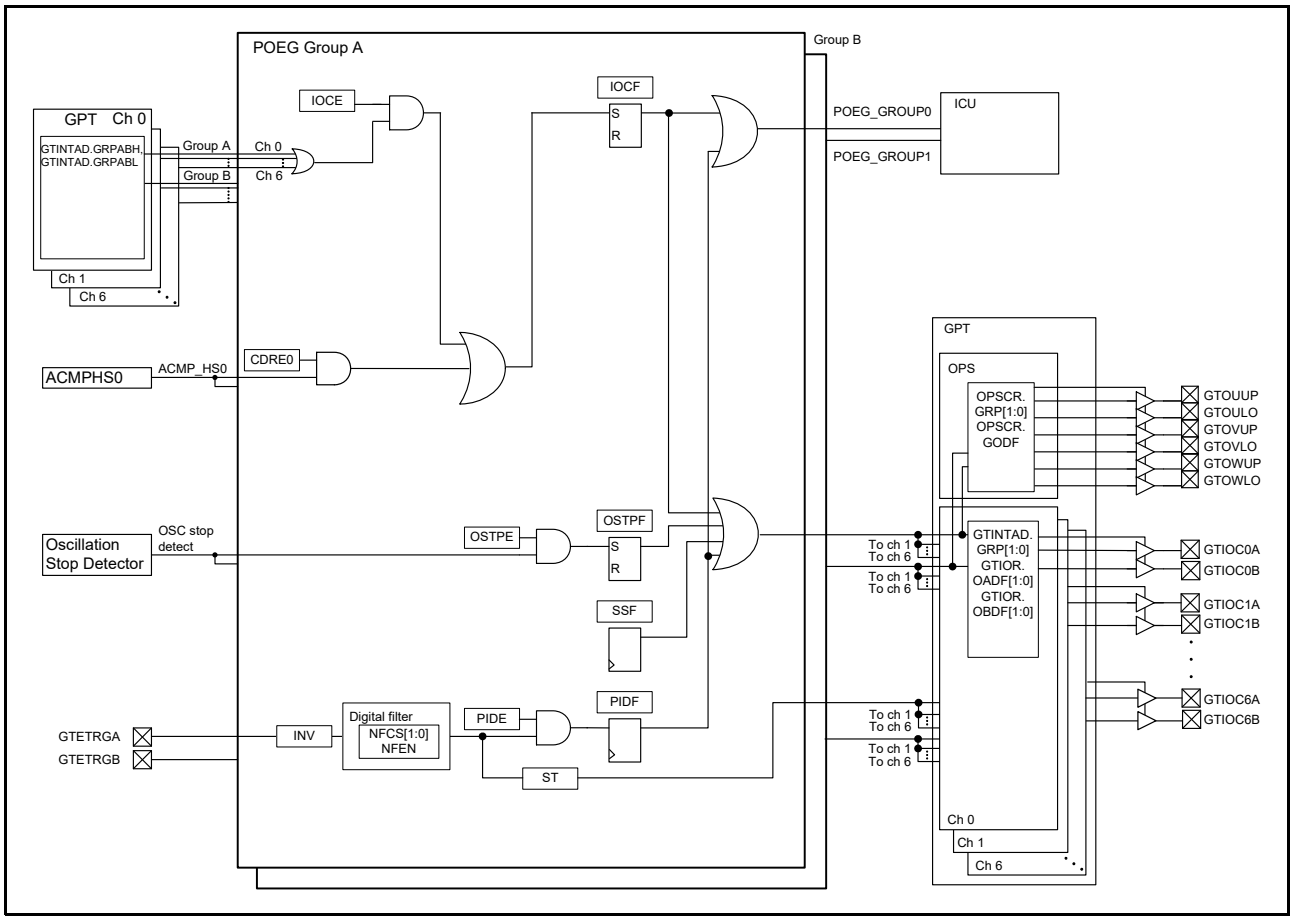


Figure 20.1 POEG block diagram

Table 20.2 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

20.2 Register Descriptions

20.2.1 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
	—	—	—	—	—	—	—	CDRE0	—	OSTPE	IOCE	PIDE	SSF	OSTPF	IOCF	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PIDF	Port Input Detection Flag	0: No output-disable request from the GTETRGn pin occurred 1: Output-disable request from the GTETRGn pin occurred.	R(/W)*1
b1	IOCF	Detection Flag for GPT or ACMPHS Output-Disable Request	0: No output-disable request from the GPT disable request or the ACMPHS occurred 1: Output-disable request from the GPT disable request or the ACMPHS occurred.	R(/W)*1
b2	OSTPF	Oscillation Stop Detection Flag	0: No output-disable request from the oscillation stop detection occurred 1: Output-disable request from the 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	Enable for GPT Output-Disable Request	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
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	CDRE0	ACMP_HS0 Enable	0: Comparator 0 disable requests disabled 1: Comparator 0 disable requests enabled.	R/W*2
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	ST	GTETRGn Input Status Flag	0: GTETRGn input after filtering is 0 1: GTETRGn input after filtering is 1.	R
b27 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	INV	GTETRGn Input Reverse	0: GTETRGn input 1: GTETRGn input reversed.	R/W
b29	NFEN	Noise Filter Enable	0: Noise filtering disabled 1: Noise filtering enabled.	R/W
b31, b30	NFCS[1:0]	Noise Filter Clock Select	b31 b30 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 and POEGGB 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 and POEGGB registers.

20.3 Output-Disable Control Operation

If any of the following conditions is satisfied, the GTIOCxA, GTIOCxB, and the 3-phase 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 register. The GTINTAD.GRPABH and GTINTAD.GRPABL settings apply to the group selected in the GPT registers GTINTAD.GRP[1:0] and OPSCR.GRP[1:0].
- Comparator (ACMPHS) interrupt request detection
  Comparator interrupt detection is activated when any of the POEGGn.CDREi (i = 0, 4, 5) is 1. When the associated comparator interrupt is generated, the GPT output pins are disabled. POEGGn.IOCF indicates the detection status.
- Oscillation stop detection for the clock generation circuit
  When POEGGn.OSTPE is 1, the POEGGn.OSTPF flag is set to 1.
- SSF bit setting
  When POEGGn.SSF is set to 1, the GPT and PWM output is disabled.

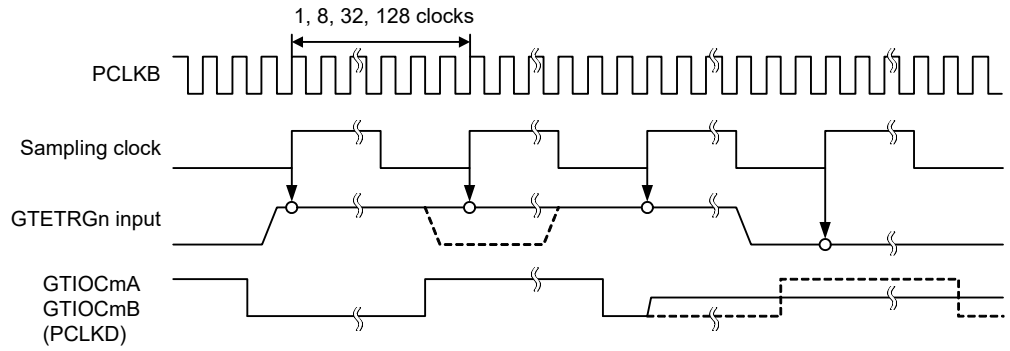
The output-disable state is controlled in the GPT. The output-disable of the GTIOCxA and GTIOCxB pins is set in the GTINTAD.GRP[1:0], GTIOR.OADF[1:0], and GTIOR.OBDF[1:0] bits in the GPT. The output-disable of the 3-phase PWM output for the BLDC motor control pins is set to the OPSCR.GRP[1:0] and OPSCR.GODF bits in GPT_OPS.

20.3.1 Pin Input Level Detection Operation

If the input conditions set in POEGGn.PIDE, POEGGn.NFCS[1:0], POEGGn.NFEN, and POEGGn.INV occur on the GTETRGn pins, the GPT output pins are output-disabled.

20.3.1.1 Digital filter

Figure 20.2 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], 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 output, changes of the levels on the GTETRGn pins are ignored.



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 20.2 Example of digital filter operation

20.3.2 Output-Disable Request from the GPT

For details on the operation, see the description for GTIOC Pin Output Negate Control in section 21, General PWM Timer (GPT).

20.3.3 Comparator Interrupt Detection

If POEGGn.CDREi (i = 0, 4, 5) is 1 when an associated comparator interrupt request is generated, the GPT output pins are output-disabled for each group. The status flag is POEGGn.IOCF, which is shared with GPT output-disable detection.

20.3.4 Output-Disable Control Using 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.

20.3.5 Output-Disable Control Using Registers

The GPT output pins can be directly controlled by writing to the Software Stop Flag, POEGGn.SSF.

20.3.6 Release from Output Disable

To release the GPT output pins in the output-disabled 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 20.3 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.

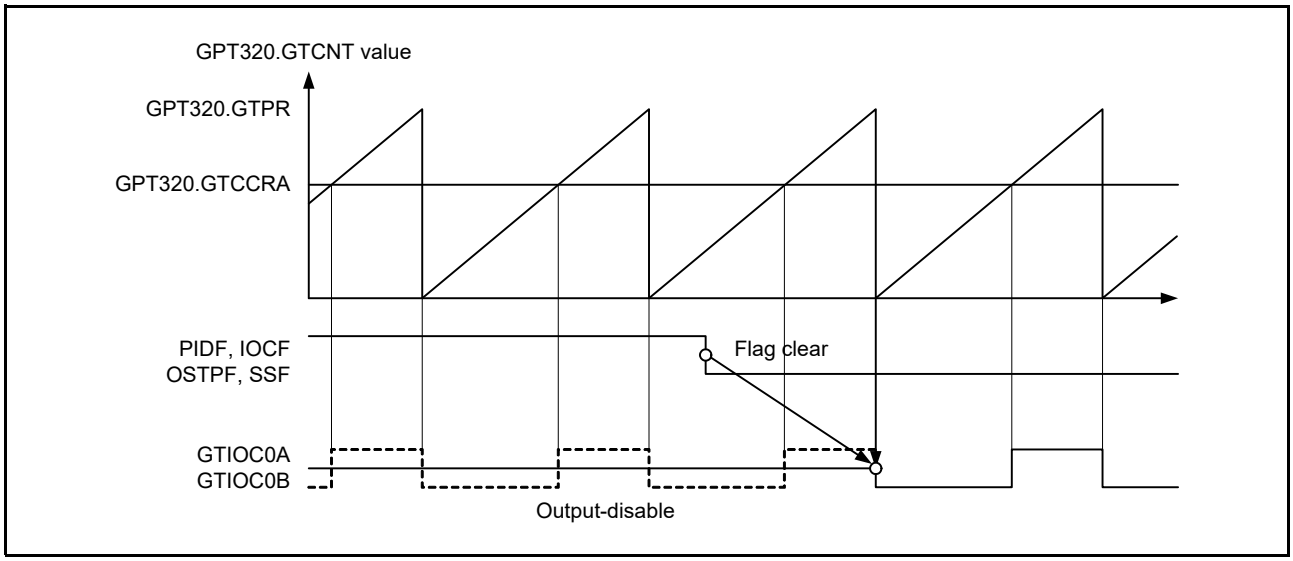


Figure 20.3 Output-disable release timing for the GPT pin outputs

20.4 Interrupt Sources

The POEG generates an interrupt request for the following factors:

- Output-disable control by the input level detection
- Output-disable request from the GPT
- Output-disable request from the comparator interrupt request detection.

Table 20.3 lists the conditions for interrupt requests.

Table 20.3 Interrupt sources and condition

Interrupt source	Symbol	Associated flag	Trigger conditions
POEG group A interrupt	POEG_GROUP0	POEGGA.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from a comparator interrupt 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
			An output-disable request from a comparator interrupt request occurred
		POEGGB.PIDF	An output-disable request from the GTETRGB pin occurred

20.5 External Trigger Output to the GPT

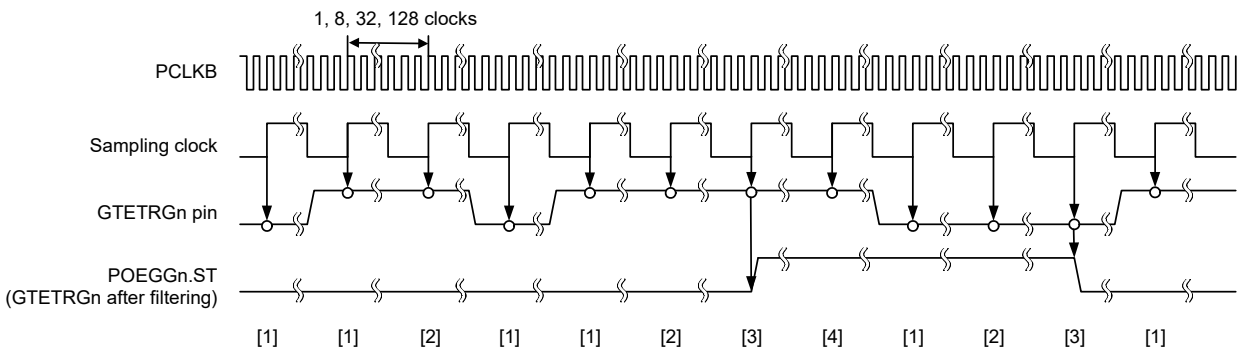
The POEG outputs the GTETRGn signals as the GPT operation trigger signal for the following operations:

- Count start
- Count stop
- Count clear
- Up-count

- Down-count
- Input capture.

For the POEGGn.INV polarity setting signal, when the same level is input three times continuously with the sampling clock selected in the POEGGn.NFCS[1:0] bits, that value is output. Set the control registers the same as for the input level detection operation described in section 20.3.1, Pin Input Level Detection Operation. The state after filtering can be monitored in POEGGn.ST.

Figure 20.4 shows the output timing of an external trigger to the GPT.



Note: Each channel output can be set in the GPT settings.
Polarity can be reversed in the POEGGn.INV setting.

Figure 20.4 Output timing of external trigger to the GPT

20.6 Usage Notes

20.6.1 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.

20.6.2 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[4:0] settings. When the pin is specified as a general I/O pin, the POEG does not perform output-disable control.

21. General PWM Timer (GPT)

21.1 Overview

The General PWM Timer (GPT) is a 32-bit timer with one GPT32 channel and a 16-bit timer with six 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 21.1 lists the GPT specifications, Table 21.2 shows the GPT functions, and Figure 21.1 shows a block diagram.

Table 21.1 GPT specifications

Parameter	Specifications
Functions	• 32 bits × 1 channel • 16 bits × 6 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 I/O 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 four 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 two 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 D 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.

Table 21.2 GPT functions (1 of 2)

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	GTIOCnA GTIOCnB (n = 0 to 6)
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

Table 21.2 GPT functions (2 of 2)

Parameter		GPT32, GPT16
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		6 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) • GTCNT overflow (GTPR compare match) (GPTn_OVF) • GTCNT underflow (GPTn_UDF). Note: n = 0 to 6
Event linking (ELC) function		Available
Noise filtering function		Available

Note 1. GTETRGn connects to GPT through the POEG module. Therefore, to use the GPT function, supply the POEG clock by clearing the MSTPCRD.MSTPD14 bit.

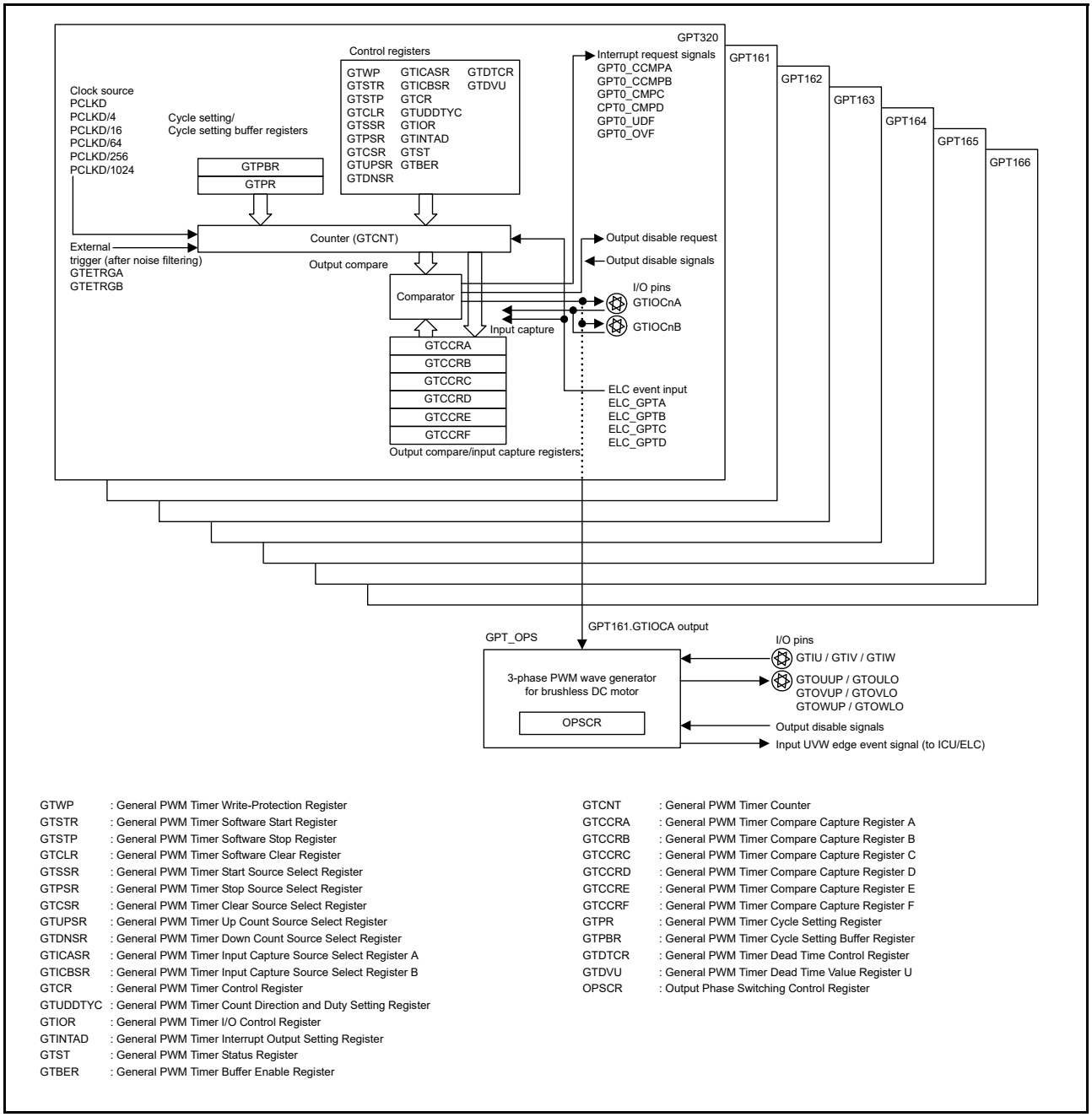


Figure 21.1 GPT block diagram

Figure 21.2 shows an example using multiple GPTs.

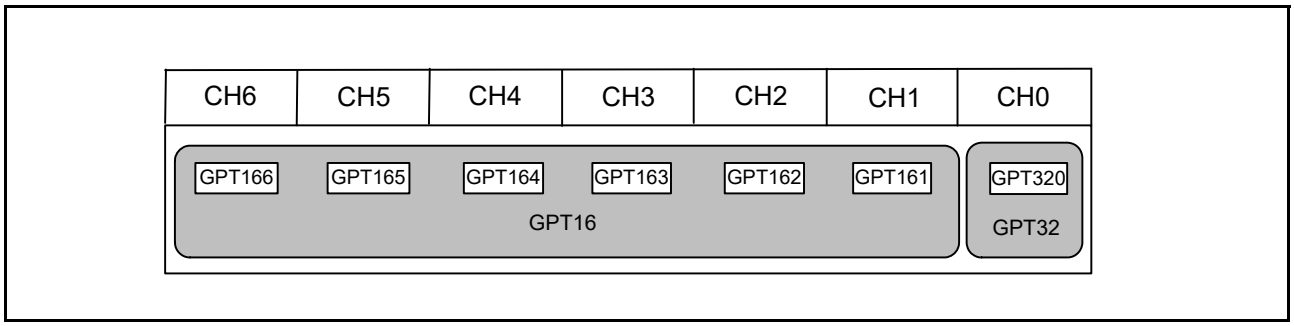


Figure 21.2 Association between GPT channels and module names

Table 21.3 lists the I/O pins.

Table 21.3 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
GPT161	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
GPT162	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
GPT163	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
GPT166	GTIOC6A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC6B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
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)

21.2 Register Descriptions

Table 21.4 lists the registers in the GPT.

Table 21.4 GPT registers

Module symbol	Register name	Register symbol	Reset value	Address	Access size
GPT320, GPT16m	General PWM Timer Write Protection Register	GTWP	0000 0000h	4007 8000h + 0100h × m	32
	General PWM Timer Software Start Register	GTSTR	0000 0000h	4007 8004h + 0100h × m	32
	General PWM Timer Software Stop Register	GTSTP	FFFF FFFFh	4007 8008h + 0100h × m	32
	General PWM Timer Software Clear Register	GTCLR	0000 0000h	4007 800Ch + 0100h × m	32
	General PWM Timer Start Source Select Register	GTSSR	0000 0000h	4007 8010h + 0100h × m	32
	General PWM Timer Stop Source Select Register	GTPSR	0000 0000h	4007 8014h + 0100h × m	32
	General PWM Timer Clear Source Select Register	GTCSR	0000 0000h	4007 8018h + 0100h × m	32
	General PWM Timer Up Count Source Select Register	GTUPSR	0000 0000h	4007 801Ch + 0100h × m	32
	General PWM Timer Down Count Source Select Register	GTDNSR	0000 0000h	4007 8020h + 0100h × m	32
	General PWM Timer Input Capture Source Select Register A	GTICASR	0000 0000h	4007 8024h + 0100h × m	32
	General PWM Timer Input Capture Source Select Register B	GTICBSR	0000 0000h	4007 8028h + 0100h × m	32
	General PWM Timer Control Register	GTCR	0000 0000h	4007 802Ch + 0100h × m	32
	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	0000 0001h	4007 8030h + 0100h × m	32
	General PWM Timer I/O Control Register	GTIOR	0000 0000h	4007 8034h + 0100h × m	32
	General PWM Timer Interrupt Output Setting Register	GTINTAD	0000 0000h	4007 8038h + 0100h × m	32
	General PWM Timer Status Register	GTST	0000 8000h	4007 803Ch + 0100h × m	32
	General PWM Timer Buffer Enable Register	GTBER	0000 0000h	4007 8040h + 0100h × m	32
	General PWM Timer Counter	GTCNT	0000 0000h	4007 8048h + 0100h × m	32
	General PWM Timer Compare Capture Register A	GTCCRA	FFFF FFFFh*1	4007 804Ch + 0100h × m	32
	General PWM Timer Compare Capture Register B	GTCCRB	FFFF FFFFh*1	4007 8050h + 0100h × m	32
	General PWM Timer Compare Capture Register C	GTCCRC	FFFF FFFFh*1	4007 8054h + 0100h × m	32
	General PWM Timer Compare Capture Register E	GTCCRE	FFFF FFFFh*1	4007 8058h + 0100h × m	32
	General PWM Timer Compare Capture Register D	GTCCRD	FFFF FFFFh*1	4007 805Ch + 0100h × m	32
	General PWM Timer Compare Capture Register F	GTCCRF	FFFF FFFFh*1	4007 8060h + 0100h × m	32
	General PWM Timer Cycle Setting Register	GTPR	FFFF FFFFh*1	4007 8064h + 0100h × m	32
	General PWM Timer Cycle Setting Buffer Register	GTPBR	FFFF FFFFh*1	4007 8068h + 0100h × m	32
	General PWM Timer Dead Time Control Register	GTDTCR	0000 0000h	4007 8088h + 0100h × m	32
	General PWM Timer Dead Time Value Register U	GTDVU	FFFF FFFFh*1	4007 808Ch + 0100h × m	32
GPT_OPS	Output Phase Switching Control Register	OPSCR	0000 0000h	4007 8FF0h	32

m = 0 to 6, n = 1 to 6

Note 1. The reset value of GPT16m is 0000 FFFFh.

21.2.1 General PWM Timer Write Protection Register (GTWP)

Address(es): GPT320.GTWP 4007 8000h,
GPT16m.GTWP 4007 8000h + 0100h × m (m = 1 to 6)

	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, writes to the WP bit are 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 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.

21.2.2 General PWM Timer Software Start Register (GTSTR)

Address(es): GPT320.GTSTR 4007 8004h,
GPT16m.GTSTR 4007 8004h + 0100h × m (m = 1 to 6)

	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 6	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 (n = 0 to 6).

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 where 1 is written. Writing 0 has no effect on the status of the GTCNT counter and the value of GTSTR register.

For the association between GTSTR bit number and a channel number, see Figure 21.2.

CSTRT[6:0] bits (Channel n GTCNT Count Start) (n = 0 to 6)

The CSTRT[6:0] bits start channel n of the GTCNT counter operation. Writing to the GTSTR.CSTRTn bit (n = 0 to 6) has no effect unless GPTm.GTSSR.CSTRT bit is set to 1 (m = 320, 161 to 166).

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.

21.2.3 General PWM Timer Software Stop Register (GTSTP)

Address(es): GPT320.GTSTP 4007 8008h,
GPT16m.GTSTP 4007 8008h + 0100h × m (m = 1 to 6)

	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 6	CSTOP 5	CSTOP 4	CSTOP 3	CSTOP 2	CSTOP 1	CSTOP 0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTSTP stops the GTCNT counter operation for each channel n, where n = 0 to 6.

The GTSTP bit number represents the channel number. The GTSTP register is shared by all of the channels. The GTCNT counter stops for the channel associated with the GTSTP bit 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 GTSTP bit number and a channel number, see Figure 21.2.

CSTOP[6:0] bits (Channel n GTCNT Count Stop) (n = 0 to 6)

The CSTOP[6:0] bits stop channel n of the GTCNT counter operation. Writing to the GTSTP.CSTOPn bit (n = 0 to 6) has no effect unless the GPTm.GTPSR.CSTOPn bit is set to 1 (m = 320, 161 to 166). 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.

21.2.4 General PWM Timer Software Clear Register (GTCLR)

Address(es): GPT320.GTCLR 4007 800Ch,
GPT16m.GTCLR 4007 800Ch + 0100h × m (m = 1 to 6)

	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 6	CCLR 5	CCLR 4	CCLR 3	CCLR 2	CCLR 1	CCLR 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The GTCLR is a write-only register that clears the GTCNT counter operation for each channel n, where n = 0 to 6.

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 GTCNT counter.

For the association between GTCLR bit number and a channel number, see Figure 21.2.

CCLR[6:0] bits (Channel n GTCNT Count Clear) (n = 0 to 6)

Channel n of the GTCNT counter value is cleared on writing 1 to the CCLRn bit. This bit is read as 0.

21.2.5 General PWM Timer Start Source Select Register (GTSSR)

Address(es): GPT320.GTSSR 4007 8010h,
GPT16m.GTSSR 4007 8010h + 0100h × m (m = 1 to 6)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CSTRT	—	—	—	—	—	—	—	—	—	—	—	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 event input 1: Counter start enabled at the ELC_GPTA event input.	R/W
b17	SSELCB	ELC_GPTB Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTB event input 1: Counter start enabled at the ELC_GPTB event input.	R/W
b18	SSELCC	ELC_GPTC Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTC event input 1: Counter start enabled at the ELC_GPTC event input.	R/W
b19	SSELCD	ELC_GPTD Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTD event input 1: Counter start enabled at the ELC_GPTD event input.	R/W
b30 to b20	—	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 the GTETRGA pin input.

SSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Start Enable)

The SSGTRGAF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGA pin input.

SSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Start Enable)

The SSGTRGBR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGB pin input.

SSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Start Enable)

The SSGTRGBF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGB pin input.

SSCARBL bit (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 the 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 the 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 the 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 the 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 the 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 the 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 the 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 the GTIOCB pin input, when GTIOCA input is 1.

SSELCm bit (ELC_GPTm Event Source Counter Start Enable) (m = A to D)

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 the GTSTR register.

21.2.6 General PWM Timer Stop Source Select Register (GTPSR)

Address(es): GPT320.GTPSR 4007 8014h,
GPT16m.GTPSR 4007 8014h + 0100h × m (m = 1 to 6)

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PSCBFAH	PSCBFAL	PSCBRAH	PSCBRAL	PSCAFBH	PSCAFBL	PSCARBH	PSCARBL	—	—	—	—	PSGTRGBF	PSGTRGBR	PSGTRGAF	PSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PSGTRGAR	GTETRGA Pin Rising Input Source Counter Stop Enable	0: Counter stop disabled on the rising edge of GTETRGA input 1: Counter stop enabled on the rising edge of GTETRGA input.	R/W
b1	PSGTRGAF	GTETRGA Pin Falling Input Source Counter Stop Enable	0: Counter stop disabled 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
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

Bit	Symbol	Bit name	Description	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 event input 1: Counter stop enabled at the ELC_GPTA event input.	R/W
b17	PSELCB	ELC_GPTB Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTB event input 1: Counter stop enabled at the ELC_GPTB event input.	R/W
b18	PSELCC	ELC_GPTC Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTC event input 1: Counter stop enabled at the ELC_GPTC event input.	R/W
b19	PSELCD	ELC_GPTD Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTD event input 1: Counter stop enabled at the ELC_GPTD event input.	R/W
b30 to b20	—	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 the GTETRGA pin input.

PSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Stop Enable)

The PSGTRGAF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGA pin input.

PSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Stop Enable)

PSGTRGBR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGB pin input.

PSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Stop Enable)

The PSGTRGBF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGB pin input.

PSCARBL bit (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 the GTIOCA pin input, when GTIOCB input is 0.

PSCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Stop Enable)

This bit enables or disables the GTCNT counter stop on the rising edge of the 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 the 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 the GTIOCA pin input, when GTIOCB input is 1.

PSCBRAL bit (GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Stop Enable)

The PSCBRA bit enables or disables the GTCNT counter stop on the rising edge of the 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 the 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 the 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 the GTIOCB pin input, when GTIOCA input is 1.

PSELCm bit (ELC_GPTm Event Source Counter Stop Enable) (m = A to D)

The PSELCm bit enables or disables the GTCNT counter stop at the ELC_GPTm event input.

CSTOP bit (Software Source Counter Stop Enable)

The CSTOP bit enables or disables the GTCNT counter stop by the GTSTP register.

21.2.7 General PWM Timer Clear Source Select Register (GTCSR)

Address(es): GPT320.GTCSR 4007 8018h,
GPT16m.GTCSR 4007 8018h + 0100h × m (m = 1 to 6)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CCLR	—	—	—	—	—	—	—	—	—	—	—	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

Bit	Symbol	Bit name	Description	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: Disable counter clear on the rising edge of GTETRGB input 1: Enable counter clear 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
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 event input 1: Counter clear enabled at the ELC_GPTA event input.	R/W
b17	CSELCB	ELC_GPTB Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTB event input 1: Counter clear enabled at the ELC_GPTB event input.	R/W
b18	CSELCC	ELC_GPTC Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTC event input 1: Counter clear enabled at the ELC_GPTC event input.	R/W
b19	CSELCD	ELC_GPTD Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTD event input 1: Counter clear enabled at the ELC_GPTD event input.	R/W
b30 to b20	—	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 the GTETRGA pin input.

CSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Clear Enable)

The CSGTRGAF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGA pin input.

CSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Clear Enable)

The CSGTRGBR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGB pin input.

CSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Clear Enable)

The CSGTRGBF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGB pin input.

CSCARBL bit (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 the 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 the 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 the 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 the GTCNT counter clear on the falling edge of the 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 the GTCNT counter clear on the rising edge of the 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 the GTCNT counter clear on the rising edge of the 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 the GTCNT counter clear on the falling edge of the 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 the GTCNT counter clear on the falling edge of the GTIOCB pin input, when GTIOCA input is 1.

CSELCm bit (ELC_GPTm Event Source Counter Clear Enable) (m = A to D)

The CSELCm bit enables or disables the GTCNT counter clear at the ELC_GPTm event input.

CCLR bit (Software Source Counter Clear Enable)

The CCLR bit enables or disables the GTCNT counter clear by the GTCLR register.

21.2.8 General PWM Timer Up Count Source Select Register (GTUPSR)

Address(es): GPT320.GTUPSR 4007 801Ch,
GPT16m.GTUPSR 4007 801Ch + 0100h × m (m = 1 to 6)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	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
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

Bit	Symbol	Bit name	Description	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 event input 1: Counter count up enabled at the ELC_GPTA event input.	R/W
b17	USELCB	ELC_GPTB Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTB event input 1: Counter count up enabled at the ELC_GPTB event input.	R/W
b18	USELCC	ELC_GPTC Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTC event input 1: Counter count up enabled at the ELC_GPTC event input.	R/W
b19	USELCD	ELC_GPTD Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTD event input 1: Counter count up enabled at the ELC_GPTD event input.	R/W
b31 to b20	—	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 the GTETRGA pin input.

USGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Up Enable)

The USGTRGAF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGA pin input.

USGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Up Enable)

The USGTRGBR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGB pin input.

USGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Up Enable)

The USGTRGBF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGB pin input.

USCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Count Up Enable)

The USCARBL bit enables or disables GTCNT counter count up on the rising edge of GTIOCA pin input, when GTIOCB input is 0.

USCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Count Up Enable)

The USCARBH bit enables or disables the GTCNT counter count up on the rising edge of the 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 the 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 the 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 the 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 the GTIOCB pin input, when the 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 the GTIOCB pin input, when the 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 the GTIOCB pin input, when the GTIOCA input is 1.

USELCm bit (ELC_GPTm Event Source Counter Count Up Enable) (m = A to D)

The USELCm bit enables or disables the GTCNT counter count up at the ELC_GPTm event input.

21.2.9 General PWM Timer Down Count Source Select Register (GTDNSR)

Address(es): GPT320.GTDNSR 4007 8020h,
GPT16m.GTDNSR 4007 8020h + 0100h × m (m = 1 to 6)

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DSCBFAH	DSCBFAL	DSCBRAH	DSCBRAL	DSCAFBH	DSCAFBL	DSCARBH	DSCARBL	—	—	—	—	DSGTRGBF	DSGTRGBR	DSGTRGAF	DSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DSGTRGAR	GTETRGA Pin Rising Input Source Counter Count Down Enable	0: Counter count down disabled on the rising edge of GTETRGA input 1: Counter count down enabled on the rising edge of GTETRGA input.	R/W
b1	DSGTRGAF	GTETRGA Pin Falling Input Source Counter Count Down Enable	0: Counter count down disabled 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

Bit	Symbol	Bit name	Description	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
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 event input 1: Counter count down enabled at the ELC_GPTA event input.	R/W
b17	DSELCB	ELC_GPTB Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTB event input 1: Counter count down enabled at the ELC_GPTB event input.	R/W
b18	DSELCC	ELC_GPTC Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTC event input 1: Counter count down enabled at the ELC_GPTC event input.	R/W
b19	DSELCD	ELC_GPTD Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTD event input 1: Counter count down enabled at the ELC_GPTD event input.	R/W
b31 to b20	—	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. In this case, GTCR.TPCS has no effect.

DSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Down Enable)

The DSGTRGAR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGA pin input.

DSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Down Enable)

The DSGTRGAF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGA pin input.

DSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Down Enable)

The DSGTRGBR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGB pin input.

DSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Down Enable)

The DSGTRGBF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGB pin input.

DSCARBL bit (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 the GTIOCA pin input, when the 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 the 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 the 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 the 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 the 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 the 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 the GTIOCB pin input, when GTIOCA input is 1.

DSELCm bit (ELC_GPTm Event Source Counter Count Down Enable) (m = A to D)

The DSELCm bit enables or disables the GTCNT counter count down at the ELC_GPTm event input.

21.2.10 General PWM Timer Input Capture Source Select Register A (GTICASR)

Address(es): GPT320.GTICASR 4007 8024h,
GPT16m.GTICASR 4007 8024h + 0100h × m (m = 1 to 6)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	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
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

Bit	Symbol	Bit name	Description	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 event input 1: GTCCRA input capture enabled at the ELC_GPTA event input.	R/W
b17	ASELCB	ELC_GPTB Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTB event input 1: GTCCRA input capture enabled at the ELC_GPTB event input.	R/W
b18	ASELCC	ELC_GPTC Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTC event input 1: GTCCRA input capture enabled at the ELC_GPTC event input.	R/W
b19	ASELCD	ELC_GPTD Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTD event input 1: GTCCRA input capture enabled at the ELC_GPTD event input.	R/W
b31 to b20	—	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 the input capture for GTCCRA on the rising edge of the GTETRGA pin input.

ASGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGAF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGA pin input.

ASGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGBR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGB pin input.

ASGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGBF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGB pin input.

ASCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source GTCCRA Input Capture Enable)

The ASCARBL bit enables or disables the input capture for GTCCRA on the rising edge of the 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 the input capture for GTCCRA on the rising edge of the 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 the input capture for GTCCRA on the falling edge of the 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 the input capture for GTCCRA on the falling edge of the GTIOCA pin input, when the 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 the input capture for GTCCRA on the rising edge of the GTIOCB pin input, when the 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 the input capture for GTCCRA on the rising edge of the 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 the input capture for GTCCRA on the falling edge of the 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 the input capture for GTCCRA on the falling edge of the GTIOCB pin input, when GTIOCA input is 1.

ASELCm bit (ELC_GPTm Event Source Counter GTCCRA Input Capture Enable) (m = A to D)

The ASELCm bit enables or disables the input capture for GTCCRA at the ELC_GPTm event input.

21.2.11 General PWM Timer Input Capture Source Select Register B (GTICBSR)

Address(es): GPT320.GTICBSR 4007 8028h,
GPT16m.GTICBSR 4007 8028h + 0100h × m (m = 1 to 6)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	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

Bit	Symbol	Bit name	Description	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
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 event input 1: GTCCRB input capture enabled at the ELC_GPTA event input.	R/W
b17	BSELCB	ELC_GPTB Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTB event input 1: GTCCRB input capture enabled at the ELC_GPTB event input.	R/W
b18	BSELCC	ELC_GPTC Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTC event input 1: GTCCRB input capture enabled at the ELC_GPTC event input.	R/W
b19	BSELCD	ELC_GPTD Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTD event input 1: GTCCRB input capture enabled at the ELC_GPTD event input.	R/W
b31 to b20	—	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 the input capture for GTCCRB on the rising edge of the GTETRGA pin input.

BSGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGAF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGA pin input.

BSGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGBR bit enables or disables the input capture for GTCCRB on the rising edge of GTETRGB pin input.

BSGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGBF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGB pin input.

BSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source GTCCRB Input Capture Enable)

The BSCARBL bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCA pin input, when the 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 the input capture for GTCCRB on the rising edge of the 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 the input capture for GTCCRB on the falling edge of the 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 the input capture for GTCCRB on the falling edge of the 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 the input capture for GTCCRB on the rising edge of the 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 the input capture for GTCCRB on the rising edge of the 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 the input capture for GTCCRB on the falling edge of the 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 the input capture for GTCCRB on the falling edge of the GTIOCB pin input, when GTIOCA input is 1.

BSELCm bit (ELC_GPTm Event Source Counter GTCCRB Input Capture Enable) (m = A to D)

The BSELCm bit enables or disables the input capture for GTCCRB at the ELC_GPTm event input.

21.2.12 General PWM Timer Control Register (GTCR)

Address(es): GPT320.GTCR 4007 802Ch,
GPT16m.GTCR 4007 802Ch + 0100h × m (m = 1 to 6)

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

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

Bit	Symbol	Bit name	Description	R/W
b0	CST	Count Start	0: Count operation is stopped 1: Count operation is performed.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	MD[2:0]	Mode Select	b18 b16 0 0 0: Saw-wave PWM mode (single buffer or double buffer possible) 0 0 1: Saw-wave one-shot pulse mode (fixed buffer operation) 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer is possible) 1 0 1: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer is 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 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 event 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 GTPSR.CSTOP bit at 1

- The ELC event input or the GTIOCA/GTIOCB/GTETRGn port input event enabled by GTPSR 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 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.

21.2.13 General PWM Timer Count Direction and Duty Setting Register (GTUDDTYC)

Address(es): GPT320.GTUDDTYC 4007 8030h,
GPT16m.GTUDDTYC 4007 8030h + 0100h × m (m = 1 to 6)

	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, b16	OADTY[1:0]	GTIOCA Output Duty Setting	b17 b16 0 x: GTIOCA pin duty depends on compare match 1 0: GTIOCA pin duty 0% 1 1: GTIOCA pin duty 100%.	R/W
b18	OADTYF	Forcible GTIOCA Output Duty Setting	0: Not forcibly set 1: Forcibly set.	R/W
b19	OADTYR	GTIOCA Output Value Selecting after Releasing 0%/100% Duty Setting	0: Apply output value set in 0%/100% duty to GTIOA[3:2] function after releasing 0%/100% duty setting 1: Apply masked compare match output value to GTIOA[3:2] function after releasing 0%/100% duty setting.	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	OBDTY[1:0]	GTIOCB Output Duty Setting	b25 b24 0 x: GTIOCB pin duty is depended on the compare match 1 0: GTIOCB pin duty 0% 1 1: GTIOCB pin duty 100%.	R/W
b26	OBDTYF	Forcible GTIOCB Output Duty Setting	0: Not forcibly set 1: Forcibly set.	R/W
b27	OBDTYR	GTIOCB Output Value Selecting after Releasing 0%/100% Duty Setting	0: Apply output value set in 0%/100% duty to GTIOB[3:2] function after releasing 0%/100% duty setting 1: Apply masked compare match output value to GTIOB[3:2] function after releasing 0%/100% duty setting.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

The GTUDDTYC sets the direction in which the 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 the GTCNT value becomes 0).

When the UD value changes from 1 to 0 with the UDF bit being 0 and while counting stops, the counter starts up-counting and the count direction changes at an overflow (the timing synchronous with count clock after the GTCNT value becomes the GTPR value). When the UD value changes from 0 to 1 with the UDF bit being 0 and while counting stops, the counter starts down-counting and the count direction changes at an underflow (the timing synchronous with count clock after the GTCNT value becomes 0).

When the UDF bit is set to 1 while counting stops, the UD bit value is reflected in the count direction when counting starts.

- In triangle-wave mode.

When the UD value changes during counting, the count direction does not change. When the UD value changes while the UDF bit is 0 and counting stops, the change is not reflected in the count direction when counting starts.

When the UDF bit is set to 1 while counting is stopped, the UD value is reflected in the count direction when counting starts.

UD bit (Count Direction Setting)

The UD bit sets the count direction (up-counting or down-counting) for GTCNT.

UDF bit (Forcible Count Direction Setting)

The UDF bit forcibly sets the count direction when GTCNT starts operation as the UD value. Only 0 should be written to this bit during counter operation. When 1 is written to the UDF bit while counting stops, return this bit to 0 before counting starts.

Output duty

- In saw-wave mode.

When the OADTY/OBDTY value changes during up-counting, the duty is reflected at an overflow (GTCNT = GTPR). When the OADTY/OBDTY value changes during down-counting, the duty is reflected at an underflow (GTCNT = 0).

When the OADTY/OBDTY value changes 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 with the OADTYF/OBDTYF bit being 1 and while counting stops, the output duty is reflected at starting counter operation.

- In triangle-wave mode.

When the OADTY/OBDTY value changes during counting, the duty is reflected at an underflow.

When the OADTY/OBDTY value changes 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 with the OADTYF/OBDTYF bit being 1 and while counting stops, the output duty is reflected at 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. Set this bit to 0 during counter operation. When OmDTYF bit is set to 1 while counting stops, return this bit to 0 until the first period ends after the counter starts.

OmDTYR bit (GTIOCm Output Value Selecting after Releasing 0%/100% Duty Setting) (m = A, B)

The OmDTYR bit selects the value that is the object of output retained or toggled at cycle end, when the control changes from 0%/100% duty setting to compare match for the 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, the compare match operation continues 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] bits at the end of the cycle in the compare match operation.

21.2.14 General PWM Timer I/O Control Register (GTIOR)

Address(es): GPT320.GTIOR 4007 8034h,
GPT16m.GTIOR 4007 8034h + 0100h × m (m = 1 to 6)

	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 21.5.	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: None of the below options are specified 0 1: GTIOCA pin is set to Hi-Z in response to control the output negation 1 0: GTIOCA pin is set to 0 in response to control the output negation 1 1: GTIOCA pin is set to 1 in response to control the output negation.	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 21.5	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

Bit	Symbol	Bit name	Description	R/W
b23	OBHLD	GTIOCB Pin Output Setting at the Start/Stop Count	0: The GTIOCB pin output level at the start or stop of counting depends on the register setting 1: The GTIOCB pin output level is retained at the start or 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: None of the below options are specified 0 1: GTIOCB pin is set to Hi-Z in response to control the output negation 1 0: GTIOCB pin is set to 0 in response to control the output negation 1 1: GTIOCB pin is set to 1 in response to control the output negation.	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: Noise filter for the GTIOCB pin is disabled 1: Noise filter for the GTIOCB pin is enabled.	R/W
b31, b30	NFCSB[1:0]	Noise Filter B Sampling Clock Select	b31 b30 0 0: PCLKD/1 0 1: PCLKD/4 1 0: PCLKD/16 1 1: PCLKD/64.	R/W

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 21.5.

OADFLT bit (GTIOCA Pin Output Value Setting at the Count Stop)

The OADFLT bit selects 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 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 the GTIOCA pin in response to a request to disable output from the POEG.

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 21.5.

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 the GTICBSR register is set to 1), the GTIOCB pin does not output regardless of the OBE bit value.

OBDF[1:0] bits (GTIOCB Pin Disable Value Setting)

The OBDF[1:0] bits select the output value of the GTIOCB pin in response to a request to disable output from the POEG.

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 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 21.5 Settings of GTIOA[4:0] and GTIOB[4:0] bits

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
1	0	0	0	0	Initial output is high.	Output retained at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	0	1	0			High output at GTCCRA/GTCCRB compare match
1	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	1	0	0		Low output at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	1	1	0			High output at GTCCRA/GTCCRB compare match
1	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	0	0	0		High output at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	0	1	0			High output at GTCCRA/GTCCRB compare match
1	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	1	0	0		Output toggled at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	1	1	0			High output at GTCCRA/GTCCRB compare match
1	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Note 1. The cycle end means an overflow (GTCNT changes from GTPR to 0 in up-counting), an underflow (GTCNT changes from 0 to GTPR in down-counting), or counter clearing for saw-wave mode, and means a trough (GTCNT changes from 0 to 1) for trianglewave mode.

Note 2. When the timing of a cycle end and the timing of a GTCCRA/GTCCRB compare match are the same in a compare-match operation, 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.

21.2.15 General PWM Timer Interrupt Output Setting Register (GTINTAD)

Address(es): GPT320.GTINTAD 4007 8038h,
GPT16m.GTINTAD 4007 8038h + 0100h × m (m = 1 to 6)

	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
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 source.

The output disable request to POEG outputs to the group which is selected in the GRP[1:0] bits when same time output level high or same time output level low occurs based on the output disable request enable bit.

GTST.ODF shows the request of the output disable source group that is selected in 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 pin and GTIOCB pin output 1 at the same time.

GRPABL bit (Same Time Output Level Low Disable Request Enable)

The GRPABL bit enables or disables the output disable request when the GTIOCA pin and GTIOCB pin output 0 at the same time.

21.2.16 General PWM Timer Status Register (GTST)

Address(es): GPT320.GTST 4007 803Ch,
GPT16m.GTST 4007 803Ch + 0100h × m (m = 1 to 6)

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TUCF	—	—	—	—	—	—	—	TCFPU	TCFPO	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TCFA	Input Capture/Compare Match Flag A	0: No input capture/compare match of GTCCRA is generated 1: An input capture/compare match of GTCCRA is generated.	R/(W)*1
b1	TCFB	Input Capture/Compare Match Flag B	0: No input capture/compare match of GTCCRB is generated 1: An input capture/compare match of GTCCRB is generated.	R/(W)*1
b2	TCFC	Input Compare Match Flag C	0: No compare match of GTCCRC is generated 1: A compare match of GTCCRC is generated.	R/(W)*1
b3	TCFD	Input Compare Match Flag D	0: No compare match of GTCCRD is generated 1: A compare match of GTCCRD is generated.	R/(W)*1
b4	TCFE	Input Compare Match Flag E	0: No compare match of GTCCRE is generated 1: A compare match of GTCCRE is generated.	R/(W)*1
b5	TCFF	Input Compare Match Flag F	0: No compare match of GTCCRF is generated 1: A compare match of GTCCRF is generated.	R/(W)*1
b6	TCFPO	Overflow Flag	0: No overflow (crest) 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 pin and GTIOCB pin do not output 1 at the same time 1: GTIOCA pin and GTIOCB pin output 1 at the same time.	R
b30	OABLF	Same Time Output Level Low Flag	0: GTIOCA pin and GTIOCB pin do not output 0 at the same time 1: GTIOCA pin and GTIOCB pin output 0 at the same time.	R
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit. Do not write 1.

The GTST indicates the status of the GPT.

TCFA flag (Input Capture/Compare Match Flag A)

The TCFA flag indicates the status for the input capture or compare match of GTCCRA.

[Setting conditions]

- GTCNT = GTCCRA, when the GTCCRA register functions as a compare match register

- GTCNT counter value is transferred to GTCCRA by the input capture signal when the GTCCRA register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFB flag (Input Capture/Compare Match Flag B)

The TCFB flag indicates the status for the input capture or compare match of GTCCRB.

[Setting conditions]

- GTCNT = GTCCRB, when the GTCCRB register functions as a compare match register
- GTCNT counter value is transferred to GTCCRB by the input capture signal when the GTCCRB register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFC flag (Input Compare Match Flag C)

The TCFC flag indicates the status for the compare match of GTCCRC.

[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)

The TCFD flag indicates the status 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 flag indicates the status 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)

The TCFF flag indicates the status 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 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 crest (GTCNT changes from 0 to 1) has occurred
- In counting by hardware sources, an underflow (GTCNT changes from 0 to GTPR in down-counting) has occurred.

[Clearing condition]

- 0 is written to this bit.

TUCF flag (Count Direction Flag)

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 the 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 the GTIOCA or GTIOCB pin outputs 0, this flag returns 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 bit and OBE bit are set to 1.

[Clearing conditions]

- The GTIOCA pin output value is different from the GTIOCB pin output value when both OAE bit and OBE bit are set to 1
- The GTIOCA and GTIOCB pins output 0 at the same time when both OAE bit and OBE bit are set to 1
- Either the OAE bit or OBE bit is set to 0.

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 returns 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 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.

21.2.17 General PWM Timer Buffer Enable Register (GTBER)

Address(es): GPT320.GTBER 4007 8040h,
GPT16m.GTBER 4007 8040h + 0100h × m (m = 1 to 6)

	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

Bit	Symbol	Bit name	Description	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 writen. 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 provides settings for the buffer operation and must be set while the GTCNT operation is stopped.

BD[0] bit (GTCCR Buffer Operation Disable)

The BD[0] bit disables buffer operation using GTCCRA, GTCCRB, GTCCRC, GTCCRD, 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 the writing of 1. This bit is read as 0.

This bit 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).

21.2.18 General PWM Timer Counter (GTCNT)

Address(es): GPT320.GTCNT 4007 8048h,
GPT16m.GTCNT 4007 8048h + 0100h × m (m = 1 to 6)

	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 GPT320. For GPT16m (m = 1 to 6), GTCNT is a 16-bit register. GTCNT can only be written to after counting is stopped. GTCNT must be accessed in 32-bit units. Access in 8-bit/16-bit units is prohibited.

For GPT16m (m = 1 to 6), 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 ≤ GTCNT ≤ GTPR.

21.2.19 General PWM Timer Compare Capture Register n (GTCCRn) (n = A to F)

Address(es): GPT320.GTCCRA 4007 804Ch,
GPT320.GTCCRB 4007 8050h,
GPT320.GTCCRC 4007 8054h,
GPT320.GTCCRD 4007 805Ch,
GPT320.GTCCRE 4007 8058h,
GPT320.GTCCRF 4007 8060h,

GPT16m.GTCCRA 4007 804Ch + 0100h × m (m = 1 to 6),
GPT16m.GTCCRB 4007 8050h + 0100h × m (m = 1 to 6),
GPT16m.GTCCRC 4007 8054h + 0100h × m (m = 1 to 6),
GPT16m.GTCCRD 4007 805Ch + 0100h × m (m = 1 to 6),
GPT16m.GTCCRE 4007 8058h + 0100h × m (m = 1 to 6),
GPT16m.GTCCRF 4007 8060h + 0100h × m (m = 1 to 6)

	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

	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 = 1 to 6), 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-bit 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 compare match registers that can also function as buffer registers for GTCCRA and GTCCRB.

GTCCRD and GTCCRF are compare match registers that can also function as buffer registers for GTCCRC and GTCCRE (double-buffer registers for GTCCRA and GTCCRB).

21.2.20 General PWM Timer Cycle Setting Register (GTPR)

Address(es): GPT320.GTPR 4007 8064h,
GPT16m.GTPR 4007 8064h + 0100h × m (m = 1 to 6)

	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

	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 = 1 to 6), the 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-bit 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.

21.2.21 General PWM Timer Cycle Setting Buffer Register (GTPBR)

Address(es): GPT320.GTPBR 4007 8068h,
GPT16m.GTPBR 4007 8068h + 0100h × m (m = 1 to 6)

	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

	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 = 1 to 6), 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-bit 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.

21.2.22 General PWM Timer Dead Time Control Register (GTDTCR)

Address(es): GPT320.GTDTCR 4007 8088h,
GPT16m.GTDTCR 4007 8088h + 0100h × m (m = 1 to 6)

	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

The 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.

21.2.23 General PWM Timer Dead Time Value Register U (GTDVU)

Address(es): GPT320.GTDVU 4007 808Ch,
GPT16m.GTDVU 4007 808Ch + 0100h x m (m = 1 to 6)

	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

	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 = 1 to 6), 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-bit 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 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.

21.2.24 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	—	—	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 = 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 inputs that are synchronized by PCLKD OPSCR.FB = 1: The OPSCR.U, OPSCR.V, and OPSCR.W bits can read the OPSCR.UF. OPSCR.VF, and OPSCR.WF bits.	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 on external pin) 1: Output*1.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	FB	External Feedback Signal Enable	This bit selects the input phase from the software settings or external input. 0: Select the external input 1: Select the software settings (OPSCR.UF, VF, WF).	R/W
b17	P	Positive-Phase Output (P) Control	0: Output level signal 1: Output level signal (PWM of GPT161).	R/W
b18	N	Negative-Phase Output (N) Control	0: Output level signal 1: Output level signal (PWM of GPT161).	R/W
b19	INV	Invert-Phase Output Control	0: Output positive logic (active-high) 1: Output negative logic (active-low).	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
b24	GRP	Output Disabled Source Selection	0: Select group A output disable source 1: Select group B output disable source.	R/W

Bit	Symbol	Bit name	Description	R/W
b25	—	Reserved	This bit is read as 0. The write value should be 0.	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 bit is high, the OPSCR.EN bit is set to 0.

The OPSCR 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.RV, OPSCR.ALIGN, OPSCR.GRP, OPSCR.GODF, OPSCR.NFEN, OPSCR.NFCS. Then, set this bit to 1. Also when OPSCR.GODF is 1 and the signal value selected by the OPSCR.GRP bit is high, the OPSCR.EN bit is set to 0.

FB bit (External Feedback Signal Enable)

The FB bit selects the input phase from the software settings (OPSCR.UF, VF, WF) and external input such as a Hall element.

P bit (Positive-Phase Output (P) Control)

The P bit selects one of the level signal output or PWM signal output for the positive-phase output (GTOUUP pin, GTOVUP pin, GTOWUP pin).

N bit (Negative-Phase Output (N) Control)

The N bit selects one of the level signal output or PWM signal output for the negative-phase output (GTOULO pin, GTOVLO pin, GTOWLO pin).

INV bit (Invert-Phase Output Control)

The INV bit selects either 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 might be short-pulsed.

Note: When OPSCR.ALIGN bit is 1, input phase is aligned with PWM output.

GRP bit (Output Disabled Source Selection)

The GRP bit selects the output disable source A or B.

GODF bit (Group Output Disable Function)

When OPSCR.GODF is 1 and the signal value selected by the OPSCR.GRP bit is high, the OPSCR.EN bit is set to 0. When 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 is not used for the external input.

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.

21.3 Operation

21.3.1 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.

21.3.1.1 Counter operation

(1) Counter start and stop

The counter for 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 set to 1
- Writing 1 to the bit in GTSTP associated with the GPT channel number when the GTPSR.CSTOP bit set to 1
- The hardware source selected in the GTSSR register
- The hardware source selected in the GTPSR register.

(2) Periodic count operation in up-counting by count clock

The GTCNT counter in each channel starts up-counting when the associated GTCR.CST bit is set to 1 with GTUPSR and GTDNSR registers set to 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 21.3 shows an example of a periodic count operation in up-counting.

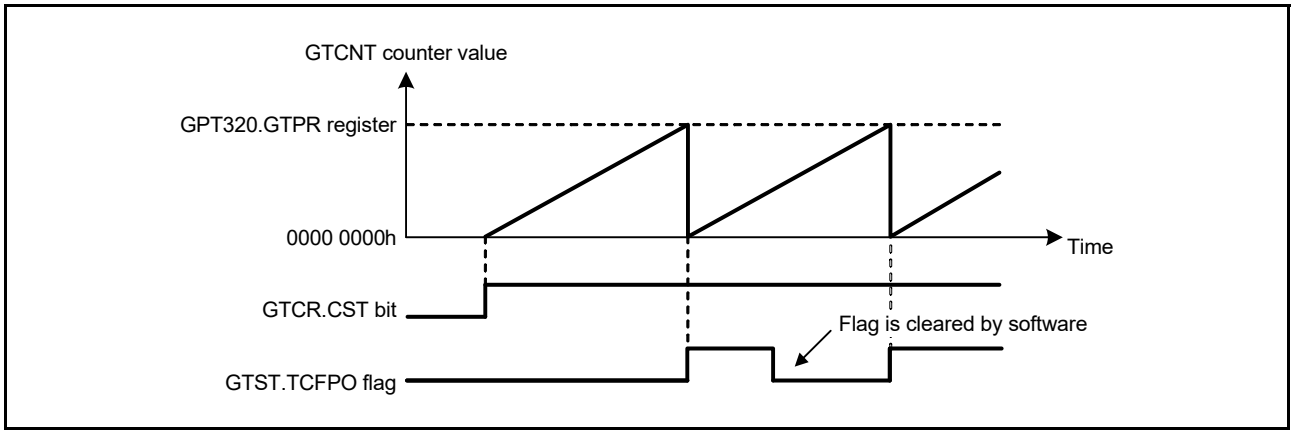


Figure 21.3 Example of periodic count operation in up-counting by the count clock

Figure 21.4 shows an example for setting periodic count operation in up-counting.

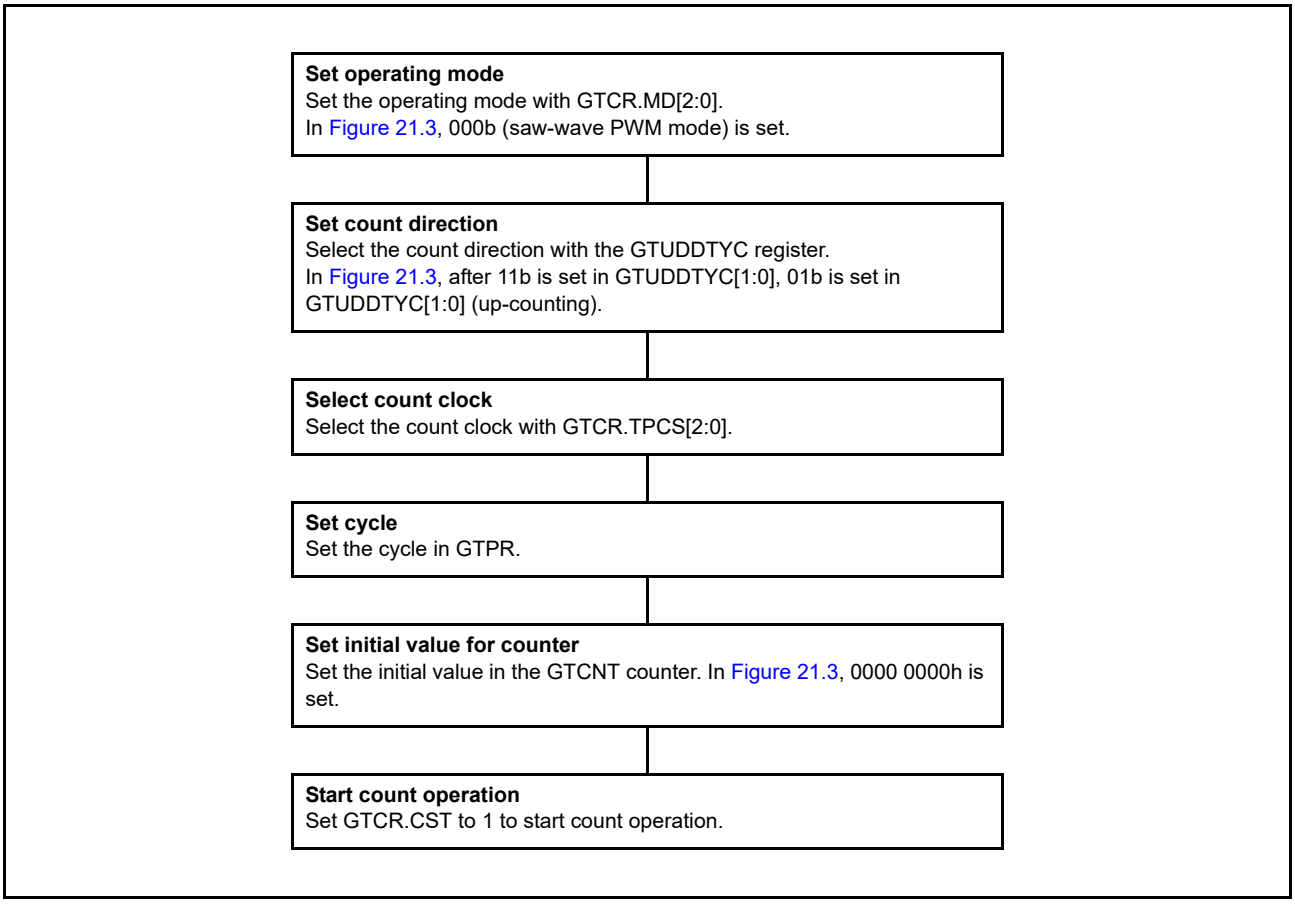


Figure 21.4 Example setting for a periodic count operation in up-counting by the count clock

(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 the GTUPSR and GTDNSR registers set to 0000 0000h. When GTCNT changes from 0 to the GTPR value (underflow), GTST.TCFPU is set to 1. After the GTCNT counter underflows, down-counting resumes from the GTPR value.

Figure 21.5 shows an example of periodic count operation in down-counting by the count clock.

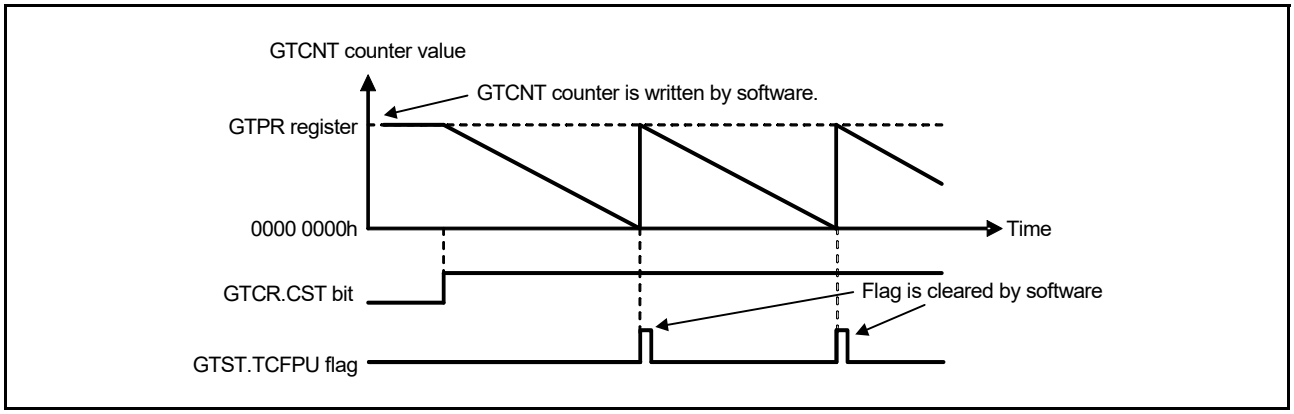


Figure 21.5 Example of periodic count operation in down-counting by the count clock

Figure 21.6 shows an example setting for periodic count operation in down-counting by the count clock.

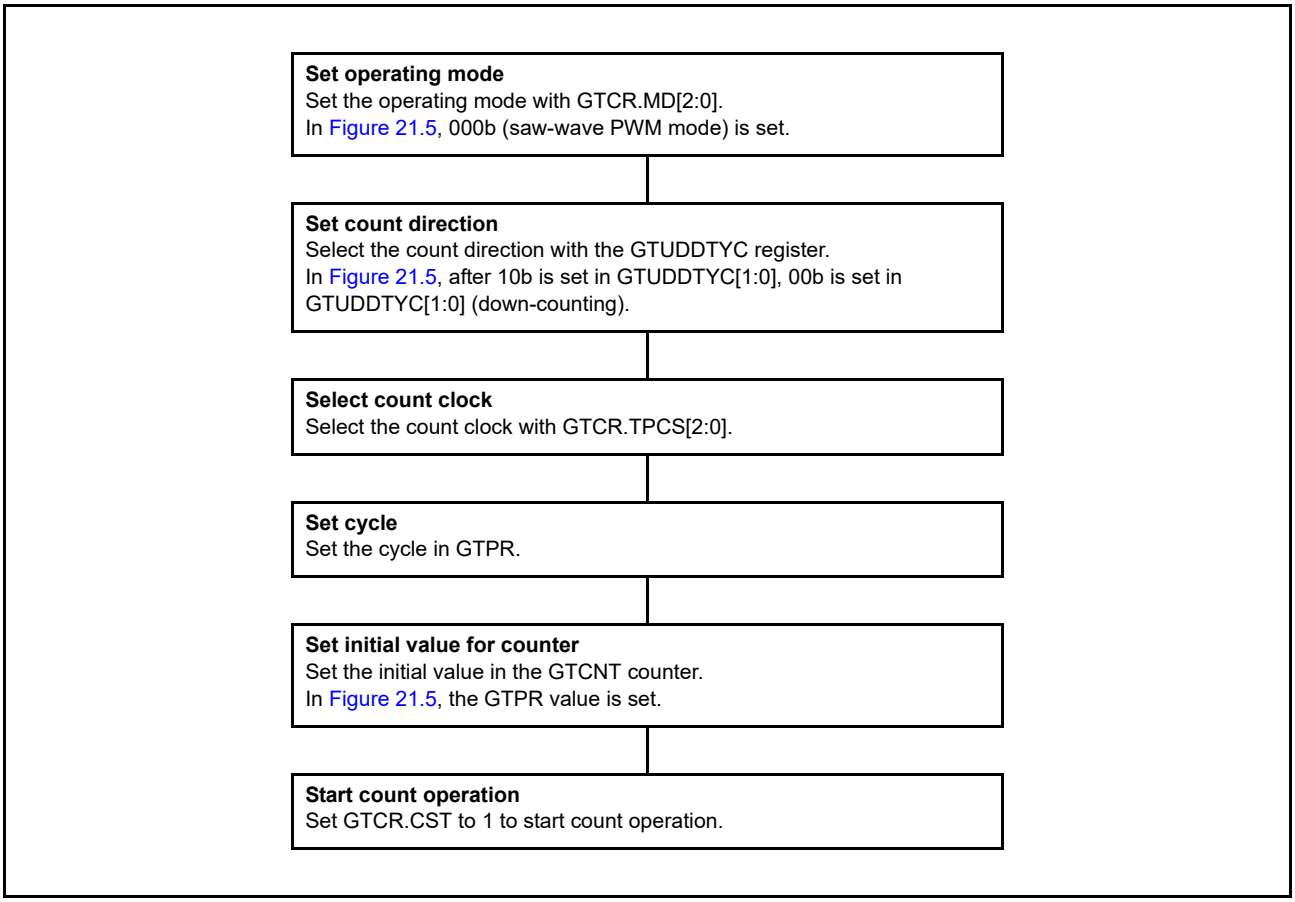


Figure 21.6 Example setting for periodic count operation in down-counting by count clock

(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 when up-counting using hardware sources is the same as when up-counting by the count clock.

If you are using a hardware source to count up, set the GTCR.CST bit to 1 to enable the counting operation. After GTCR.CST is set to 1, the counter cannot count up for 1 clock cycle as specified by GTCR.TPCS[2:0] because the count

operation is synchronized by the count clock selected in GTCR.TPCS[2:0]. Set GTCR.TPCS[2:0] to 000b to count up with a 1 PCLKD delay after GTCR.CST is set to 1.

Figure 21.7 shows an example of a periodic count operation in up-counting by a hardware resource (rising edge of GTETRGA pin).

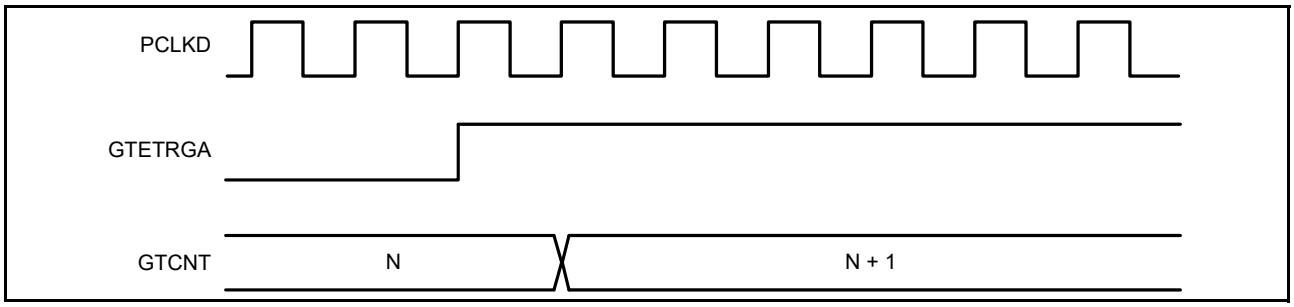


Figure 21.7 Example of periodic count operation in up-counting using hardware sources

Figure 21.8 shows an example setting for periodic count operation in down-counting by the count clock.

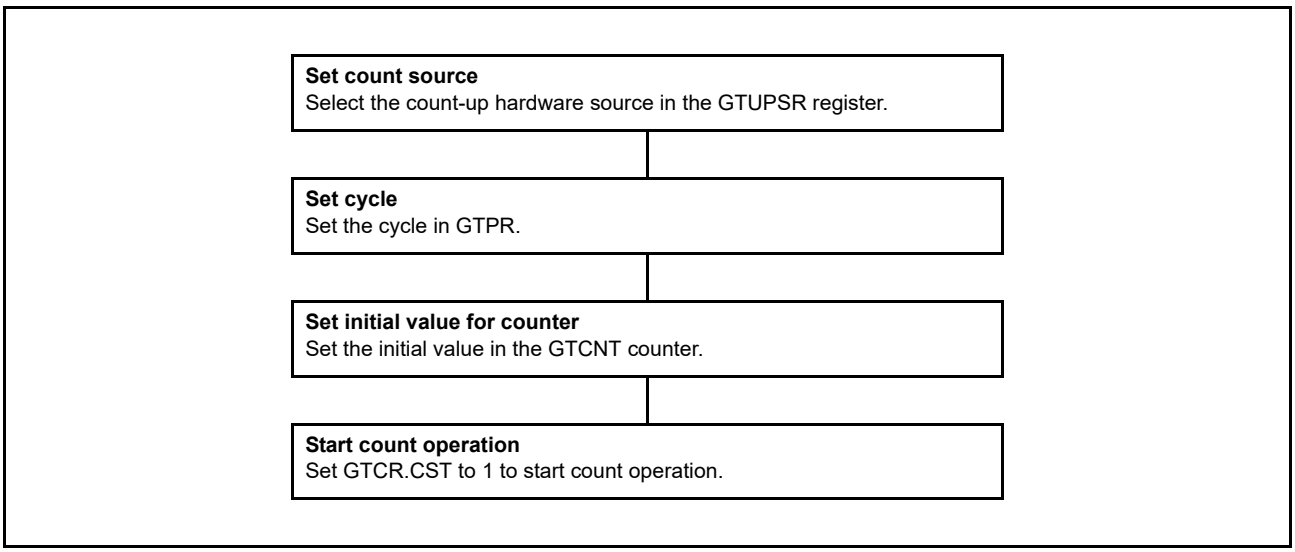


Figure 21.8 Example setting for an event count operation in up-counting using hardware sources

(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, the 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 in 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 21.9 shows an example of a periodic count operation in down-counting by a hardware resource (rising edge of GTETRGA pin).

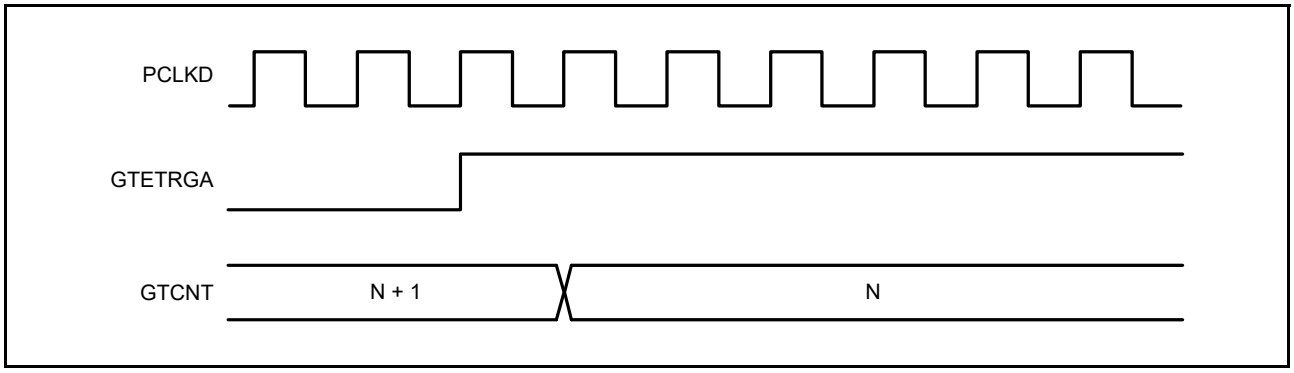


Figure 21.9 Example of event count operation in down-counting using hardware sources

Figure 21.10 shows an example setting for a periodic count operation in down-counting using a hardware resource.

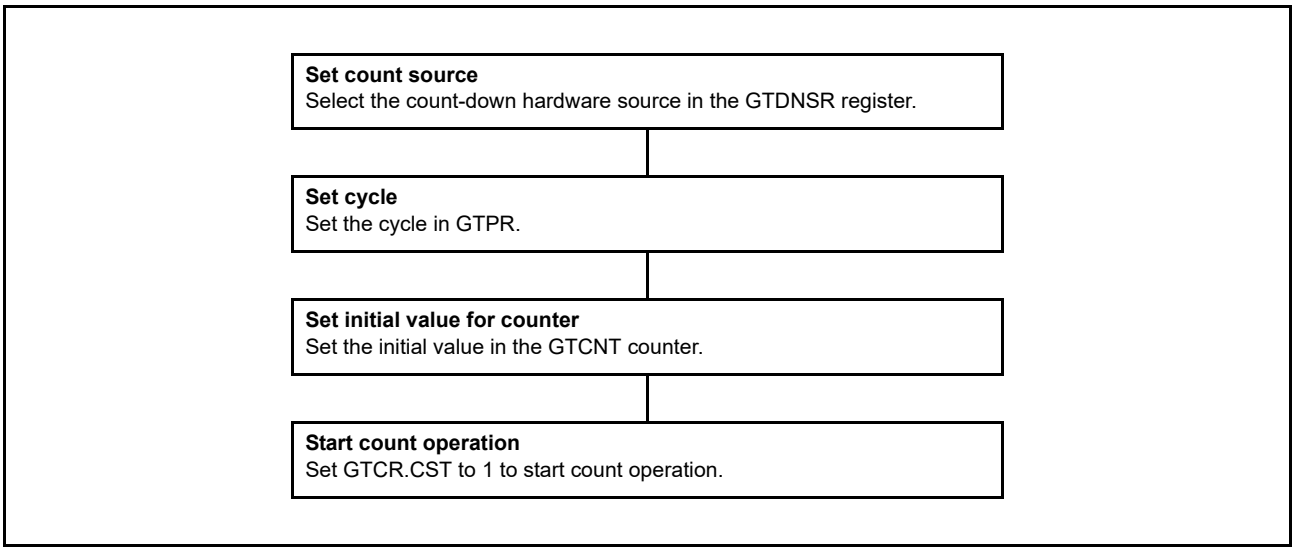


Figure 21.10 Example setting for an event count operation in down-counting using hardware sources

(6) Counter clear operation

The counter of each channel is cleared by following sources:

- Writing 0 to GTCNT register
- Writing 1 to the bit in GTCLR associated with the GPT channel number when the GTCSR.CCLR bit set to 1
- The hardware source selected in GTCSR register.

Writing to the GTCNT register is prohibited during count operation. The GTCNT counter can be cleared both by writing 1 to the GTCLR and by the clear request of hardware sources, whether GTCNT is counting (GTCR.CST is 1) or not (GTCR.CST is 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 is performed.

In event count operation when at least 1 bit in the GTUPSR or GTDNSR is set to 1, after clear sources occur, both writing to GTCLR register and clearing by hardware sources are performed immediately to synchronize with PCLKD. If other settings are used, clear is synchronized with the counter clock selected in GTCR.TPCS[2:0].

21.3.1.2 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 toggle 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 21.11 shows an example of low output and high output operation by a compare match of GTCCRA and GTCCRB.

In this example, the GPT320.GTCNT counter performs up-counting, and settings are made so that high is output from the GTIOC0A pin by a GPT320.GTCCRA compare match, and low is output from the GTOC0B pin by a GPT320.GTCCRB compare match. The pin level does not change when the specified level and pin level match.

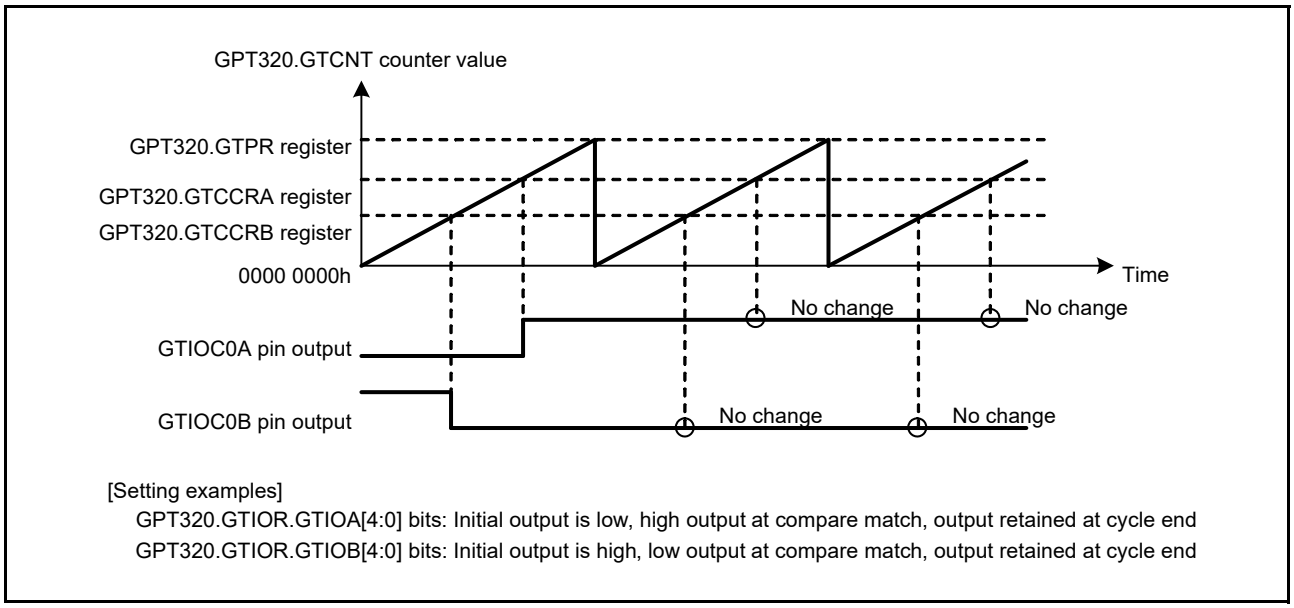


Figure 21.11 Example of low output and high output operation

Figure 21.12 shows an example setting for low output and high output operation.

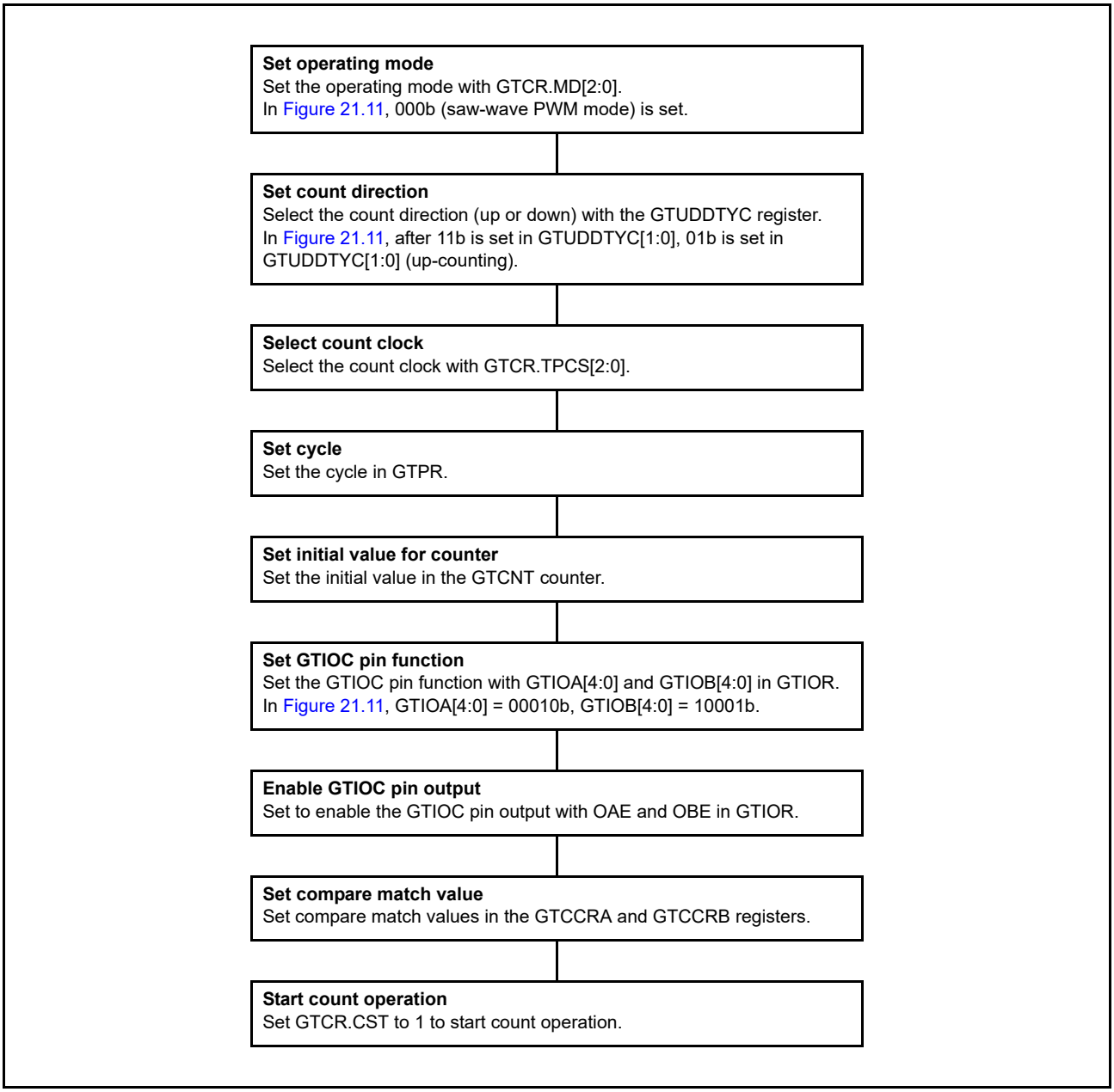


Figure 21.12 Example for setting low output and high output operation

(2) Toggled output

Figure 21.13 and Figure 21.14 show examples of toggled output operation by compare matches of GTCCRA and GTCCRB. In Figure 21.13, 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 21.14, 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.

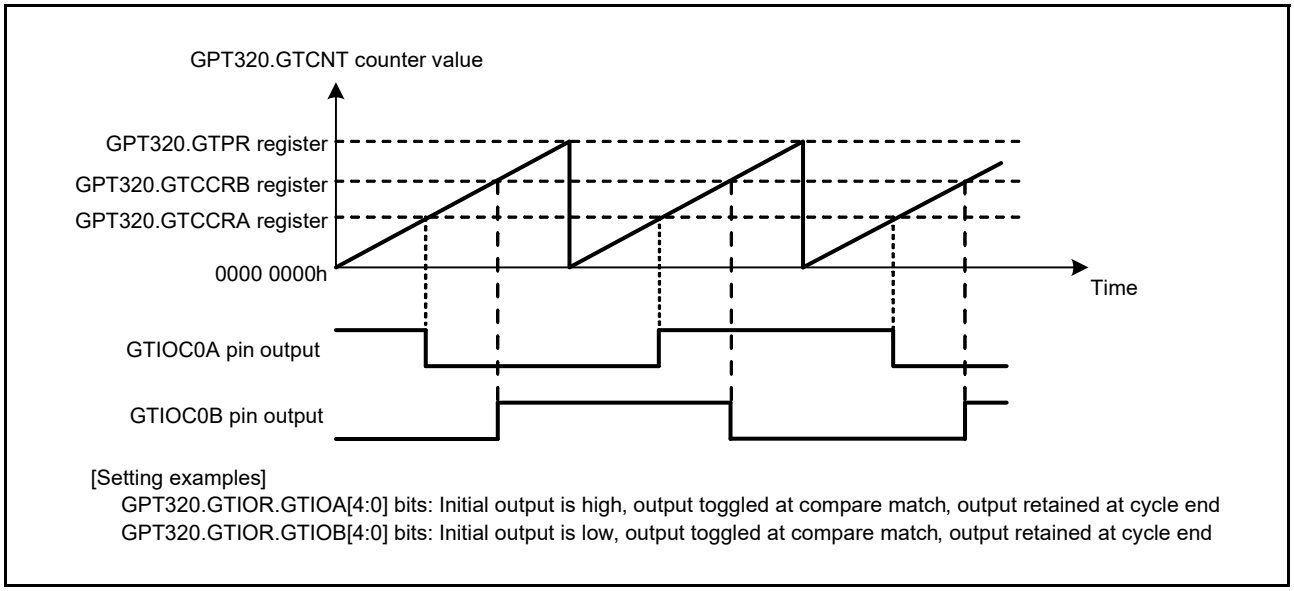


Figure 21.13 Example of toggled output operation (1)

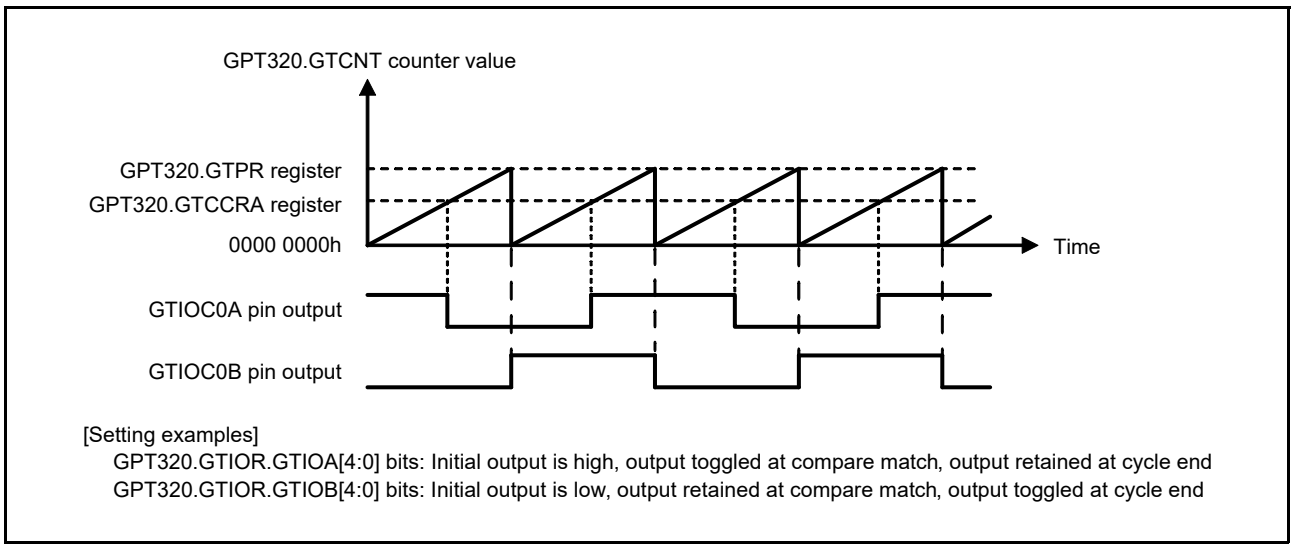


Figure 21.14 Example of toggled output operation (2)

Figure 21.15 shows an example setting for toggled output operation.

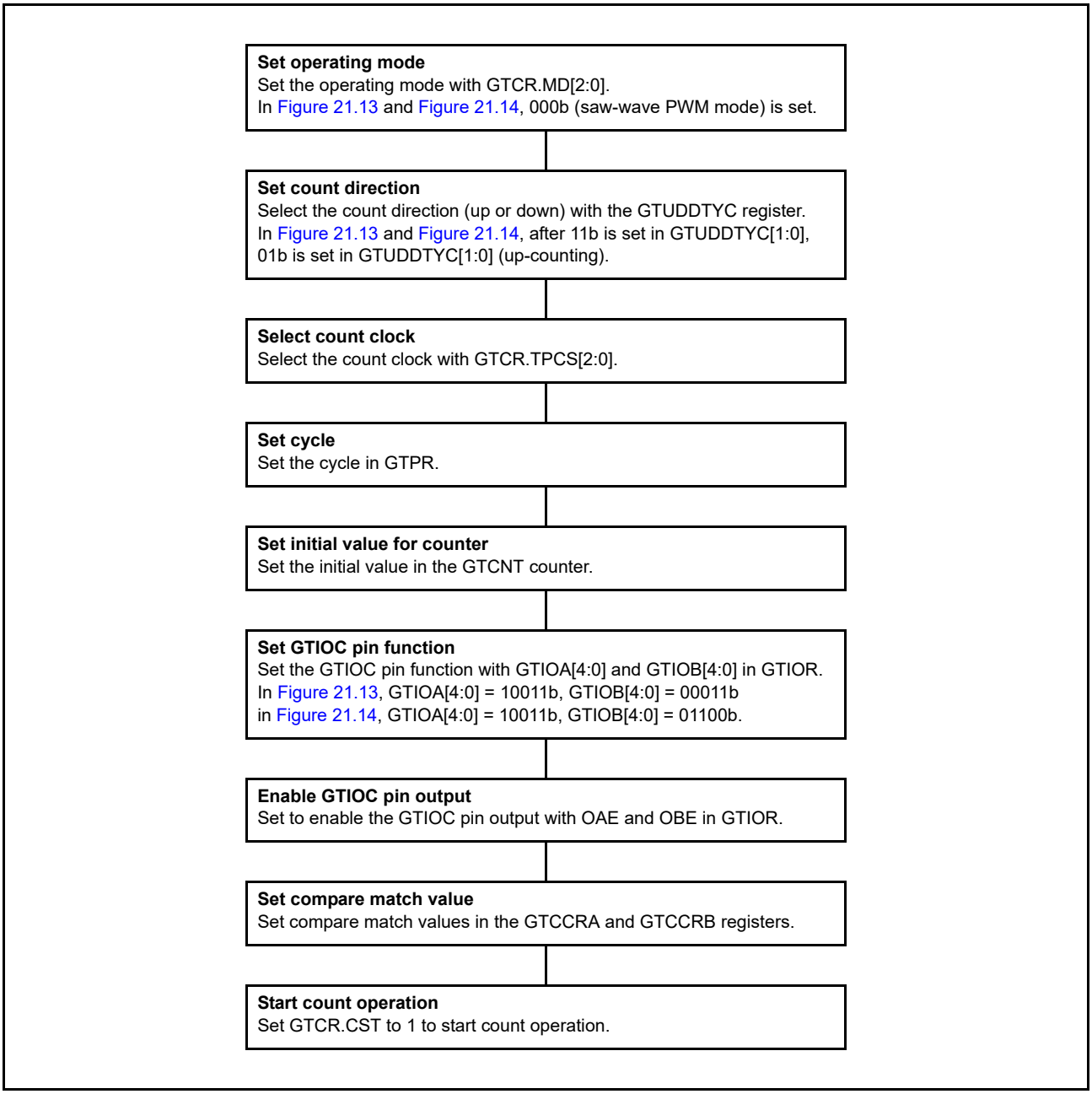


Figure 21.15 Example setting for toggled output operation

21.3.1.3 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 21.16 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 GTCCRA at both edges of the GTIOC0A input pin and to GTCCRB on the rising edge of the GTIOC0B input pin.

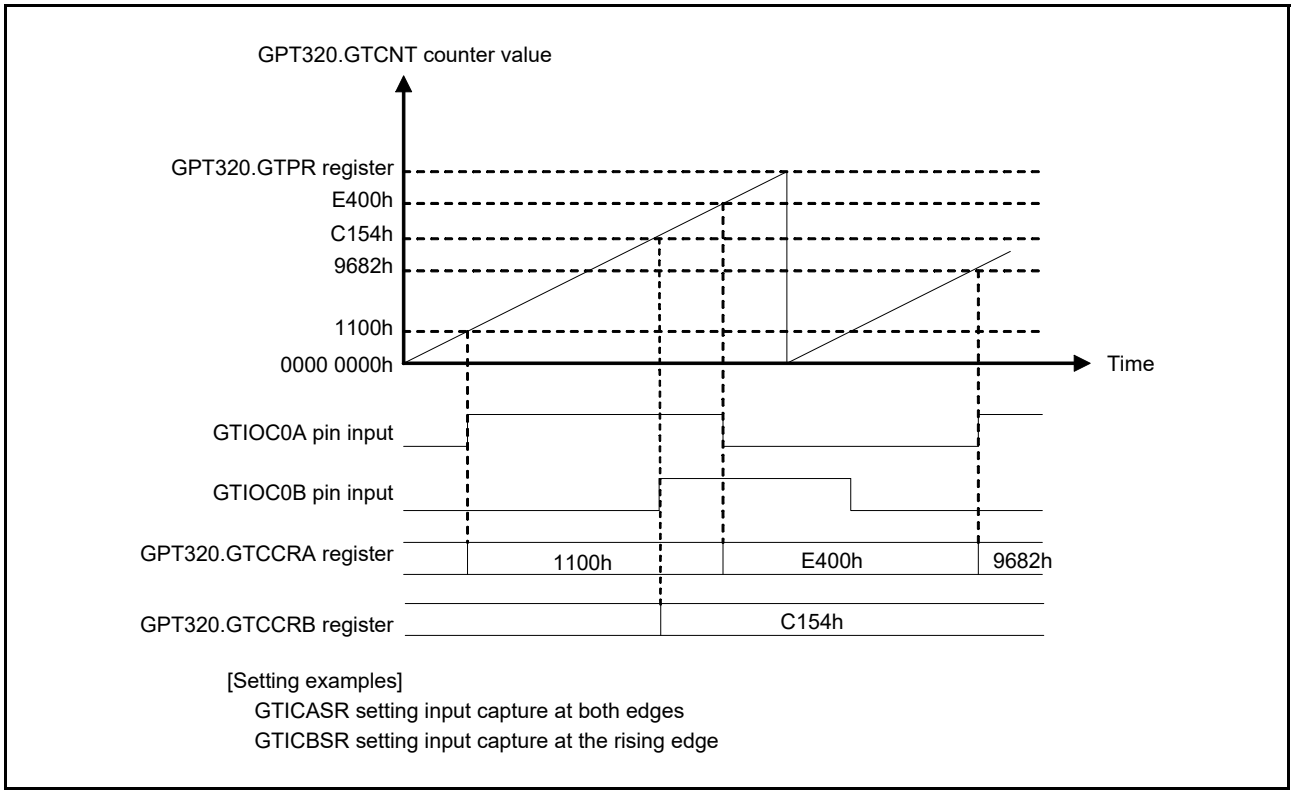


Figure 21.16 Example of input capture operation

Figure 21.17 shows an example for setting an input capture operation with count operation by the count clock.

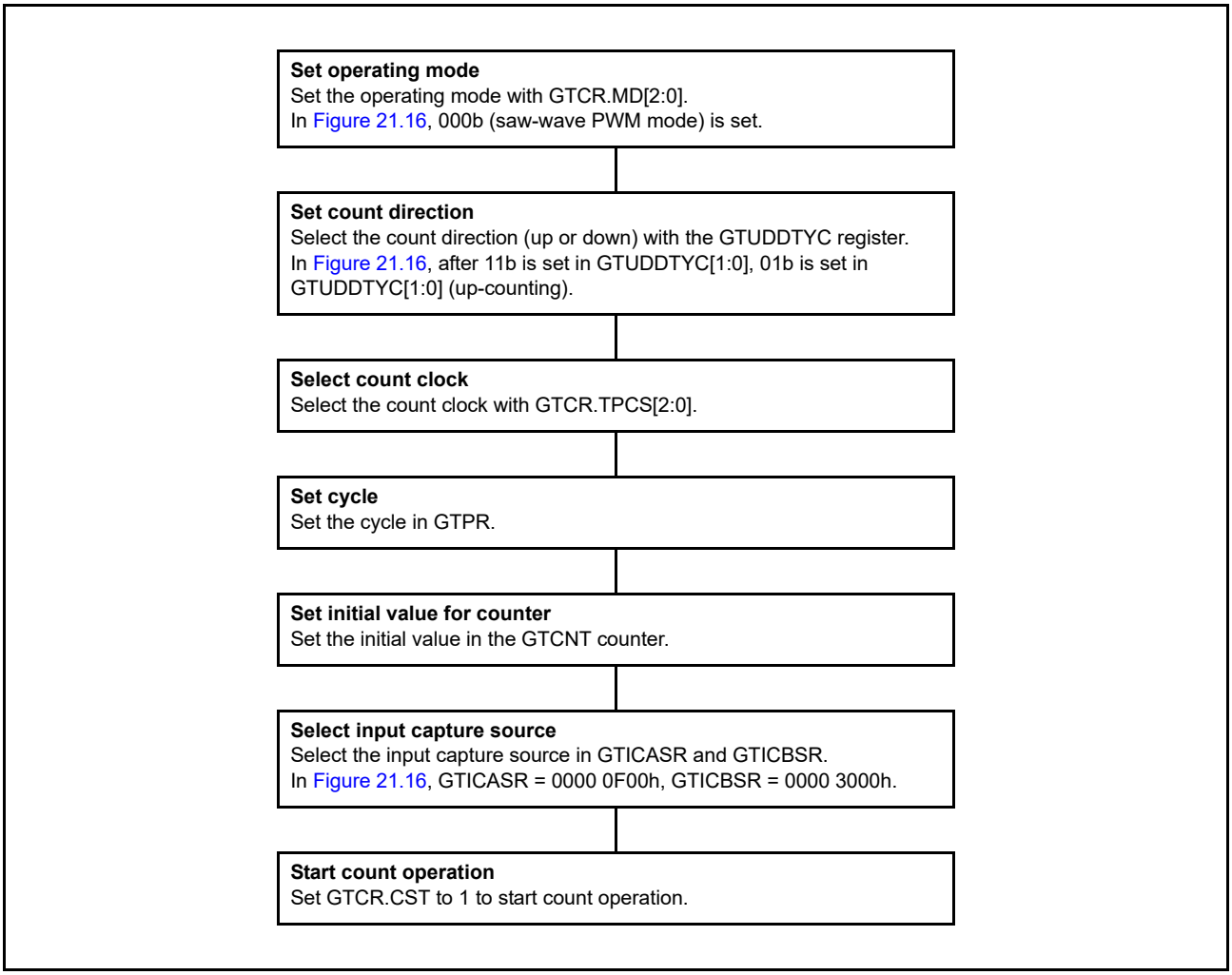


Figure 21.17 Example setting for input capture operation

21.3.2 Buffer Operation

The following buffer operations can be set with GTBER:

- GTPR and GTPBR
- GTCCRA, GTCCRC, and GTCCRD
- GTCCRB, GTCCRE, and GTCCRF.

21.3.2.1 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 21.18 to Figure 21.20 show examples of GTPR buffer operation, and Figure 21.21 shows an example setting for GTPR buffer operation.

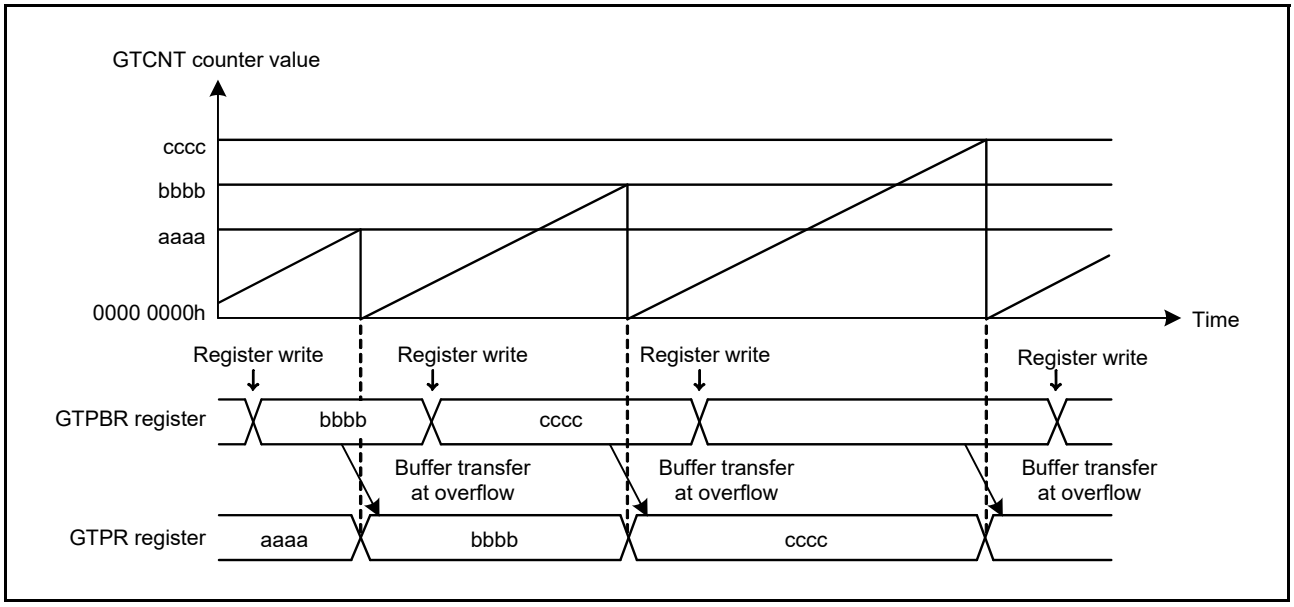


Figure 21.18 Example of GTPR buffer operation with saw waves in up-counting

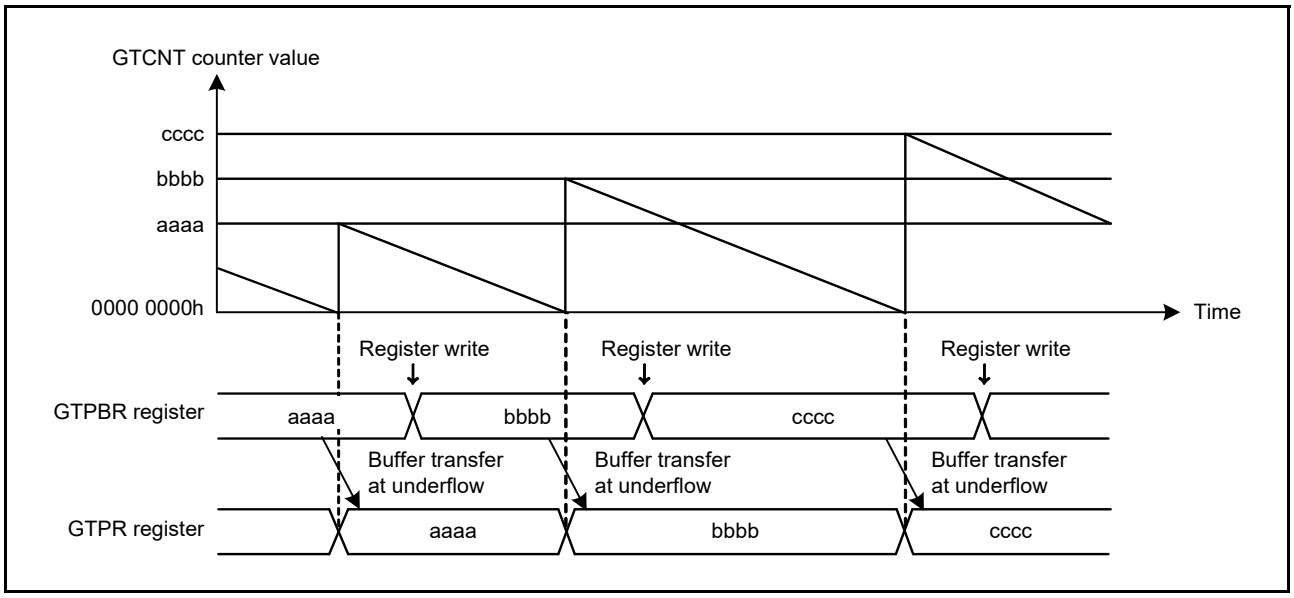


Figure 21.19 Example of GTPR buffer operation with saw waves in down-counting

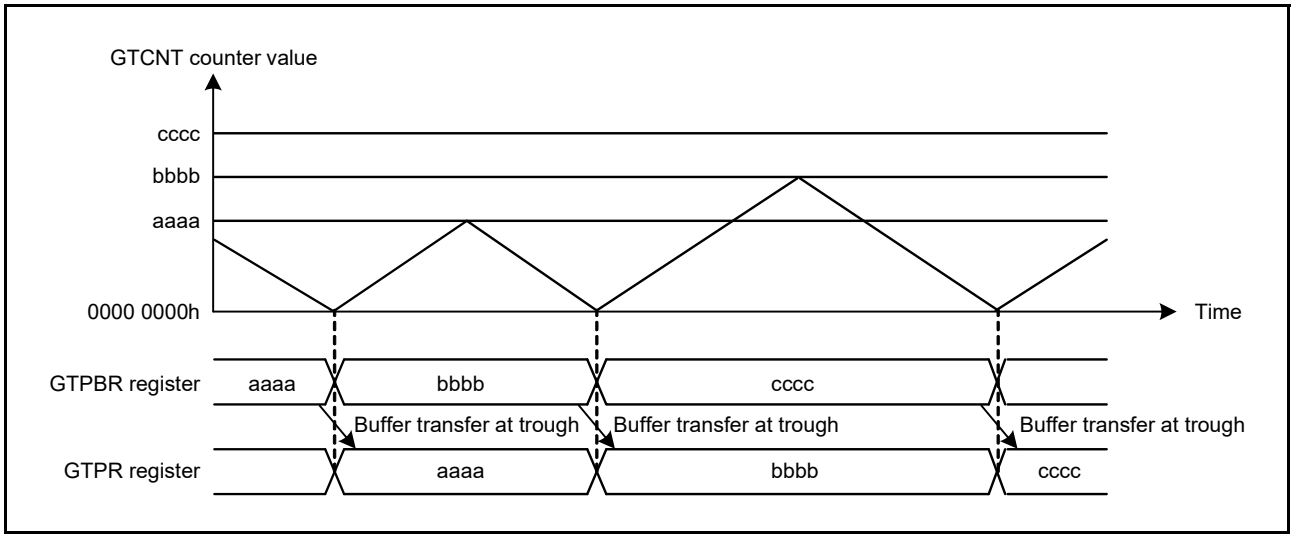


Figure 21.20 Example of GTPR buffer operation with triangle waves

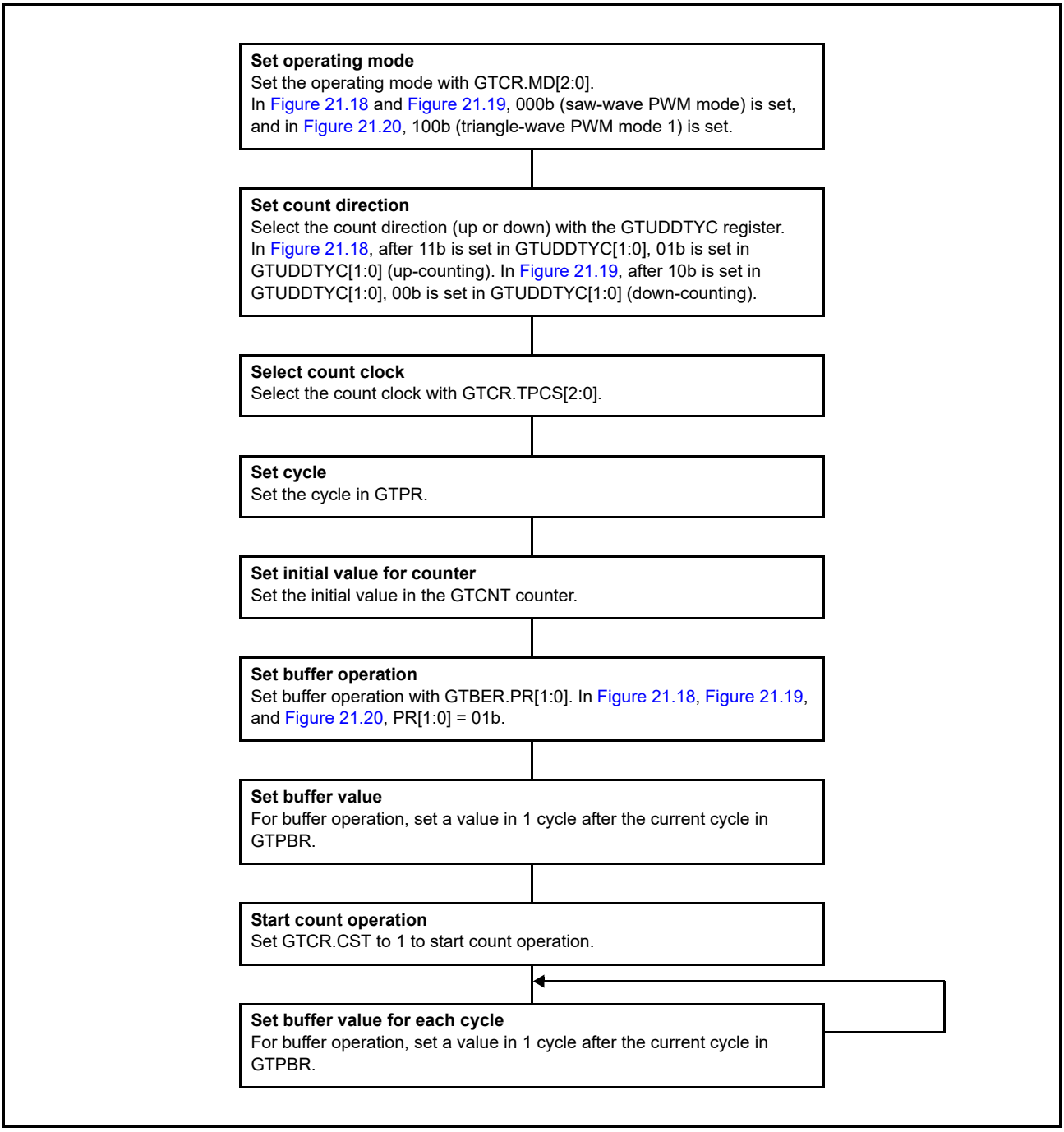


Figure 21.21 Example for setting GTPR buffer operation

21.3.2.2 Buffer operation for GTCCRA and GTCCRB

GTCCRC can function as the GTCCRA buffer register and GTCCRD can function as the GTCCRC buffer register (double-buffer register for GTCCRA). Similarly, GTCCRE can function as the GTCCRB buffer register and GTCCRF can function as the GTCCRE buffer register (double-buffer register for GTCCRB).

To set GTCCRA or GTCCRB to function as a double buffer, set GTBER.CCRA[1:0] or GTBER.CCRB[1:0] to 10b or 11b. For single-buffer operation, set 01b. To set GTCCRA or GTCCRB to not function as a buffer, set 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 similar to the case shown in section 21.3.2.1, 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 one-shot pulse mode or triangle-wave PWM mode 3.

Figure 21.22 to Figure 21.24 show examples of GTCCRA and GTCCRB buffer operation and Figure 21.25 shows an example setting for GTCCRA and GTCCRB buffer operation.

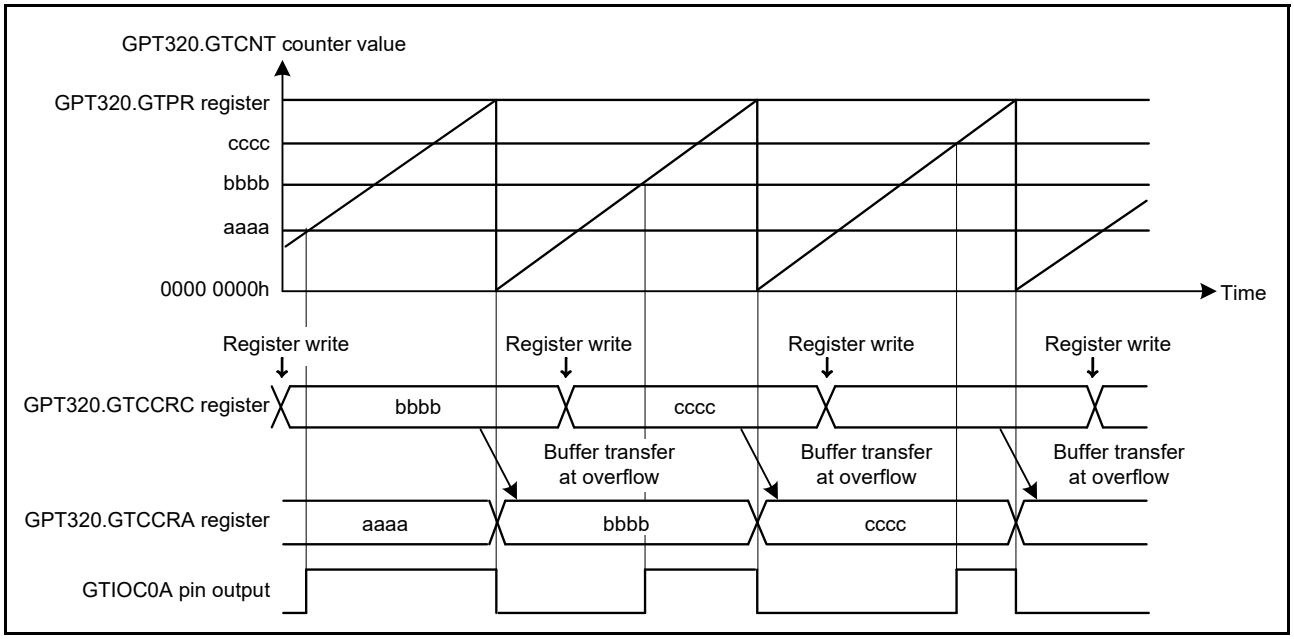


Figure 21.22 Example of GTCCRA and GTCCRB buffer operation with output compare, saw waves in up-counting, high output at GTCCRA compare match, and low output at cycle end

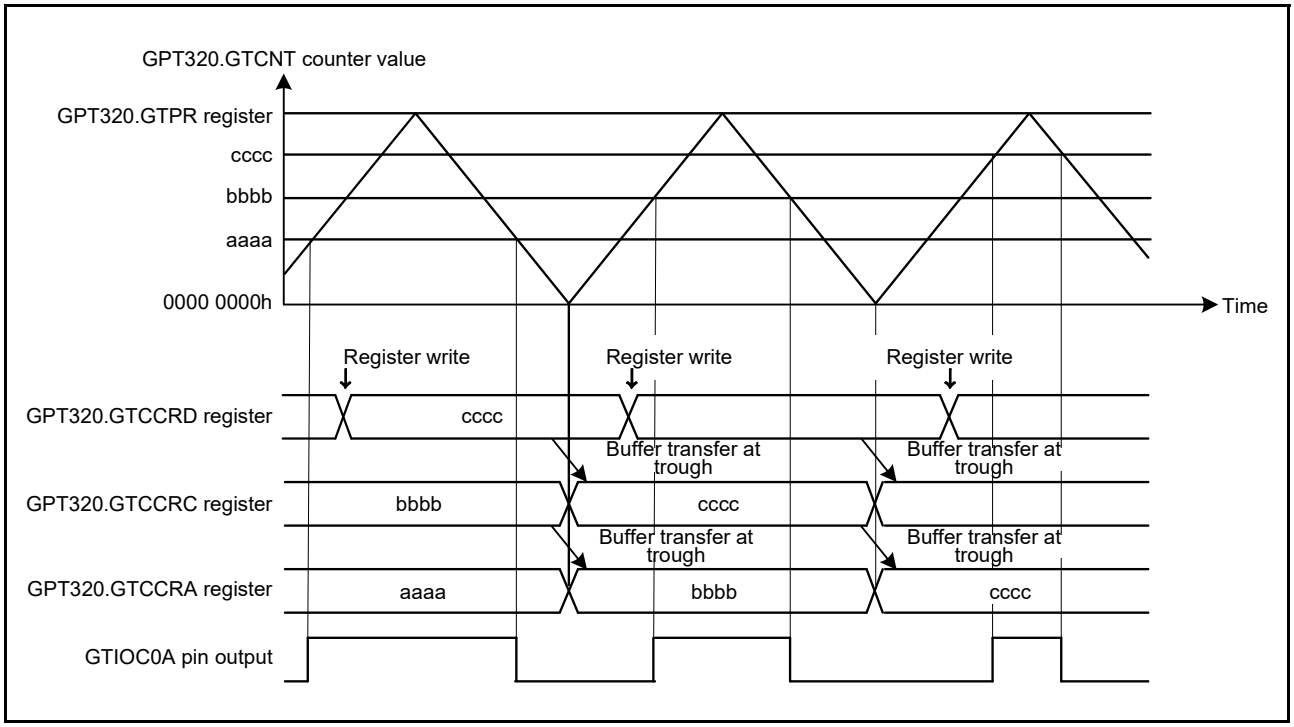


Figure 21.23 **Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at trough, output toggled at GTCCRA compare match, and output retained at cycle end**

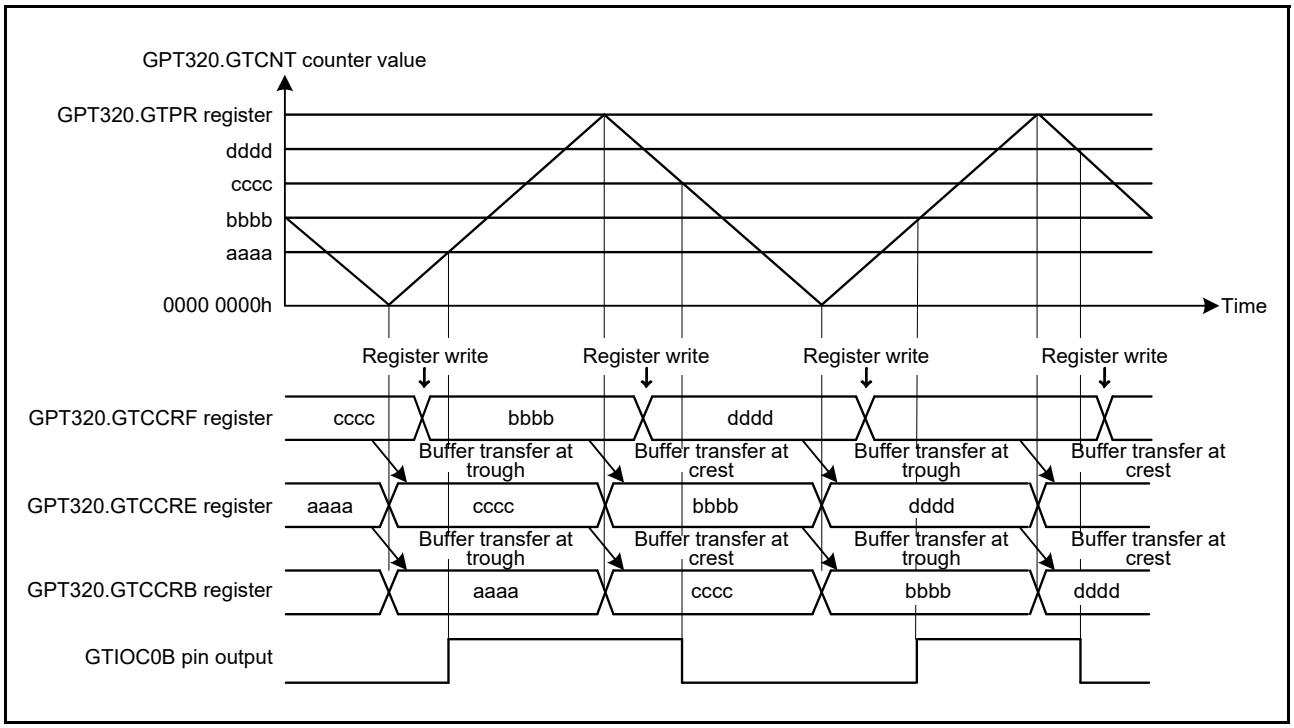


Figure 21.24 **Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at both troughs and crests, output toggled at GTCCRB compare match, and output retained at cycle end**

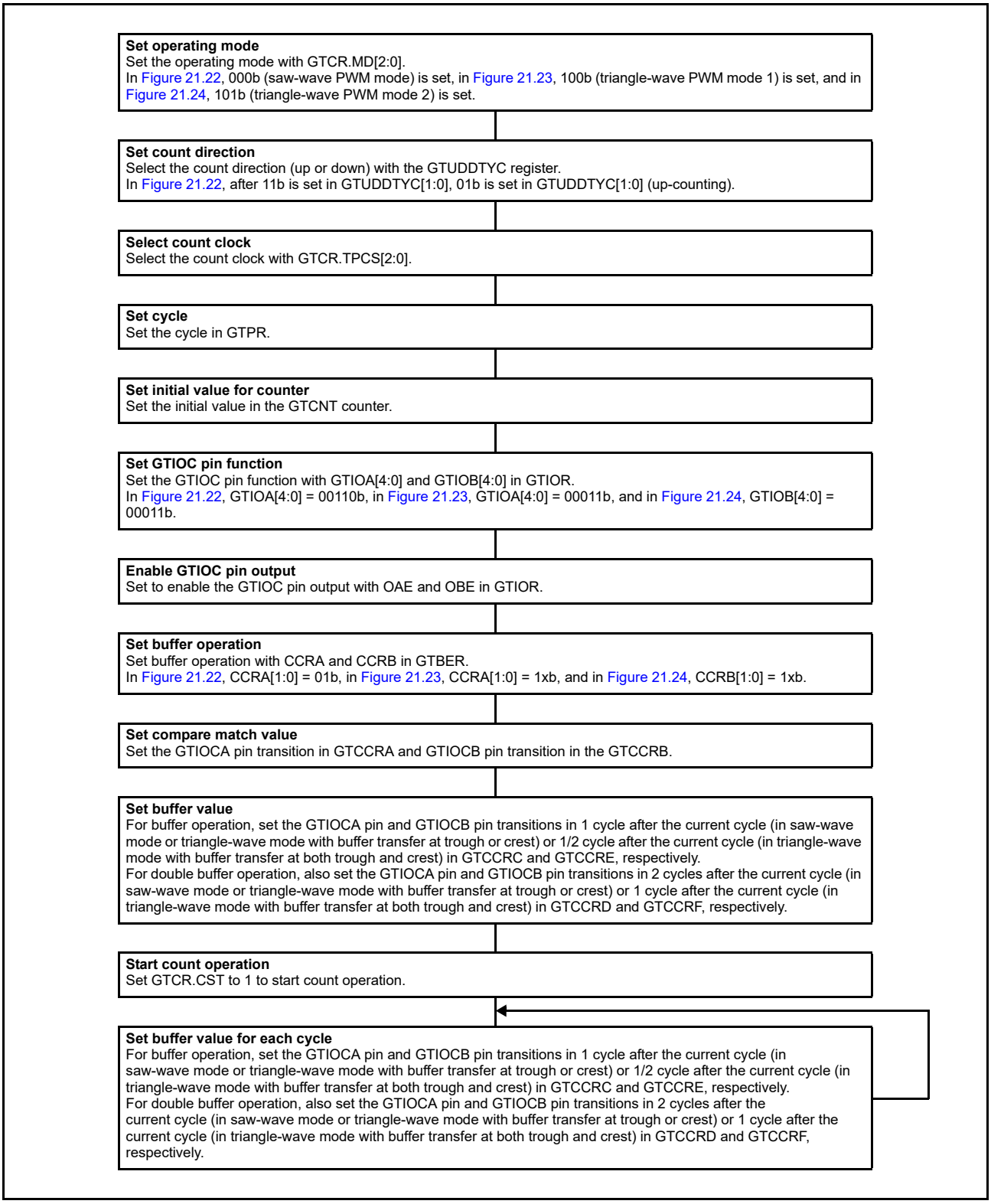


Figure 21.25 Example setting for GTCCRA and GTCCRB buffer operation with output compare

(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 21.26 and Figure 21.27 show examples of GTCCRA and GTCCRB buffer operation, and Figure 21.28 shows an example setting for GTCCRA and GTCCRB buffer operation.

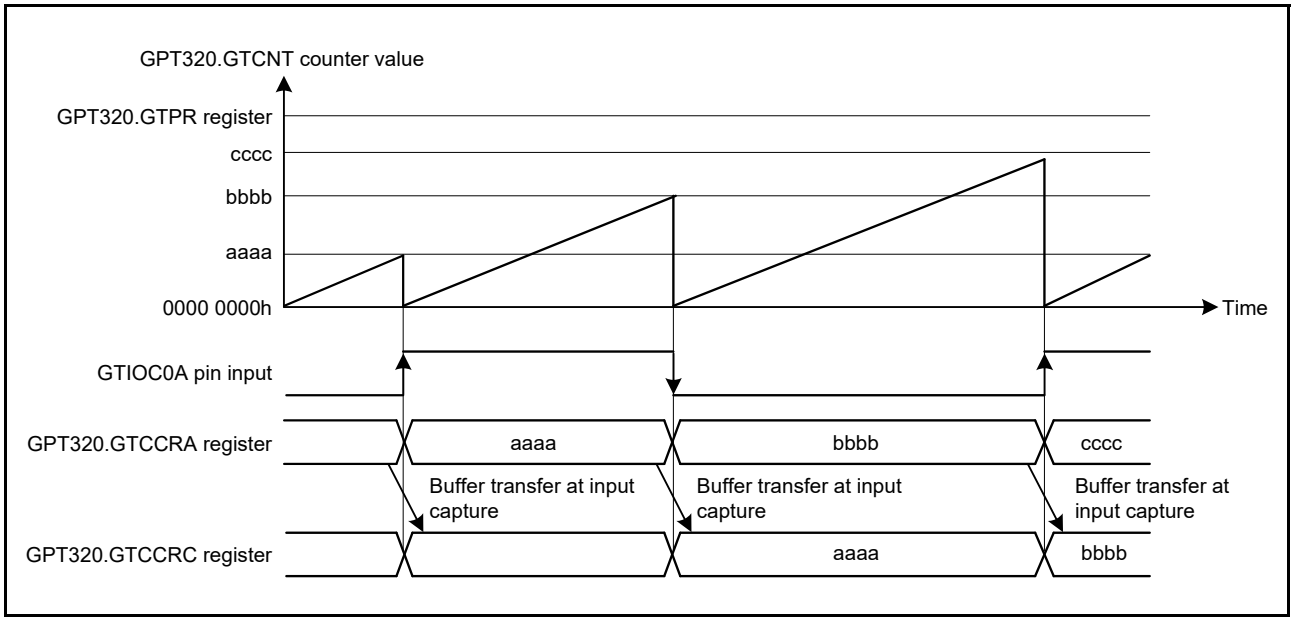


Figure 21.26 Example of GTCCRA and GTCCRB buffer operation with input capture at both edges of GTIOC0A input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOC0A input

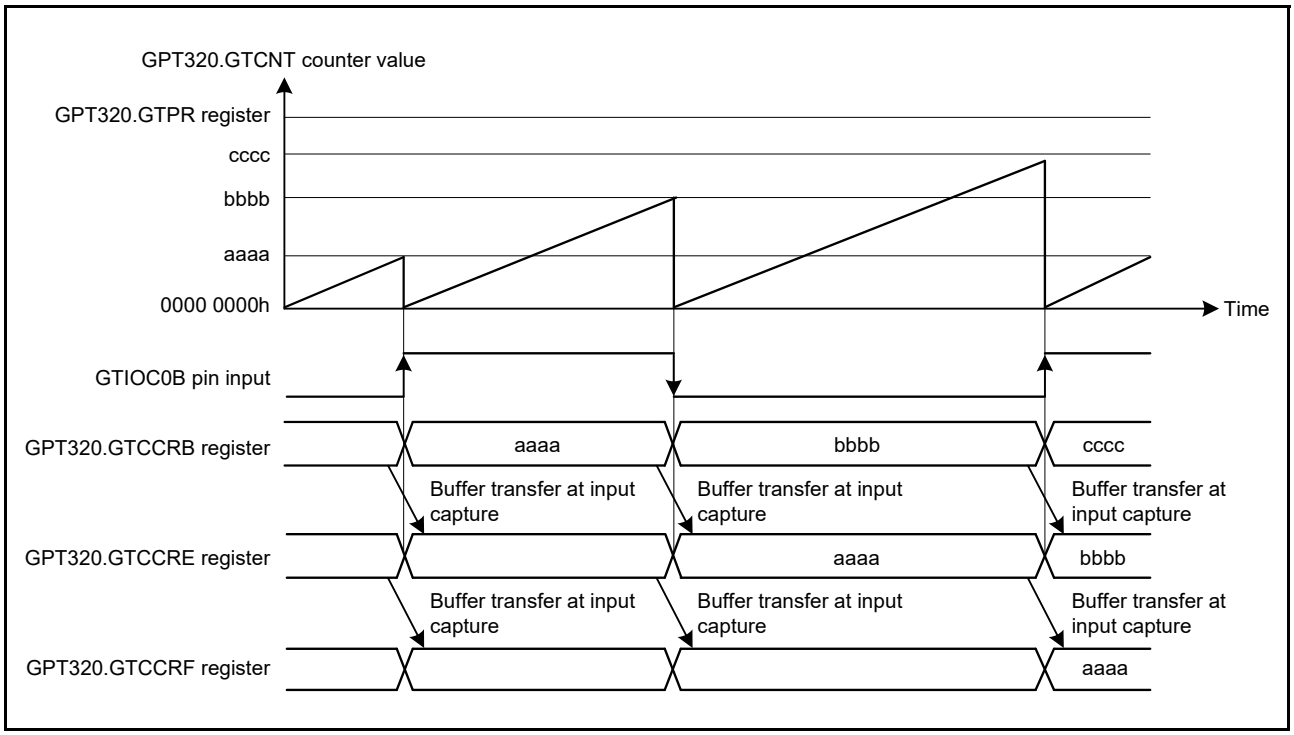


Figure 21.27 Example of GTCCRA and GTCCRB double buffer operation with input capture at both edges of GTIOC0B input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOC0B input

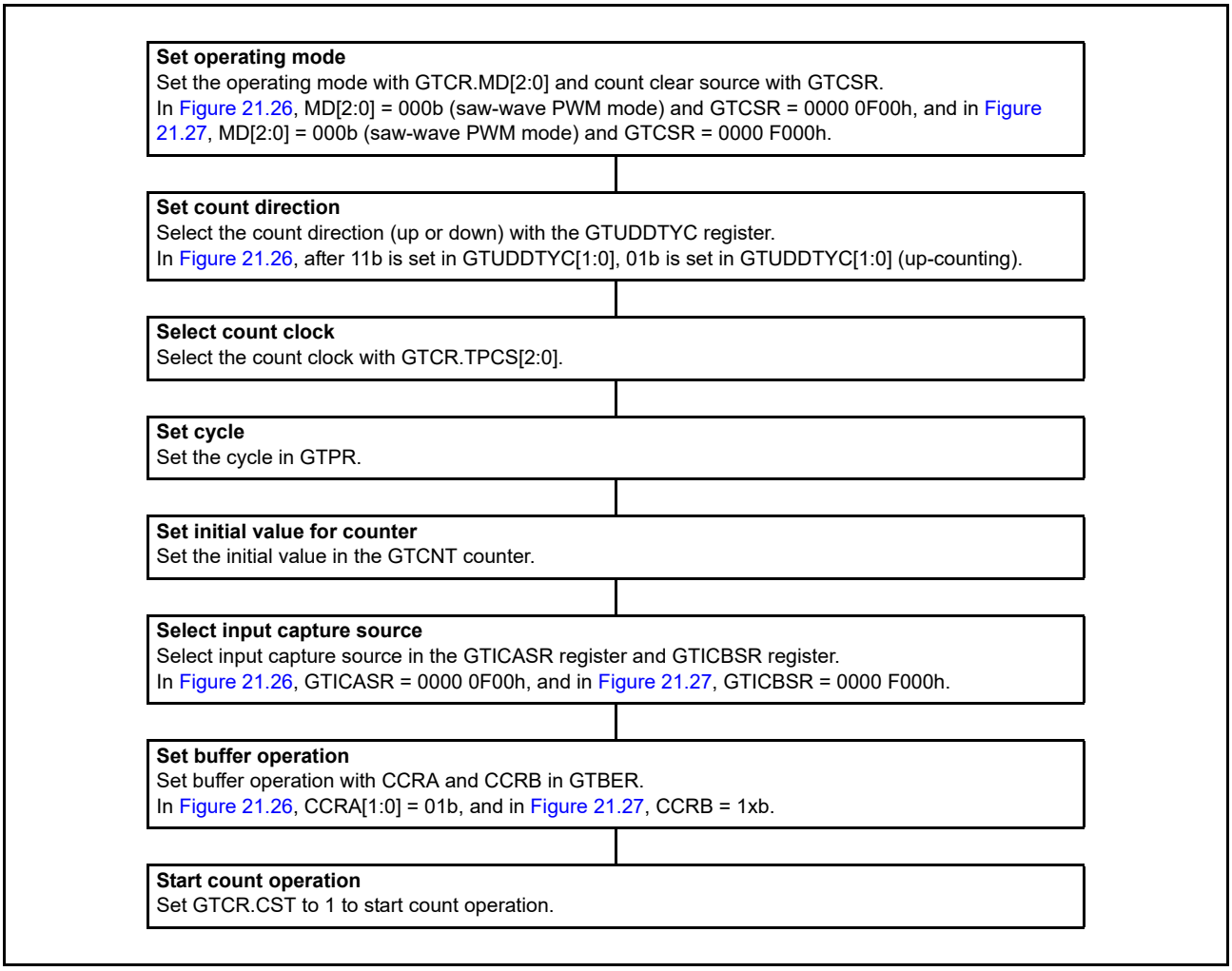


Figure 21.28 Example setting for GTCCRA and GTCCRB buffer operation with input capture

21.3.3 PWM Output Operating Mode

The GPT can output PWM waveforms to the GTIOCA pin 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.

21.3.3.1 Saw-wave PWM mode

In saw-wave PWM mode, GTCNT performs saw-wave (half-wave) operation by setting the cycle in GTPR and 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 21.29 shows an example of saw-wave PWM mode operation, and Figure 21.30 shows an example for setting saw-wave PWM mode.

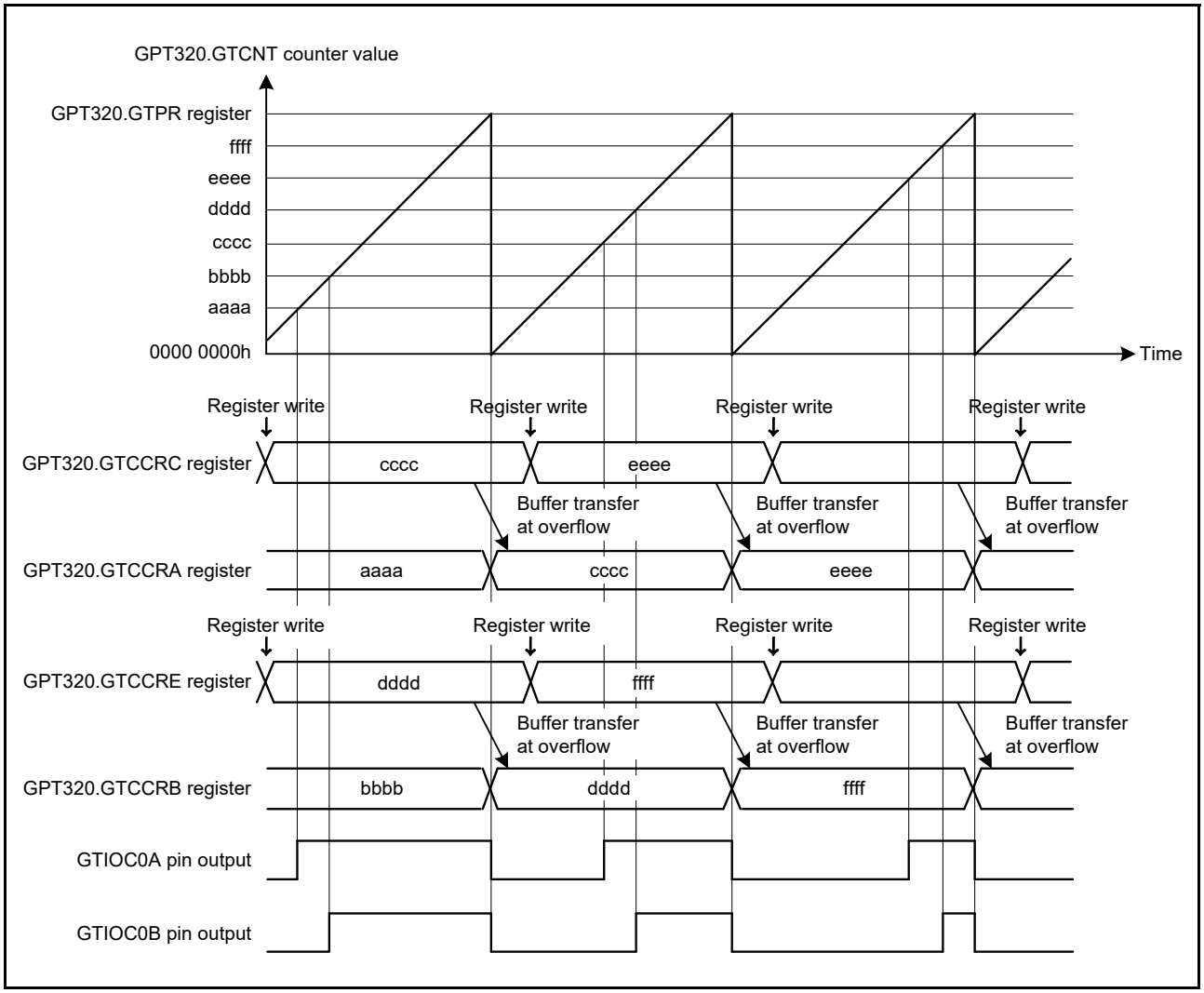


Figure 21.29 Example of saw-wave PWM mode operation with up-counting, buffer operation, high output at GTCCRA/GTCCRB compare match, and low output at cycle end

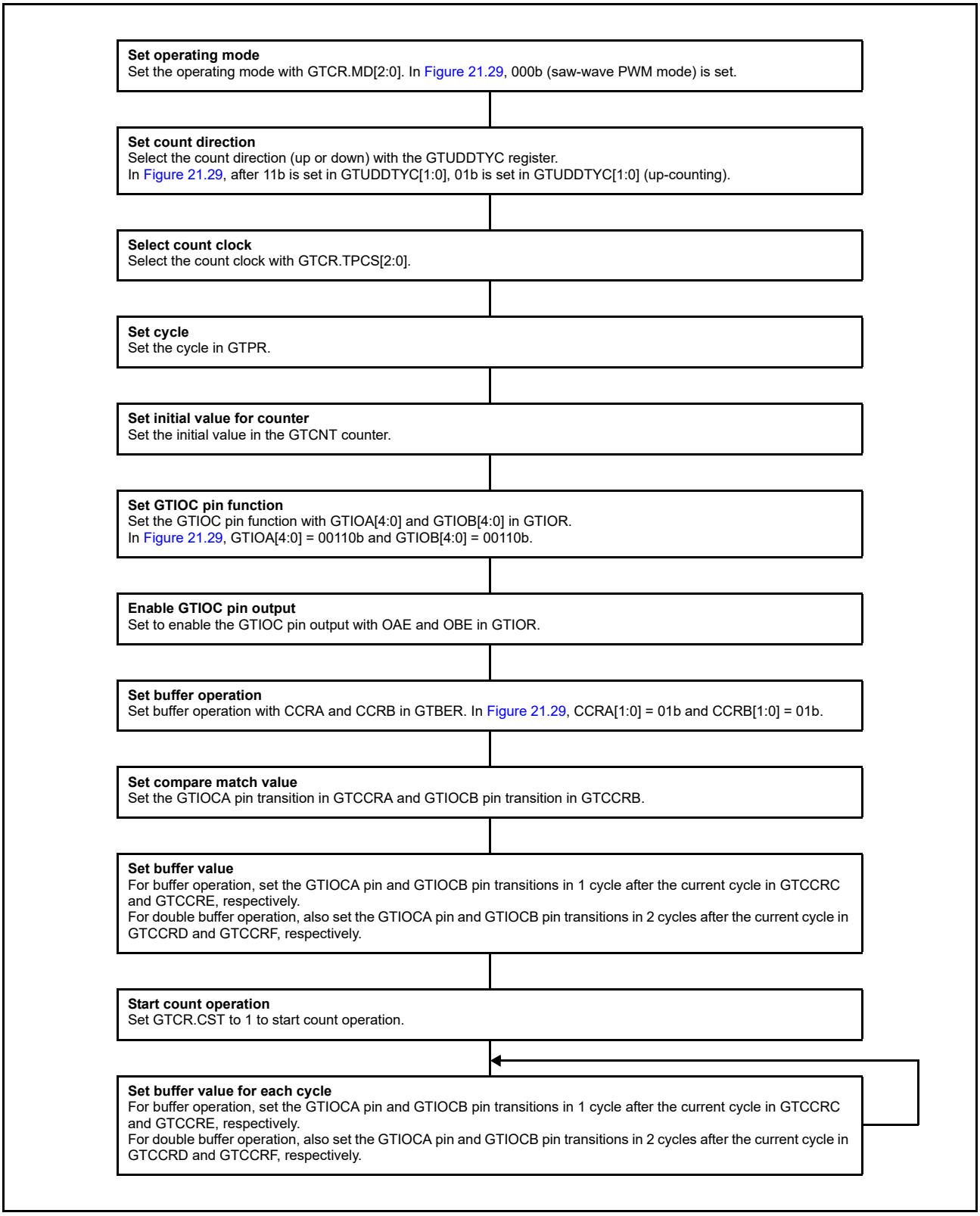


Figure 21.30 Example setting for saw-wave PWM mode

21.3.3.2 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 the 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 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 21.31 shows an example of saw-wave one-shot pulse mode operation, and Figure 21.32 shows an example setting for saw-wave one-shot pulse mode.

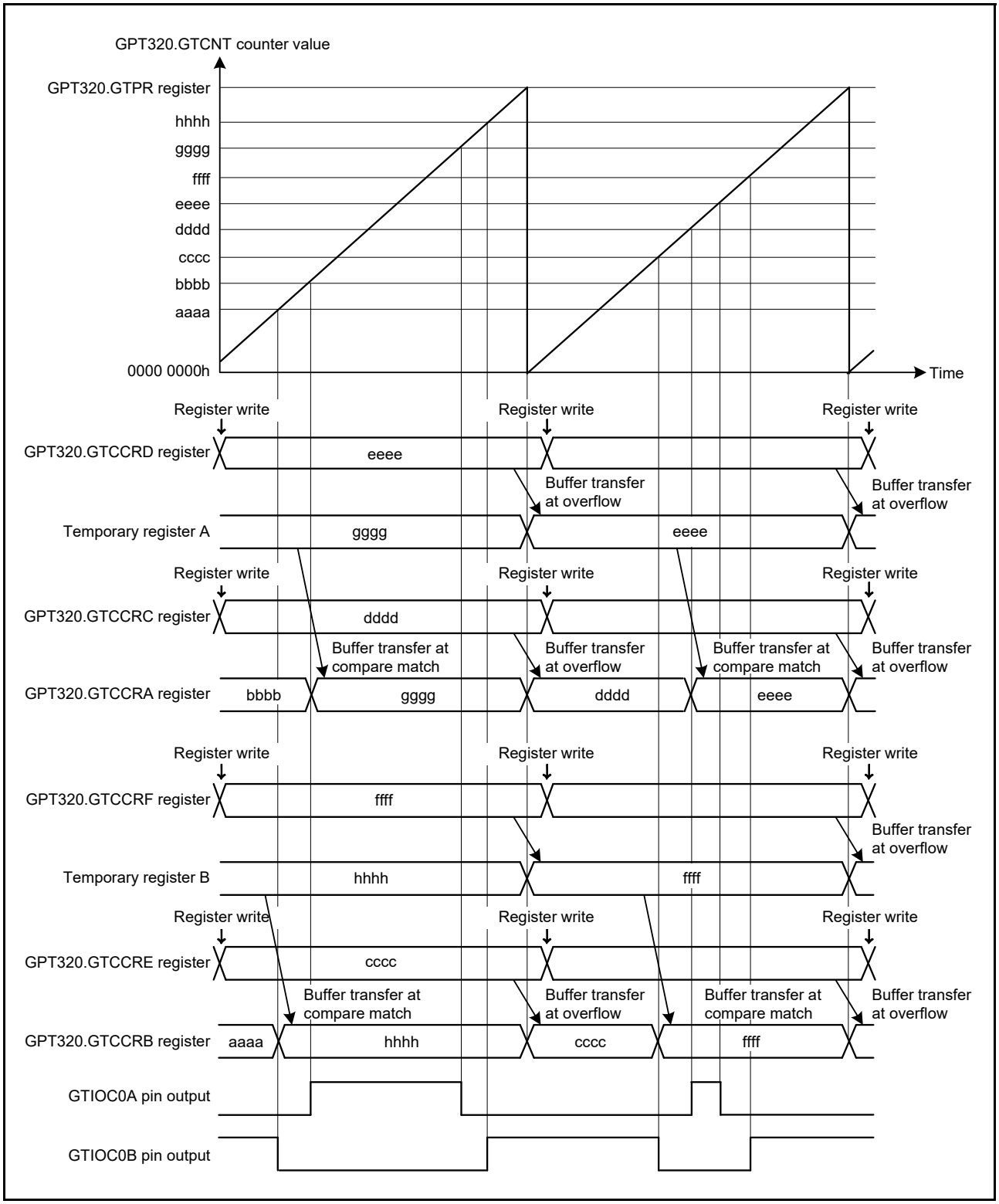


Figure 21.31 **Example of saw-wave one-shot pulse mode operation with up-counting, low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/ GTCCRB compare match, and output retained at cycle end**

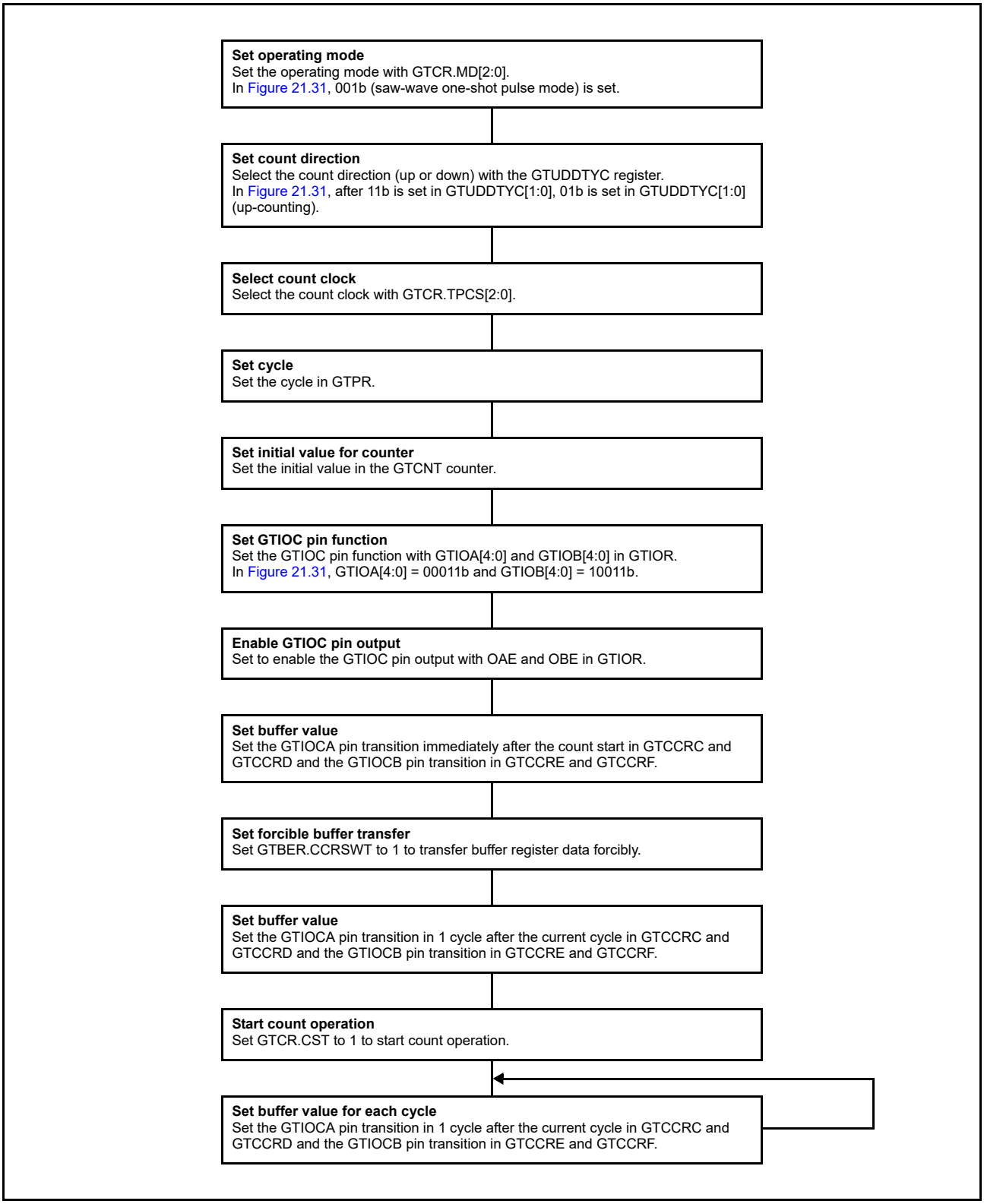


Figure 21.32 Example setting for saw-wave one-shot pulse mode

21.3.3.3 Triangle-wave PWM mode 1 (32-bit transfer at trough)

The triangle-wave PWM mode 1 is a mode in which the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCA or GTIOCB pin when a GTCCRA or GTCCRB compare match occurs. Buffer transfer is performed at the trough. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 21.33 shows an example of a triangle-wave PWM mode 1 operation, and Figure 21.34 shows an example for setting a triangle-wave PWM mode 1.

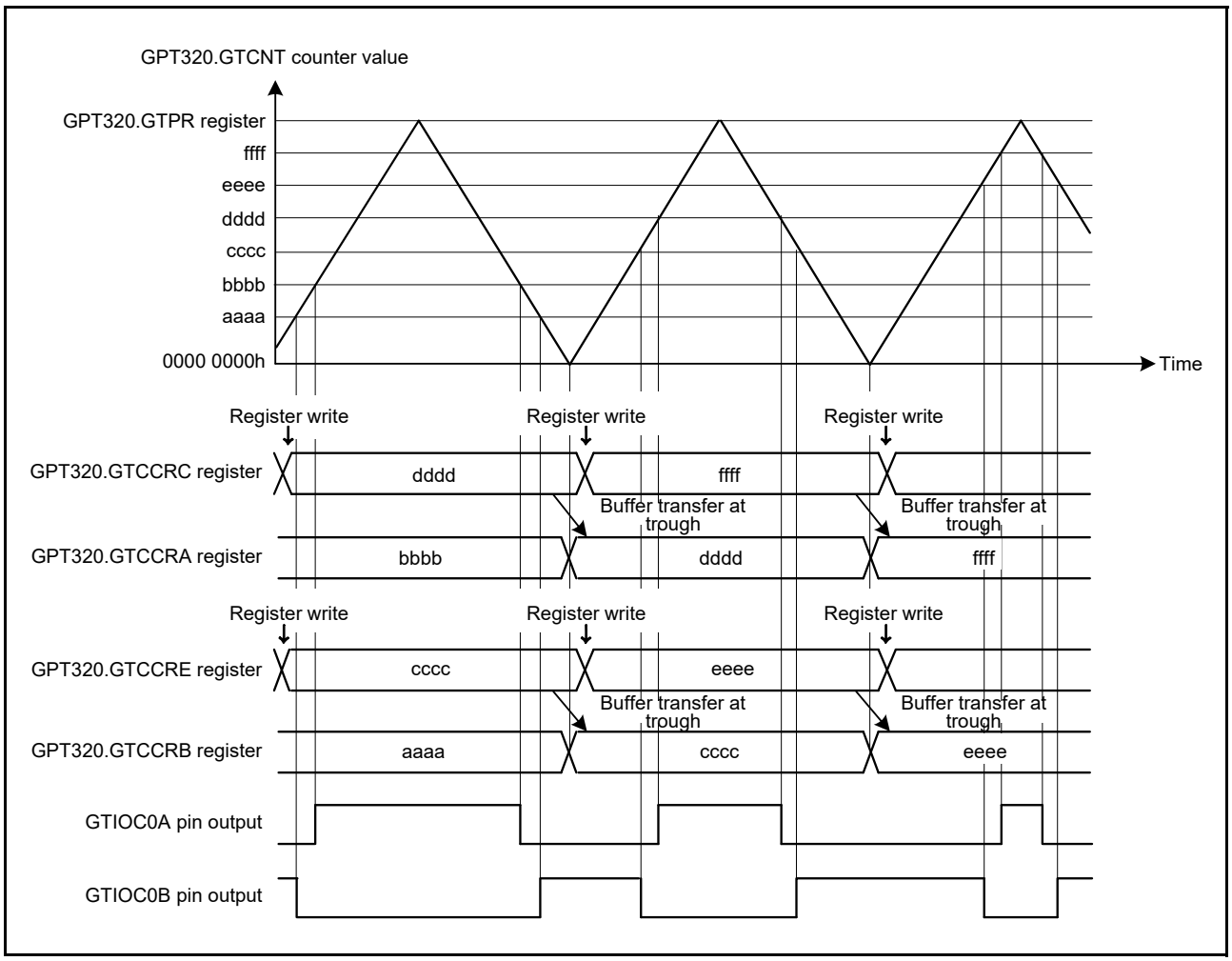


Figure 21.33 Example of triangle-wave PWM mode 1 operation with buffer operation, low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/ GTCCRB register compare match, and output retained at cycle end

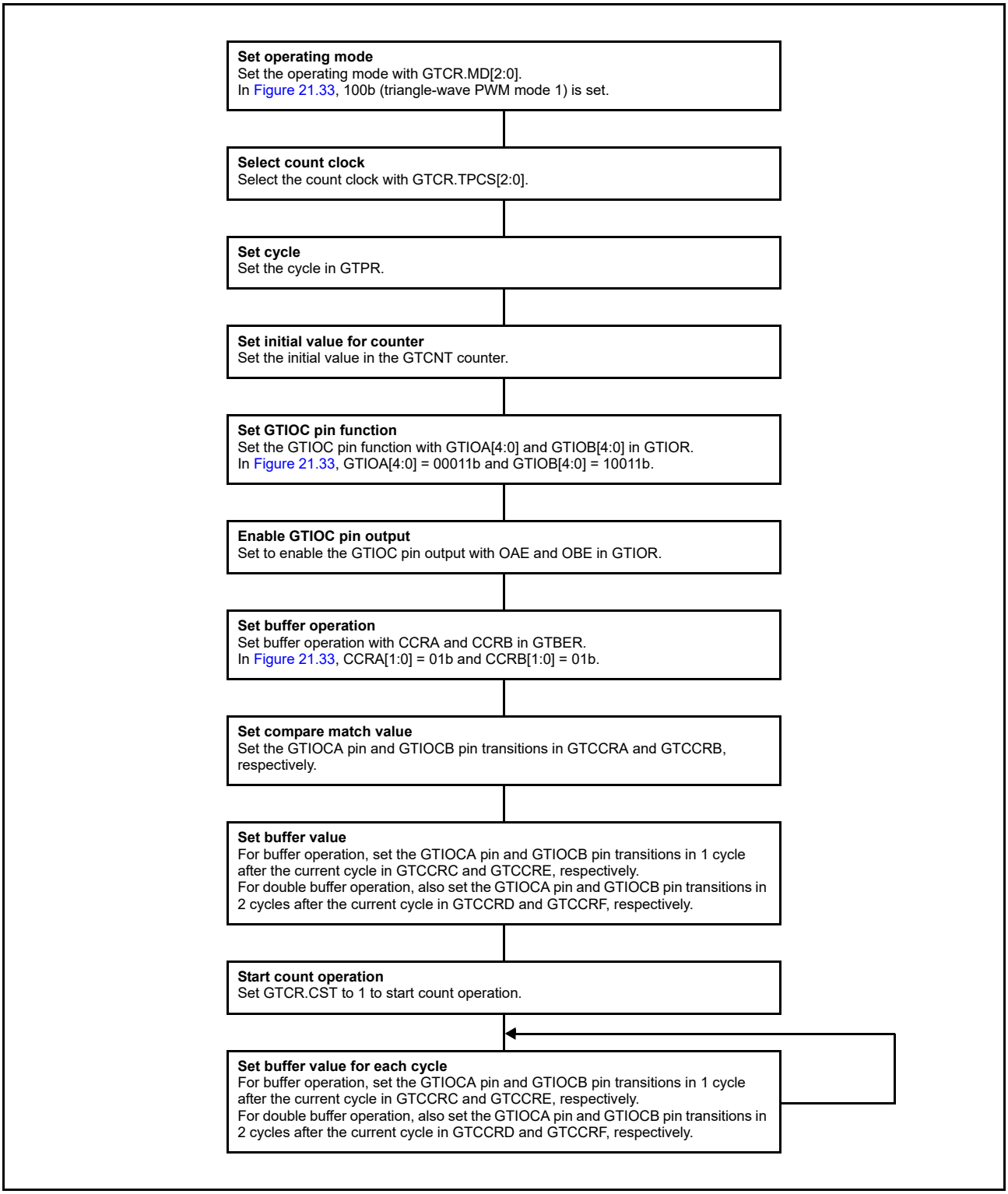


Figure 21.34 Example setting for triangle-wave PWM mode 1

21.3.3.4 Triangle-wave PWM mode 2 (32-bit transfer at crest and trough)

Similarly to triangle-wave PWM mode 1, in triangle-wave PWM mode 2 the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCA or GTIOCB pin when a GTCCRA or GTCCRB compare match occurs. The buffer transfer is performed at both crests and troughs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 21.35 shows an example of triangle-wave PWM mode 2 operation, and Figure 21.36 shows an example setting for triangle-wave PWM mode 2.

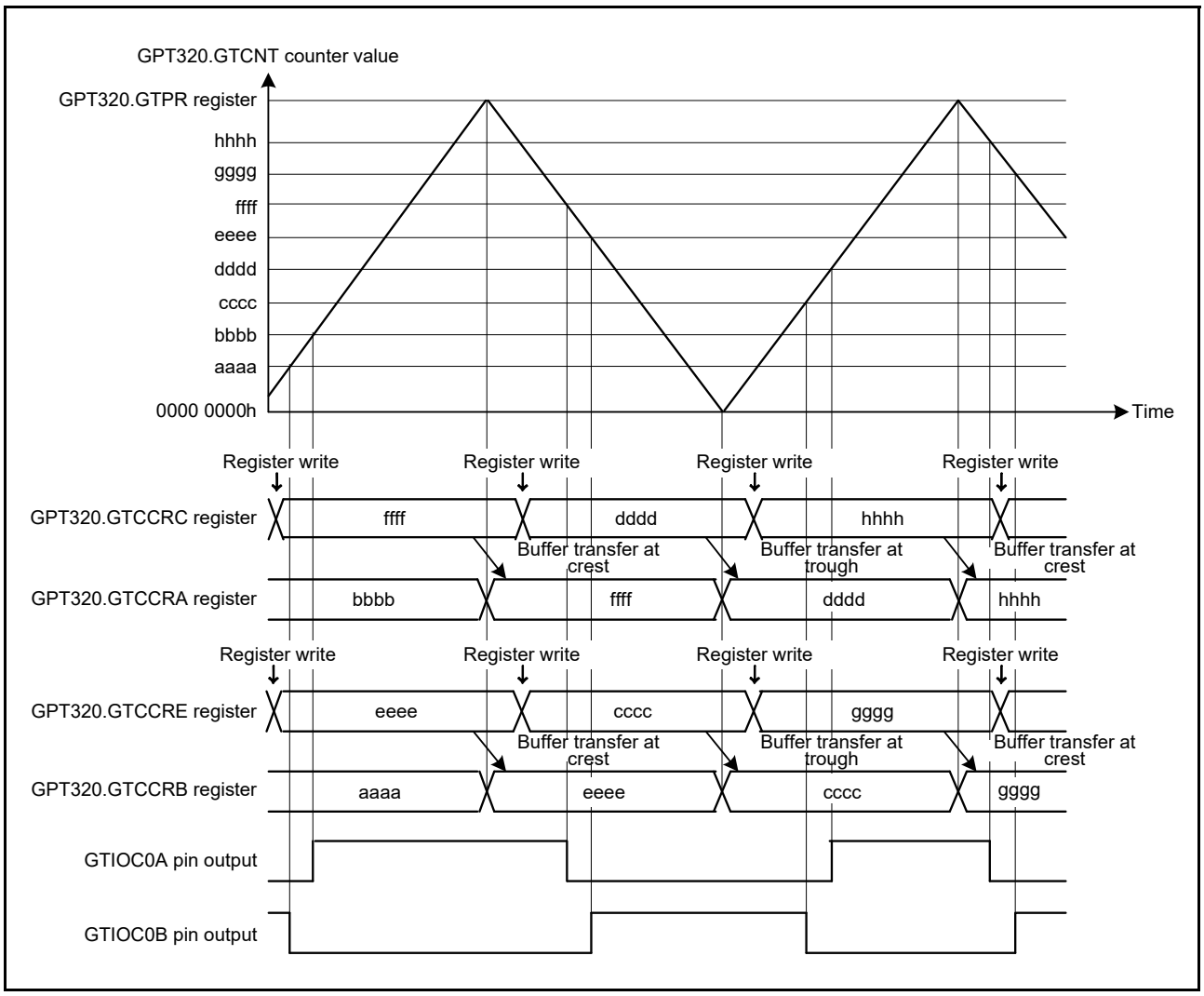


Figure 21.35 Example of triangle-wave PWM mode 2 operation with buffer operation, low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/GTCCRB compare match, and output retained at cycle end

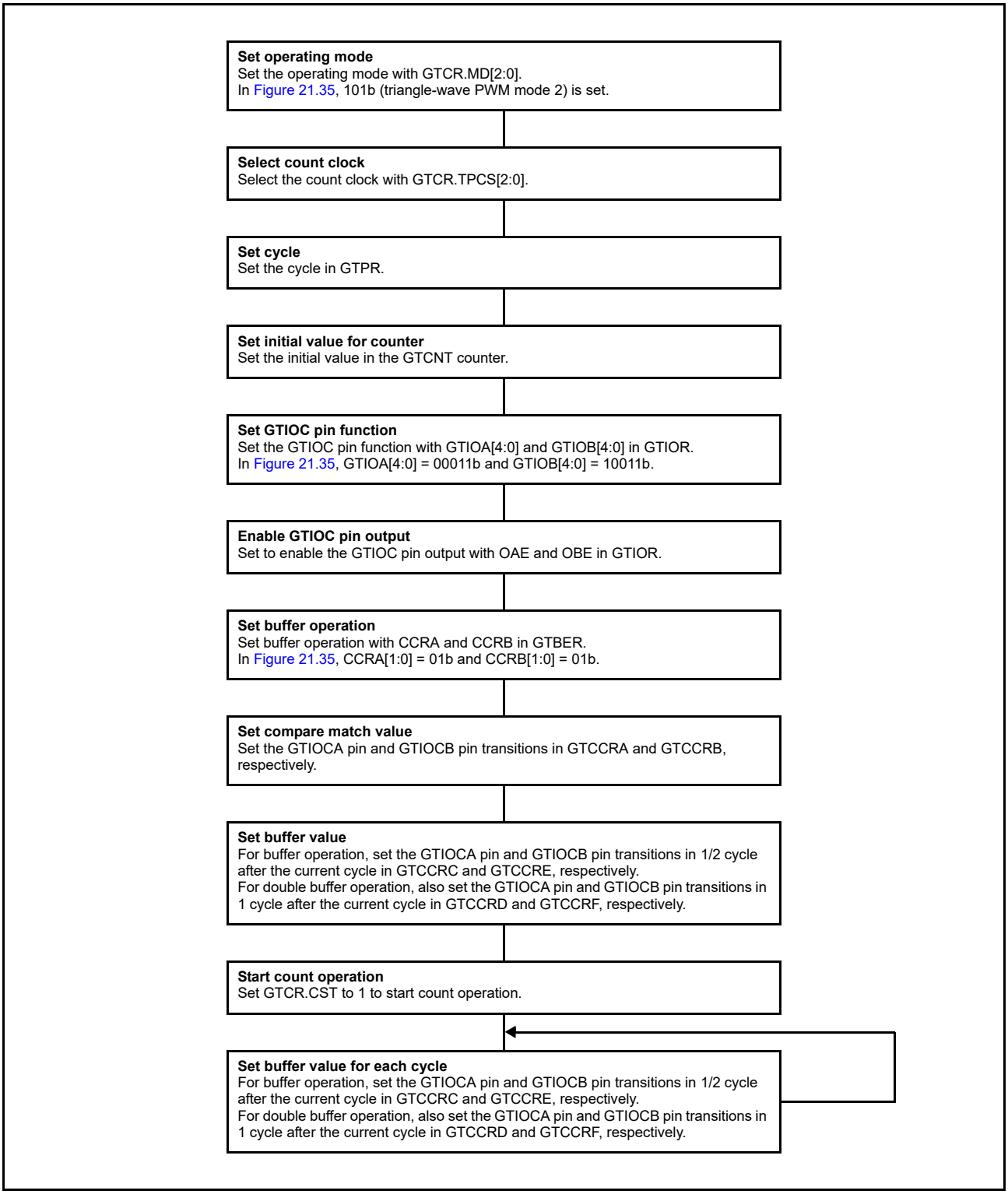


Figure 21.36 Example for setting triangle-wave PWM mode 2

21.3.3.5 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 21.37 shows an example of triangle-wave PWM mode 3 operation, and Figure 21.38 shows an example setting for triangle-wave PWM mode 3.

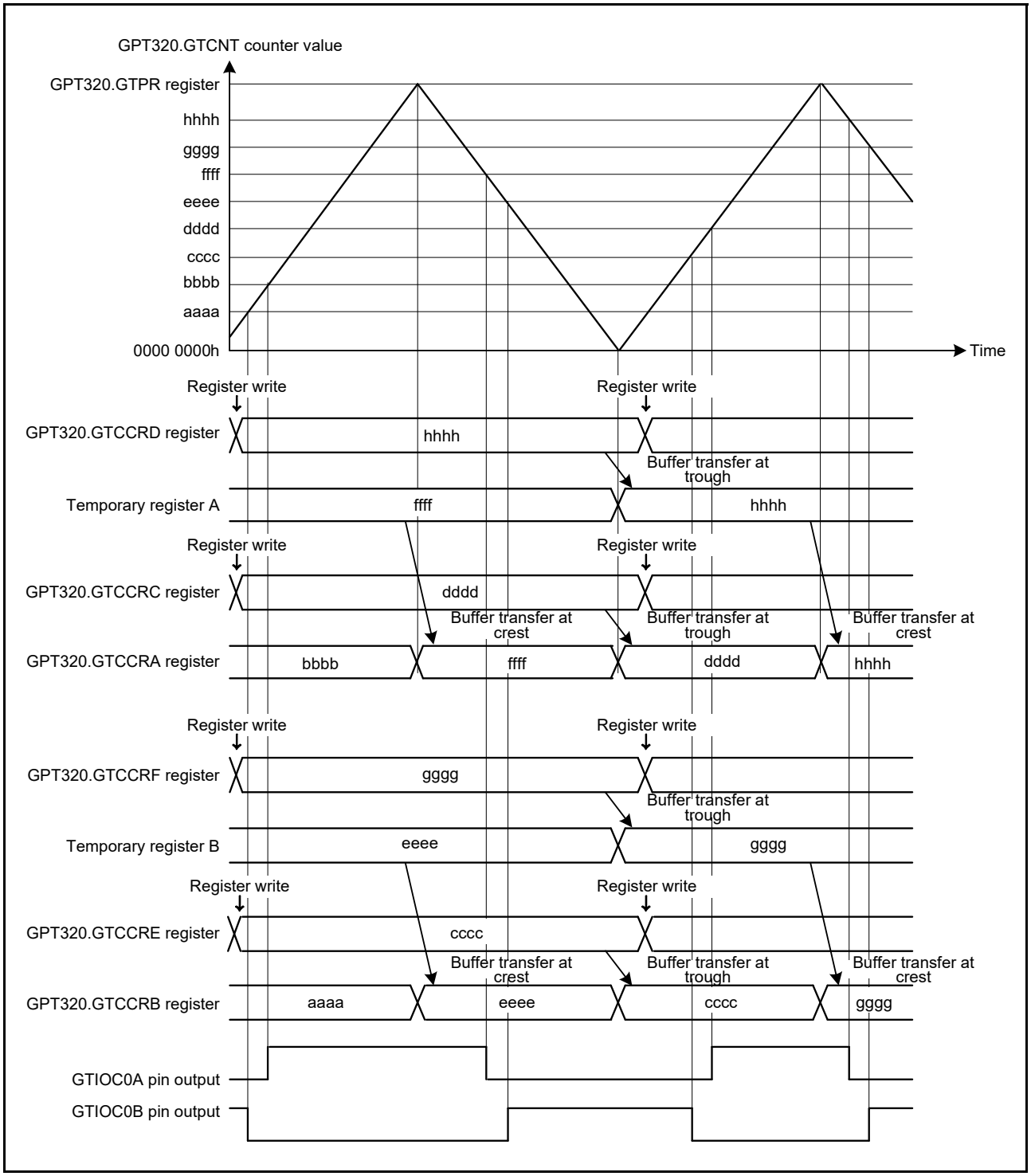


Figure 21.37 **Example of triangle-wave PWM mode 3 operation with low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/GTCCRB compare match, and output retained at cycle end**

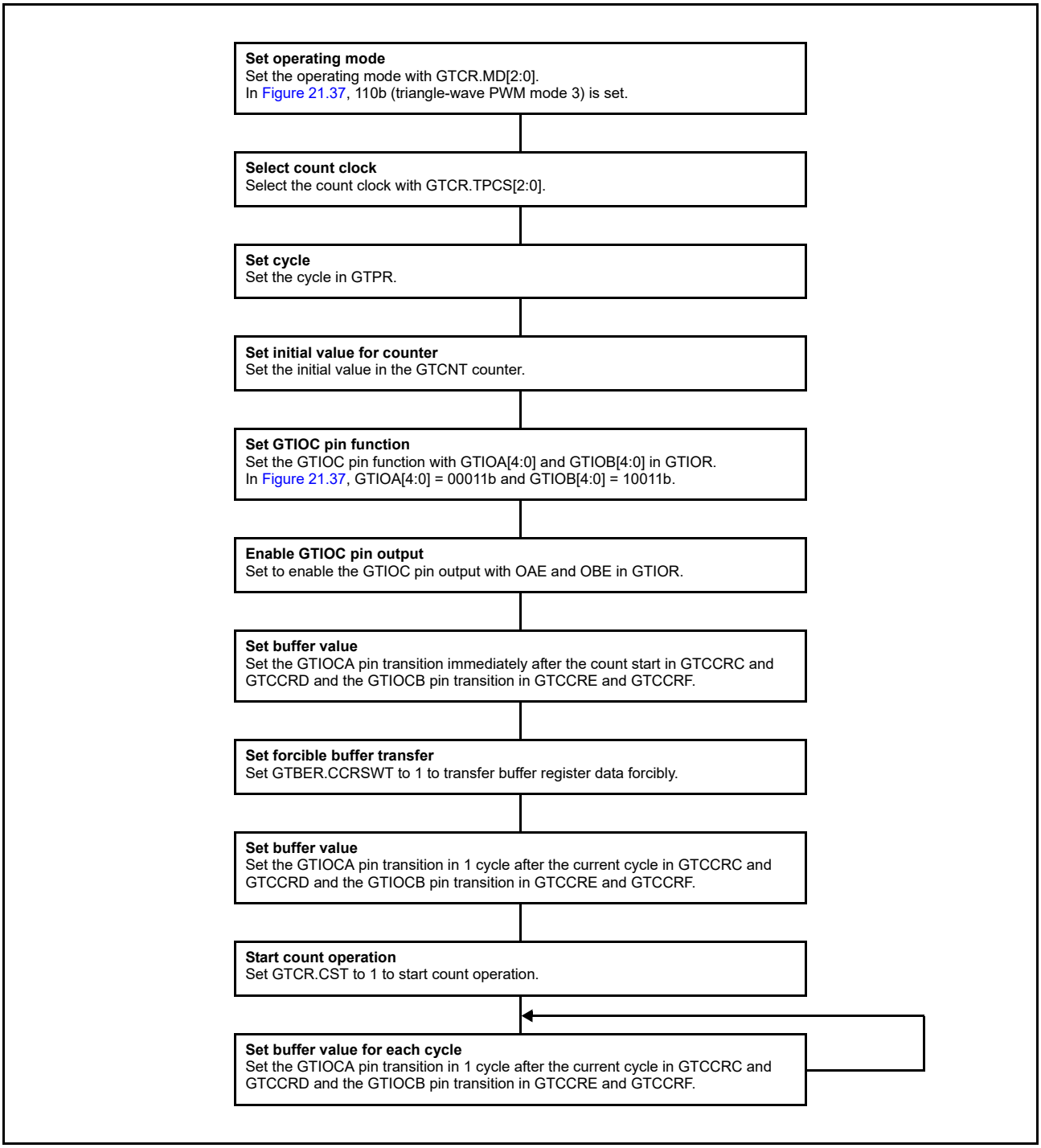


Figure 21.38 Example setting for triangle-wave PWM mode 3

21.3.4 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 21.39 to Figure 21.42 show examples of automatic dead time setting function operation. Figure 21.43 and Figure 21.44 show the setting examples.

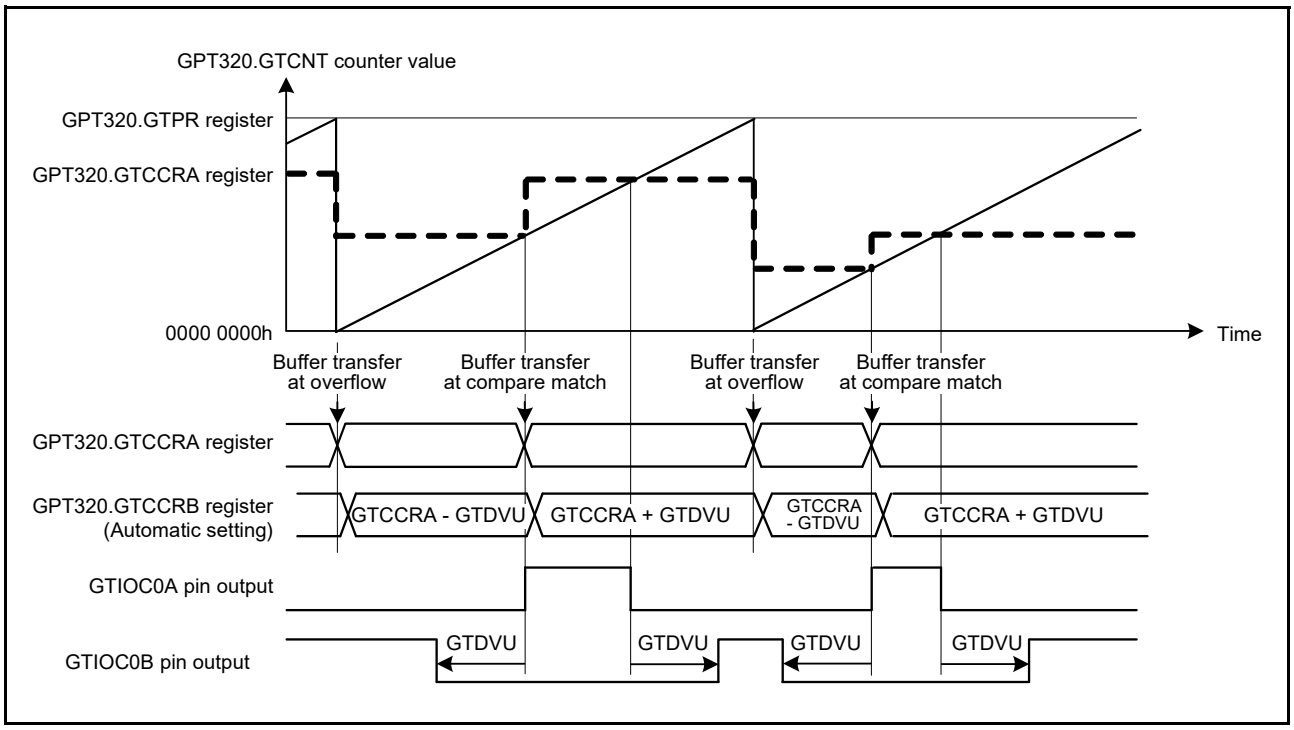


Figure 21.39 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, up-counting, and active-high

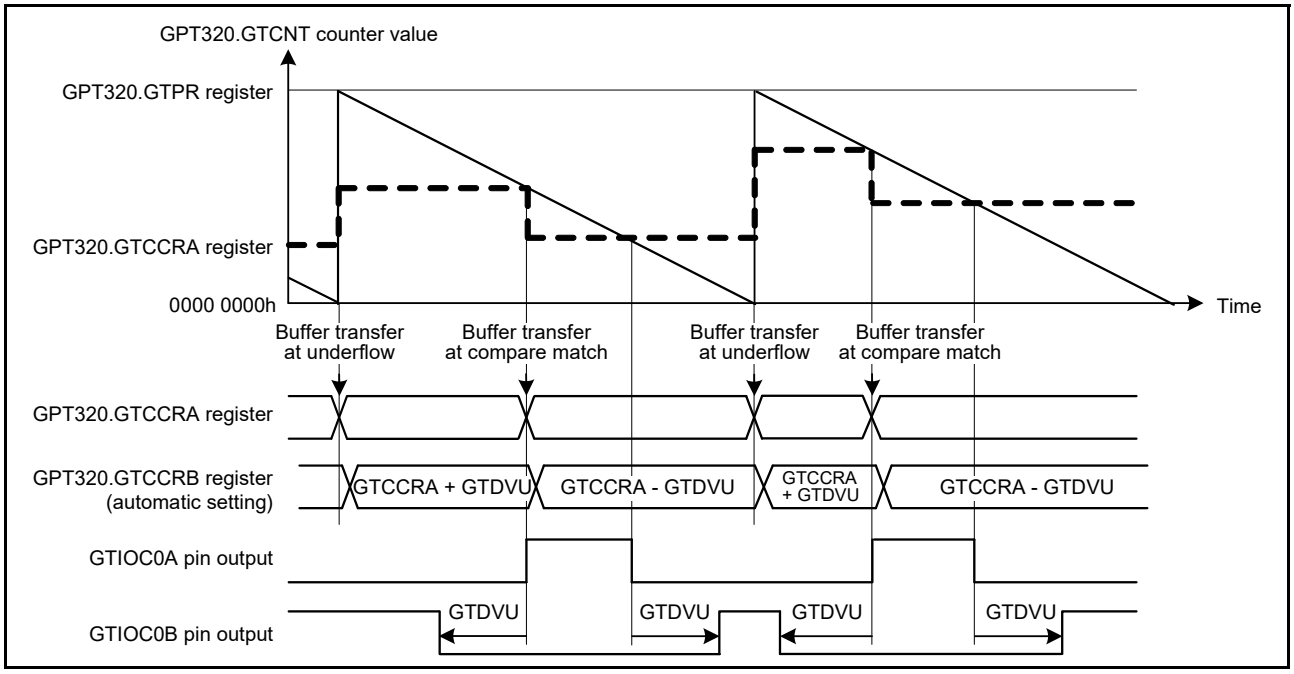


Figure 21.40 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, down-counting, and active-high

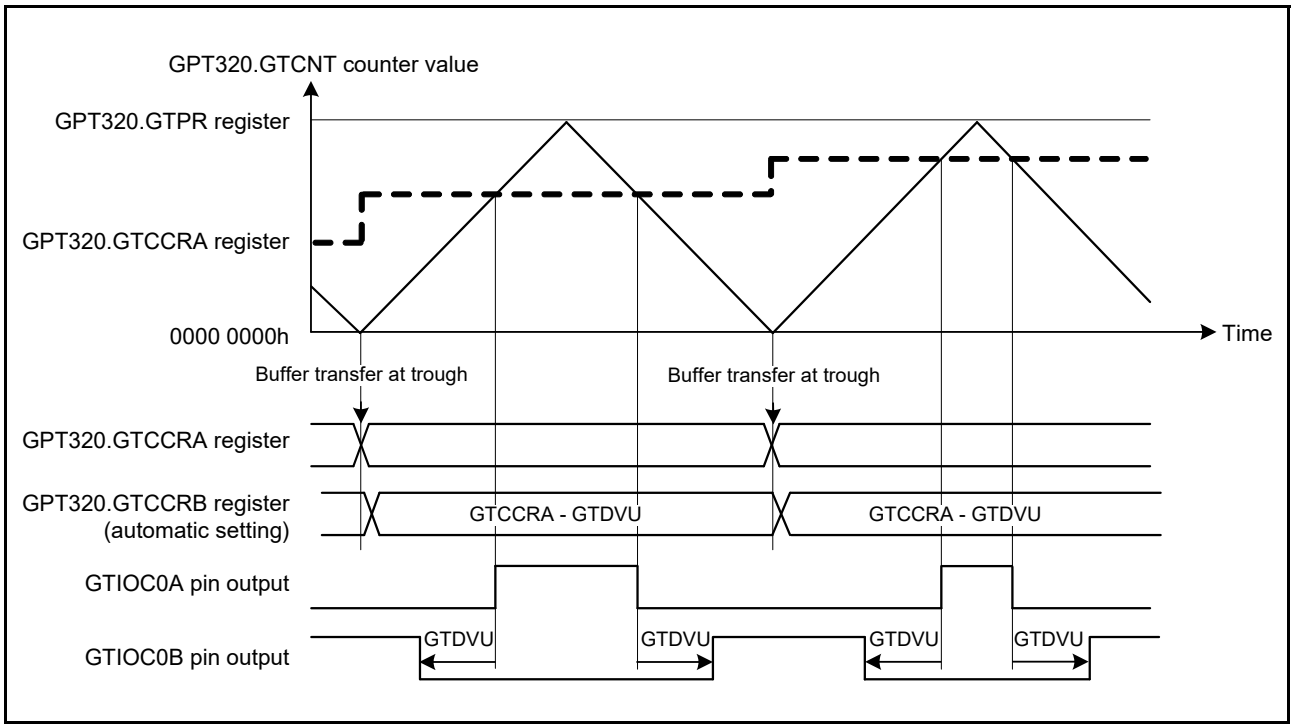


Figure 21.41 Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 1, and active-high

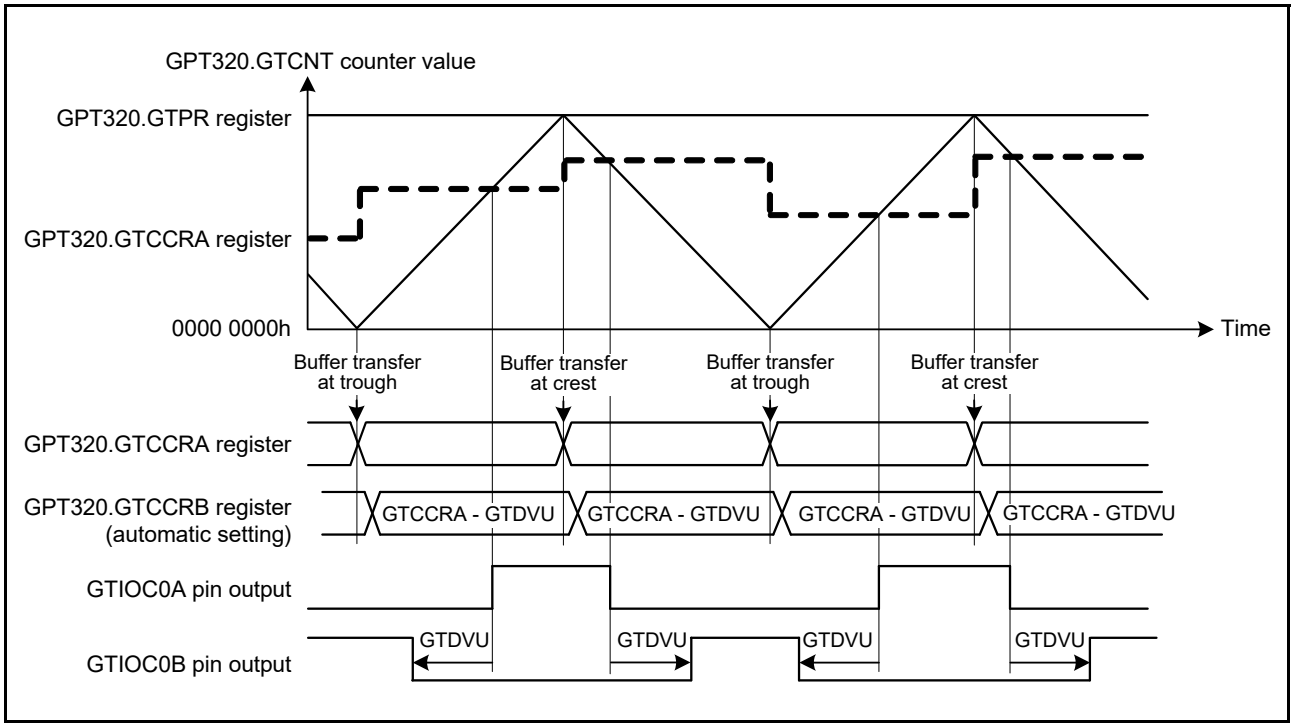


Figure 21.42 Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 2 or 3, and active-high

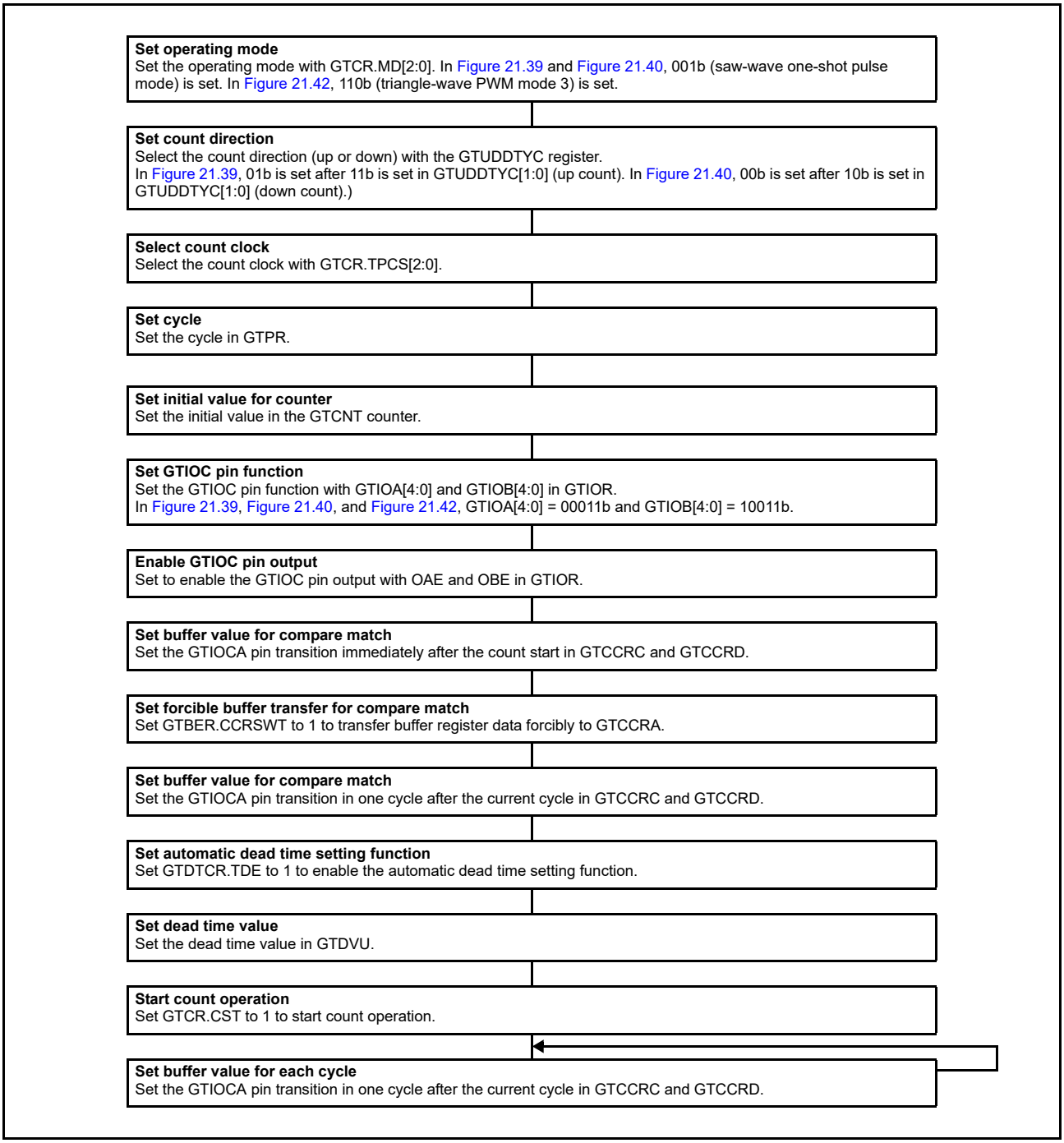


Figure 21.43 Example setting for automatic dead time setting function in saw-wave one-shot pulse mode, and triangle-wave PWM mode 3

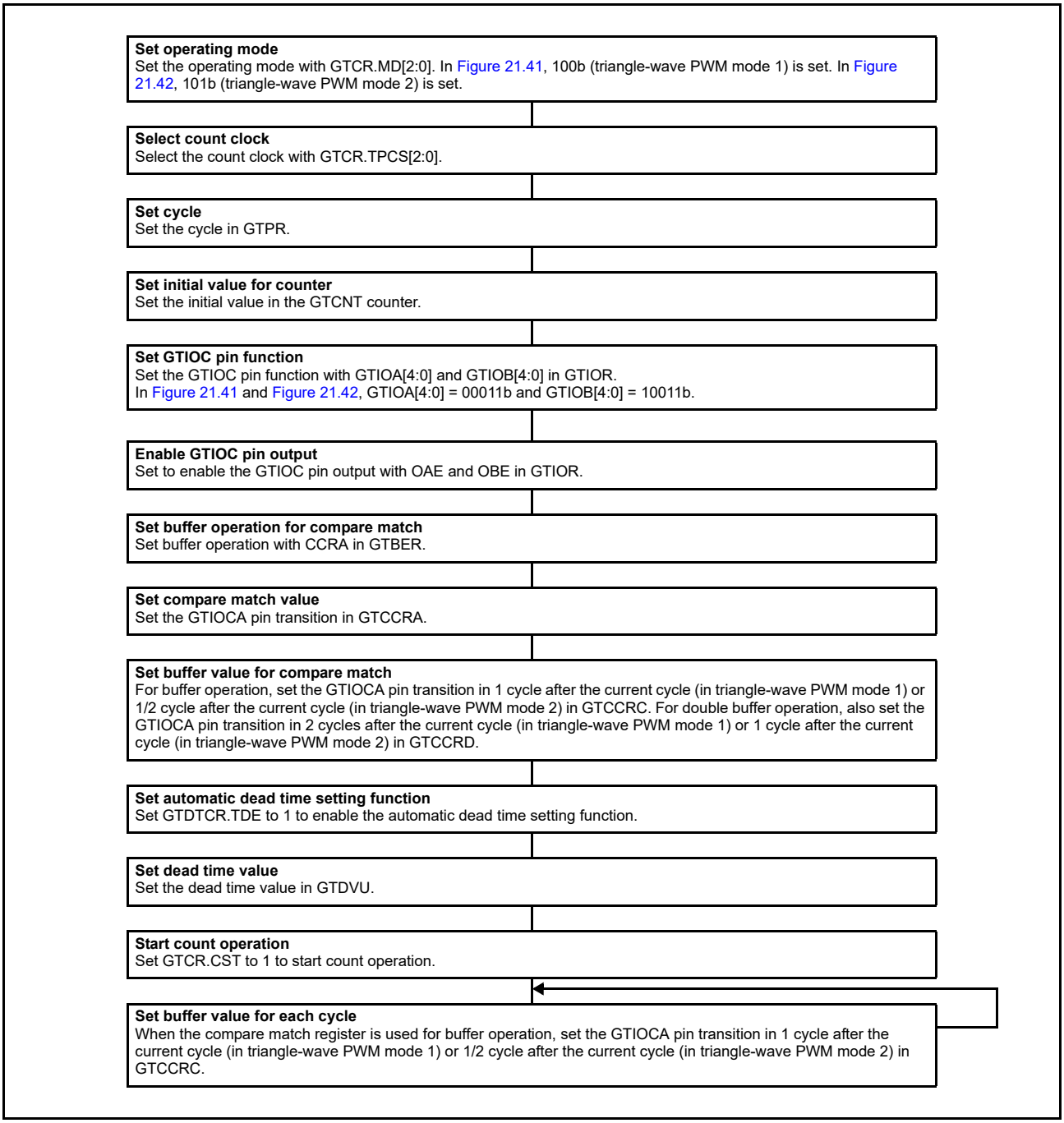


Figure 21.44 Example setting for automatic dead time setting function in triangle-wave PWM mode 1 or 2

21.3.5 Count Direction Changing Function

The count direction of the GTCNT counter can be changed by modifying the UD bit in GTUDDTYC.

In saw-wave mode, if the UD bit in GTUDDTYC is modified during count operation, the count direction is changed at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the GTUDDTYC.UD bit is modified while the count operation stops and the GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit modification is not reflected at the start of counting and the count direction is changed at an overflow or an underflow. If the UDF bit is set to 1 while the count operation 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 21.45 shows an example operation of count direction changing function.

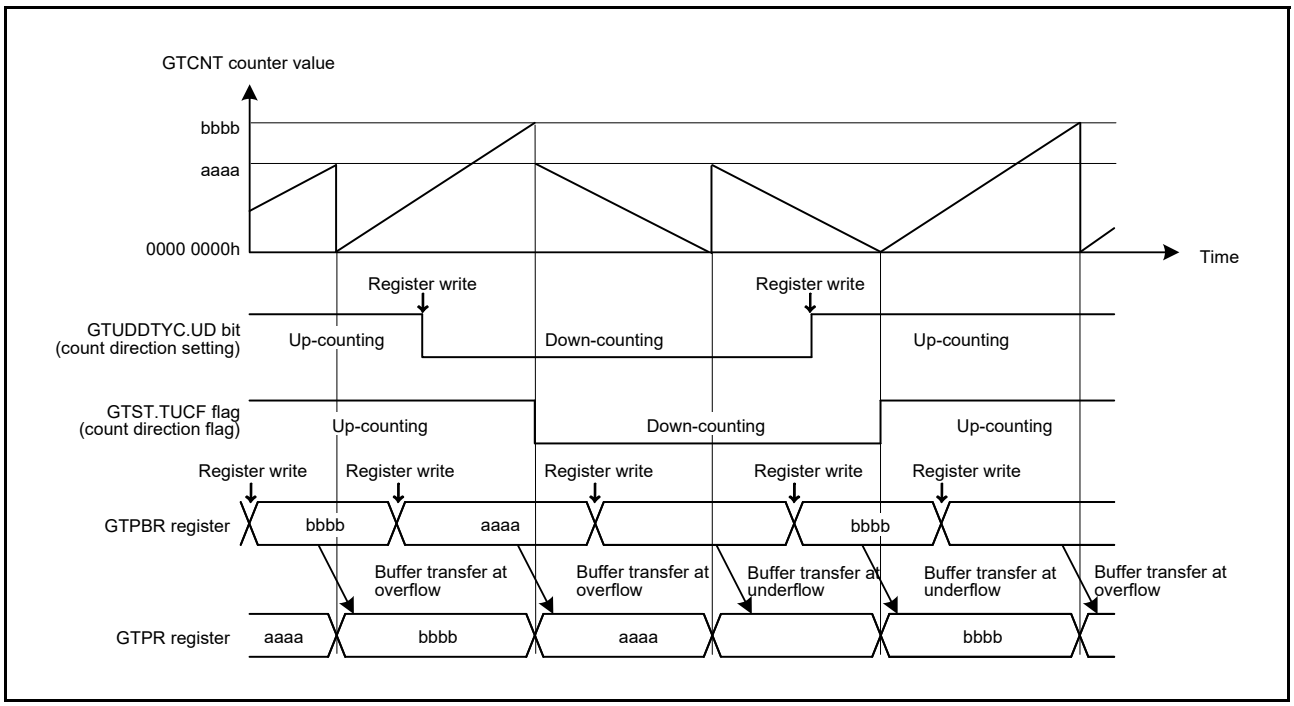


Figure 21.45 Example operation of a count direction changing function during buffer operation

21.3.6 Function of Output Duty 0% and 100%

The output duty of the GTIOCA pin and the GTIOCB pin are set to 0% or 100% by changing the GTUDDTYC.OADTY bit or GTUDDTYC.OBDTY bit.

In saw-wave mode, if the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified during the count operation, the output duty setting is reflected at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation 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 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%/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 determined by GTIOR.GTIOB[3:2] and GTUDDTYC.OBDTYR.

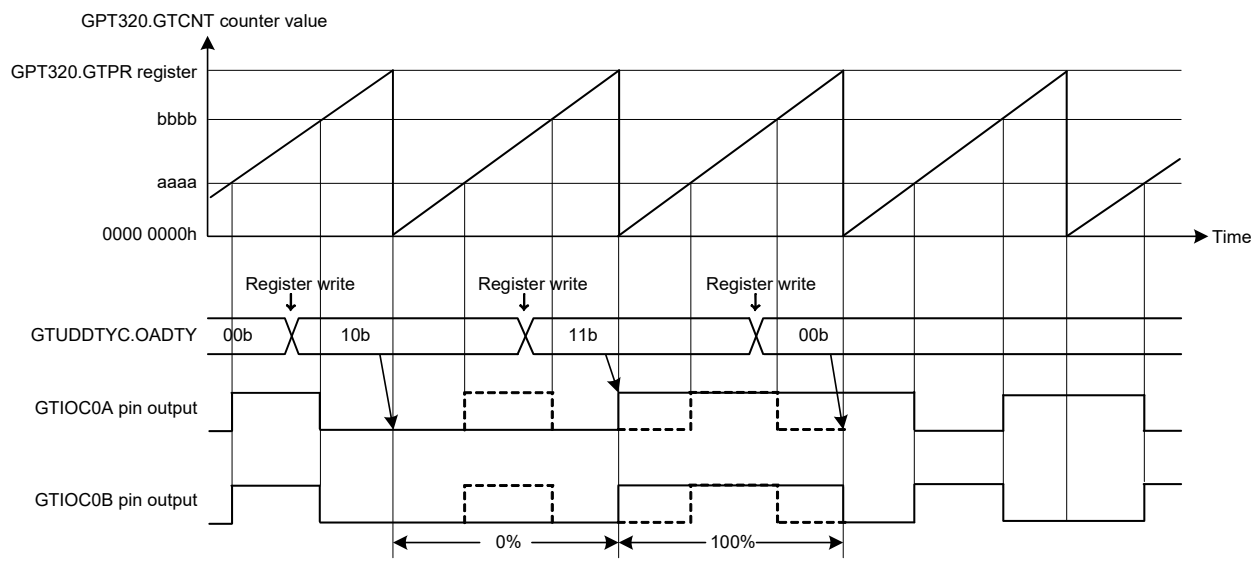
When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 01b, the output pins output low at cycle end. When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 10b, the output pins output high at cycle end.

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 21.6 shows the values of GTIOCA/GTIOCB pin output at cycle end.

Table 21.6 Output values after releasing 0%/100% duty setting (m = A, B)

GTIOR.GTIOm[3:2]	Compare match value at cycle end masked by 0%/100% duty setting	GTUDDTYC.OmDTYR in duty 0% setting		GTUDDTYC.OmDTYR in duty 100% setting	
		0	1	0	1
00 (output retained at cycle end)	0	0	0	1	0
	1	0	1	1	1
01 (low output at cycle end)	—	0	0	0	0
10 (high output at cycle end)	—	1	1	1	1
11 (output toggled at cycle end)	0	1	1	0	1
	1	1	0	0	0

Figure 21.46 shows an example of output duty 0% and 100% function.



[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 21.46 Example of output duty 0% and 100% function

21.3.7 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.

21.3.7.1 Hardware start operation

The GTCNT counter can be started by selecting a hardware source using GTSSR.

Figure 21.47 shows an example of a count start operation by a hardware source. Figure 21.48 shows the setting example.

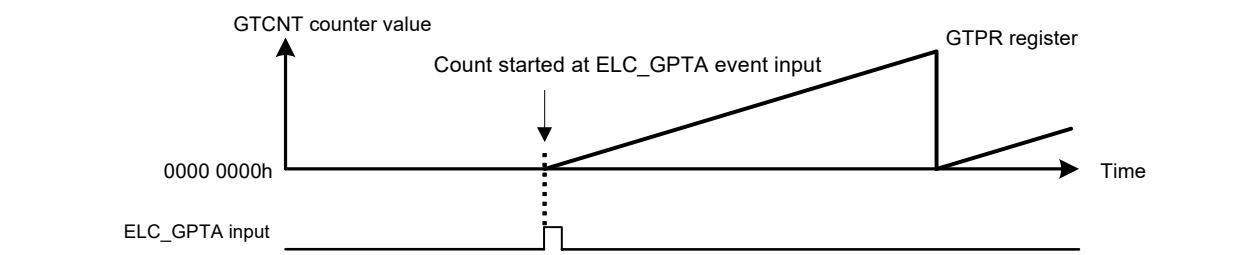


Figure 21.47 Example of count start operation by a hardware source, started at the input of the signal from the ELC_GPTA event

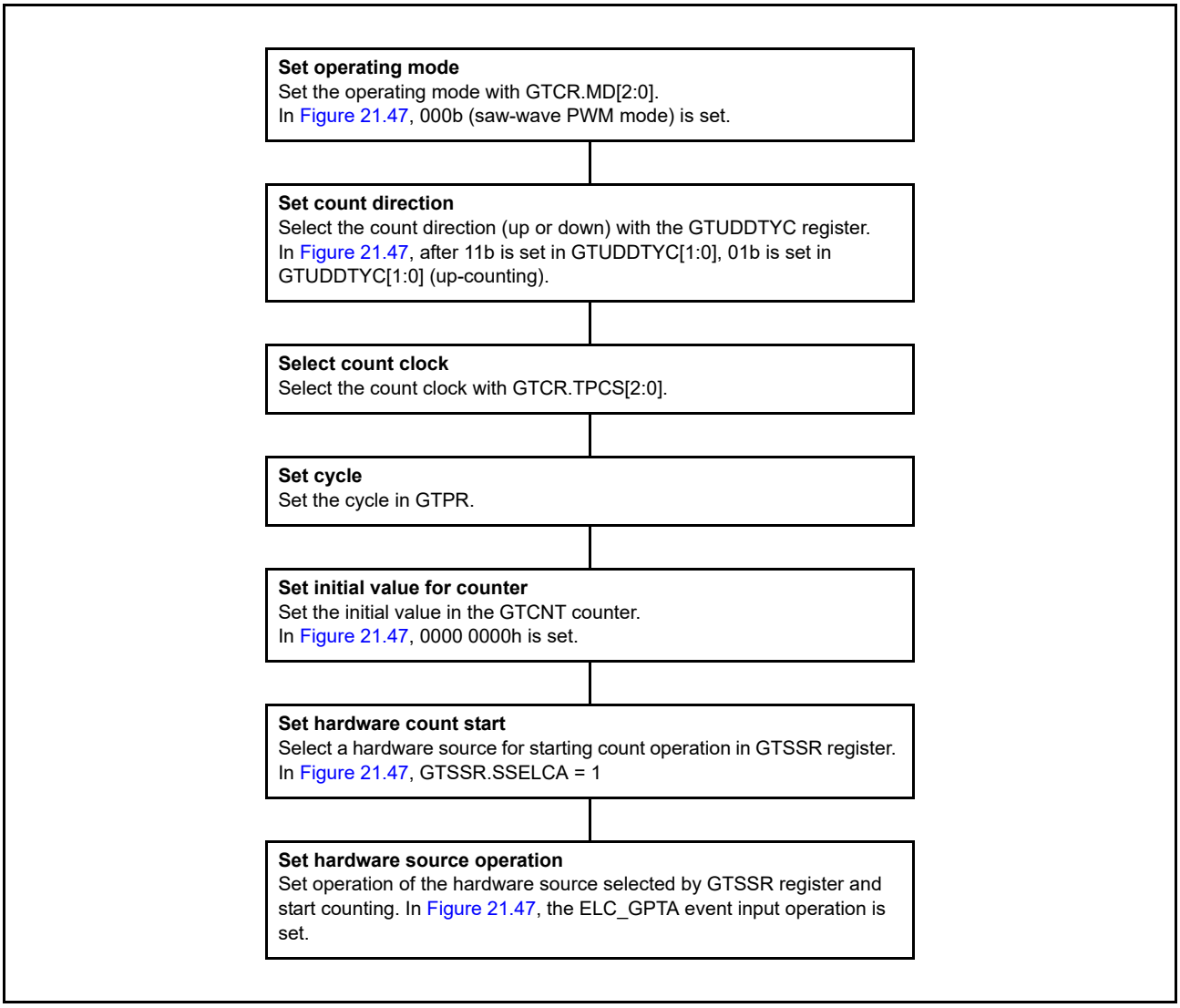


Figure 21.48 Example setting for count start operation by a hardware source

21.3.7.2 Hardware stop operation

The GTCNT counter can be stopped by selecting a hardware source using GTPSR.

Figure 21.49 shows an example of a count stop operation by a hardware source. Figure 21.50 shows the setting example. In this example, the count operation stops at the edge of the ELC_GPTA event input and restarts at the edge of the ELC_GPTB event input.

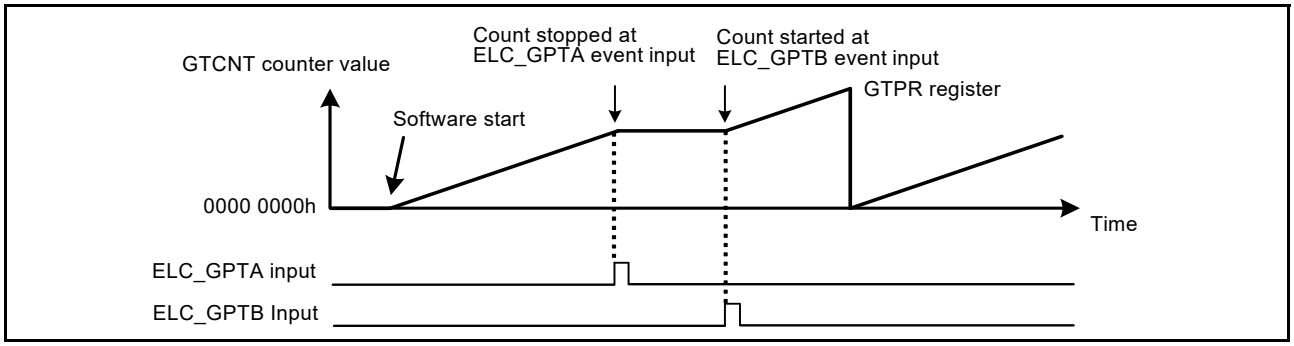


Figure 21.49 Example of count stop operation by hardware source started by software, stopped at ELC_GPTA event input, and restarted at ELC_GPTB event input

Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 21.49, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 21.49, 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 21.49, 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 21.49, 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 21.49, 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 21.49, ELC_GPTA input operation and ELC_GPTB input operation are set.

Figure 21.50 Example setting for count stop operation by a hardware source

Figure 21.51 shows an example of a count start/stop operation by a hardware source. Figure 21.52 shows the setting example. In this example, the counter operates during the high-level periods of the external trigger input GTETRGA.

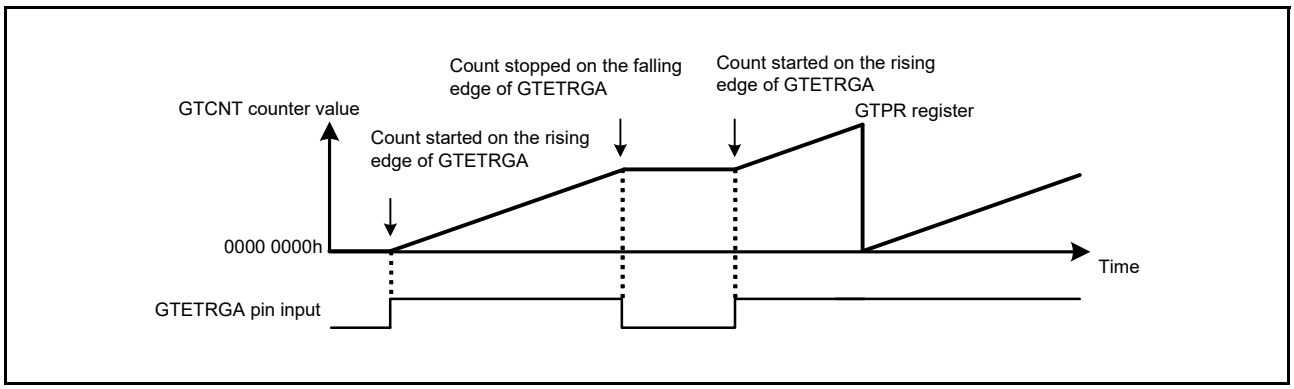


Figure 21.51 Example of count start/stop operation by a hardware source, started on the rising edge of GTETRGA pin input, and stopped on the falling edge of GTETRGA pin input

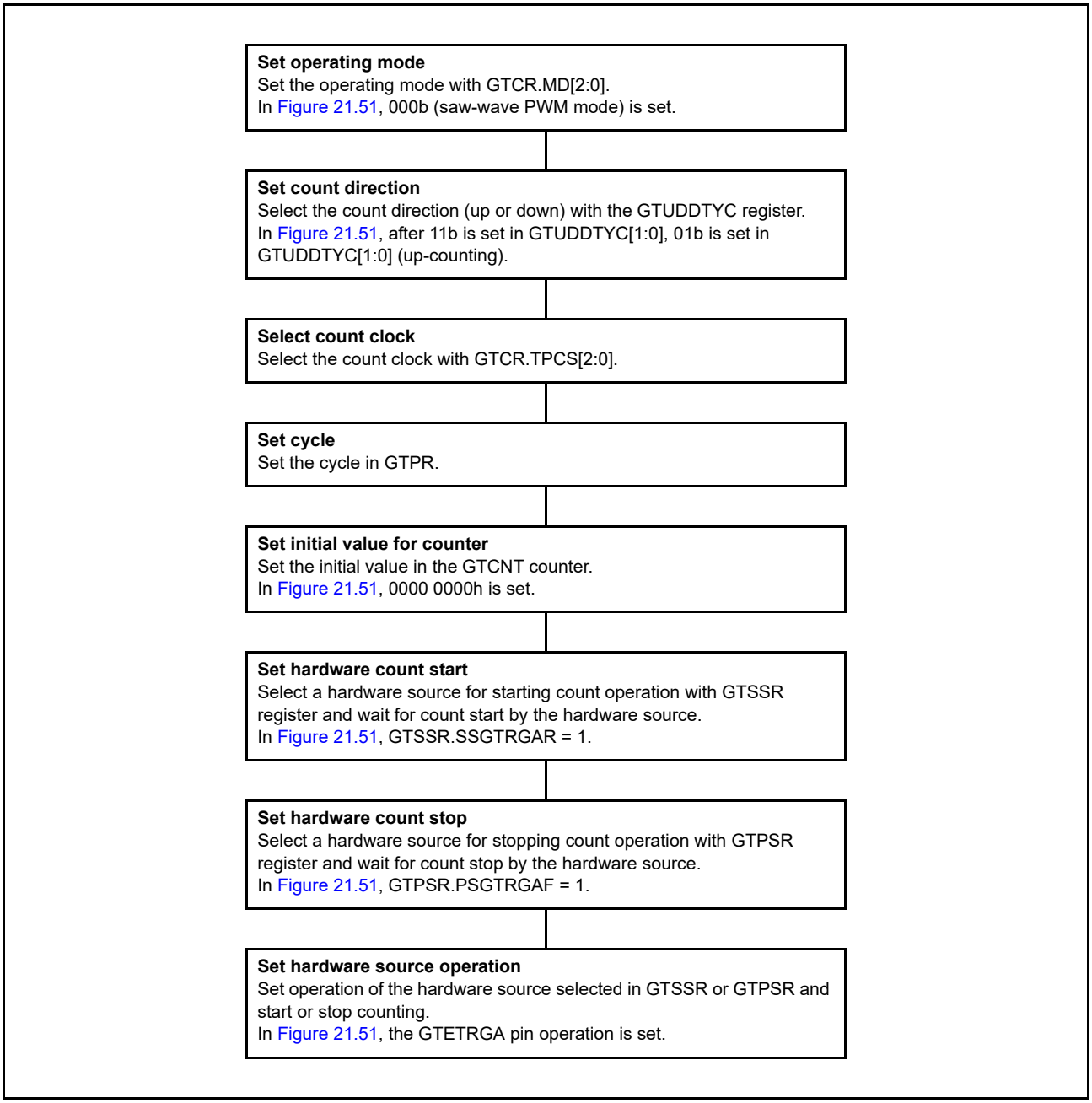


Figure 21.52 Example setting for count start/stop operation by a hardware source

21.3.7.3 Hardware clear operation

The GTCNT counter can be cleared by selecting a hardware source using GTCSR. The GPTn_OVF/GPTn_UDF (n = 0 to 6) interrupt (overflow/underflow interrupt) is not generated when the GTCNT counter is cleared by a hardware source or by software.

Figure 21.53 and Figure 21.54 show examples of the GTCNT counter clearing operation by a hardware source. Figure 21.55 shows the setting example. In this example, the GTCNT counter starts at the edge of the ELC_GPTA input, and the counter stops and clears at the edge of the ELC_GPTB input.

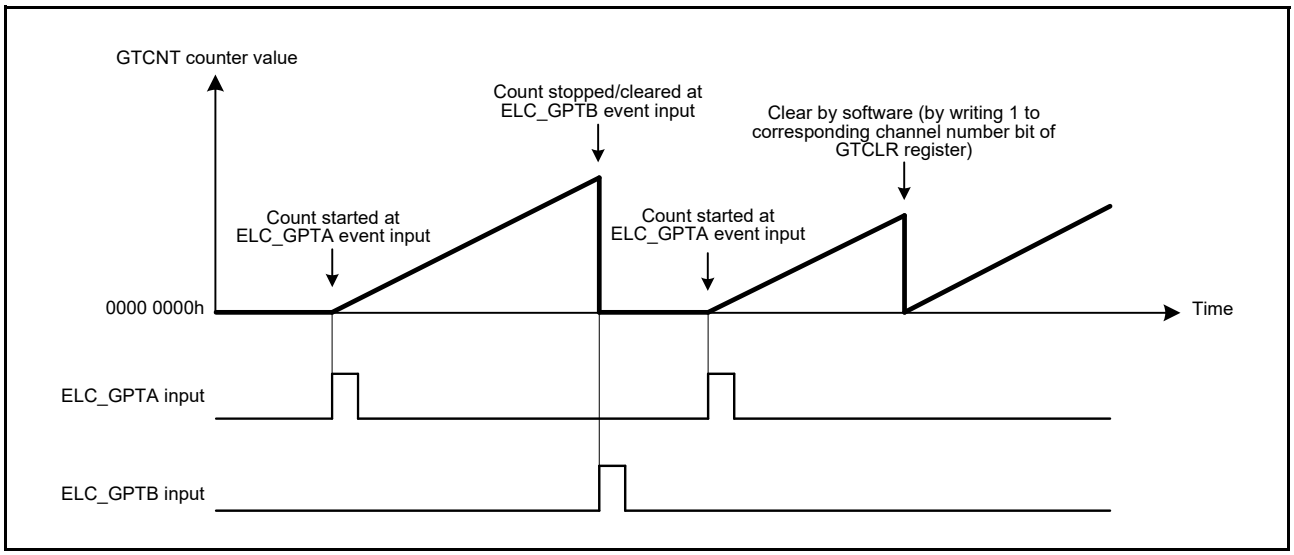


Figure 21.53 Examples of count clearing operation by hardware source in saw-wave up-counting, started at ELC_GPTA event input, and stopped/cleared at ELC_GPTB event input

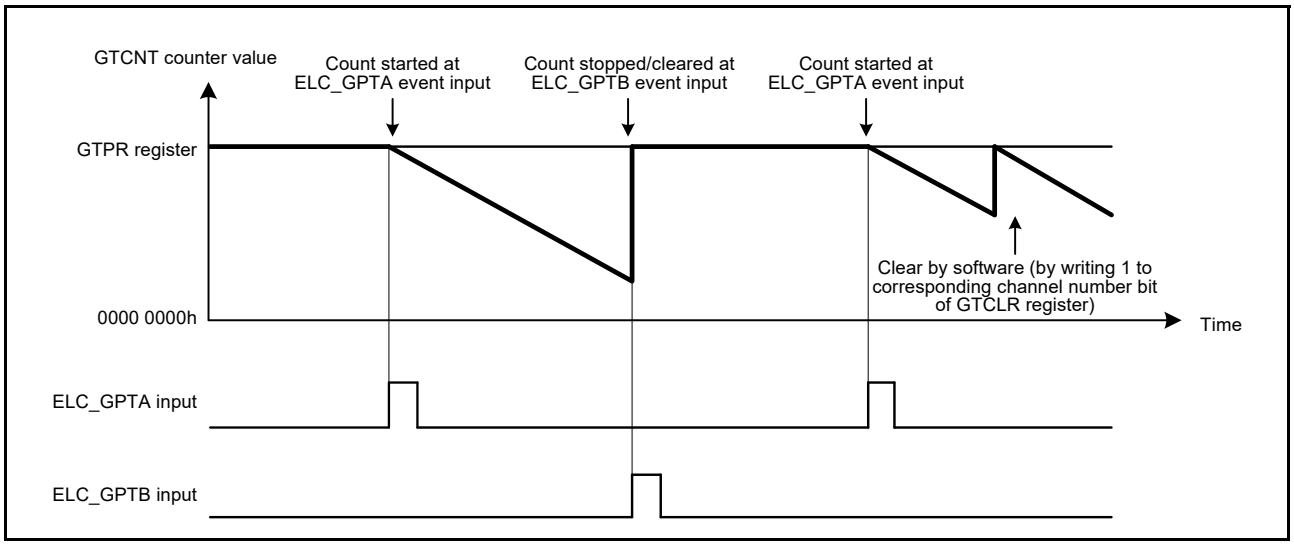


Figure 21.54 Examples of count clearing operation by hardware source in saw-wave down-counting, started at ELC_GPTA event input, and stopped/cleared at ELC_GPTB event input

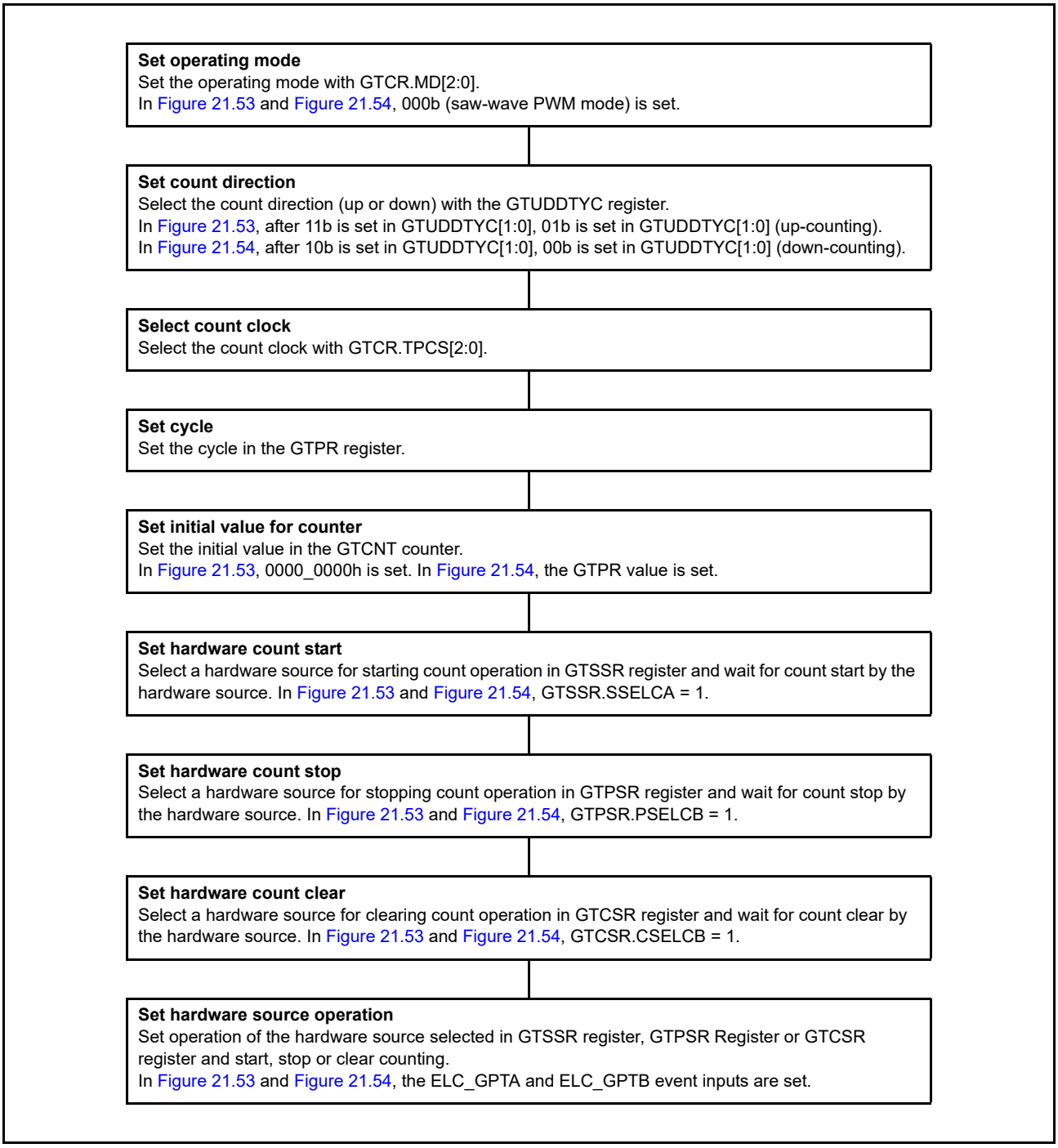


Figure 21.55 Example setting for count clearing operation by a hardware source

The GPTn_OVF/GPTn_UDF (n = 0 to 6) interrupt (overflow/underflow interrupt) is not generated when the counter is cleared by a hardware source or by software.

Figure 21.56 shows the relationship between the counter clearing by a hardware source and the GPTn_OVF (n = 0 to 6) interrupt.

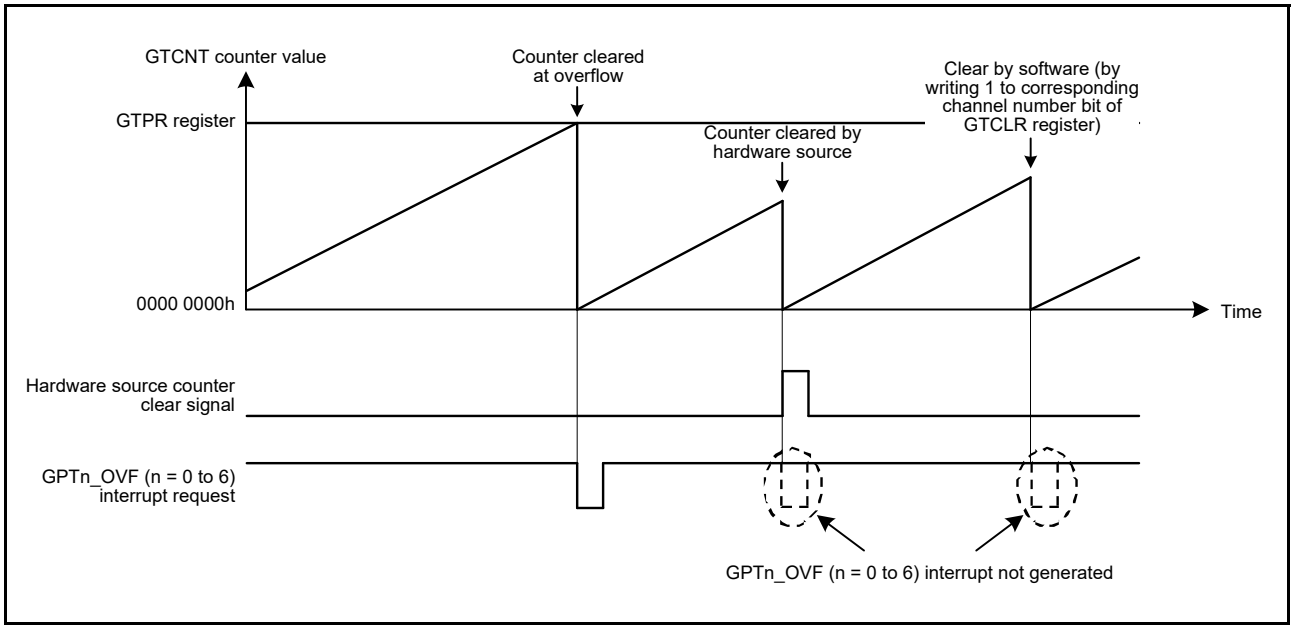


Figure 21.56 Relationship between counter clearing by hardware source and GPTn_OVF (n = 0 to 6) interrupt

21.3.8 Synchronized Operation

Synchronized operation on channels such as a synchronized start, stop, and clear operation can be performed.

21.3.8.1 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 21.57 shows an example of a simultaneous start, stop, and clear by software. Figure 21.58 shows an example of phase start operation by software.

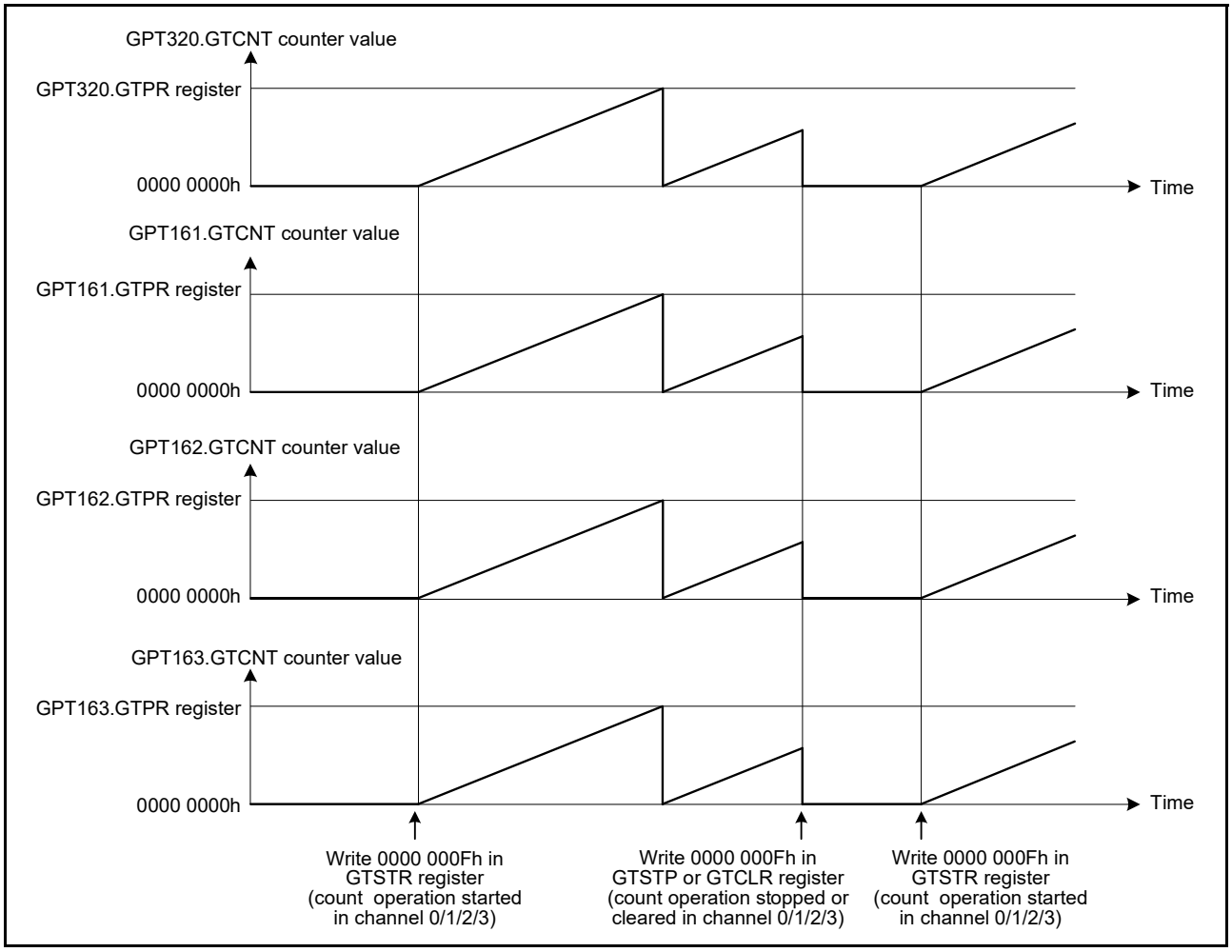


Figure 21.57 Example of a simultaneous start, stop, and clear by software with the same count cycle (GTPR register value)

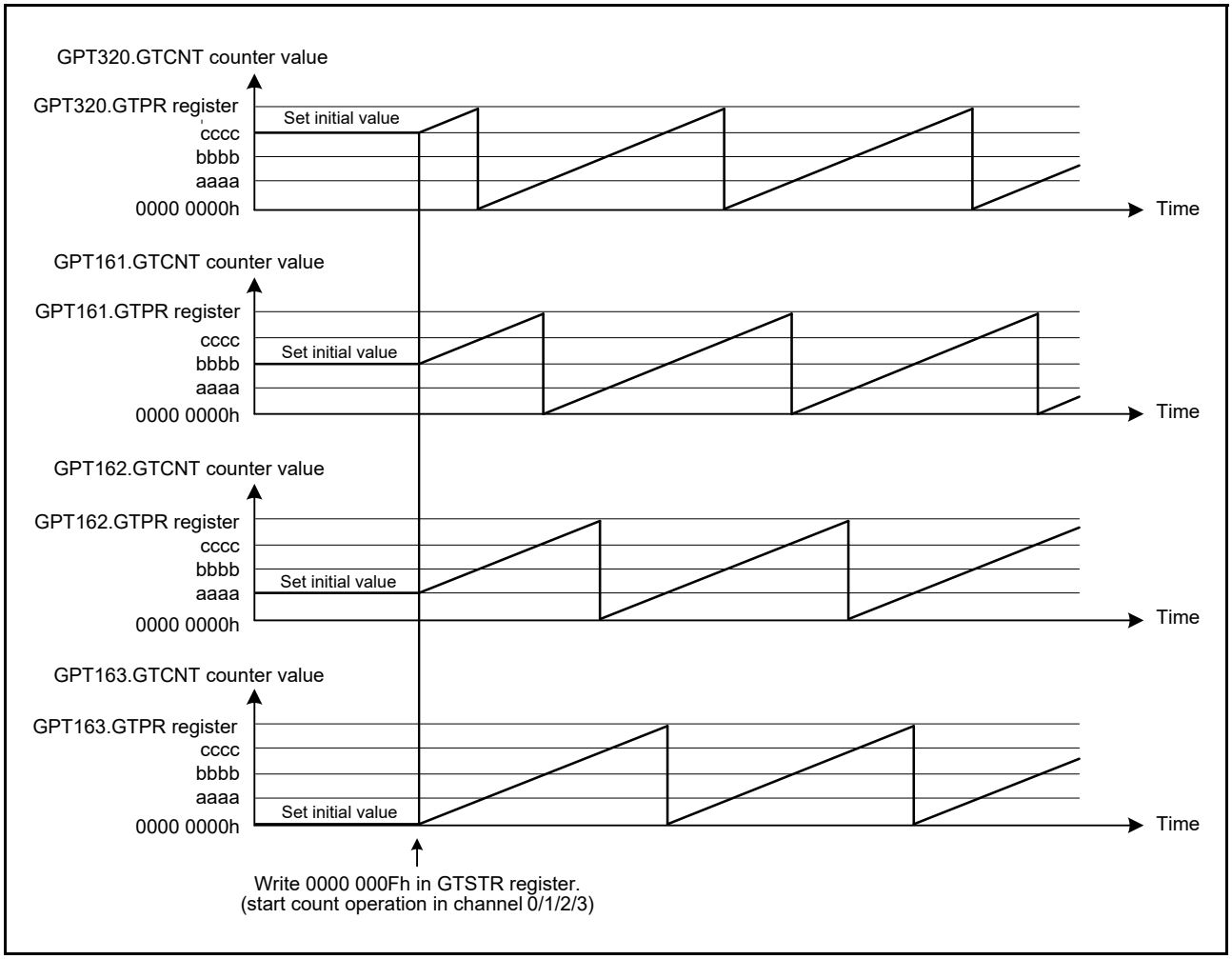


Figure 21.58 Example of software phase start with the same count cycle (GTPR register value)

21.3.8.2 Synchronized operation by hardware

The GTCNT counters can be started simultaneously by the following hardware sources:

- External trigger input
- ELC event input.

Figure 21.59 shows an example of a simultaneous start, stop, and clear operation by a hardware source. Figure 21.60 shows the setting example.

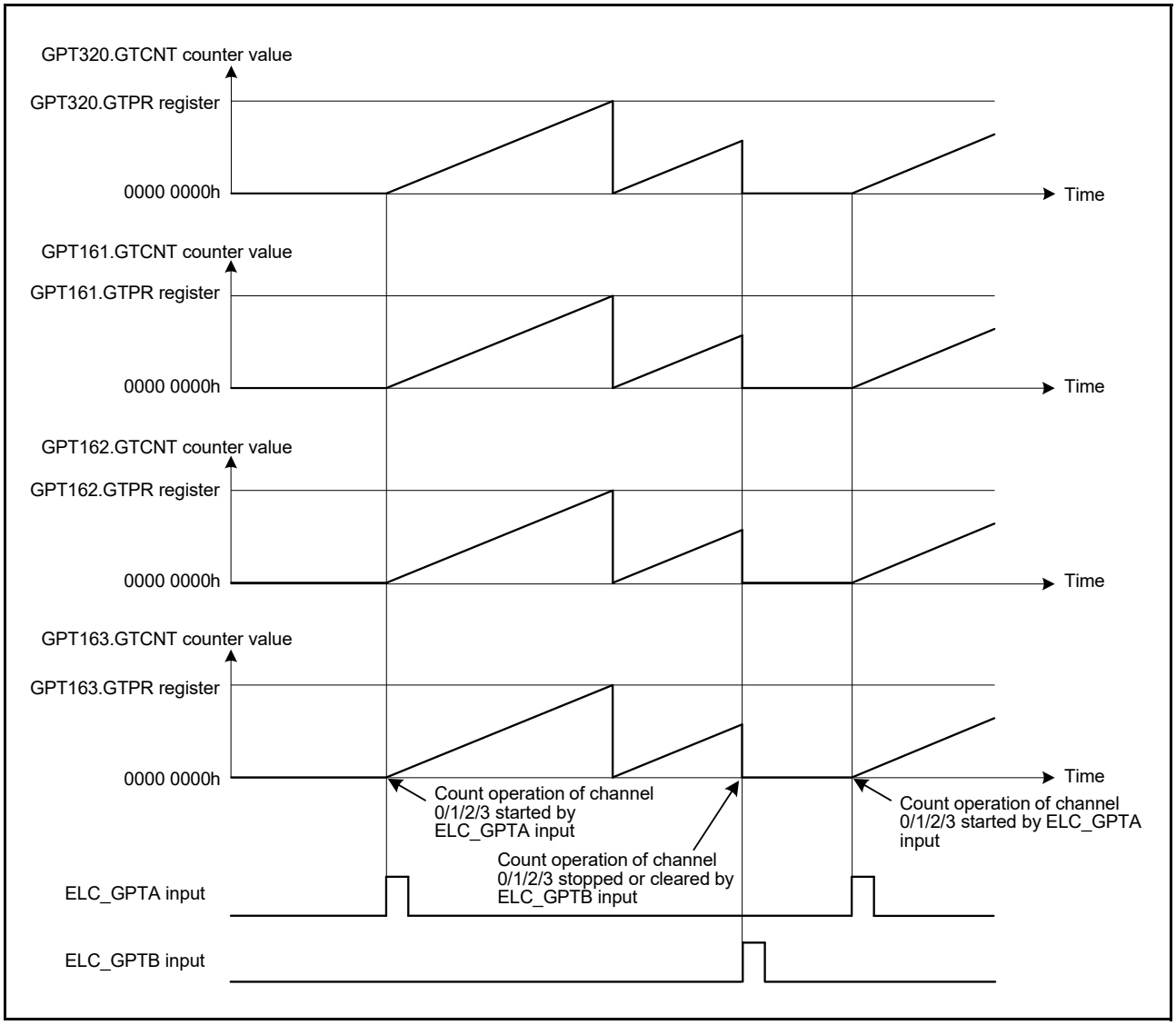


Figure 21.59 Example of a simultaneous start, stop, and clear by a hardware source with the same count cycle (GTPR register value)

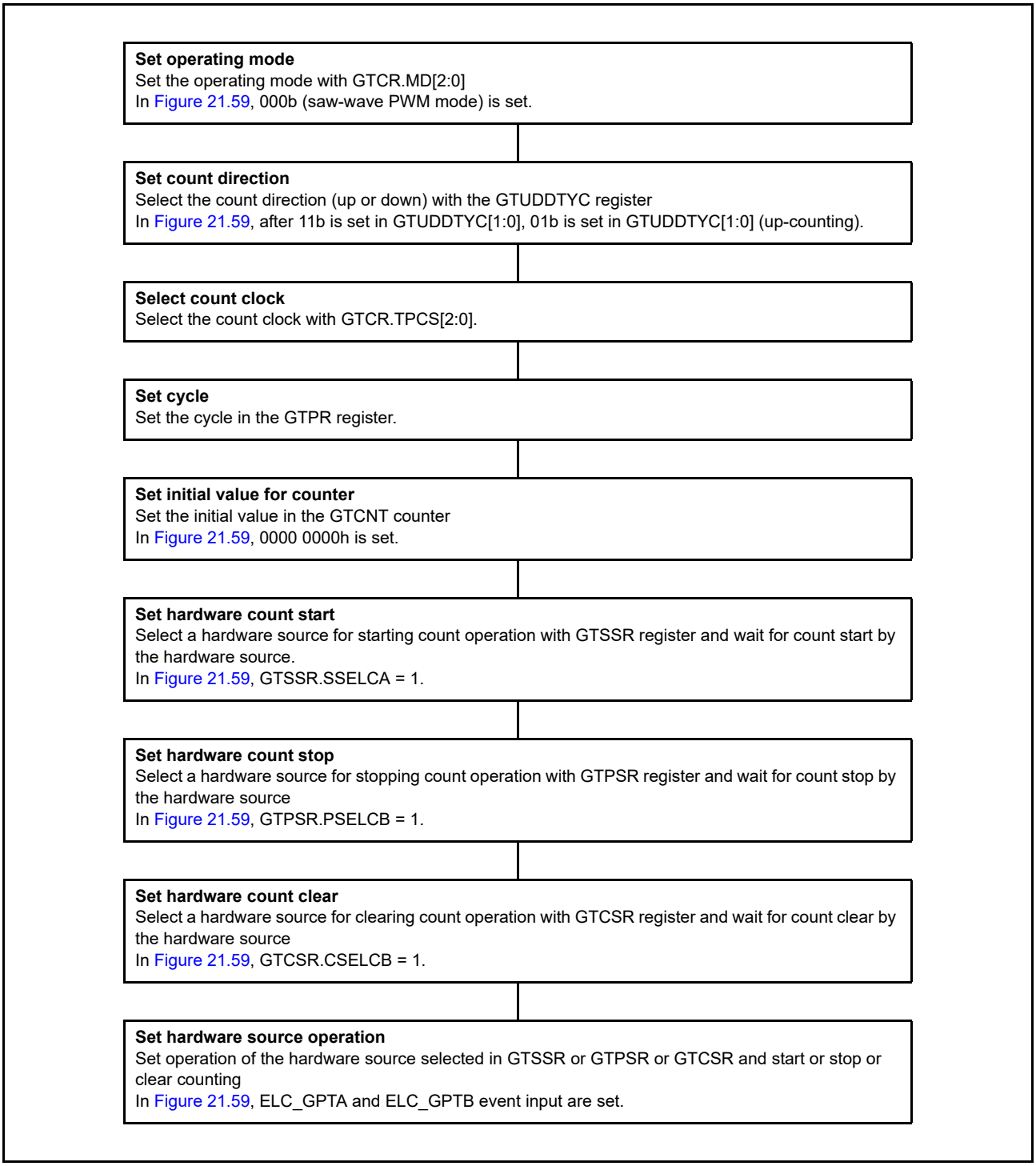


Figure 21.60 Example setting for simultaneous start by a hardware source

21.3.9 PWM Output Operation Examples

(1) Synchronized PWM output

The GPT outputs 14 phases of linked PWM waveforms for a maximum of seven channels by multiple GPTs.

Figure 21.61 shows an example in which four channels perform synchronized operation in saw-wave PWM mode and eight phases of PWM waveforms are output. The GTIOCA is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCB is set so that it outputs low as the initial value, high at a GTCCRB compare match, and low at the cycle end.

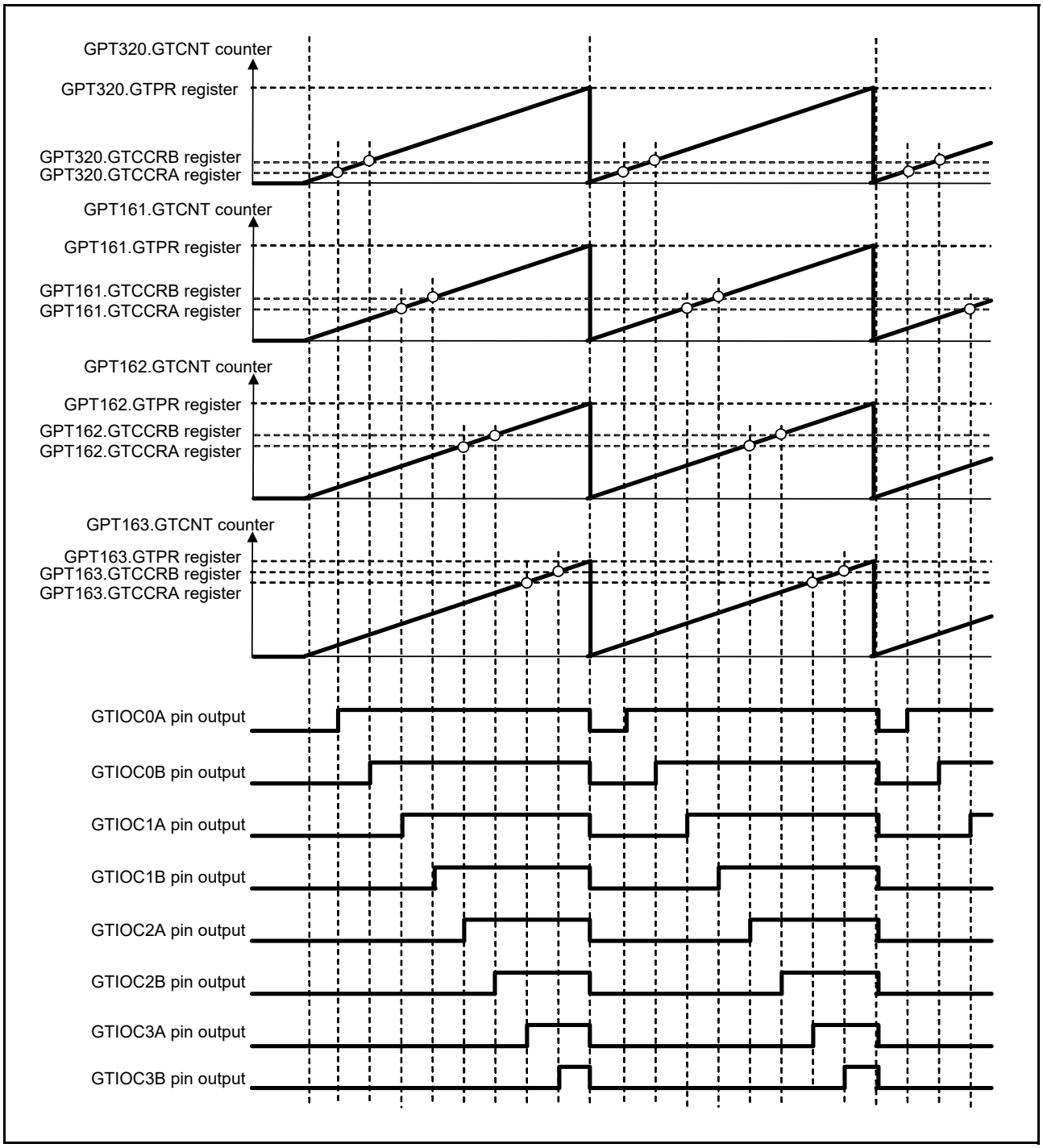


Figure 21.61 Example of synchronized PWM output

(2) 3-phase saw-wave complementary PWM output

Figure 21.62 shows an example in which three channels perform synchronized operation in saw-wave PWM mode and 3-phase complementary PWM waveforms are output. The GTIOCA pin is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCB pin is set so that it outputs high as the initial value, low at a GTCCRB compare match, and high at the cycle end.

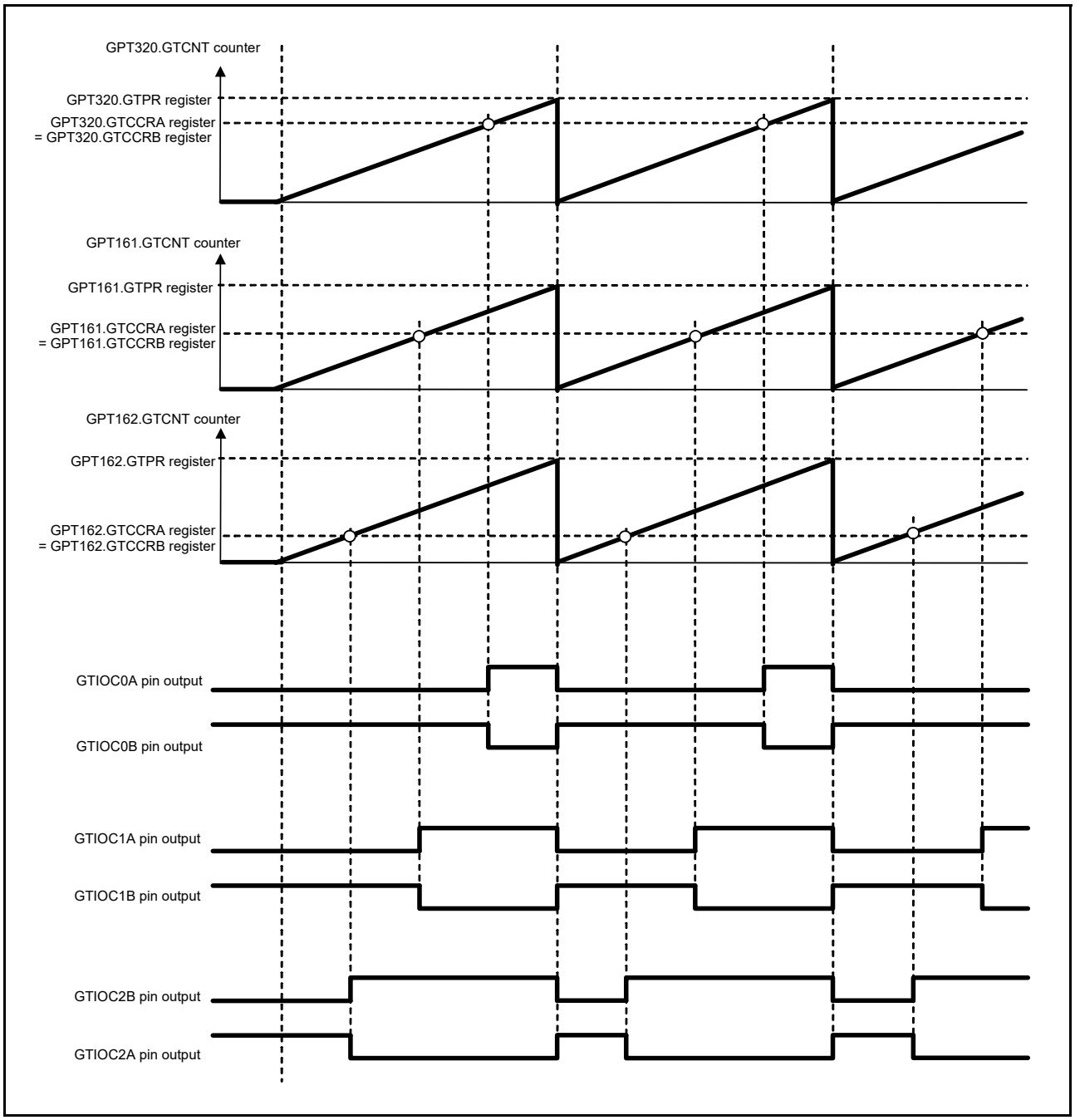


Figure 21.62 Example of 3-phase saw-wave complementary PWM output

(3) 3-phase saw-wave complementary PWM output with automatic dead time setting

Figure 21.63 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.

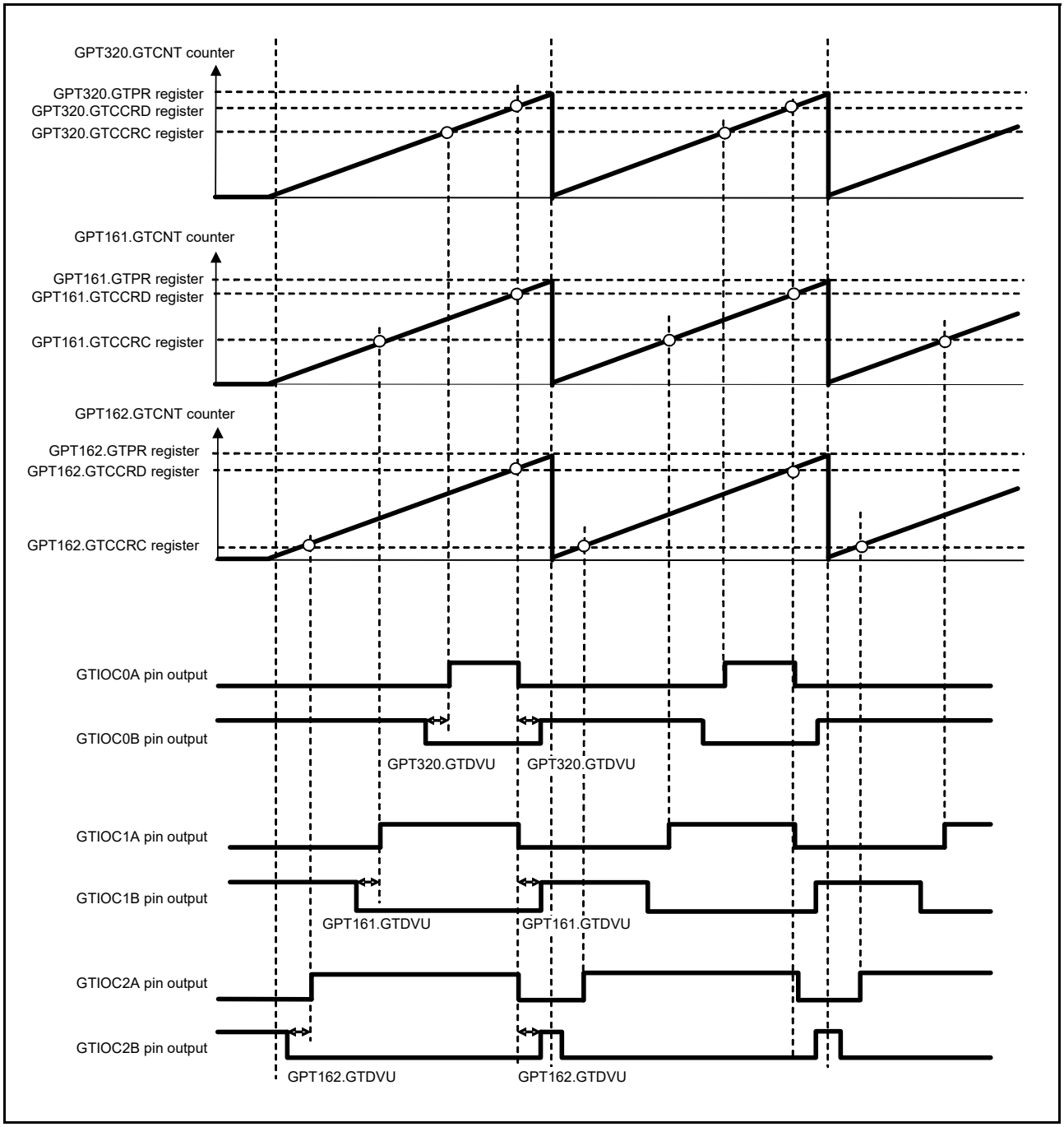


Figure 21.63 Example of 3-phase saw-wave complementary PWM output with automatic dead time setting

(4) 3-phase triangle-wave complementary PWM output

Figure 21.64 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 and 3-phase complementary PWM waveforms are output. The GTIOCA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

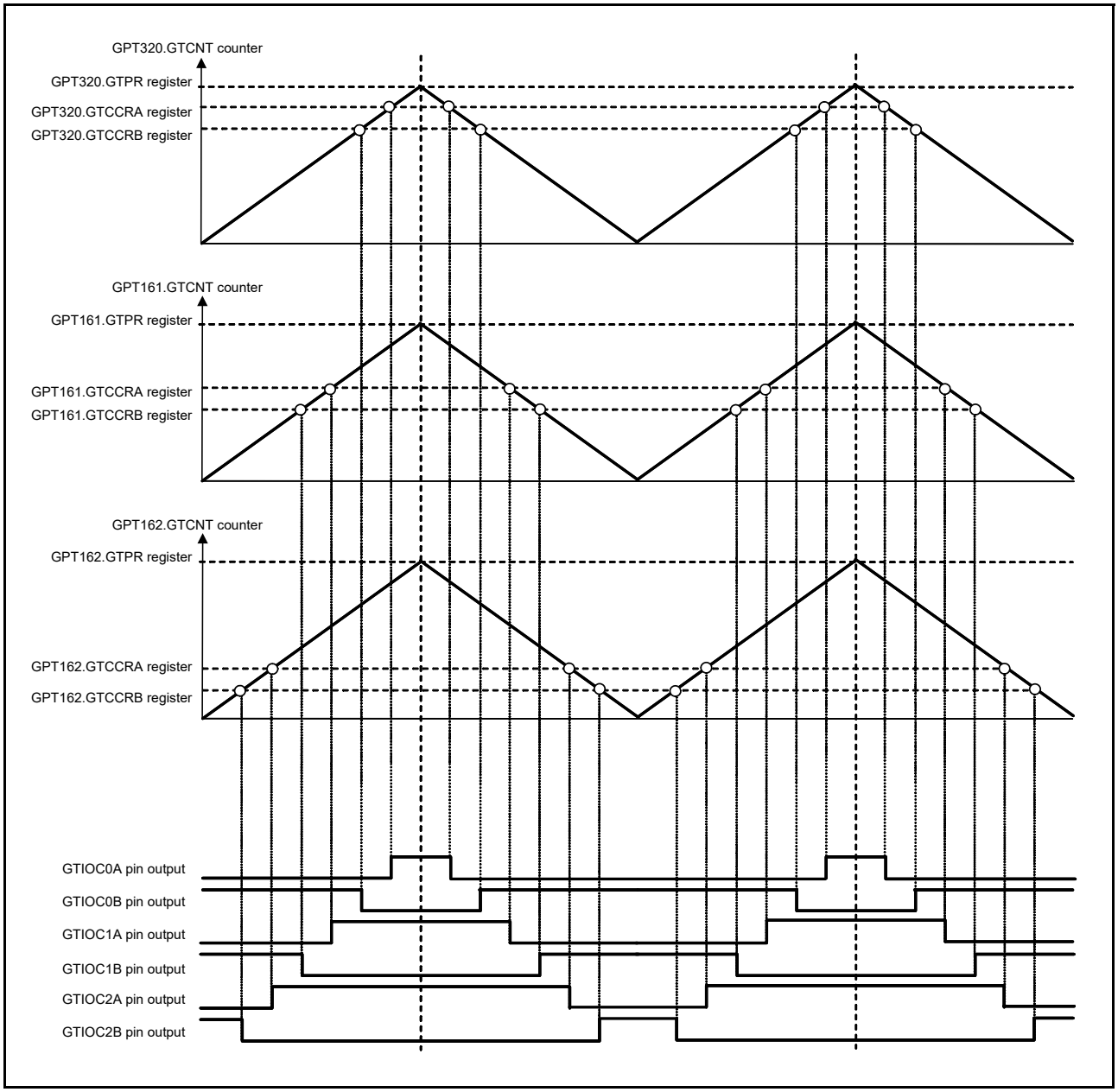


Figure 21.64 Example of 3-phase triangle-wave complementary PWM output

(5) 3-phase triangle-wave complementary PWM output with automatic dead time setting

Figure 21.65 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

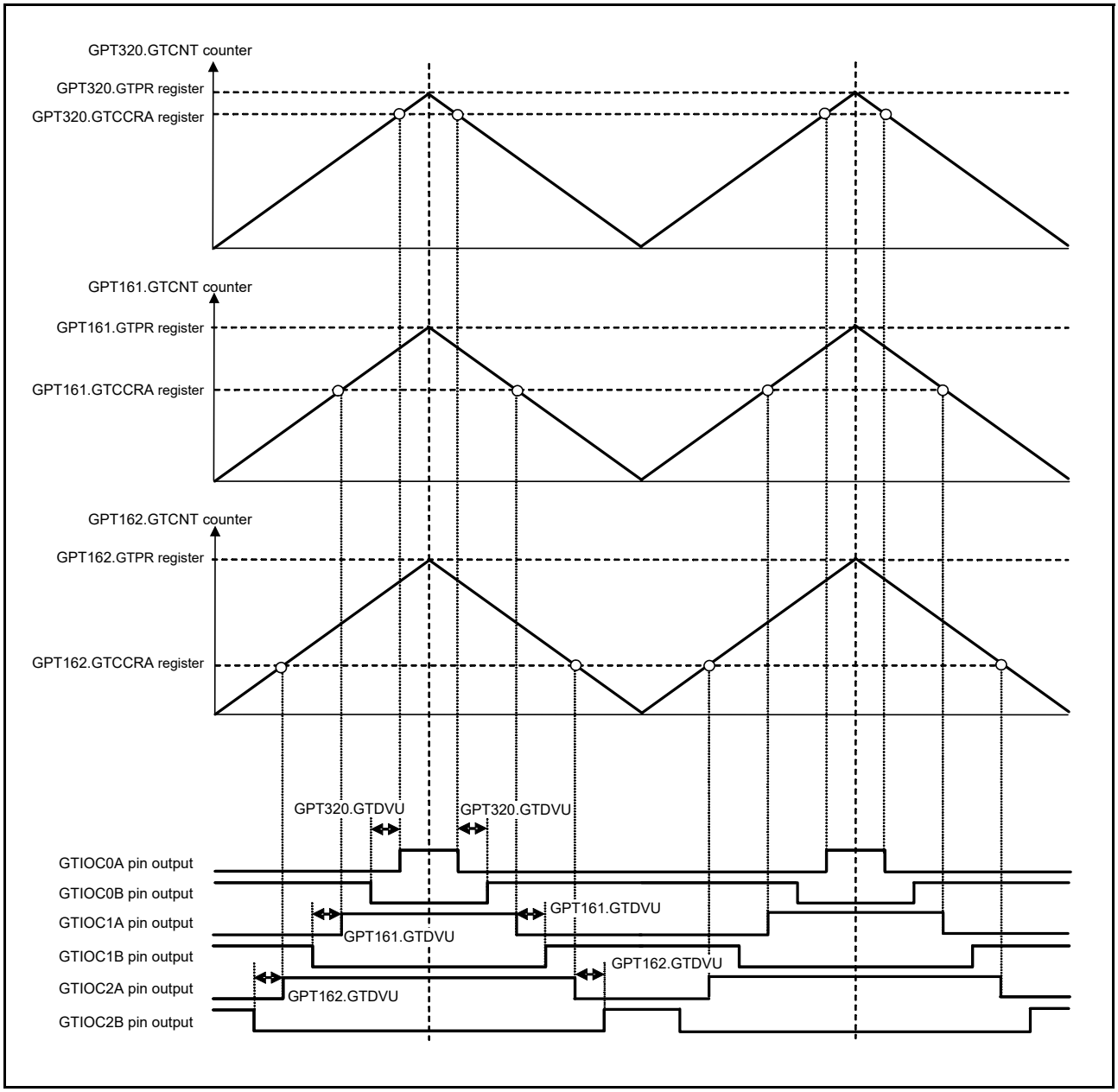


Figure 21.65 Example of 3-phase triangle-wave complementary PWM output with automatic dead time setting

(6) 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

Figure 21.66 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 3 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCA is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCB is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

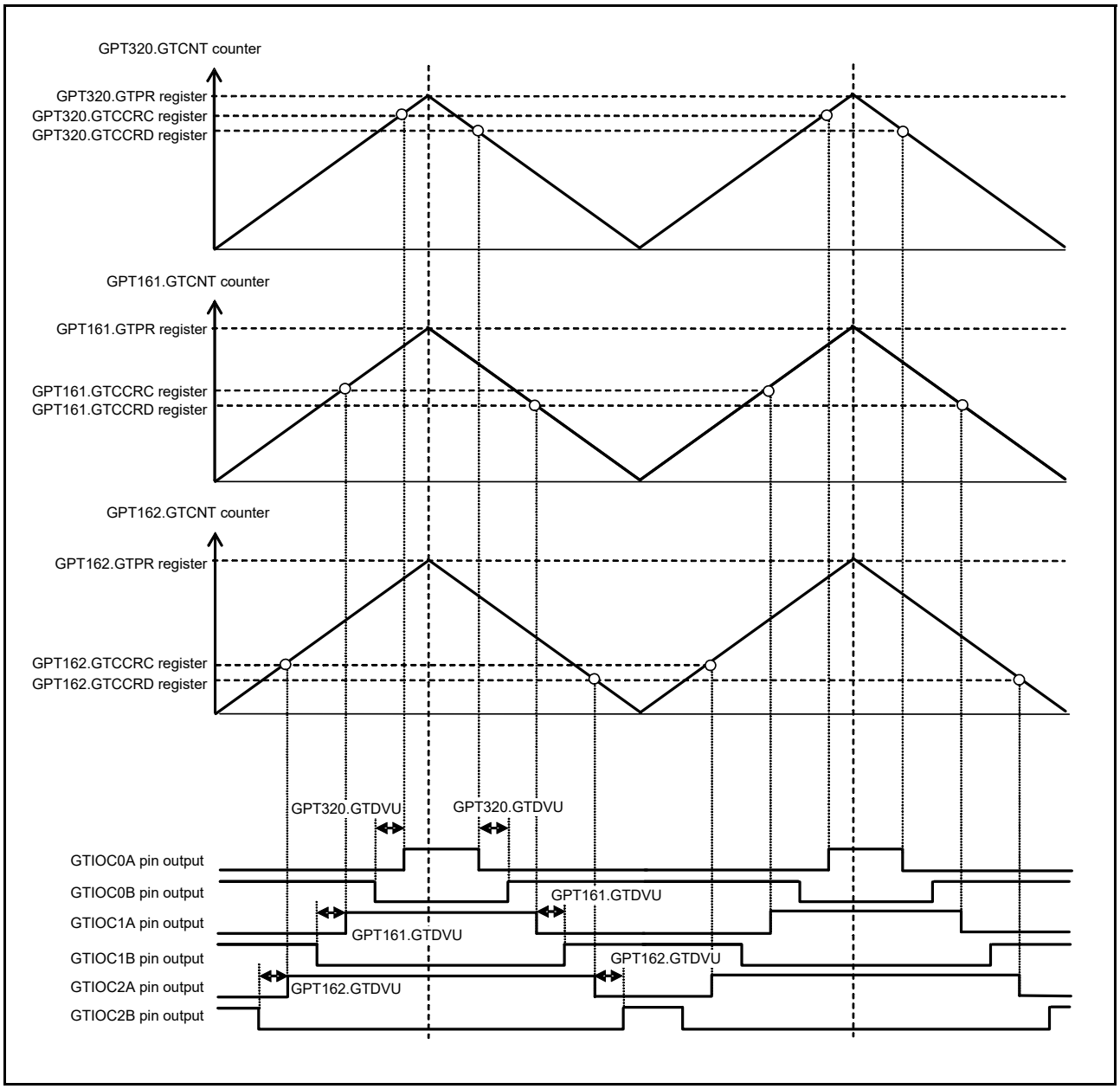


Figure 21.66 Example of 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

21.3.10 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 21.3.1.1, Counter operation.

Figure 21.67 to Figure 21.76 show phase counting modes 1 to 5. Table 21.7 to Table 21.16 show conditions of up-counting or down-counting and list settings for the GTUPSR and GTDNSR registers.

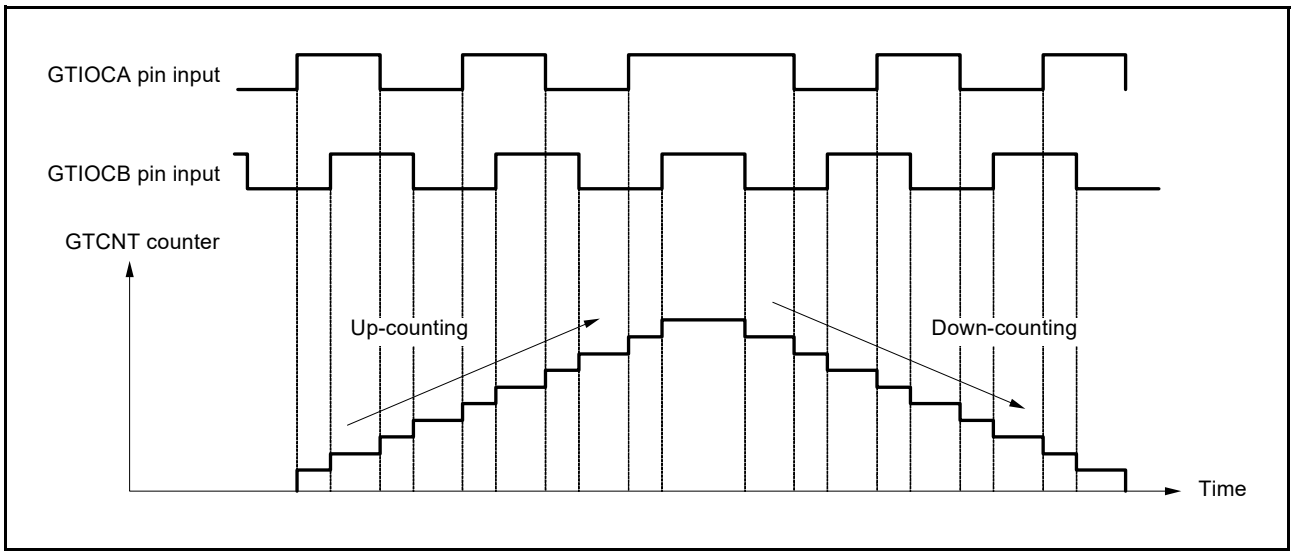


Figure 21.67 Example of phase counting mode 1

Table 21.7 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

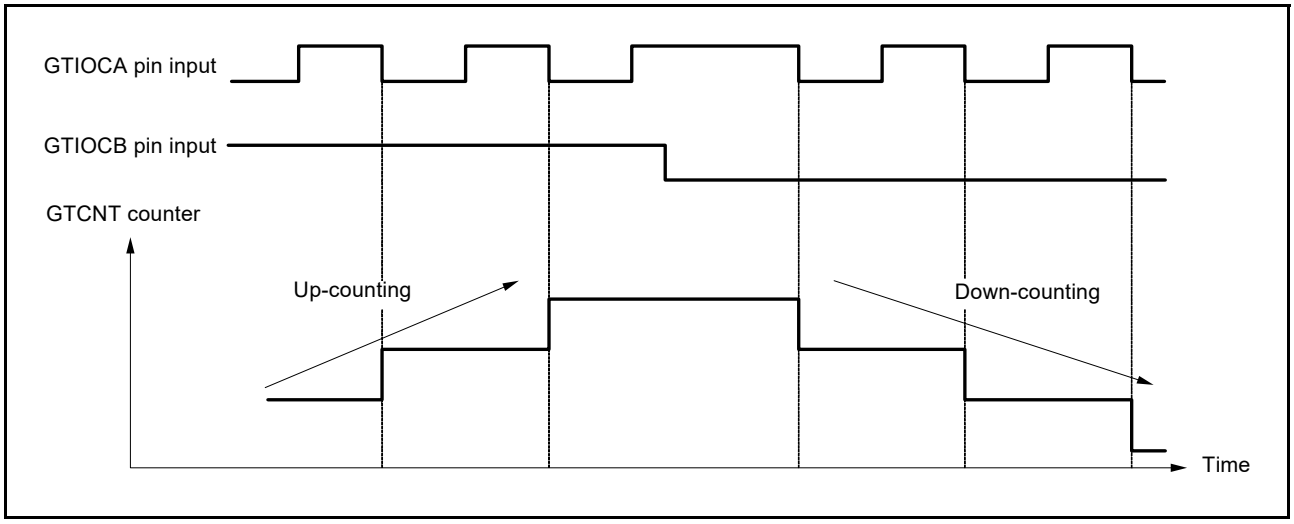


Figure 21.68 Example of phase counting mode 2 (A)

Table 21.8 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

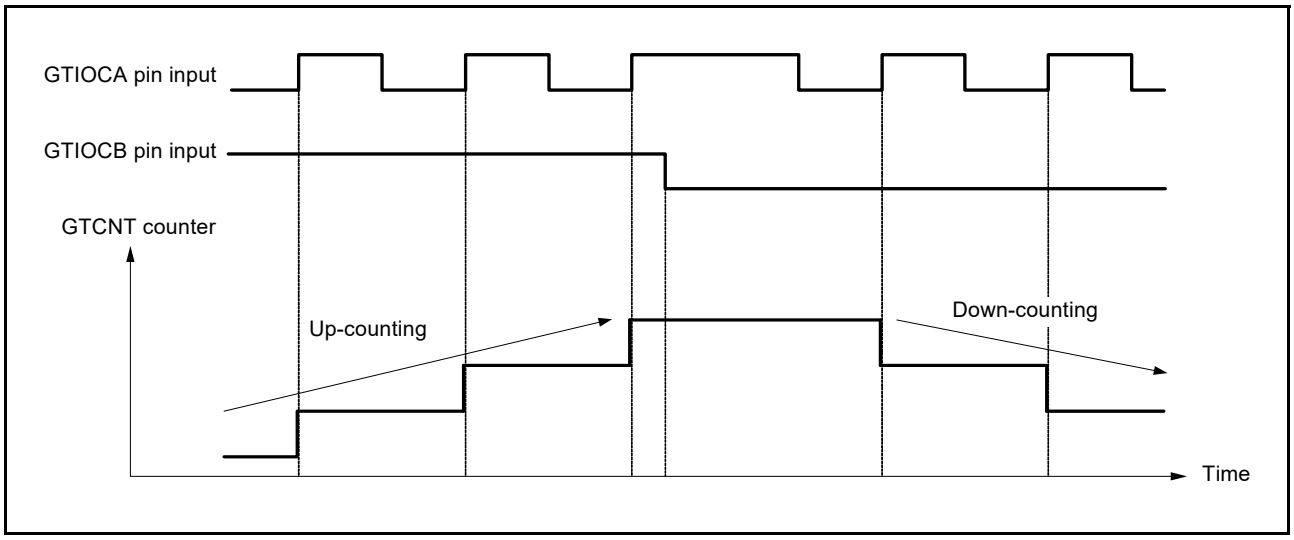


Figure 21.69 Example of phase counting mode 2 (B)

Table 21.9 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

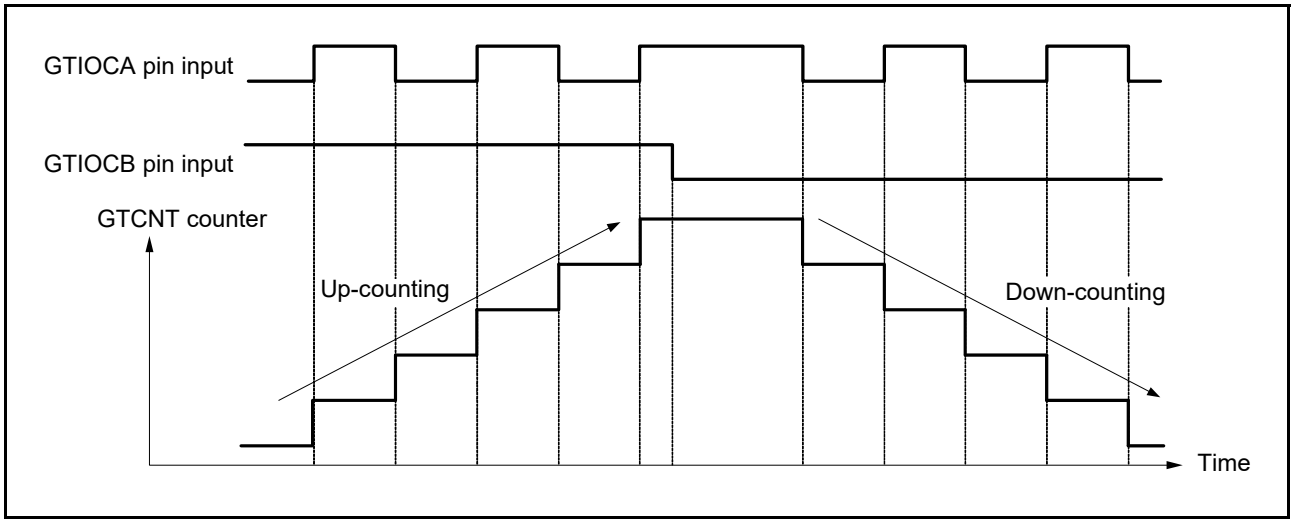


Figure 21.70 Example of phase counting mode 2 (C)

Table 21.10 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

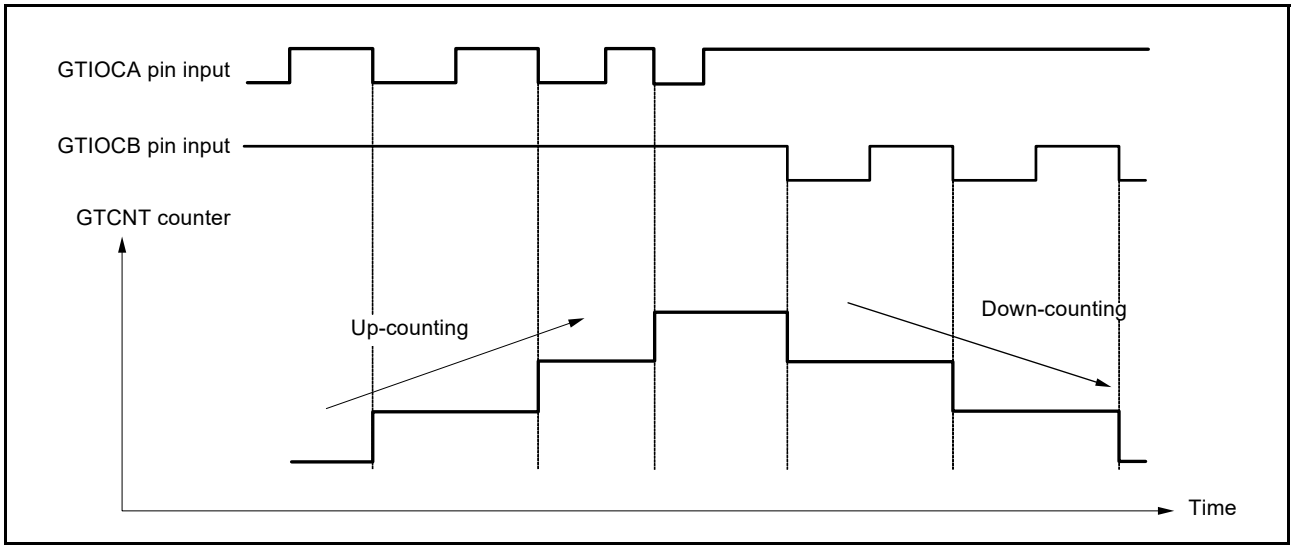


Figure 21.71 Example of phase counting mode 3 (A)

Table 21.11 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

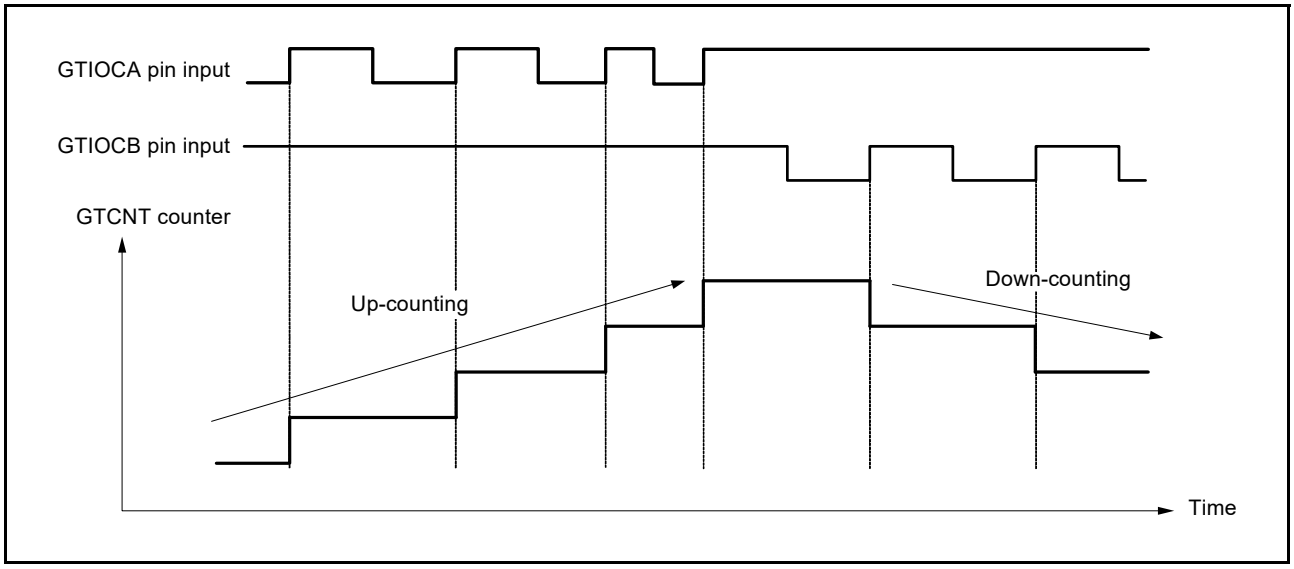


Figure 21.72 Example of phase counting mode 3 (B)

Table 21.12 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

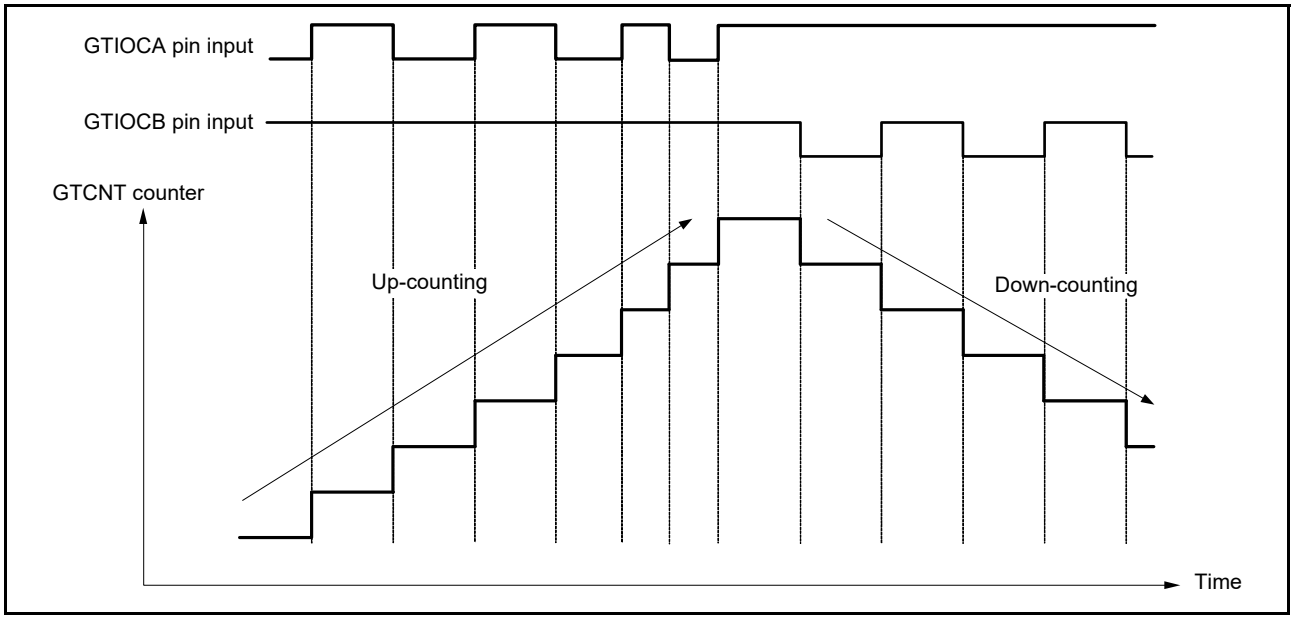


Figure 21.73 Example of phase counting mode 3 (C)

Table 21.13 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

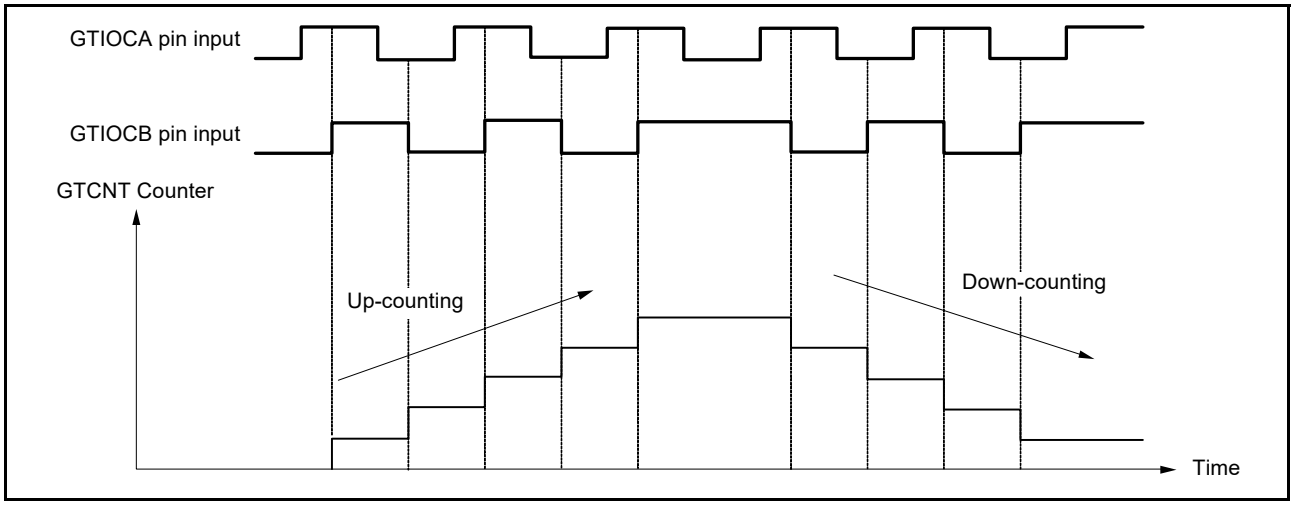


Figure 21.74 Example of phase counting mode 4

Table 21.14 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

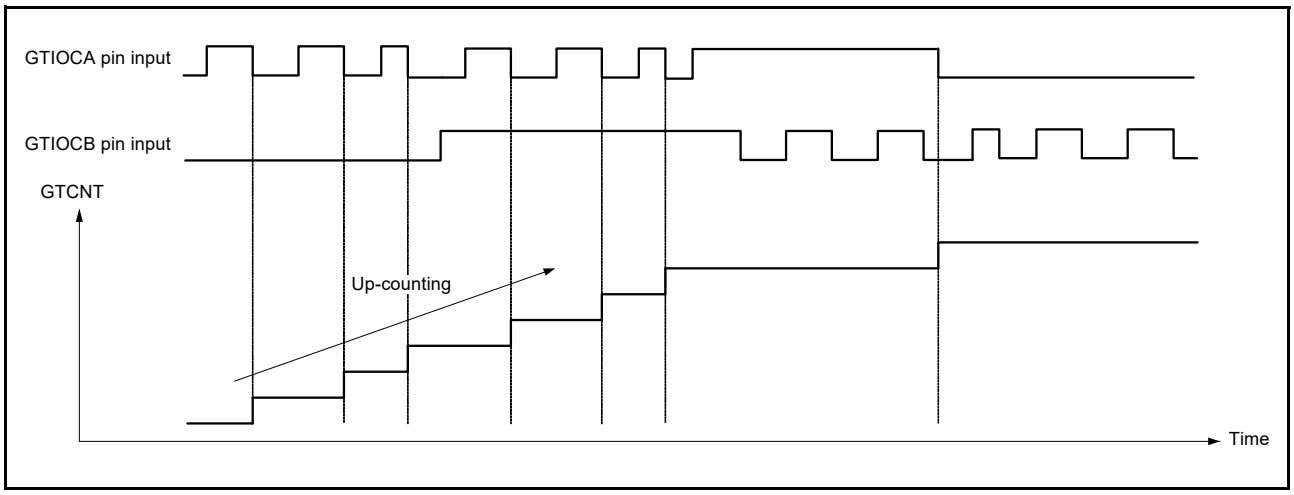


Figure 21.75 Example of phase counting mode 5 (A)

Table 21.15 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

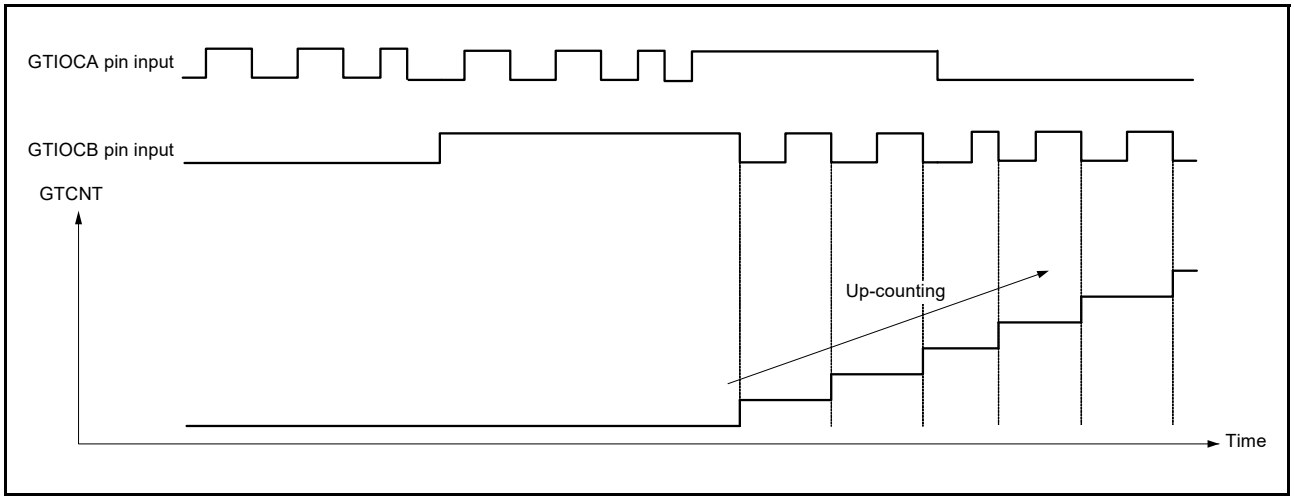


Figure 21.76 Example of phase counting mode 5 (B)

Table 21.16 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

21.3.11 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 GPT161.GTIOCA.

Figure 21.77 shows the conceptual diagram of GPT_OPS control flow.

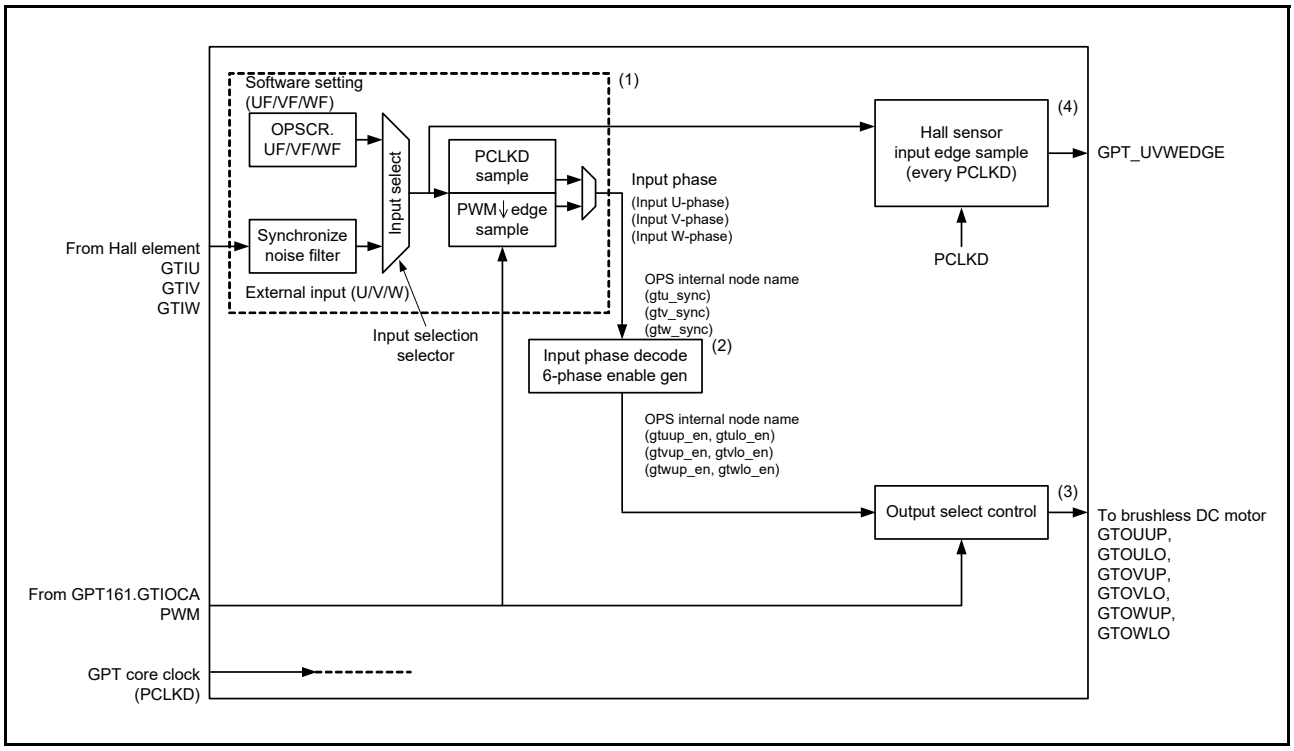


Figure 21.77 Conceptual diagram of GPT_OPS control flow

Figure 21.78 shows a 6-phase level signals output example of a GPT_OPS operation.

The GPT_UVWEDGE signal in Figure 21.78 is the Hall sensor input edge that outputs to the ELC.

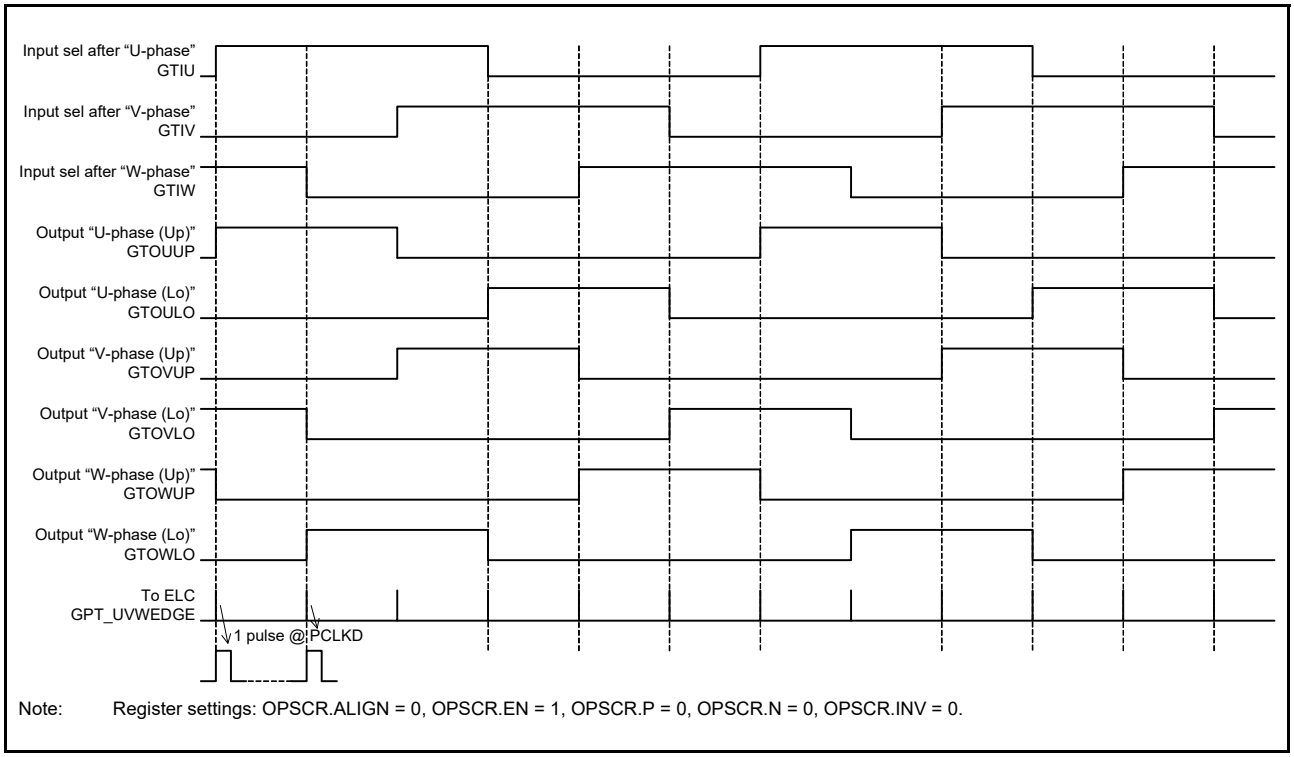


Figure 21.78 Example of 6-phase level output operation

Figure 21.79 shows a 6-phase PWM output example of a GPT_OPS operation with chopper control.

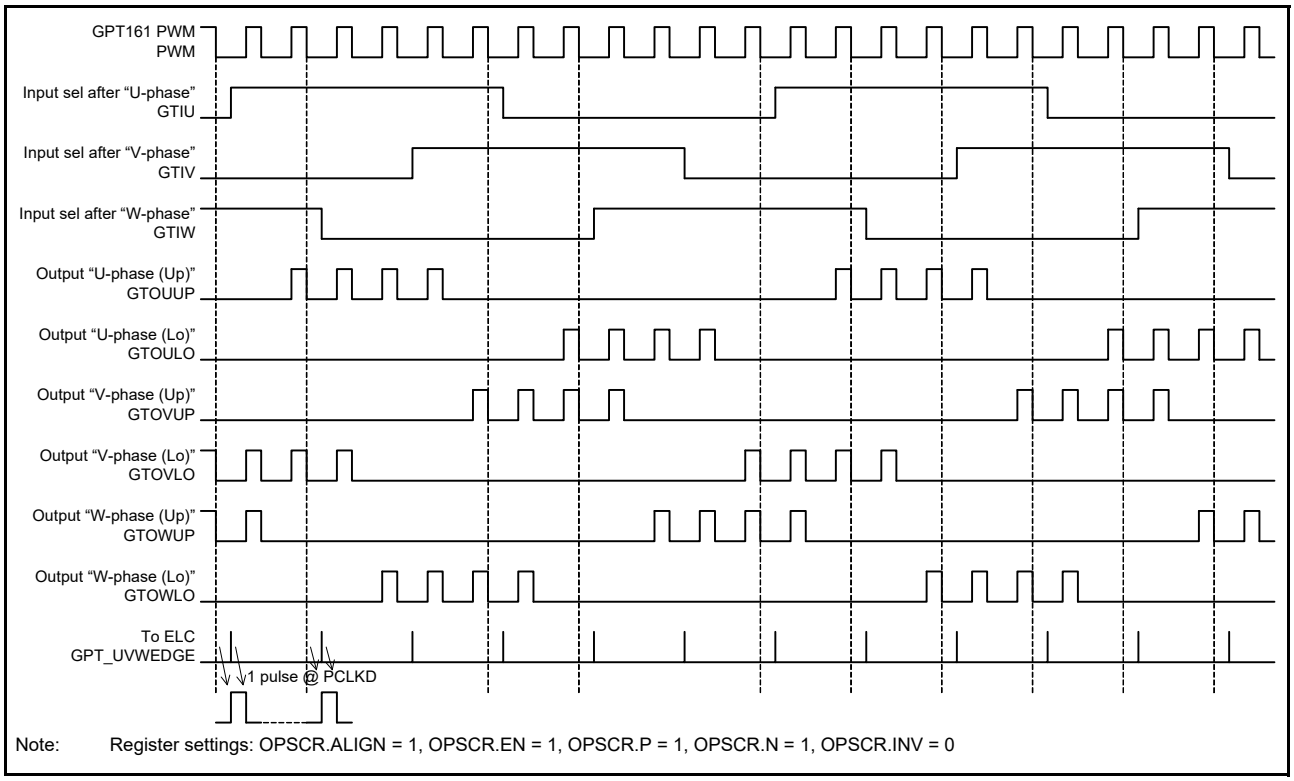


Figure 21.79 Example of 6-phase PWM output operation with chopper control

Figure 21.80 shows a 6-phase PWM output example of an output disable control operation.

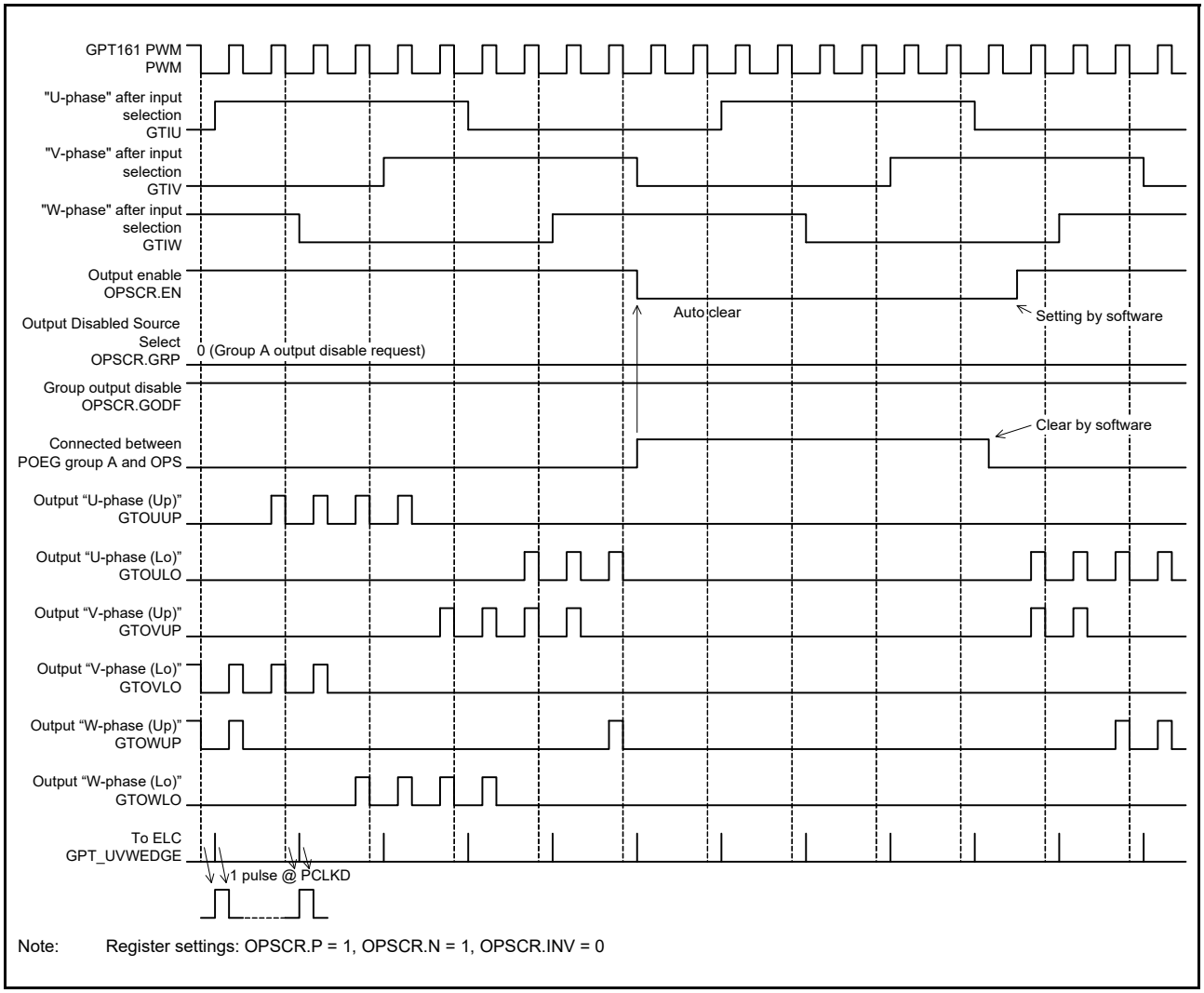


Figure 21.80 Example of group output disable control operation

21.3.11.1 Input selection and synchronization of external input signal

In the GPT_OPS control flow conceptual diagram shown in Figure 21.77, (1) is a selection of input phase from the 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 GPT161.GTIOCA) using falling edge sampling with OPSCR.ALIGN bit set to 1.

When OPSCR.FB bit is 1, select the software setting (OPSCR.UF, VF, WF) with the value of the input phase of PWM (PWM of GPT161.GTIOCA) using falling edge sampling with OPSCR.ALIGN bit set to 1.

When OPSCR.ALIGN bit is 0, GPT_OPS operates with the input phase of PCLKD synchronization with either OPSCR.FB bit set to 0 or OPSCR.FB bit set to 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 21.17 shows the input selection process and setting of the associated OPSCR bits.

Table 21.17 Input selection processing method

Register OPSCR			
FB bit	**ALIGN bit**	**Selection of input phase sampling method(U/V/W-phase)**	**Synchronization input/output selection process (GPT_OPS internal node name)**
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)	

21.3.11.2 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 software setting.

21.3.11.3 Input phase decode

In the GPT_OPS control flow conceptual diagram shown in Figure 21.77, (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 21.18 shows the decode table of input phase.

Table 21.18 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

21.3.11.4 Output selection control

In the GPT_OPS control flow conceptual diagram in Figure 21.77, (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 21.19 and Table 21.20 show the output selection control method using the OPSCR register bit.

Table 21.19 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	**OPSCR.P**	**OPSCR.INV**	**GTOUUP GTOVUP GTOWUP**	**Mode**
0	x	x	0	Output Stop (External pin: Hi-Z) GPT_OPS → 0 output
1	0	0	Level signal (gtuup_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 21.20 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	**OPSCR.N**	**OPSCR.INV**	**GTOULO GTOVLO GTOWLO**	**Mode**
0	x	x	0	Output Stop (External pin: Hi-Z) GPT_OPS → 0 output
1	0	0	Level signal (gtulo_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)

21.3.11.5 Output selection control (group output disable function)

When OPSCR.GODF is 1 and the signal value selected by the OPSCR.GRP bit is high (output disable request), the GPT_OPS output pins change 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 bit 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 21.80.

21.3.11.6 Event Link Controller (ELC) output

In the GPT_OPS control flow conceptual diagram shown in Figure 21.77, (4) 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 software setting (OPSCR.UF, VF, WF) sampled at PCLKD.

See Figure 21.78 to Figure 21.80 for examples of the output signal to the ELC.

21.3.11.7 GPT_OPS start operation setting flow

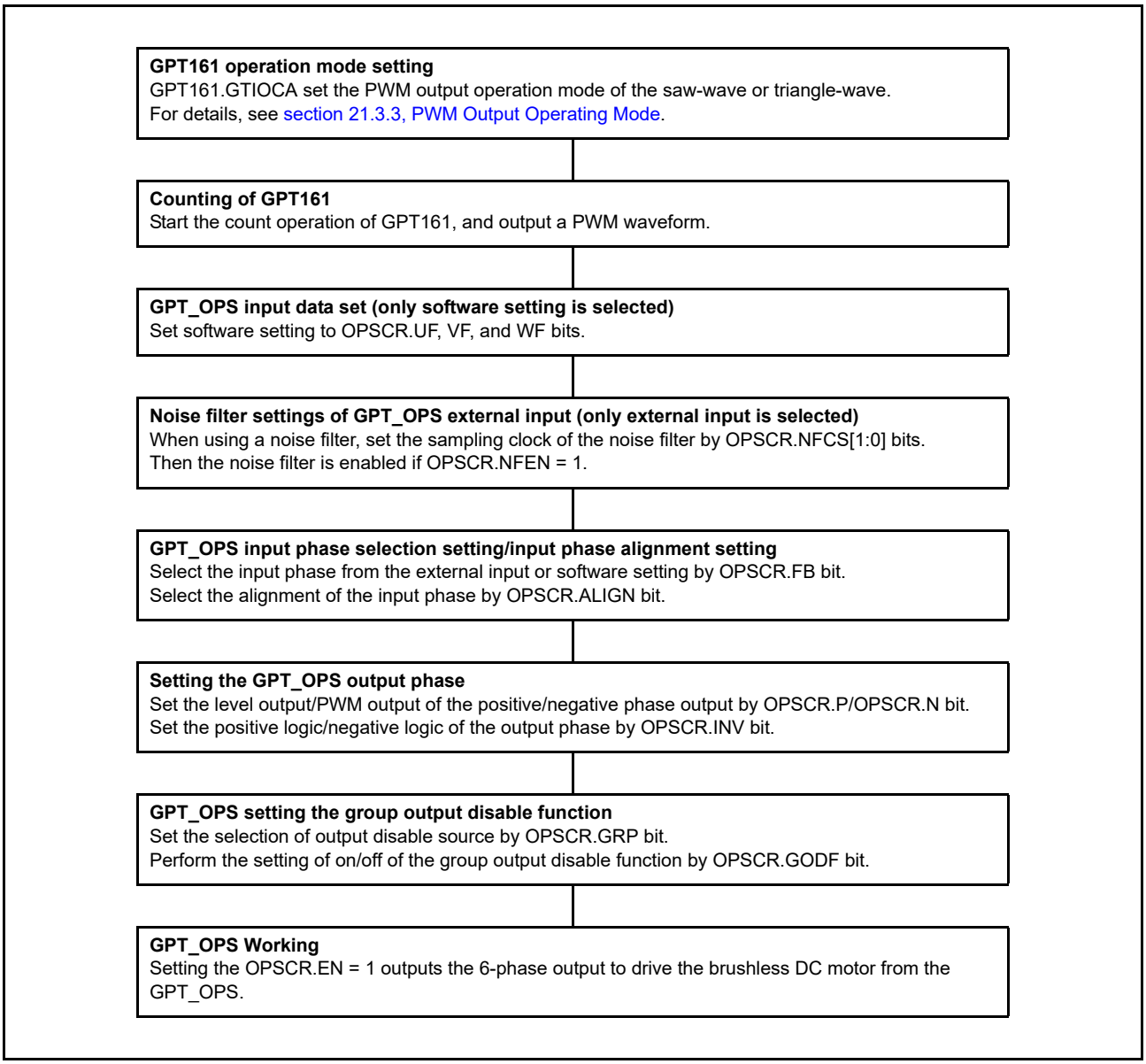


Figure 21.81 Example setting of GPT_OPS start operation

21.4 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. Those flags are automatically updated by the internal state.

Table 21.21 lists the GPT interrupt sources.

Table 21.21 Interrupt sources

Channel	Name	Interrupt source	Interrupt flag	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_OVF	GPT320.GTCNT overflow (GPT320.GTPR compare match)	TCFPO	Possible
	GPT0_UDF	GPT320.GTCNT underflow	TCFPU	Possible
1	GPT1_CCMPA	GPT161.GTCCRA input capture/compare match	TCFA	Possible
	GPT1_CCMPB	GPT161.GTCCRB input capture/compare match	TCFB	Possible
	GPT1_CMPC	GPT161.GTCCRC compare match	TCFC	Possible
	GPT1_CMPD	GPT161.GTCCRD compare match	TCFD	Possible
	GPT1_OVF	GPT161.GTCNT overflow (GPT161.GTPR compare match)	TCFPO	Possible
	GPT1_UDF	GPT161.GTCNT underflow	TCFPU	Possible
2	GPT2_CCMPA	GPT162.GTCCRA input capture/compare match	TCFA	Possible
	GPT2_CCMPB	GPT162.GTCCRB input capture/compare match	TCFB	Possible
	GPT2_CMPC	GPT162.GTCCRC compare match	TCFC	Possible
	GPT2_CMPD	GPT162.GTCCRD compare match	TCFD	Possible
	GPT2_OVF	GPT162.GTCNT overflow (GPT162.GTPR compare match)	TCFPO	Possible
	GPT2_UDF	GPT162.GTCNT underflow	TCFPU	Possible
3	GPT3_CCMPA	GPT163.GTCCRA input capture/compare match	TCFA	Possible
	GPT3_CCMPB	GPT163.GTCCRB input capture/compare match	TCFB	Possible
	GPT3_CMPC	GPT163.GTCCRC compare match	TCFC	Possible
	GPT3_CMPD	GPT163.GTCCRD compare match	TCFD	Possible
	GPT3_OVF	GPT163.GTCNT overflow (GPT163.GTPR compare match)	TCFPO	Possible
	GPT3_UDF	GPT163.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_OVF	GPT164.GTCNT overflow (GPT164.GTPR compare match)	TCFPO	Possible
	GPT4_UDF	GPT164.GTCNT underflow	TCFPU	Possible
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_OVF	GPT165.GTCNT overflow (GPT165.GTPR compare match)	TCFPO	Possible
	GPT5_UDF	GPT165.GTCNT underflow	TCFPU	Possible
6	GPT6_CCMPA	GPT166.GTCCRA input capture/compare match	TCFA	Possible
	GPT6_CCMPB	GPT166.GTCCRB input capture/compare match	TCFB	Possible
	GPT6_CMPC	GPT166.GTCCRC compare match	TCFC	Possible
	GPT6_CMPD	GPT166.GTCCRD compare match	TCFD	Possible
	GPT6_OVF	GPT166.GTCNT overflow (GPT166.GTPR compare match)	TCFPO	Possible
	GPT6_UDF	GPT166.GTCNT underflow	TCFPU	Possible

(1) GPTn_CCMPA interrupt (n = 0 to 6)

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 6)

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 6)

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 therefore, an interrupt is not requested in 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 6)

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 therefore, an interrupt is not requested in 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_OVF interrupt (n = 0 to 6)

An interrupt request is generated in 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 (the GTCNT changes from GTPR to GTPR-1)
- In counting by hardware sources, an overflow (GTCNT changes from GTPR to 0 in up count) has occurred.

(6) GPTn_UDF interrupt (n = 0 to 6)

An interrupt request is generated in 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 (the GTCNT changes from 0 to 1)
- In counting by hardware sources, underflow (GTCNT changes from 0 to GTPR in down count) has occurred.

Table 21.22 Interrupt signals and interrupt status flags

Interrupt signal	Interrupt status flag
GPTn_UDF	GTST[7] (TCFPU)
GPTn_OVF	GTST[6] (TCFPO)
GPTn_CMPD	GTST[3] (TCFD)
GPTn_CMPC	GTST[2] (TCFC)
GPTn_CCMPB	GTST[1] (TCFB)
GPTn_CCMPA	GTST[0] (TCFA)

n = 0 to 6

21.4.1 DTC Activation

The DTC can be activated by the interrupt in each channel. For details, see section 13, Interrupt Controller Unit (ICU), and section 16, Data Transfer Controller (DTC).

21.5 Operations Linked by the ELC

21.5.1 Event Signal Output to the 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 overflow interrupt (GPTn_OVF)
- Generation of underflow interrupt (GPTn_UDF).

Note: n = 0 to 6

21.5.2 Event Signal Inputs from the ELC

The GPT can perform the following operations in response to a maximum of four events from the ELC:

- Start counting, stop counting, clear counting
- Up-counting, down-counting
- Input capture.

See section 21.3, Operation for detail on hardware resources.

21.6 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 21.82 shows the timing of noise filtering.

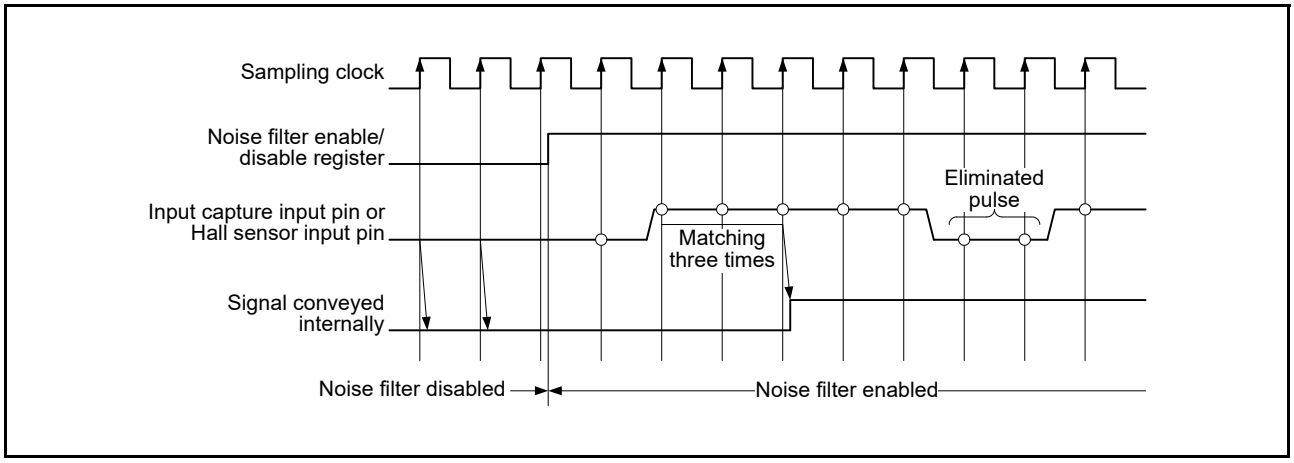


Figure 21.82 Timing of noise filtering

If noise filtering is enabled, the input capture operation or hall sensor input operation performs 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 hall sensor input operation.

21.7 Protection Function

21.7.1 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.

21.7.2 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[1] and BD[0] bit settings. Specifically, 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[n] bits to 1 (buffer operation disabled) before a buffer register write and clearing the bit to 0 (buffer operation enabled) after completion of writing to all the buffer registers.

Figure 21.83 shows an example of operation for disabling buffer operation.

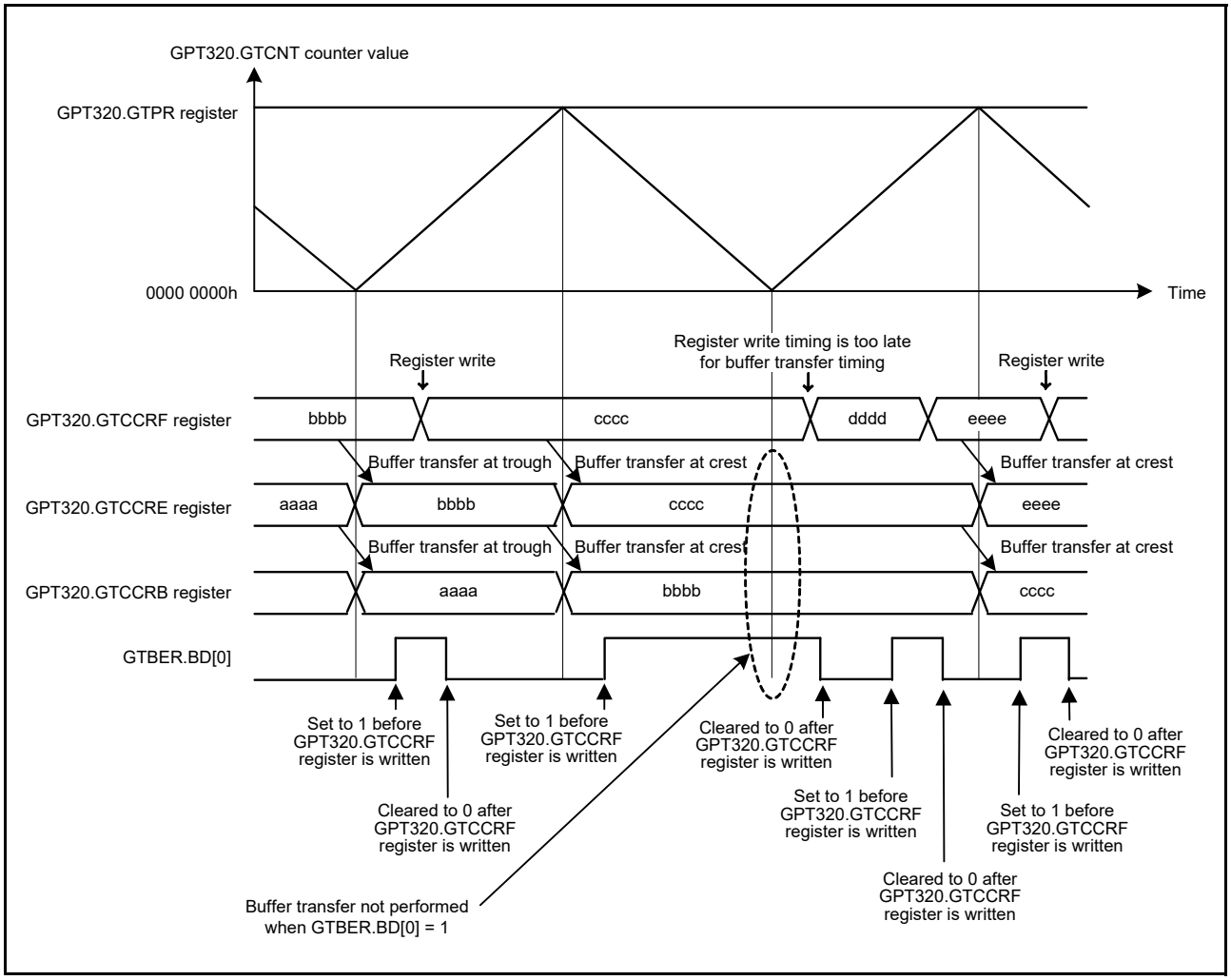


Figure 21.83 Example of operation for disabling buffer operation with triangle waves, double buffer operation, and buffer transfer at both troughs and crests

21.7.3 GTIOC Pin Output Negate Control

For protection from system failure, the output disable control that changes the GTIOC pin output value forcibly is provided for GTIOC pin output by the request of output disable from POEG. When the GTIOCA pin output value is the same as the GTIOCB pin output value, output protection is required. GPT detects such a case and generates output disable requests to POEG according to the settings in the output disable request permission bits, such as GTINTAD.GRPABH, GTINTAD.GRPABL. After 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 two 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 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 end of cycle, GTIOR.OADF[1:0] must be set to 00b (for GTIOCA pin) or GTIOR.OBDF[1:0] must be set to 00b (for the GTIOCB pin).

Figure 21.84 shows an example of the GTIOC pin output disable control operation.

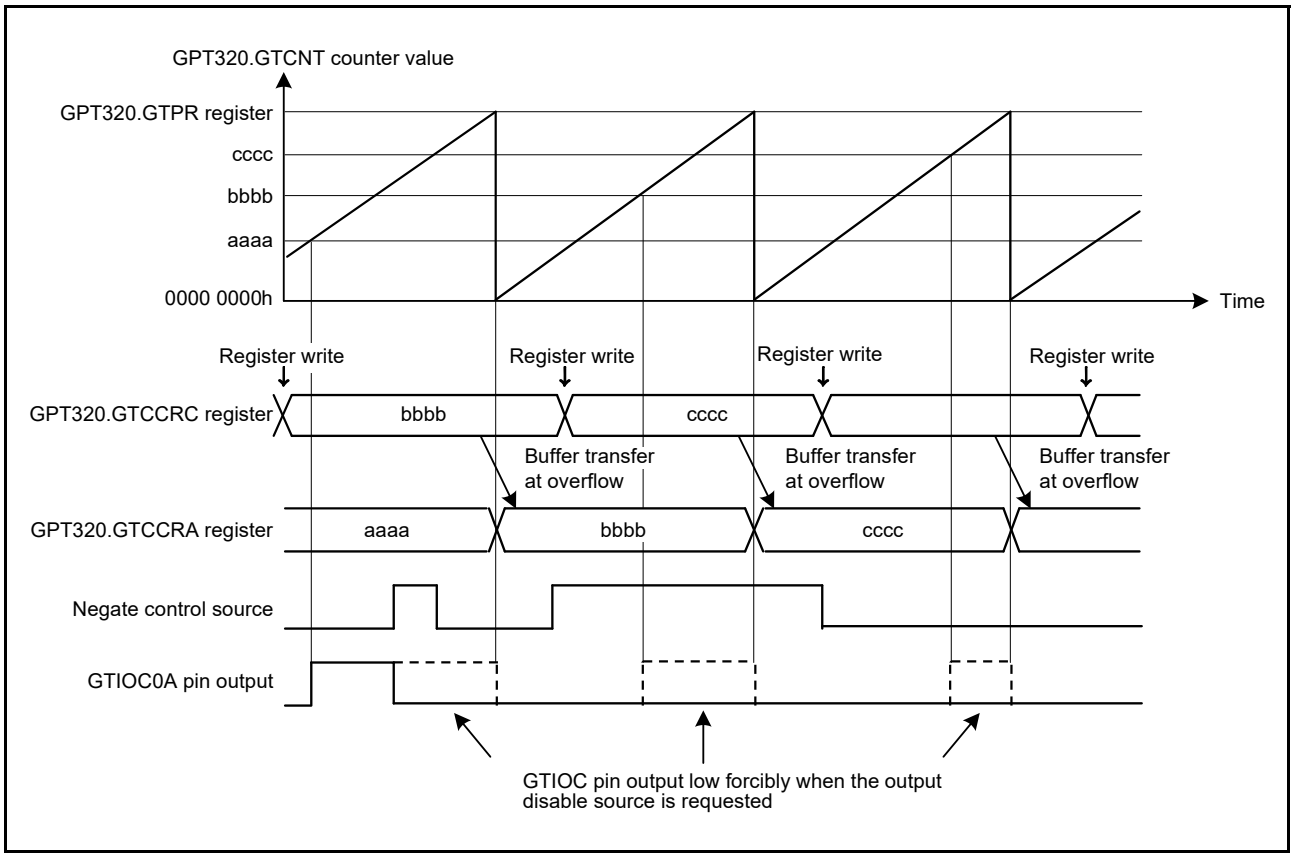


Figure 21.84 Example of GTIOC pin output disable control operation in saw-wave up-counting, buffer operation, active level 1, high output at GTCCRA compare match, low output at cycle end, and low output at output disable

21.8 Initialization Method of Output Pins

21.8.1 Pin Settings after Reset

The GPT registers are initialized at a reset. Start counting after selecting the port pin function with the PmnPFS register, setting GTIOR.OAE and GTIOR.OBE bits, and outputting the GPT function to external pins.

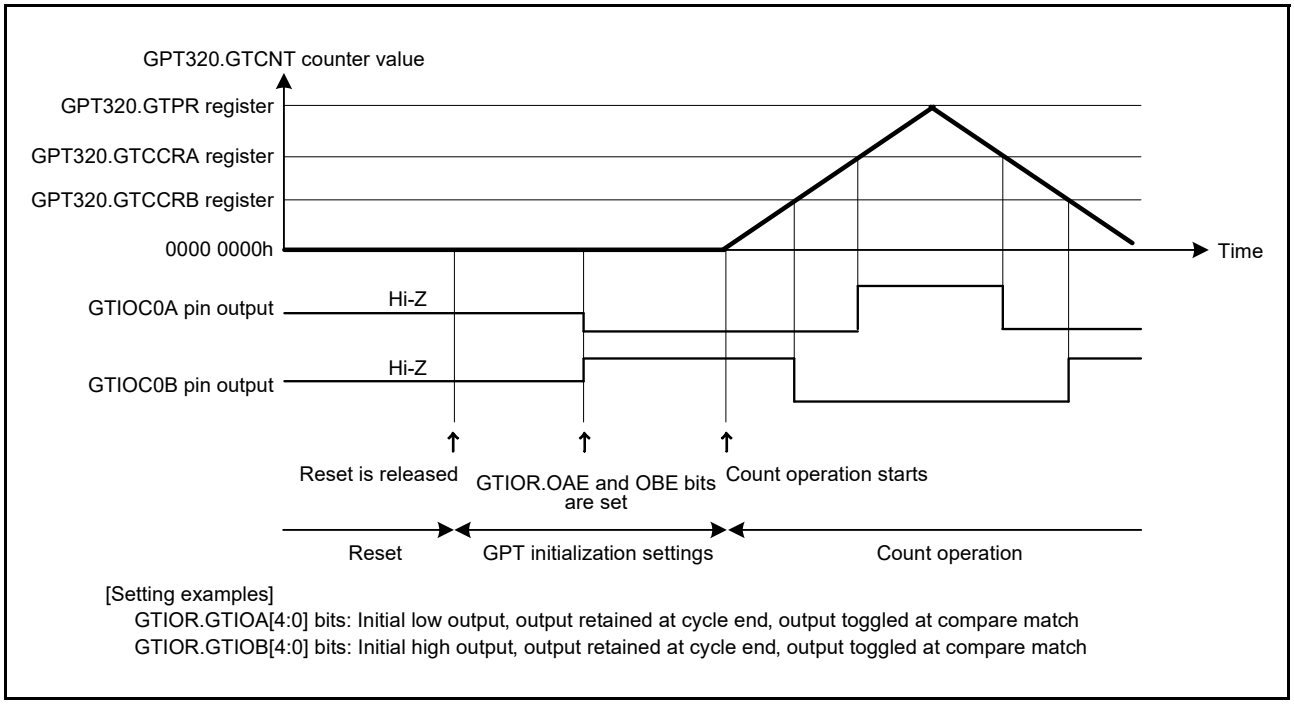


Figure 21.85 Example of pin settings after reset

21.8.2 Pin Initialization Caused by 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 of OADFLT and OBDFLT in GTIOR, and output the arbitrary values on count stop
- Set the pin to output an arbitrary value as a general output port by setting the PDR, PODR registers and PmnPFS.PMR bit of the I/O port in advance. Set the OAE and OBE bits in GTIOR to 0, and the control bit associated with the pin in 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 is resumed, operation continues from where it stopped. If counting is stopped, the registers must be initialized before counting starts.

21.9 Usage Notes

21.9.1 Module-Stop Function Setting

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

21.9.2 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 both of the following conditions:

- GTDVU < GTCCRA
- 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 must be set within the range of 0 < GTCCRB < GTPR. If GTCCRB = 0 or GTCCRB = GTPR is set, a compare match occurs within the cycle only when GTCCRB is 0 or GTCCRB = GTPR is satisfied. When GTCCRB > GTPR, no compare match occurs.

(3) When automatic dead time setting is made in saw-wave one-shot pulse mode

The GTCCRC and GTCCRD registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, the correct output waveforms with secured dead time might 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.

21.9.3 Setting Range for GTCNT Counter

The GTCNT counter register must be set with the range of 0 ≤ GTCNT ≤ GTPR.

21.9.4 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 hand, there might be cases where an event is accepted or an interrupt occurs after GTCR.CST is set to 0.

21.9.5 Priority Order of Each Event

(1) GTCNT register

Table 21.23 shows a priority order of events updating the GTCNT register.

Table 21.23 Priority order of sources updating GTCNT

Source updating GTCNT	Priority order
Writing by CPU (writing to GTCNT/GTCLR)	High
Clear by hardware sources set in GTCSR	↑
Count up or down by hardware sources set in GTUPSR/GTDNSR	
Count operation	Low

If up-counting and down-counting by hardware sources occur at the same time, the GTCNT counter value does not change. When there is a conflict between updating the GTCNT register and reading by the CPU, pre-update data is read.

(2) GTCR.CST bit

When there is a conflict between starting/stopping by hardware sources set in the GTSSR/GTPSR registers and writing by the CPU (writing to GTCR/GTSTR/GTSTP registers), 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) GTCCRn registers (n = A to F)

When there is a conflict between input capture/buffer transfer operation and writing to the GTCCRn registers, the writing to GTCCRn 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 GTCCRn 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.

22. Low Power Asynchronous General Purpose Timer (AGT)

22.1 Overview

The Low Power 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 to the same address, and can be accessed with the AGT register.

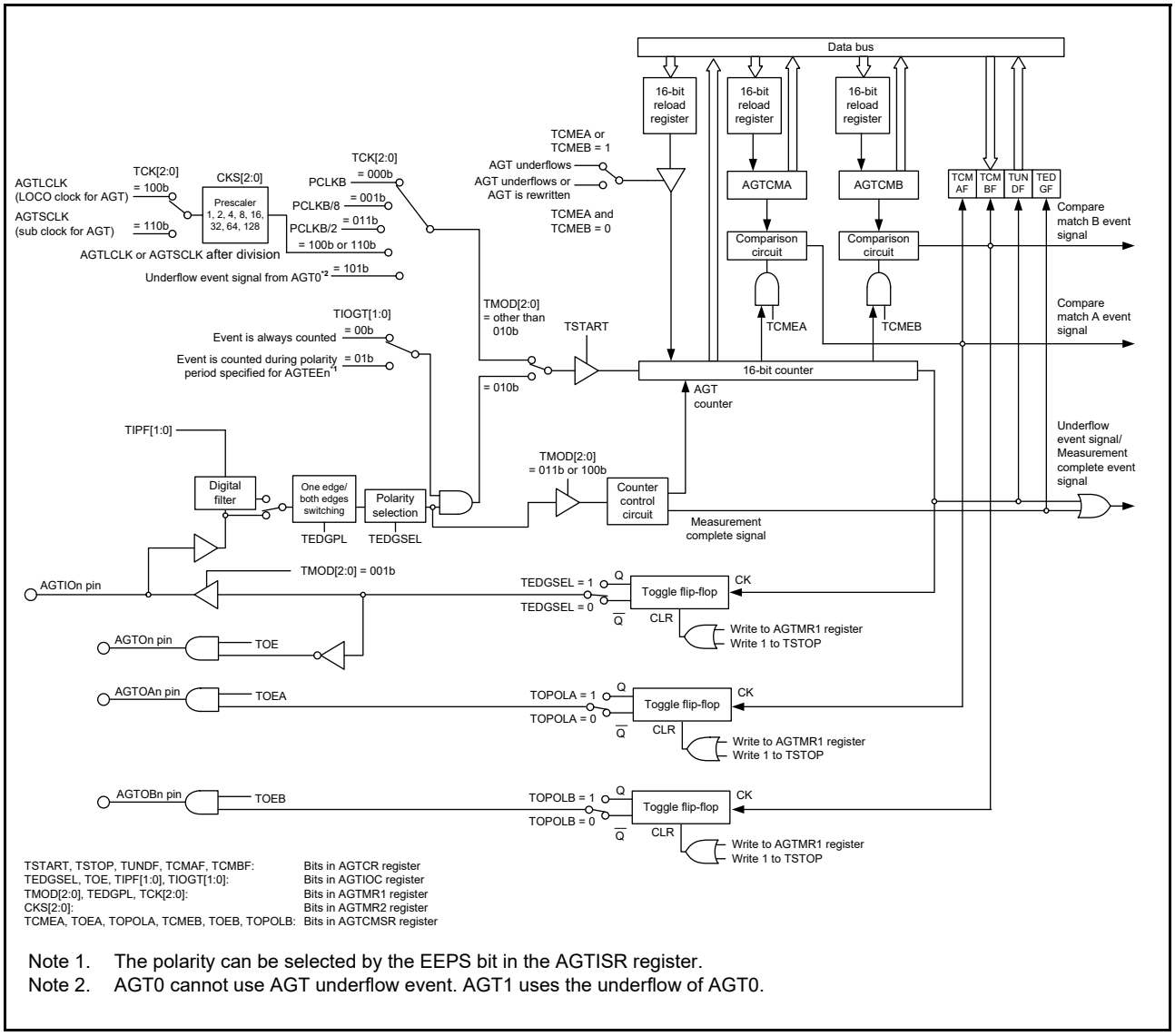
Table 22.1 lists the AGT specifications, Figure 22.1 shows a block diagram, and Table 22.2 lists the I/O pins.

Table 22.1 AGT specifications

Parameter		Specifications
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)		• Underflow event signal or measurement complete event signal – When the counter underflows – When the measurement of the active width of the external input (AGTIOn) completes in pulse width measurement mode – When the set edge of the external input (AGTIOn) is input in pulse period measurement mode. • Compare match A event signal – When the values of AGT and AGTCMA matched (compare match A function enabled). • Compare match B event signal – When the values of AGT and AGTCMB matched (compare match B function enabled). • Recovery from Software Standby mode can be performed by an 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 the AGT0 underflow signal. AGT1 connects directly with the underflow event signal from the AGT0 timer.

Note 2. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source clock.



Note 1. The polarity can be selected by the EEPS bit in the AGTISR register.
Note 2. AGT0 cannot use AGT underflow event. AGT1 uses the underflow of AGT0.

Figure 22.1 AGT block diagram

Table 22.2 AGT I/O pins

Pin name	I/O	Function
AGTEEn	Input	External event input for AGT
AGTIOn	Input/output	External event input and pulse output for AGT
AGTOn	Output	Pulse output for AGT
AGTOAn	Output	Output compare match A output for AGT
AGTOBn	Output	Output compare match B output for AGT

Note: Channel number (n = 0, 1).

22.2 Register Descriptions

22.2.1 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 outputs are 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 22.3.1, Reload Register and Counter Rewrite Operation. The AGT register can be set by a 16-bit memory manipulation instruction.

22.2.2 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 used.

The AGTCMA register is a read/write register to set a value for compare match with the AGT counter. The states of the reload register and compare register A change according to the TSTART bit in the AGTCR register. For details, see section 22.3.2, Reload Register and Compare Register A/B Rewrite Operation. The AGTCMA register can be set by a 16-bit memory manipulation instruction.

22.2.3 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 compare register B change according to the TSTART bit in the AGTCR register. For details, see section 22.3.2, Reload Register and Compare Register A/B Rewrite Operation. The AGTCMB register can be set by a 16-bit memory manipulation instruction.

22.2.4 AGT Control Register (AGTCR)

Address(es): AGT0.AGTCR 4008 4008h, AGT1.AGTCR 4008 4108h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCMBF	TCMAF	TUNDF	TEDGF	—	TSTOP	TCSTF	TSTART
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TSTART	AGT Count Start*2	0: Count stops 1: Count starts.	R/W
b1	TCSTF	AGT Count Status Flag*2	0: Count stops 1: Count in progress.	R
b2	TSTOP	AGT Count Forced Stop*1	0: Writing is invalid 1: The count is forcibly stopped.	W
b3	—	Reserved	The read value is 0. The write value should be 0.	R/W
b4	TEDGF	Active Edge Judgment Flag	0: No active edge received 1: Active edge received.	R/(W)*3
b5	TUNDF	Underflow Flag	0: No underflow 1: Underflow.	R/(W)*3
b6	TCMAF	Compare Match A Flag	0: No match 1: Match.	R/(W)*3
b7	TCMBF	Compare Match B Flag	0: No match 1: Match.	R/(W)*3

Note 1. When 1 (count is forcibly stopped) is written to the TSTOP bit, 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 the TSTART and TCSTF bits, see section 22.4.1, Count Operation Start and Stop Control.
Note 3. Only 0 can be written to clear the flag.

TSTART bit (AGT Count Start*2)

The count operation is started by writing 1 to the TSTART bit and stopped by writing 0. When this bit is set to 1 (count starts), 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 stops) in synchronization with the count source. For details, see section 22.4.1, Count Operation Start and Stop Control.

TCSTF flag (AGT Count Status Flag*2)

The TCSTF flag indicates the AGT count status.

[Setting condition]

- When 1 is written to the TSTART bit (the TCSTF bit is set to 1 in synchronization with the count source).

[Clearing conditions]

- When 0 is written to the TSTART bit (the TCSTF bit is set to 0 in synchronization with the count source)
- When 1 is written to the TSTOP bit.

TSTOP bit (AGT Count Forced Stop*1)

When 1 is written to the TSTOP bit, the count is forcibly stopped. The read value is 0.

TEDGF flag (Active Edge Judgment Flag)

The TEDGF flag indicates that an active edge was detected.

[Setting condition]

- When the measurement of the active width of the external input (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)

The TUNDF flag indicates that the counter underflowed.

[Setting condition]

- When the counter underflows.

[Clearing condition]

- When 0 is written to this flag by software.

TCMAF flag (Compare Match A Flag)

The TCMAF flag indicates that compare match A was detected.

[Setting condition]

- When the value in the AGT register matches the value in the AGTCMA register.

[Clearing condition]

- When 0 is written to this flag by software.

TCMBF flag (Compare Match B Flag)

The TCMBF flag indicates that compare match B was detected.

[Setting condition]

- When the value in the AGT register matches the value in the AGTCMB register.

[Clearing condition]

- When 0 is written to this flag by software.

22.2.5 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*4	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*5	0: Single-edge 1: Both-edge.	R/W
b6 to b4	TCK[2:0]	Count Source*2, *3, *6	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 AGTMR2 register 1 0 1: Underflow event signal from AGT0*7 1 1 0: Divided clock AGTSCLK specified by CKS[2:0] bits in AGTMR2 register. Other settings are prohibited.	R/W
b7	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note 1. Write access to the AGTMR1 register initializes the output from the AGTOn, AGTIOn, AGTOAn and AGTOBn pins of the AGT (n = 0, 1). For details on the output level at initialization, see section 22.2.7, AGT I/O Control Register (AGTIOC).

Note 2. When event counter mode is selected, the external input (AGTIOn) is selected as the count source regardless of the TCK[2:0] bit setting.

Note 3. Do not switch count sources during count operation. Count sources should be switched when both the TSTART and TCSTF bits in the AGTCR register are set to 0 (count is stopped).

Note 4. 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).

Note 5. The TEDGPL bit is enabled only in event counter mode.

Note 6. When running AGT in Software Standby mode or Snooze mode, select AGTLCLK or AGTSCLK (TCK[2:0] = 100b or 110b) as the count source.

Note 7. AGT0 cannot use AGT0 underflow (setting prohibited). AGT1 uses the AGT0 underflow.

22.2.6 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 or 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] bits during count operation. Only rewrite the CKS[2:0] bits when both the TSTART and TCSTF bits in the AGTCR register are set to 0 (count is stopped).

Note 2. When count source is AGTLCLK or AGTSCLK, the switch of CKS[2:0] 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 mode, 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.

22.2.7 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 22.3 and Table 22.4. 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 AGTOn and AGTIOn output. The AGTOn and AGTIOn output are initialized when the AGTMR1 register is written or the TSTOP bit of the AGTCR register is written with 1.	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	TOE	AGTOn Output Enable	0: AGTOn output disabled 1: AGTOn output enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	TIPF[1:0]	Input Filter*3	b5 b4 0 0: No filter 0 1: Filter sampled at PCLKB 1 0: Filter sampled at PCLKB/8 1 1: Filter sampled at PCLKB/32. These bits specifies the sampling frequency of the filter for the AGTIOn input. If the input to the AGTIOn pin is sampled and the value matches three successive times, that value is taken as the input value.	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 pin is used, the polarity to count an event can be selected with the EEPS bit in the AGTISR register.
Note 2. TIOGT[1:0] bits are enabled only in event counter mode.
Note 3. When event counter mode operation is performed during Software Standby mode, the digital filter function cannot be used.

Table 22.3 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 22.4 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).

22.2.8 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

22.2.9 AGT Compare Match Function Select Register (AGTCMSR)

Address(es): AGT0.AGTCMSR 4008 400Eh, AGT1.AGTCMSR 4008 410Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	TOPOLB	TOEB	TCMEB	—	TOPOLA	TOEA	TCMEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TCMEA	Compare Match A Register Enable*1, *2	0: Compare match A register disabled 1: Compare match A register enabled.	R/W
b1	TOEA	AGTOAn Output Enable*1, *2	0: AGTOAn output disabled 1: AGTOAn output enabled.	R/W
b2	TOPOLA	AGTOAn Polarity Select*1, *2	0: AGTOAn output is started on low 1: AGTOAn output is started on high.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	TCMEB	Compare Match B Register Enable*1, *2	0: Compare match B register disabled 1: Compare match B register enabled.	R/W
b5	TOEB	AGTOBn Output Enable*1, *2	0: AGTOBn output disabled 1: AGTOBn output enabled.	R/W
b6	TOPOLB	AGTOBn Polarity Select*1, *2	0: AGTOBn output is started 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.

22.2.10 AGT Pin Select Register (AGTIOSEL)

Address(es): AGT0.AGTIOSEL 4008 400Fh, AGT1.AGTIOSEL 4008 410Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TIES	—	—	—	—
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	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

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.

TIES bit (AGTIOn Input Enable)

The TIES bit enables or disables an external event input.

22.3 Operation

22.3.1 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 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 22.2 and Figure 22.3 show the timing of rewrite operation with TSTART bit value and TCMEA/TCMEB bit value.

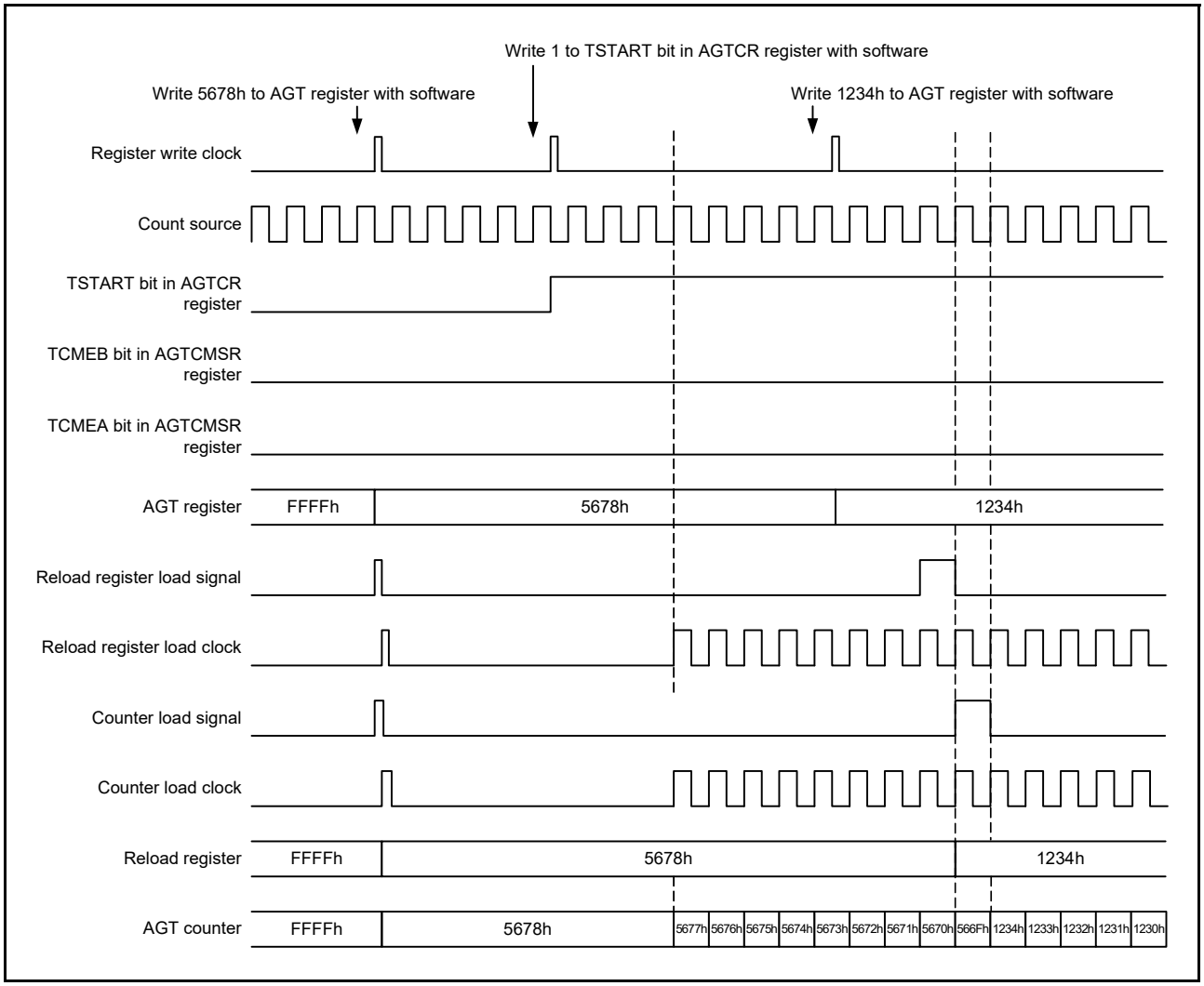


Figure 22.2 Timing of rewrite operation with TSTART, TCMEA and TCMEB bit values when compare match A register and compare match B register are invalid

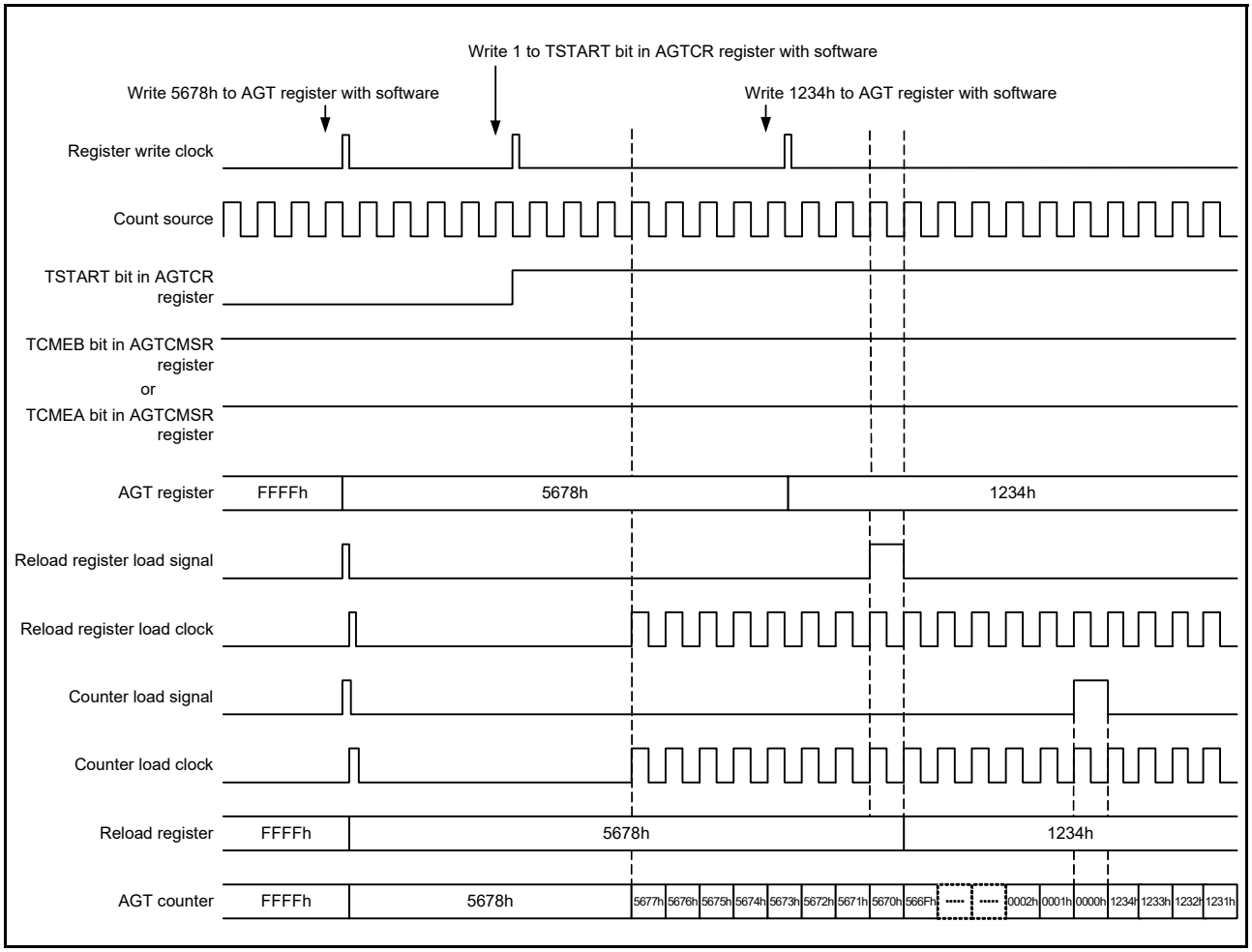


Figure 22.3 Timing of rewrite operation with TSTART bit value and TCMEA or TCMEB bit value when compare match A register or compare match B register is valid

22.3.2 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 22.4 shows the timing of rewrite operation with TSTART bit value for compare register A. Compare register B has the same timing as compare register A.

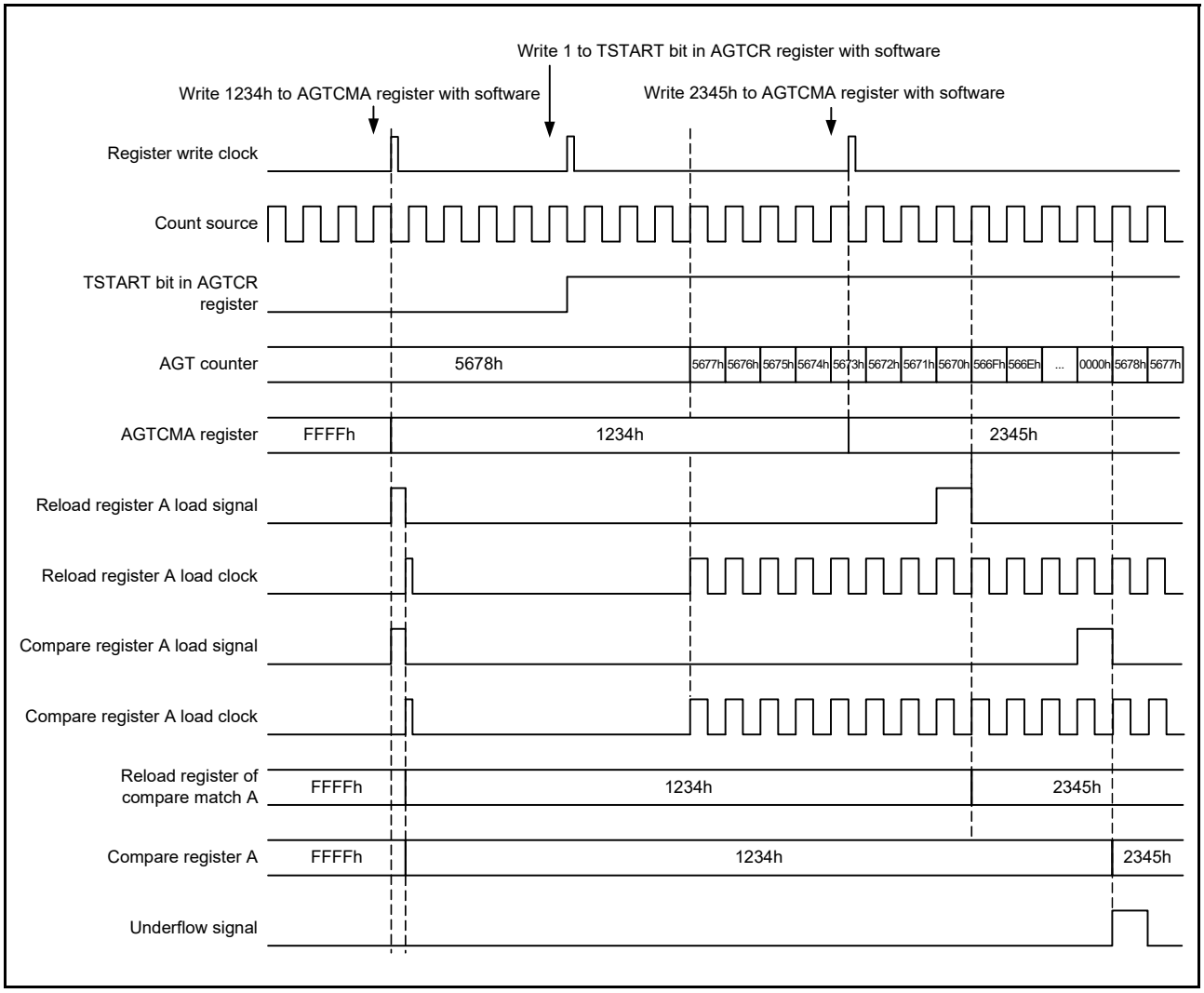


Figure 22.4 Timing of rewrite operation with the TSTART bit value for compare register A

22.3.3 Timer Mode

In timer 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 22.5 shows the operation example in timer mode.

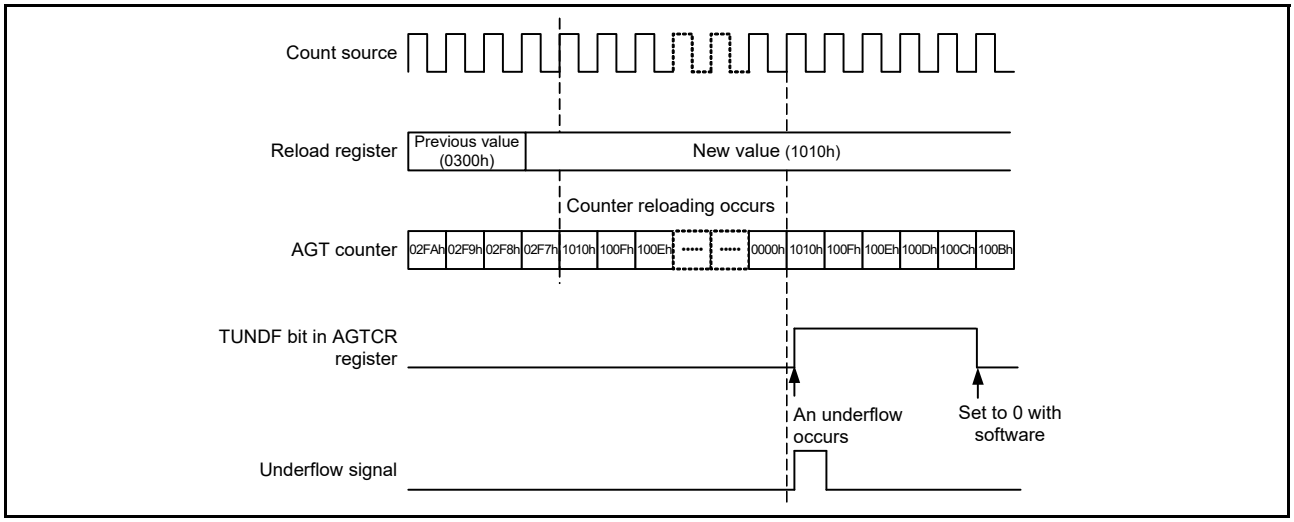


Figure 22.5 Operation example in timer mode

22.3.4 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 is 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 22.6 shows the operation example in pulse output mode.

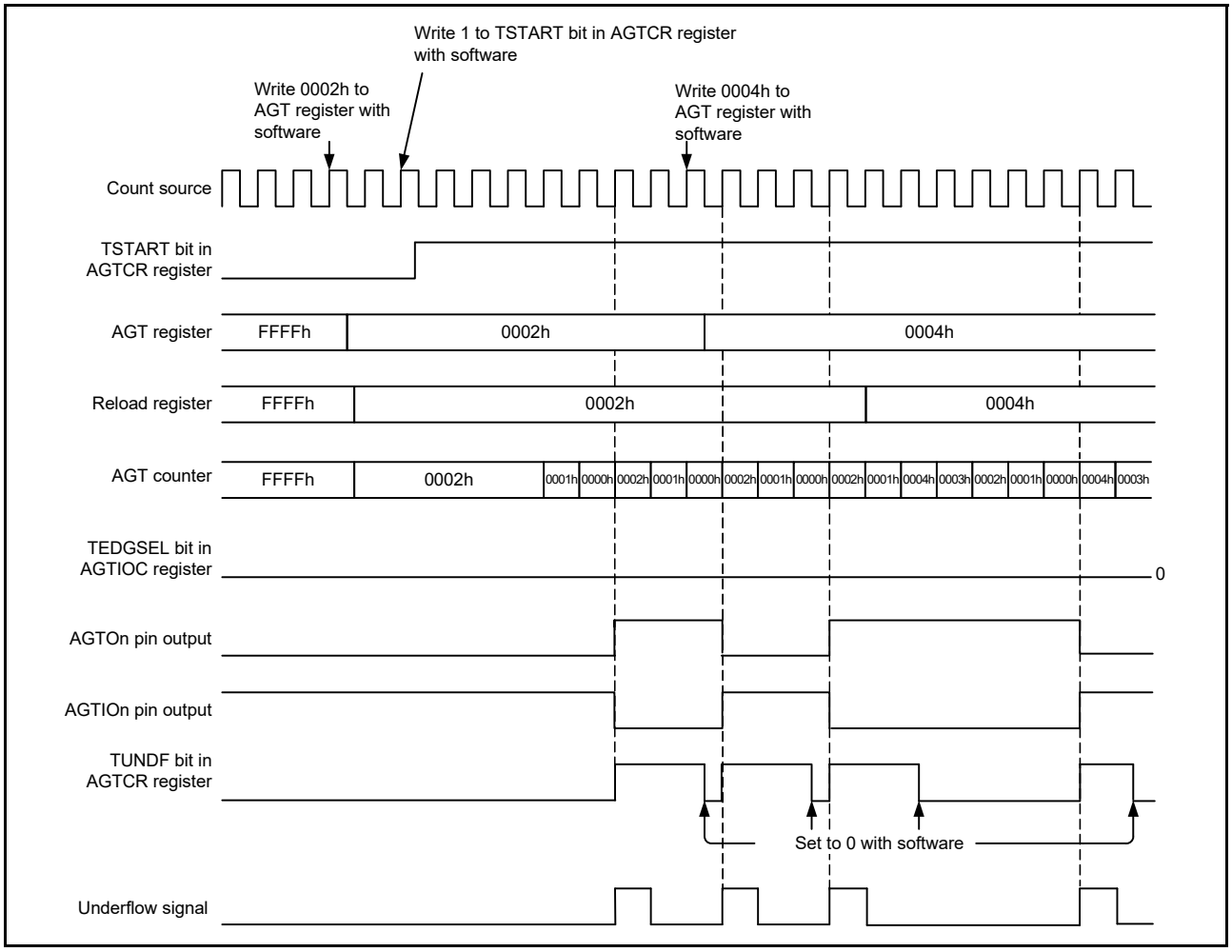


Figure 22.6 Operation example in pulse output mode

22.3.5 Event Counter Mode

In event counter mode, the counter is decremented by an external event signal (count source) input to the AGTIOn pin. Various periods for counting events can be set with the TIOGT[1:0] bits in the AGTIOC 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 22.7 shows the operation example in event counter mode.

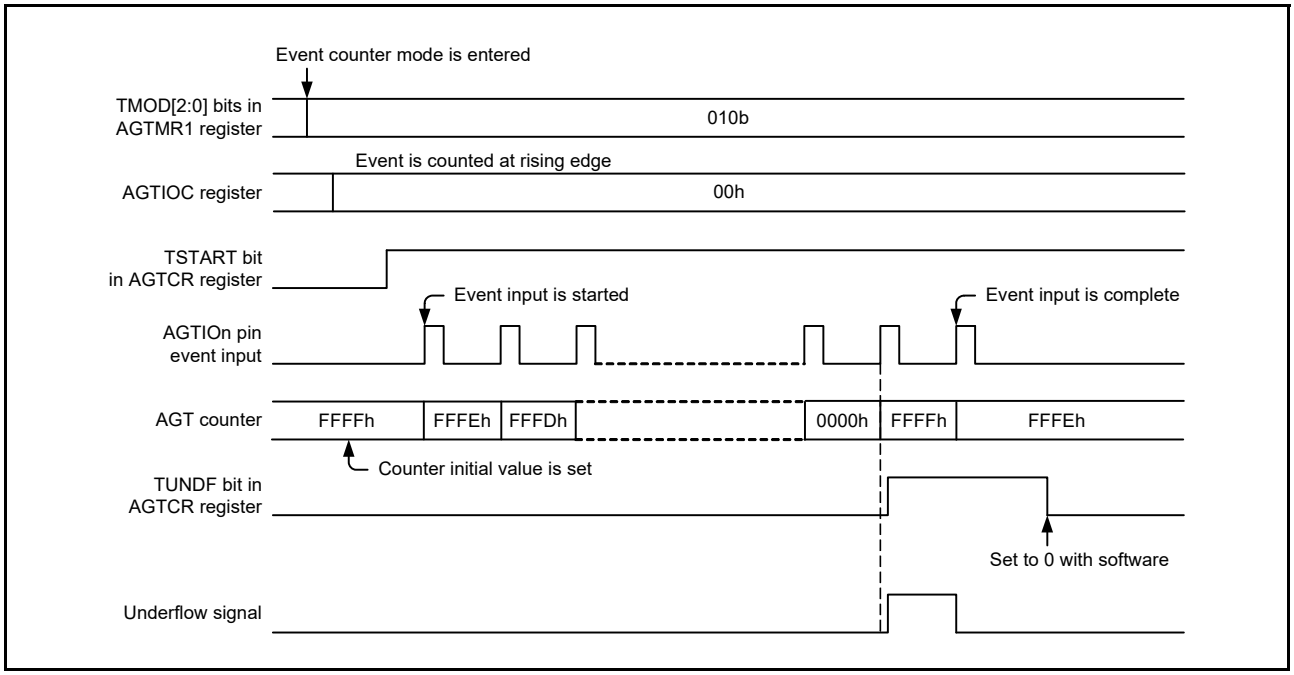


Figure 22.7 Operation example 1 in event counter mode

Figure 22.8 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).

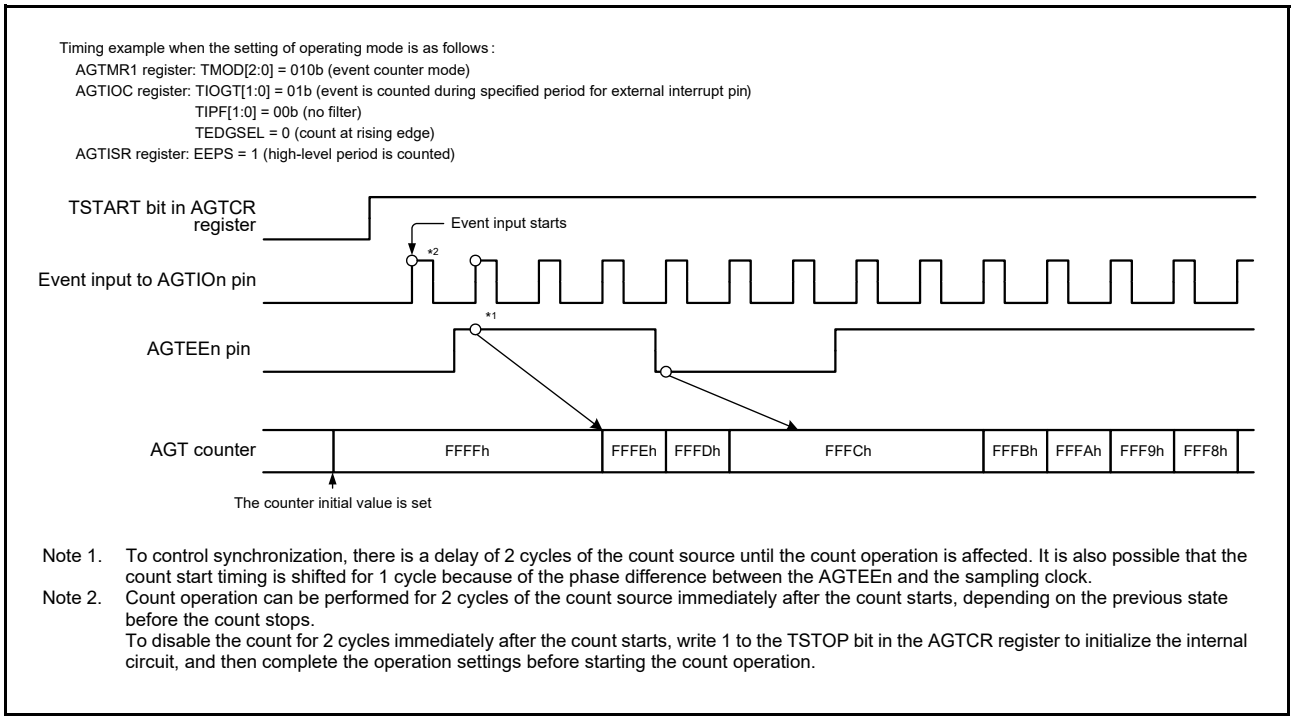


Figure 22.8 Operation example 2 in event counter mode

22.3.6 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 22.9 shows the operation example in pulse width measurement mode.

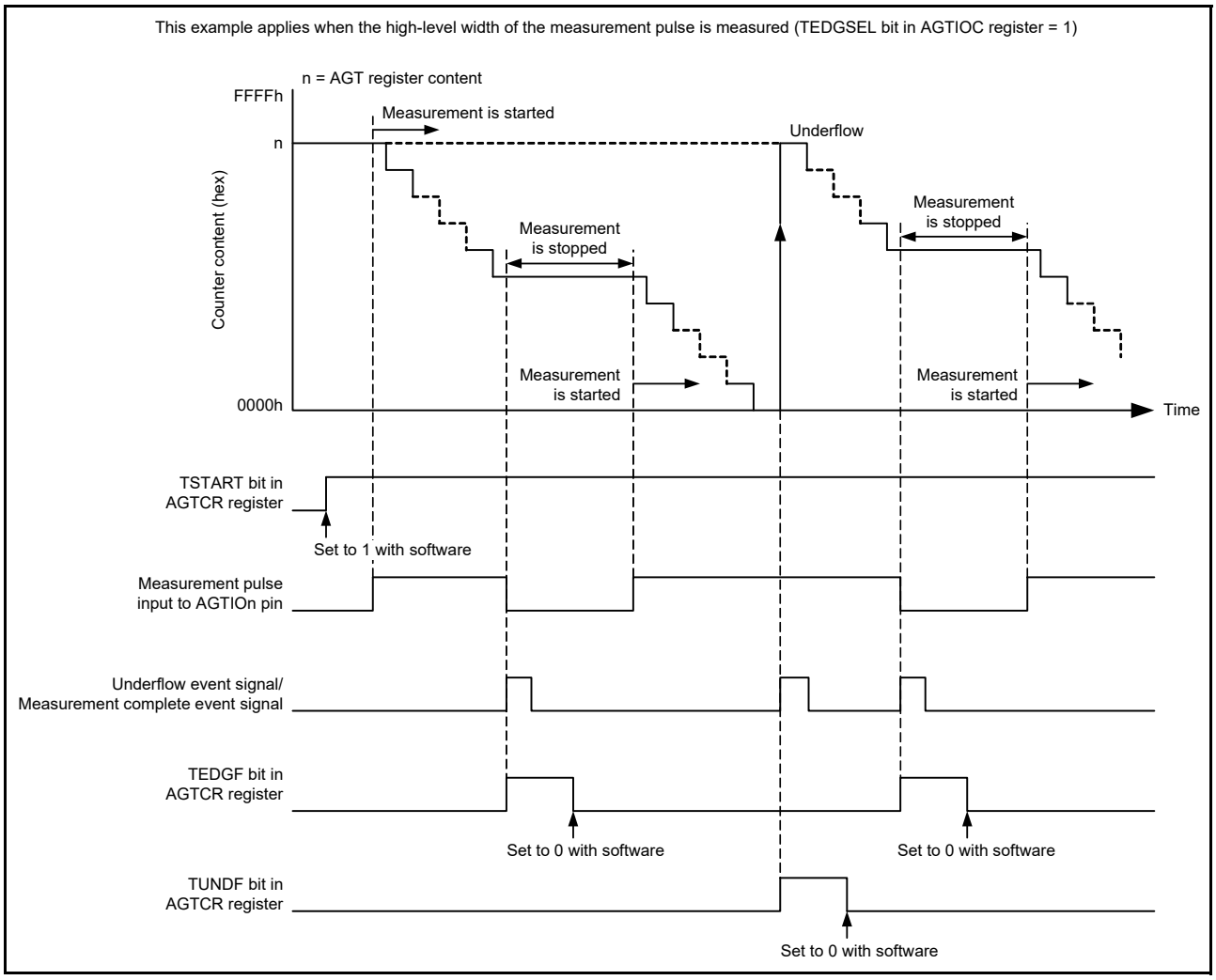


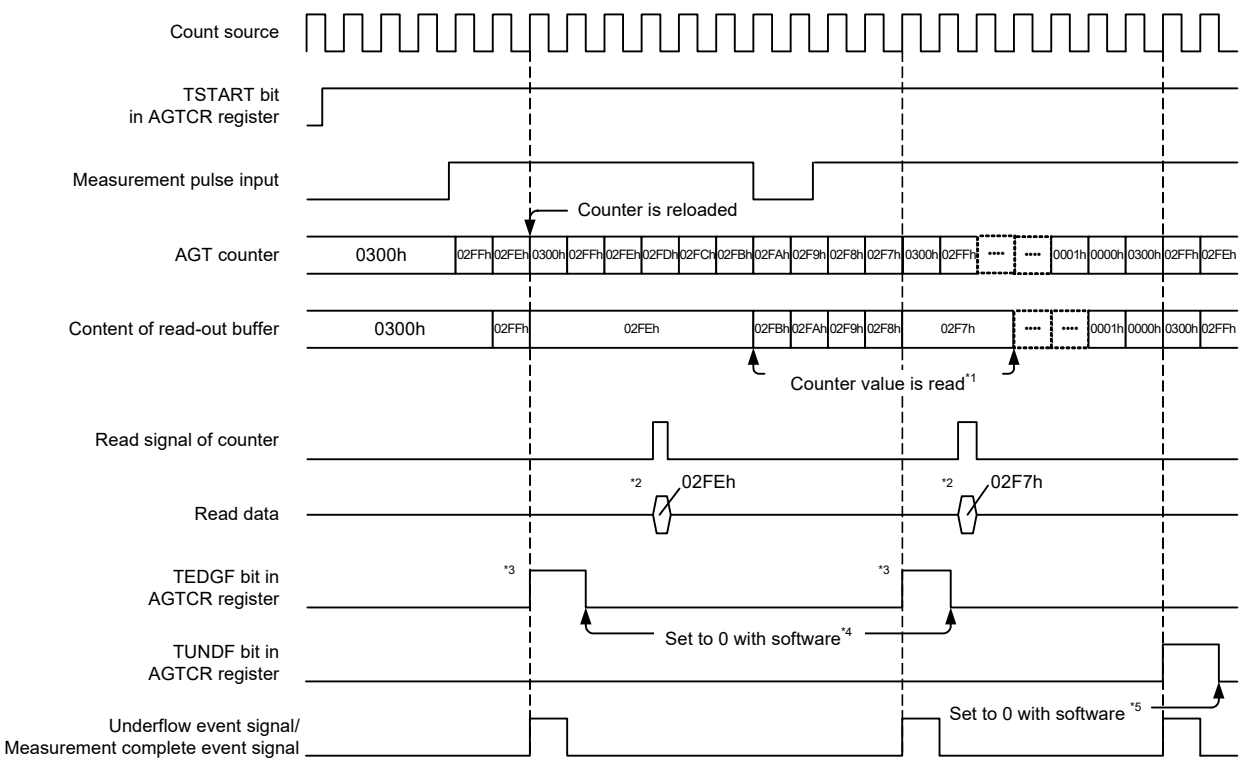
Figure 22.9 Operation example in pulse width measurement mode

22.3.7 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 the 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 22.4.5, 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 (underflow) and an interrupt request is generated.

Figure 22.10 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.



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 is input and then the set edge of an external pulse is 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 22.10 Operation example in pulse period measurement mode

22.3.8 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 22.3.1, Reload Register and Counter Rewrite Operation for details. In addition, the output level of the AGTOAn and AGTOBn pins is inverted by the match and by the underflow. The output level can be selected with the TOPOLA or TOPOLB bit in the AGTCMSR register.

Figure 22.11 shows the operation example in compare match mode.

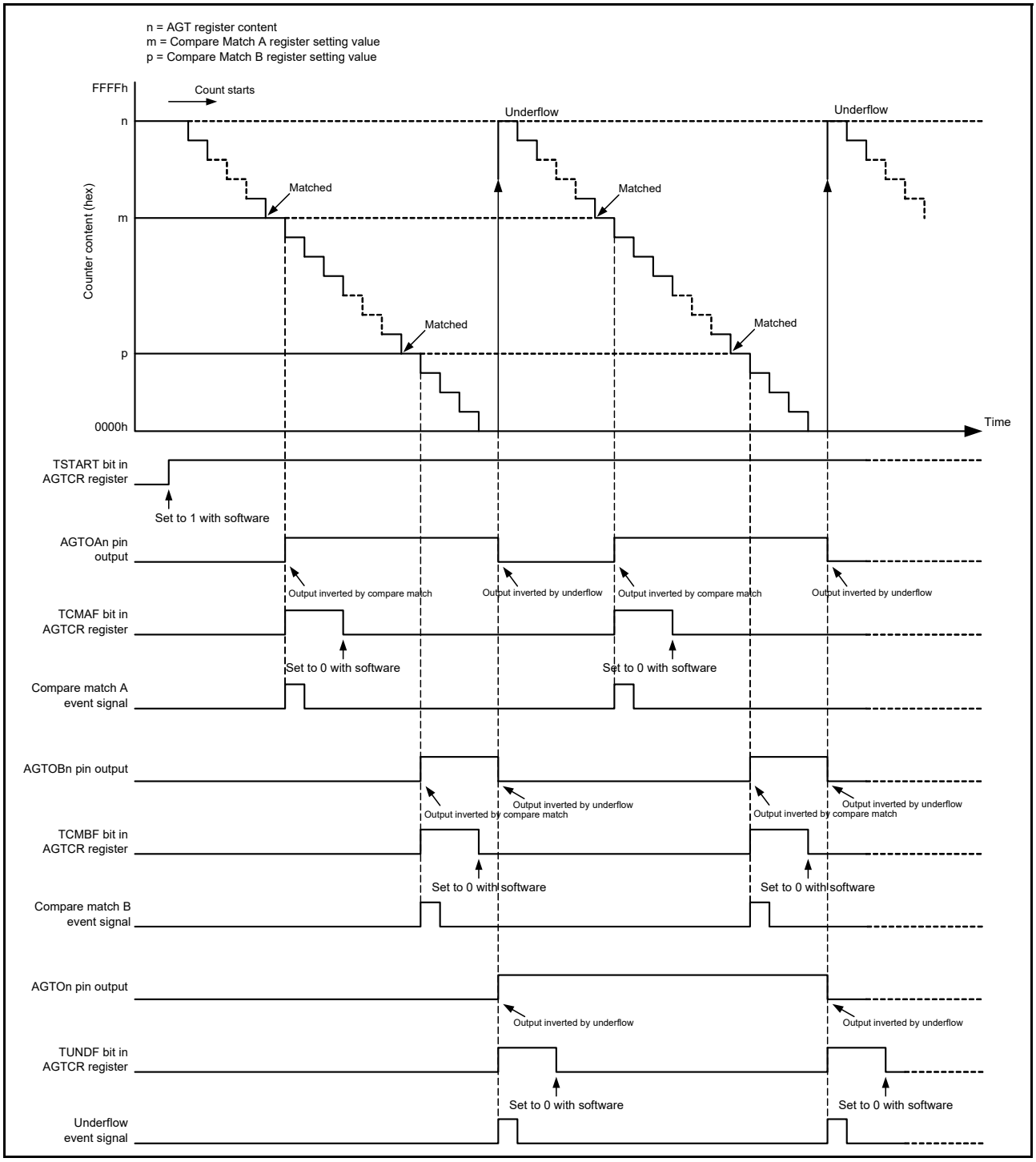


Figure 22.11 Operation example in compare match mode (TOPOLA = 0, TOPOLB = 0)

22.3.9 Output Settings for Each Mode

Table 22.5 to Table 22.8 list the states of pins AGTOn, AGTIOn, AGTOAn, and AGTOBn in each mode.

Table 22.5 AGTOn pin setting

Operating mode	AGTIOC register		AGTOn pin output
	TOE bit	TEDGSEL bit	
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

Table 22.6 AGTIOn pin setting

Operating mode	AGTIOC Register	AGTIOn pin I/O
	TEDGSEL bit	
Timer mode	0 or 1	Input (not used)
Pulse output mode	1	Normal output
	0	Inverted output
Event counter mode	0 or 1	Input
Pulse width measurement mode		
Pulse period measurement mode		

Table 22.7 AGTOAn pin setting

Operating mode	AGTCMSR register		AGTOAn pin output
	TOEA bit	TOPOLA bit	
Timer mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse output mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Event counter mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse width measurement mode	0	0	Prohibited
Pulse period measurement mode			

Table 22.8 AGTOBn pin setting (1 of 2)

Operating mode	AGTCMSR register		AGTOBn pin output
	TOEB bit	TOPOLB bit	
Timer mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse output mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Event counter mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)

Table 22.8 AGTOBn pin setting (2 of 2)

Operating mode	AGTCMSR register TOEB bit	TOPOLB bit	AGTOBn pin output
Pulse width measurement mode	0	0	Prohibited
Pulse period measurement mode			

22.3.10 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 22.9 and Table 22.10 show the settings that can be used in Software Standby mode.

Table 22.9 Usable settings 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)	AGTIO0	–
Pulse width measurement mode	100b or 110b	AGTLCLK or AGTSCLK	–
Pulse period measurement mode	100b or 110b	AGTLCLK or AGTSCLK	–

Table 22.10 Usable settings 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)	AGTIO1	• 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 22.9.

22.3.11 Interrupt Sources

The AGT has three interrupt sources described in Table 22.11.

Table 22.11 AGT interrupt sources

Name	Interrupt source	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).

22.3.12 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 17, Event Link Controller (ELC).

22.4 Usage Notes

22.4.1 Count Operation Start and Stop Control

- When the operating mode (see Table 22.1) other than the event counter mode is set, 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 13, 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 22.1) 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 13, Interrupt Controller Unit (ICU) for details.

Note 1. Registers associated with AGT: AGT, AGTCMA, AGTCMB, AGTCR, AGTMR1, AGTMR2, AGTIOC, AGTISR and AGTCMSR

22.4.2 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.

22.4.3 When Changing Mode

The registers associated with AGT operating mode (AGTMR1, AGTMR2, AGTIOC, AGTISR, and AGTCMSR) can be changed only when the count is stopped with both the TSTART 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).

22.4.4 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.

22.4.5 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.

22.4.6 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.

22.4.7 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).

22.4.8 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.

22.4.9 When Selecting PCLKB, PCLKB/8, or PCLKB/2 as the Count Source

When a reset is generated, the operation of AGT cannot be guaranteed. Set the registers associated with AGT again.

22.4.10 When Selecting AGTLCLK or AGTSCLK as the Count Source

The MSTPD2 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.

22.4.11 When Switching Source Clock

When switching a clock source by changing SCKSCR.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.

23. Realtime Clock (RTC)

23.1 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.

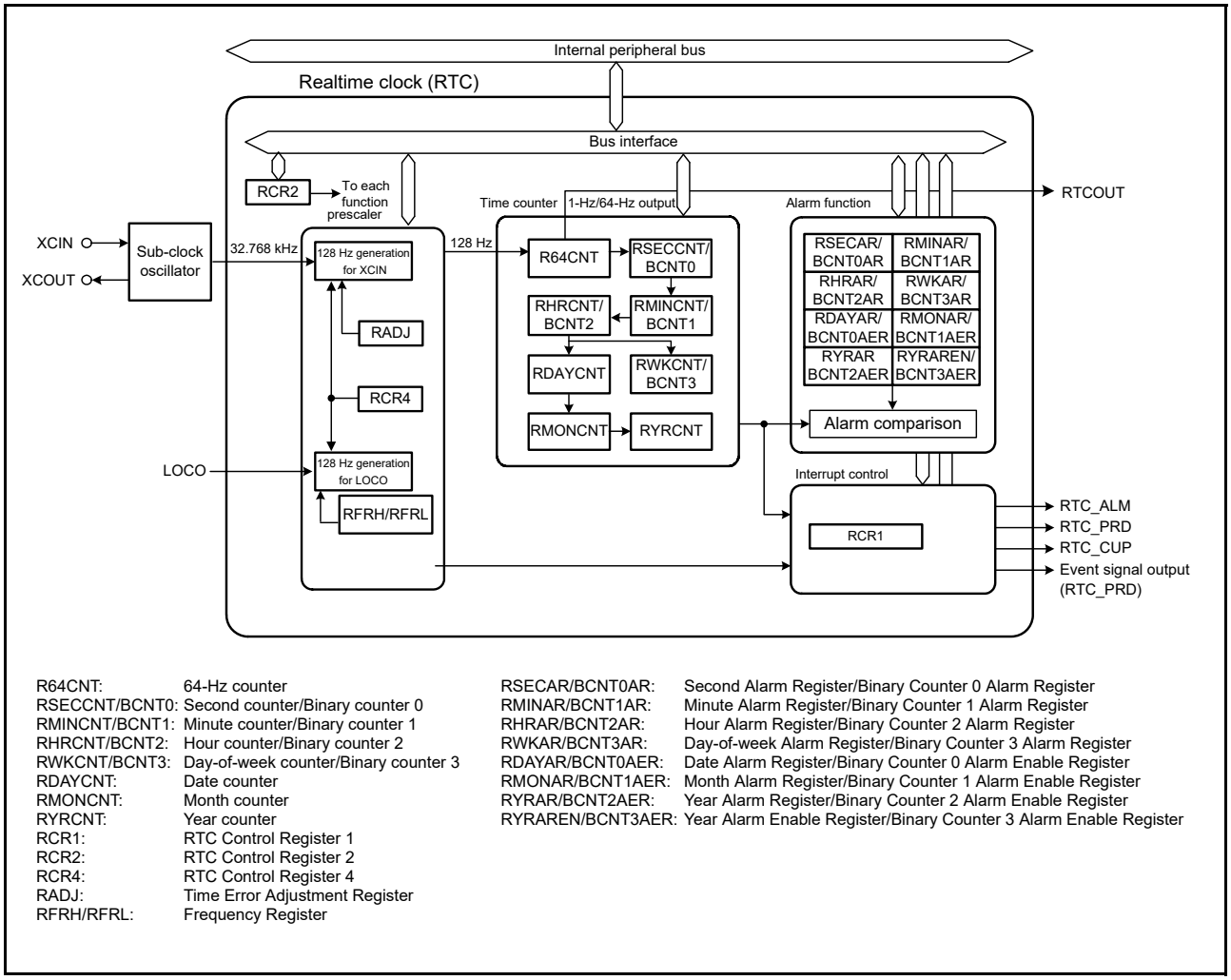
The sub-clock oscillator or LOCO can be selected as the count source of the time counters. The RTC uses a 128-Hz clock acquired by dividing the count source by a prescaler. Year, month, date, day-of-week, a.m./p.m. (in 12-hour mode), hour, minute, second, or 32-bit binary is counted by 1/128 second.

Table 23.1 lists the RTC specifications, Figure 23.1 shows a block diagram, and Table 23.2 lists the I/O pins.

Table 23.1 RTC specifications

Parameter	Specifications
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 • Shared by 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.
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.



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

Figure 23.1 RTC block diagram

Table 23.2 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

23.2 Register Descriptions

Write or read from the RTC registers as described in section 23.6.5, 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 23.6.4, Transitions to Low Power Modes after Setting Registers.

23.2.1 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	Indicate the state between 1 Hz and 64 Hz of the sub-second digit	R
b1	F32HZ	32 Hz		R
b2	F16HZ	16 Hz		R
b3	F8HZ	8 Hz		R
b4	F4HZ	4 Hz		R
b5	F2HZ	2 Hz		R
b6	F1HZ	1 Hz		R
b7	—	Reserved	This bit is read as 0	R

The R64CNT counter is used in both calendar count mode and 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 23.3.5, Reading 64-Hz Counter and Time.

23.2.2 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 23.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode:

Address(es): RTC.BCNT0 4004 4002h

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

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 23.3.5, Reading 64-Hz Counter and Time.

23.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1)

(1) In calendar count mode:

Address(es): RTC.RMINCNT 4004 4004h

	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 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 the carries generated once per minute in the second counter.

A value from 00 through 59 (in BCD) can be specified. If a value outside 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 23.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode:

Address(es): RTC.BCNT1 4004 4004h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

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 23.3.5, Reading 64-Hz Counter and Time.

23.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2)

(1) In calendar count mode:

Address(es): RTC.RHRCNT 4004 4006h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	HR1[3:0]	1-Hour Count	Counts from 0 to 9 once per hour. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	HR10[1:0]	10-Hour Count	Counts from 0 to 2 once per carry from the ones place	R/W
b6	PM	PM	Time counter setting for AM/PM: 0: AM 1: PM.	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 the carries generated once per 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 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 23.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode:

Address(es): RTC.BCNT2 4004 4006h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT2 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 23.3.5, Reading 64-Hz Counter and Time.

23.2.5 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 23.3.5, 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 23.3.5, Reading 64-Hz Counter and Time.

23.2.6 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 23.3.5, Reading 64-Hz Counter and Time.

23.2.7 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 23.3.5, Reading 64-Hz Counter and Time.

23.2.8 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 the 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 23.3.5, Reading 64-Hz Counter and Time.

23.2.9 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 set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT0AR 4004 4010h

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

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.

23.2.10 Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR)

(1) In calendar count mode:

Address(es): RTC.RMINAR 4004 4012h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	MIN10[2:0]			MIN1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	MIN1[3:0]	1 Minute	Value for the ones place of minutes	R/W
b6 to b4	MIN10[2:0]	10 Minutes	Value for the tens place of minutes	R/W
b7	ENB	ENB	0: The register value is not compared with the RMINCNT counter value 1: The register value is compared with the RMINCNT counter value.	R/W

RMINAR is an alarm register associated with the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. The RMINAR values from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT1AR 4004 4012h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT1AR is a read/write alarm register associated with the 32-bit binary counter from b15 to b8. This register is set to 00h by an RTC software reset.

23.2.11 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: AM. 1: PM.	R/W
b7	ENB	ENB	0: The register value is not compared with the RHRCNT counter value 1: The register value is compared with the RHRCNT counter value.	R/W

RHRAR is an alarm register associated with the BCD-coded hour counter RHRCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHRCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24):

- When the RCR2.HR24 bit is 0 — From 00 to 11 (in BCD)
- When the RCR2.HR24 bit is 1 — From 00 to 23 (in BCD).

If a value outside of this range is specified, the RTC does not operate correctly. When the RCR2.HR24 bit is 0, be sure to set the PM bit. When the RCR2.HR24 bit is 1, the setting in the PM bit has no effect. This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT2AR 4004 4014h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT2AR is a read/write alarm register associated with the 32-bit binary counter b23 to b16. This register is set to 00h by an RTC software reset.

23.2.12 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.

23.2.13 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 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 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.

23.2.14 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.

23.2.15 Year Alarm Register (RYRAR)/Binary Counter 2 Alarm Enable Register (BCNT2AER)

(1) In calendar count mode:

Address(es): RTC.RYRAR 4004 401Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	YR1[3:0]	1 Year	Value for the ones place of years	R/W
b7 to b4	YR10[3:0]	10 Years	Value for the tens place of years	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RYRAR is an alarm register associated with the BCD-coded year counter RYRCNT. The RYRAR values from 00 through 99 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0000h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT2AER 4004 401Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ENB[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

BCNT2AER is a read/write register for setting the alarm enable associated with the 32-bit binary counter b23 to b16. The binary counter (BCNT[31:0]) associated with the ENB[31:0] bits that are set to 1 is compared with the binary alarm register (BCNTAR[31:0]), and when all match, the IR flag associated with the RTC_ALM interrupt becomes 1. This register is set to 0000h by an RTC software reset.

23.2.16 Year Alarm Enable Register (RYRAREN)/ Binary Counter 3 Alarm Enable Register (BCNT3AER)

(1) In calendar count mode:

Address(es): RTC.RYRAREN 4004 401Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RYRCNT counter value 1: The register value is compared with the RYRCNT counter value.	R/W

When the ENB bit in RYRAREN is set to 1, the RYRAR value is compared with the RYRCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1.This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT3AER 4004 401Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT3AER is a read/write register for setting the alarm enable associated with the 32-bit binary counter b31 to b24. The binary counter (BCNT[31:0]) associated with the ENB[31:0] bits that are set to 1 is compared with the binary alarm register (BCNTAR[31:0]), and when all match, the IR flag associated with the RTC_ALM interrupt becomes 1. This register is set to 00h by an RTC software reset.

23.2.17 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 18.5.1, 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.

23.2.18 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*2	0: Automatic adjustment is disabled 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select*2	0: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every minute 1: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every 10 seconds.	R/W
b6	HR24	Hours Mode	0: 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, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP.

Note 2. When LOCO is selected, the setting of this bit is disabled.

The RCR2 register is related to hours mode, automatic adjustment function, enabling RTCOUT output, 30-second adjustment, RTC software reset, and controlling count operation.

START bit (Start)

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 completes, 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 set 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*2)

The AADJE bit 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)

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 23.3.1, 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*2	0: Automatic adjustment is disabled 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select*2	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, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP

Note 2. 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 completes, the RESET bit is automatically set to 0. When 1 is written to the RESET bit, check that the bit is 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 output from an external pin, enable the port control in addition to setting this bit.

AADJE bit (Automatic Adjustment Enable)

The AADJE bit 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)

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 set 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 23.3.1, Outline of Initial Settings of Registers after Power On.

23.2.19 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.

23.2.20 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 ≥ 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.

23.2.21 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 bit (RCR2.AADJE) 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 bit (RCR2.AADJP).

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 set 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.

23.3 Operation

23.3.1 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, and interrupt control register.

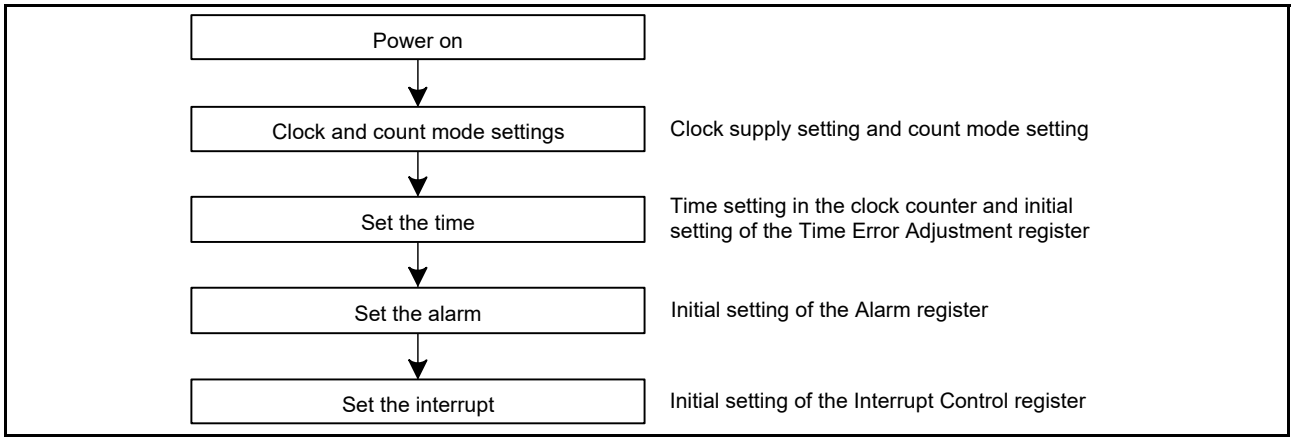


Figure 23.2 Outline of initial settings after a power on

23.3.2 Clock and Count Mode Setting Procedure

Figure 23.3 shows how to set the clock and the count mode.

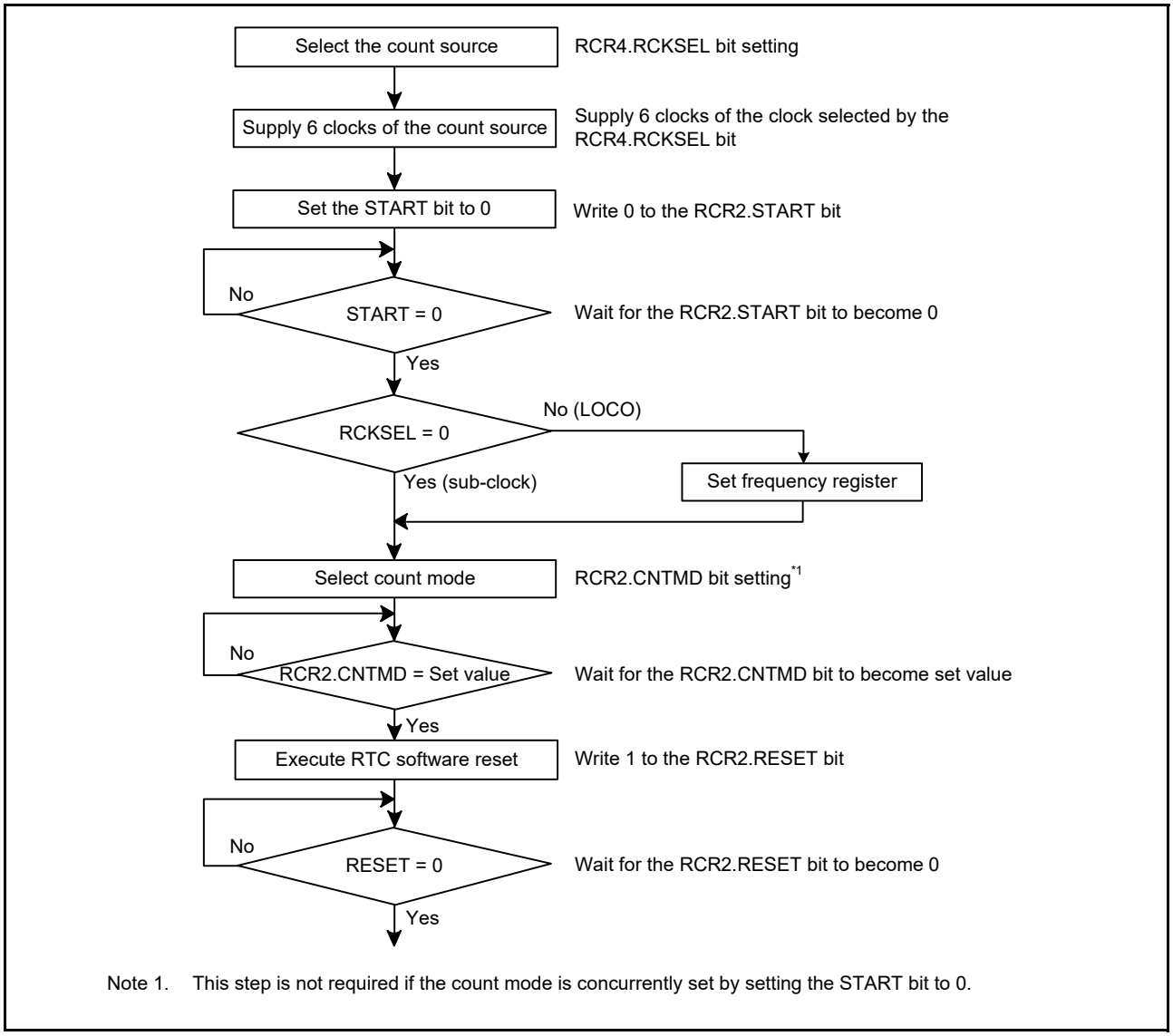
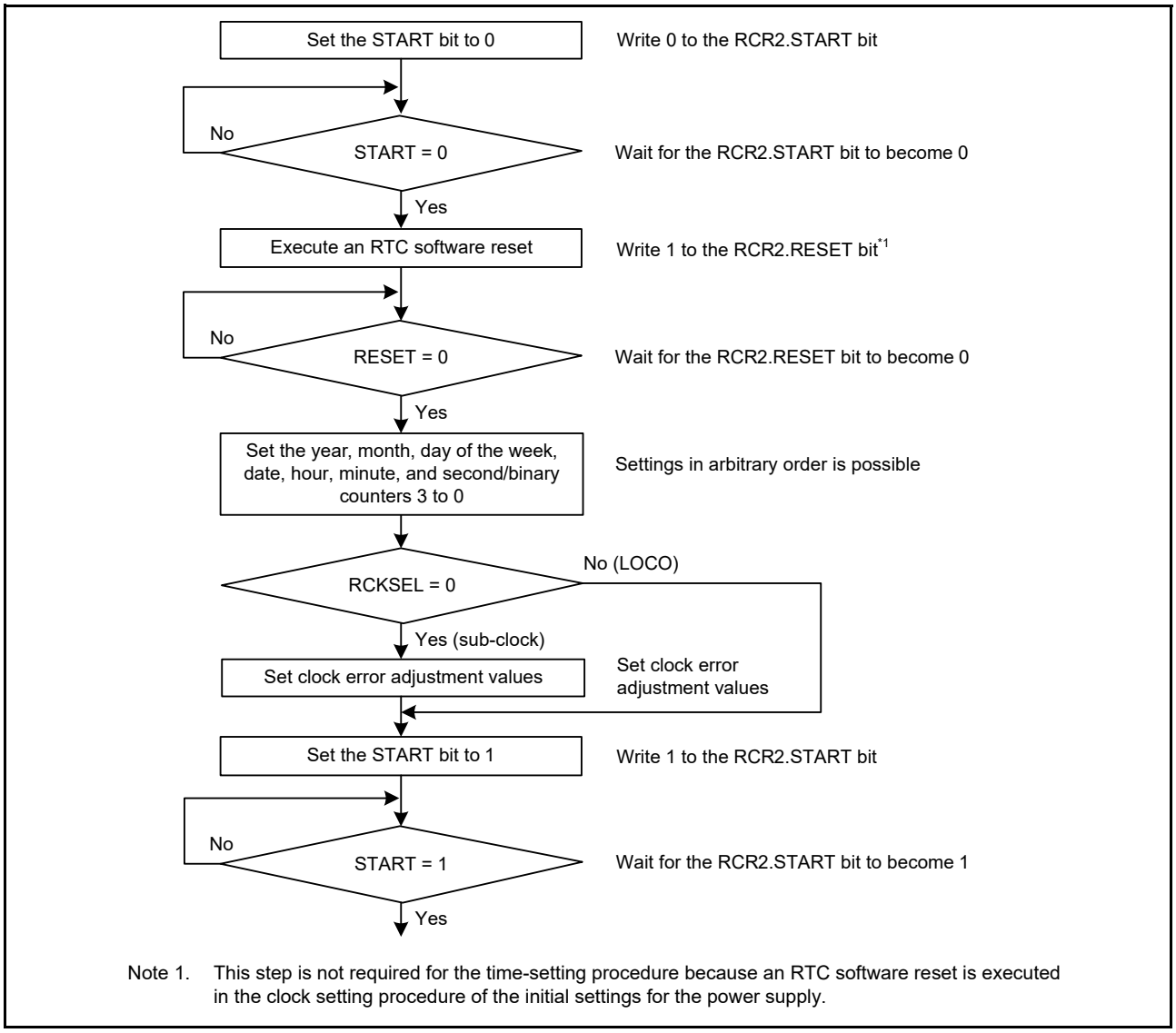


Figure 23.3 Clock and count mode setting procedure

23.3.3 Setting the Time

Figure 23.4 shows how to set the time.



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 23.4 Setting the time

23.3.4 30-Second Adjustment

Figure 23.5 shows how to execute a 30-second adjustment.

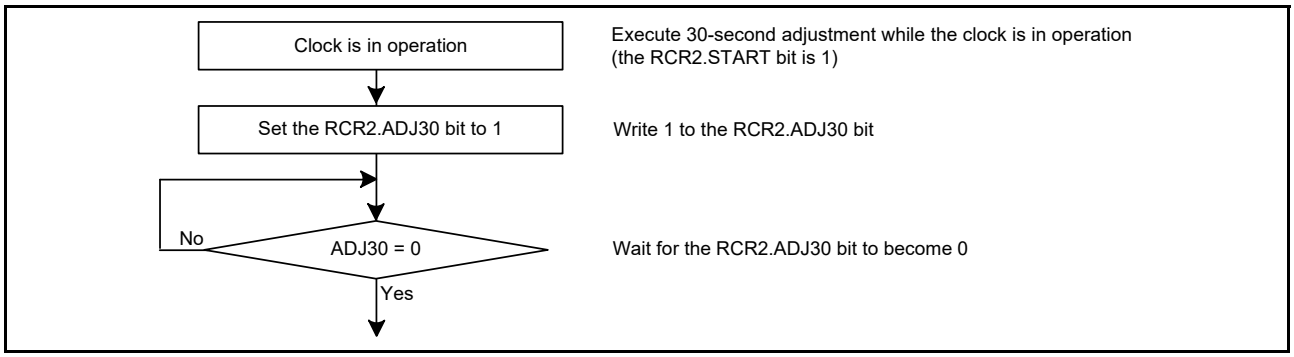


Figure 23.5 30-second adjustment

23.3.5 Reading 64-Hz Counter and Time

Figure 23.6 shows how to read a 64-Hz counter and time.

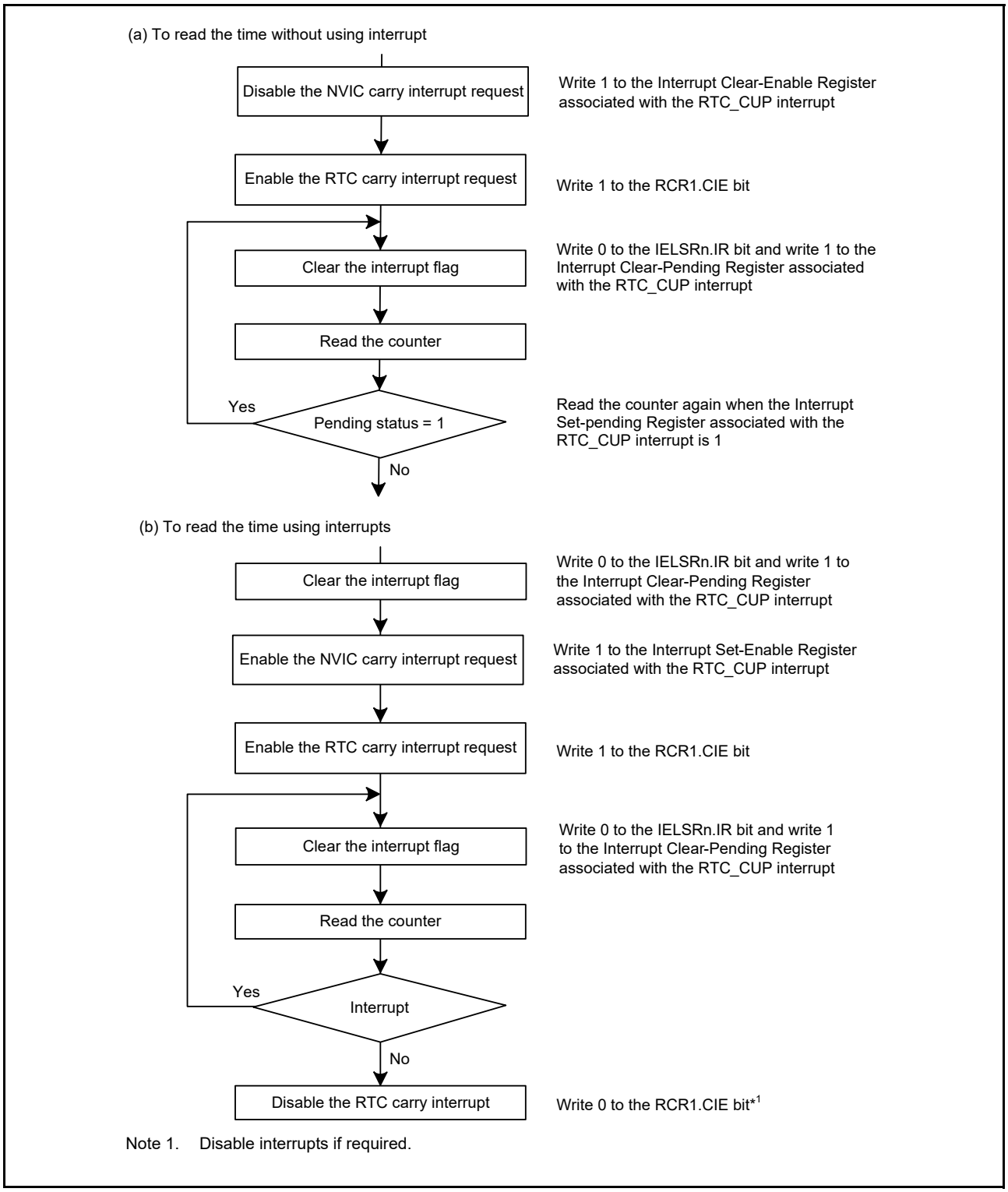


Figure 23.6 Reading time

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 23.6, and the procedure using carry interrupts is shown in (b). To keep the program simple, method (a) should be used in most cases.

23.3.6 Alarm Function

Figure 23.7 shows how to use the alarm function.

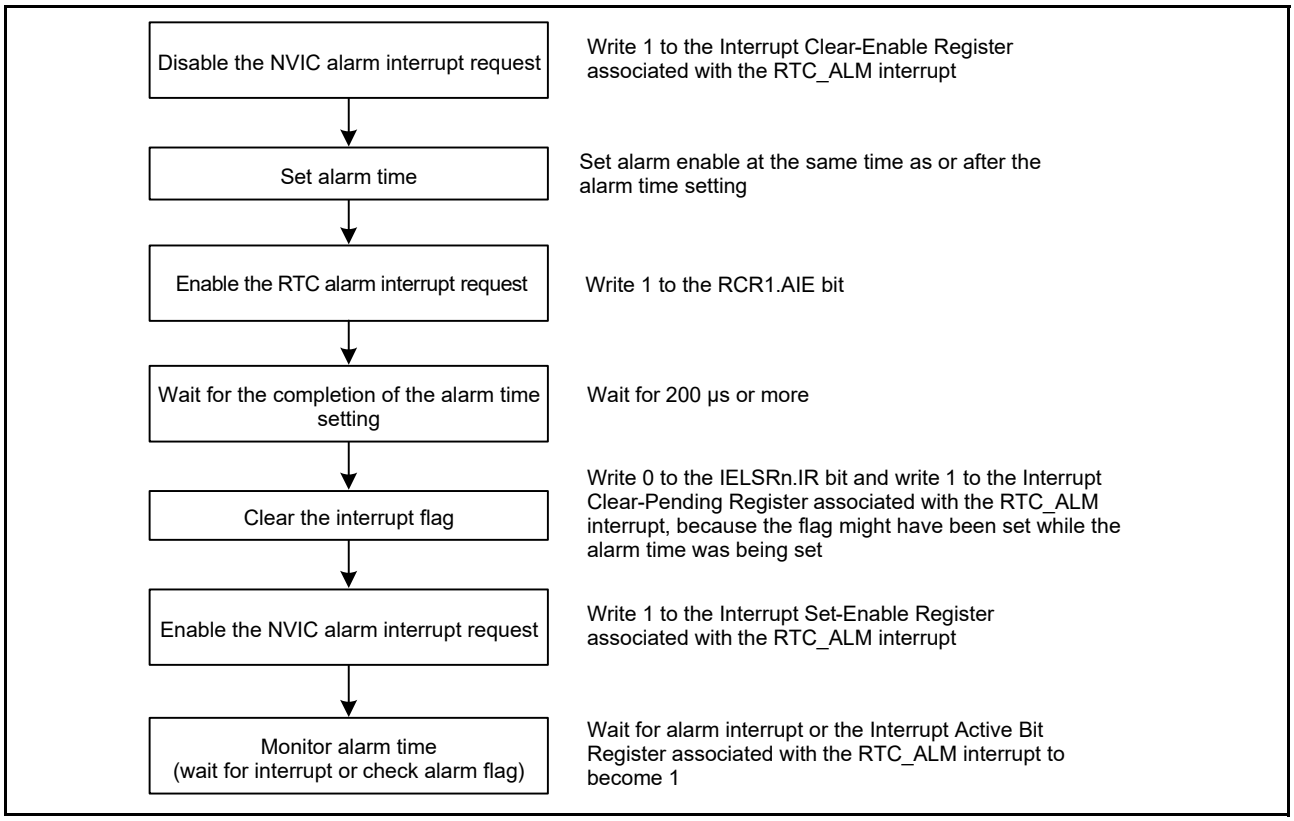


Figure 23.7 Using the alarm function

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 for 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 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 and Interrupt Active Bit 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.

23.3.7 Procedure for Disabling Alarm Interrupt

Figure 23.8 shows the procedure for disabling the enabled alarm interrupt request.

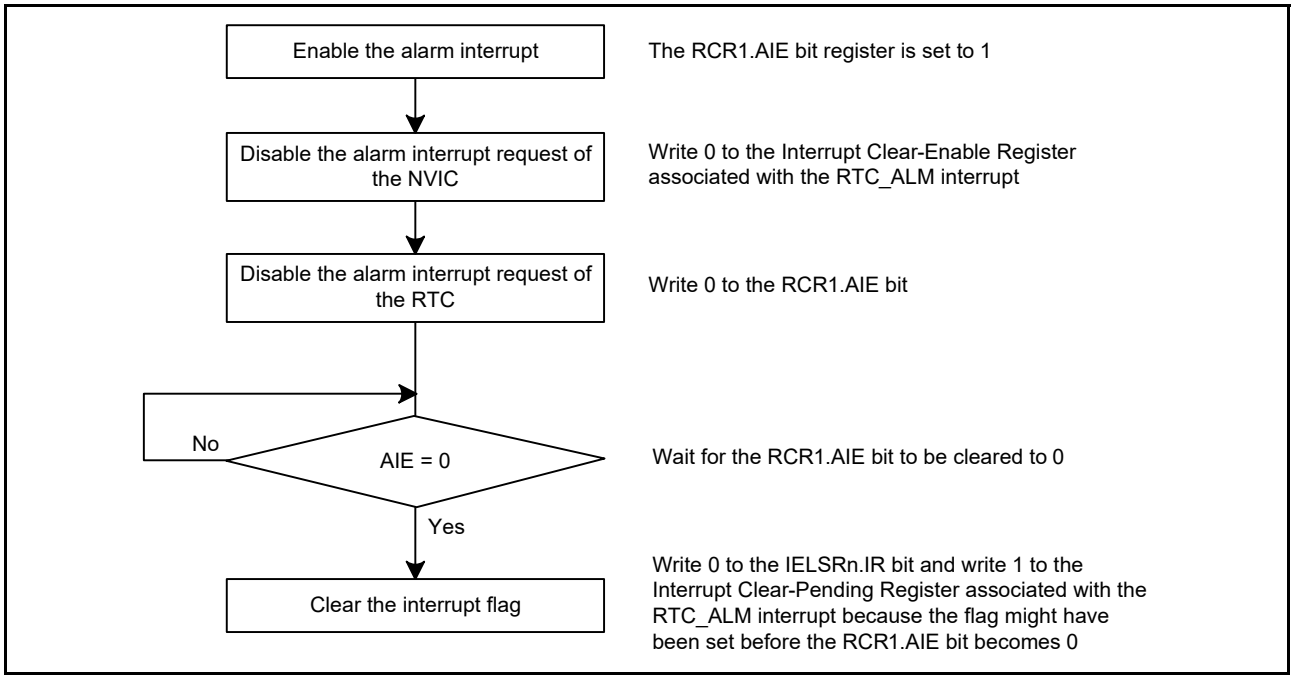


Figure 23.8 Procedure for disabling alarm interrupt request

23.3.8 Time Error Adjustment Function

The time error adjustment function is used to correct errors, running fast or slow, in the time by variation in the precision of oscillation by the sub-clock oscillator. Because 32768 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.AADJE bit to select automatic adjustment or adjustment by software.

23.3.8.1 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 32769 clock cycles. The RTC is meant to run at 32768 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 32766 clock cycles. The RTC is meant to run at 32768 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

When the sub-clock oscillator is running at 32.764 kHz, 1 second elapses on 32764 clock cycles. Because the RTC operates for 32768 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).

23.3.8.2 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 32769 clock cycles. The RTC is meant to run at 32768 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.

23.3.8.3 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 target time. After that, the time is adjusted every time a value is written to the RADJ register.

23.3.8.4 Procedure for stopping adjustment

Stop the adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

23.4 Interrupt Sources

The RTC has three interrupt sources and are listed in Table 23.3.

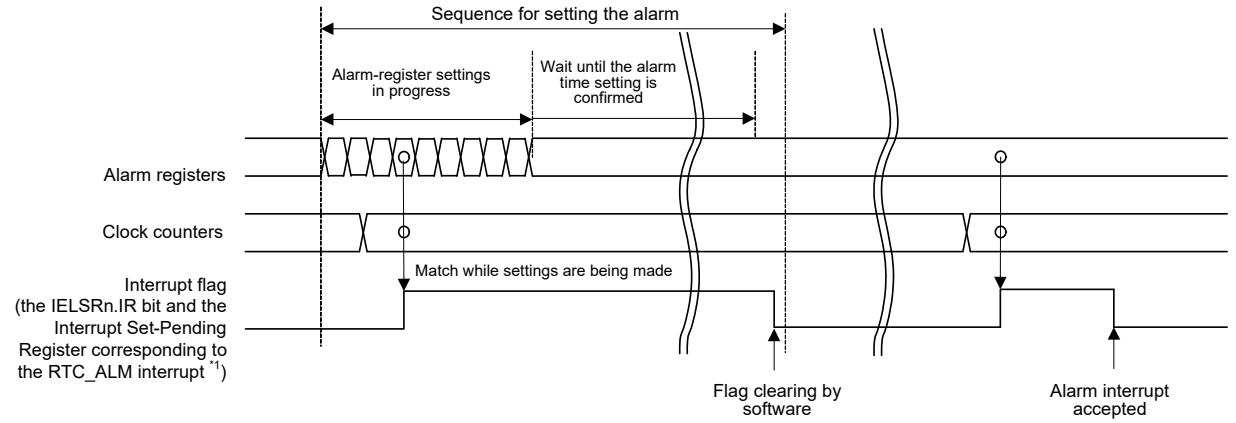
Table 23.3 RTC interrupt sources

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 23.3.6, 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 does not set again until there is another match or the values of the alarm registers are modified again.



Note 1. See section 13, Interrupt Controller Unit (ICU) for details on the associated interrupt vector number.

Figure 23.9 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 in the RCR1.PES[3:0] bits.

(3) Carry interrupt (RTC_CUP)

This interrupt is generated when a carry to the second counter/binary counter 0 occurred or a carry to the R64CNT counter occurred during read access to the 64-Hz counter.

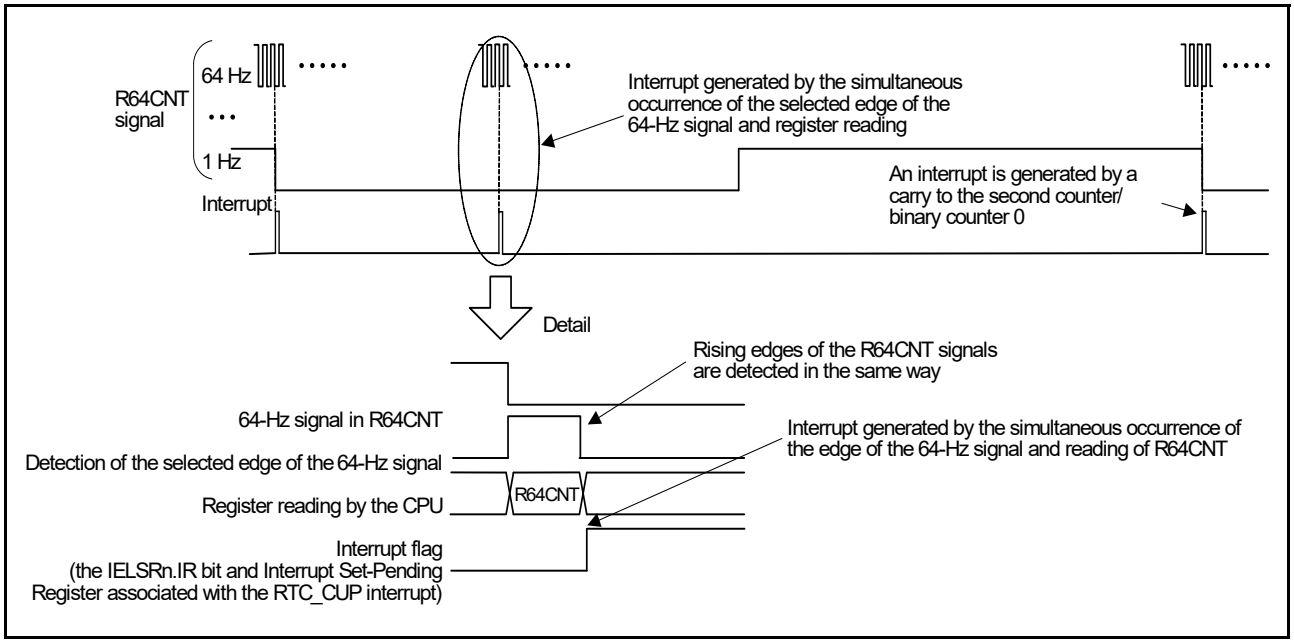


Figure 23.10 Timing for the carry interrupt (RTC_CUP)

23.5 Event Link Output

The RTC generates periodic event output (RTC_PRD) event signals for the 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.

23.5.1 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.

23.6 Usage Notes

23.6.1 Register Writing during Counting

The following registers must not be written to during counting, that is, while the RCR2.START bit is 1:

- RSECCNT/BCNT0
- RMINCNT/BCNT1
- RHRCNT/BCNT2
- RWKCNT/BCNT3
- RDAYCNT
- RMONCNT
- RYRCNT
- RCR1.RTCOS
- RCR2.RTCOE
- RCR2.HR24
- RFRL.

The counter must be stopped before writing to any of the these registers.

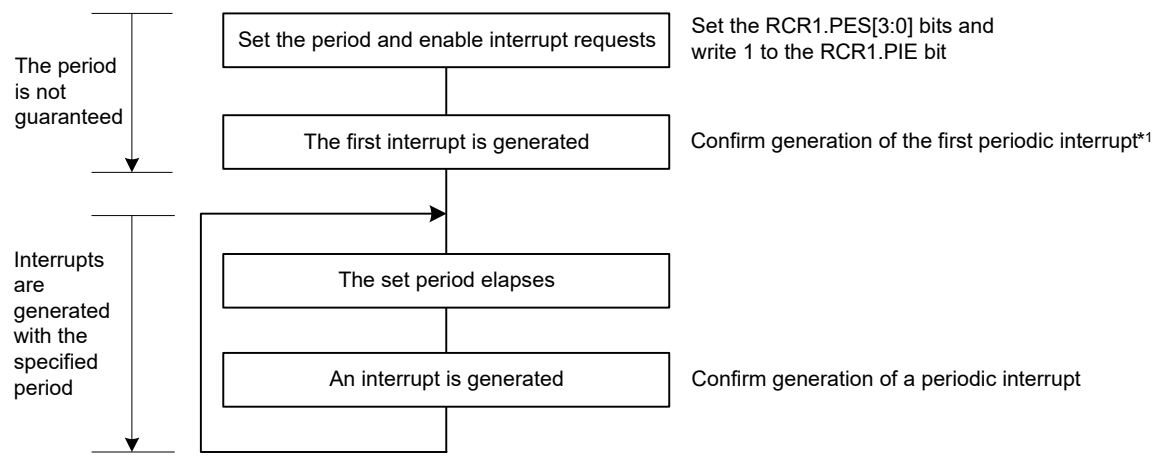
23.6.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 23.11.

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 is based on the adjustment value.



Note 1. When an interrupt generation period changes while the periodic interrupt is used, an interrupt might 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 23.11 Using periodic interrupt function

23.6.3 RTCOUT (1-Hz/64-Hz) Clock Output

Stopping/restarting or resetting the 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.

23.6.4 Transitions to Low Power Modes after Setting Registers

A transition to a low power state during a write to an RTC register might corrupt the value of the register. After setting the register, confirm that the setting is in place before initiating a transition to a low power state.

23.6.5 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 23.3.5, 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 or a period in Software Standby mode state, wait for 1/128 second while the clock is operating (RCR2.START is 1)
- After a reset is generated, write to the RTC register after 6 cycles of the count source clock have elapsed.

23.6.6 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 23.3.1, Outline of Initial Settings of Registers after Power On.

23.6.7 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 23.12.

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.

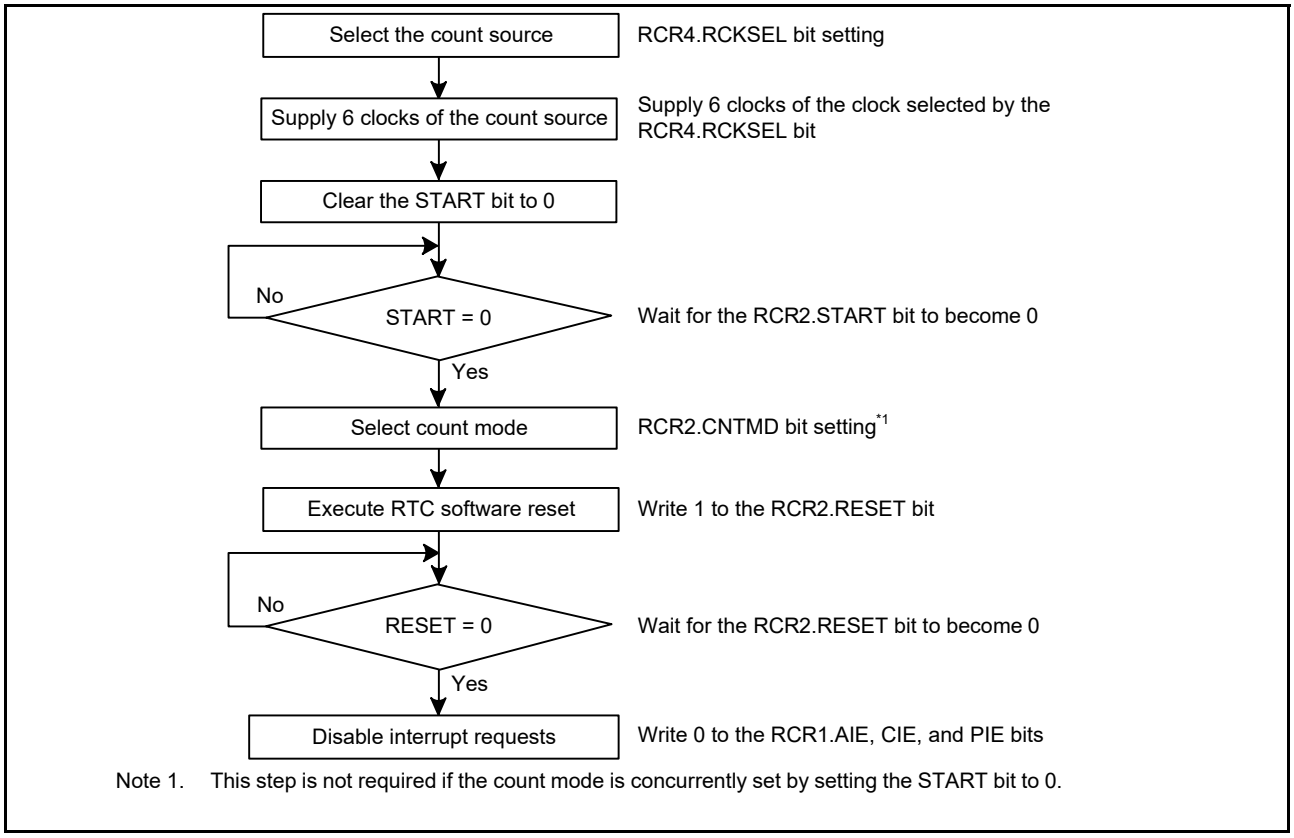


Figure 23.12 Initialization procedure

23.6.8 When Switching Source Clock

When switching a clock source by changing SCKSCR.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.

24. Watchdog Timer (WDT)

24.1 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 24.1 lists the WDT specifications and Figure 24.1 shows a block diagram.

Table 24.1 WDT specifications

Parameter	Specifications
Count source	Peripheral clock (PCLKB)
Clock division ratio	Divide by 4, 64, 128, 512, 2,048, or 8,192
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Auto start mode: Counting automatically starts after a reset, or after an underflow or refresh error occurs • Register start mode: Counting is started with a refresh by writing to the WDTRR register.
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error is generated.
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Watchdog timer Reset sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error).
Non-maskable interrupt/interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error).
Reading the counter value	The down-counter value can be read by the WDTSR register
Event link function (output)	• Down-counter underflow event output • Refresh error event output.
Output signal (internal signal)	• Reset output • Interrupt request output • Sleep mode count stop control output.

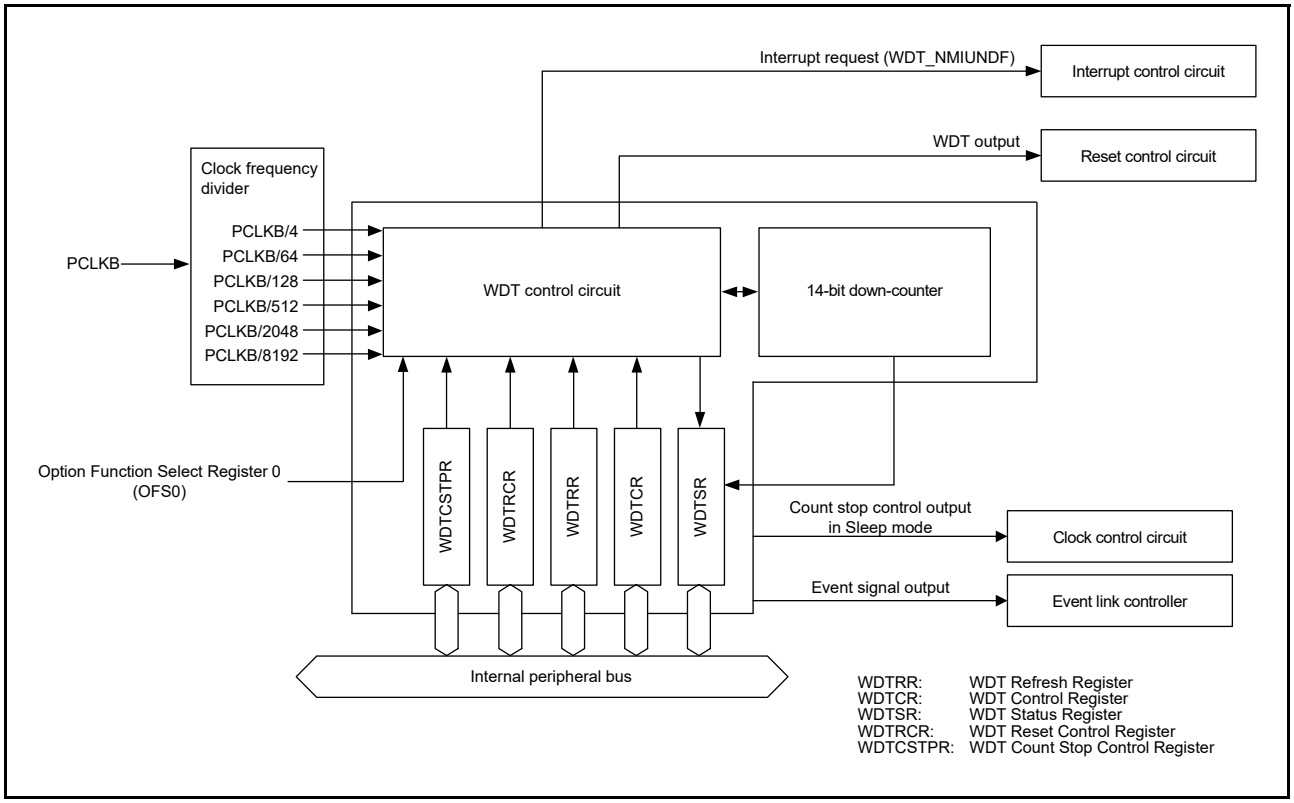


Figure 24.1 WDT block diagram

24.2 Register Descriptions

24.2.1 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 in 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 in 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 24.3.3, Refresh Operation.

24.2.2 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: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh).	R/W
b3, b2	—	Reserved	These bits are read as 0 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 settings 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 24.3.2, Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the settings in the WDTCR register are disabled, and the settings in the Option Function Select Register 0 (OFS0) are enabled. The settings for the WDTCR register can also be made in the OFS0 register. For details, see section 24.3.7, Association between Option Function Select Register 0 (OFS0) and WDT Registers.

TOPS[1:0] bits (Timeout Period Select)

The TOPS[1:0] bits select the timeout period, the period until the down-counter underflows, from 1024, 4096, 8192, and 16384 cycles, taking the divided clock specified in the CKS[3:0] bits as 1 cycle.

After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the number of PCLKB cycles until the counter underflows.

Table 24.2 lists the relationship between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLKB cycles.

Table 24.2 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 4096 and 134217728 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 set to a value greater than or equal to the window start position, the window start position setting is enabled and the window end position is set to 0%.

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

The RPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. The selected window start position should be a value greater than the value for the window end position. If the window start position is set to a value less than or equal to the window end position, the window start position setting is enable and the window end position is set to 0%.

Table 24.3 lists the counter values for the window start and end positions, and Figure 24.2 shows the refresh-permitted period set by the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 24.3 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 position setting ≥ window start position setting, the window end position setting is set to 0%.

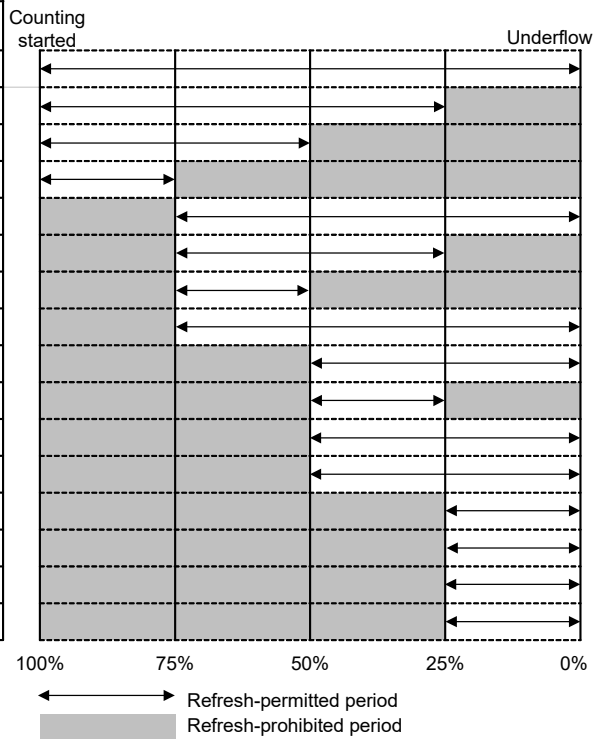


Figure 24.2 RPSS[1:0] and RPES[1:0] bit settings and refresh-permitted period

24.2.3 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 1.

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 this 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.

24.2.4 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	WDT behavior selection 0: Interrupt 1: Reset.	R/W

Some constraints apply to writes to the WDTRCR register. For details, see section 24.3.2, Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the WDTRCR register settings are disabled, and the settings in the Option Function Select register 0 (OFS0) are enabled. The settings for the WDTRCR register can also be made for the OFS0 register. For details, see section 24.3.7, Association between Option Function Select Register 0 (OFS0) and WDT Registers.

24.2.5 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 Sleep mode. Some restrictions apply to writes to the WDTCSTPR register. For details, see section 24.3.2, 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 24.3.7, Association 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.

24.2.6 Option Function Select Register 0 (OFS0)

For information on the OFS0 register, see section 24.3.7, Association between Option Function Select Register 0 (OFS0) and WDT Registers.

24.3 Operation

24.3.1 Count Operation in each Start Mode

The WDT has two start modes:

- Auto start mode, in which counting automatically starts after a release from the reset state
- Register start mode, in which counting starts with a refresh by writing to the register.

In auto start mode, counting automatically starts after a release from the reset state based on 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 a release from the reset state.

Select auto start mode or register start mode by setting the WDT Start Mode Select bit (OFS0.WDTSTRT) in the OFS0 register. When the auto start mode is selected, the settings in the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are disabled while the settings in the OFS0 register are enabled.

When the register start mode is selected, the OFS0 register setting 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.

24.3.1.1 Register start mode

When the WDT Start Mode Select bit (OFS0.WDTSTRT) is 1, register start mode is selected, the OFS0 register setting is invalid, and the WDT control register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are enabled.

After the reset state is released, set the following:

- Clock division ratio in the WDTCR register
- Window start and end positions in the WDTCR register
- Timeout period in the WDTCR register
- Reset output or interrupt request output in the WDTRCR register
- Counter stop control during transitions to Sleep mode in the WDTCSTPR register

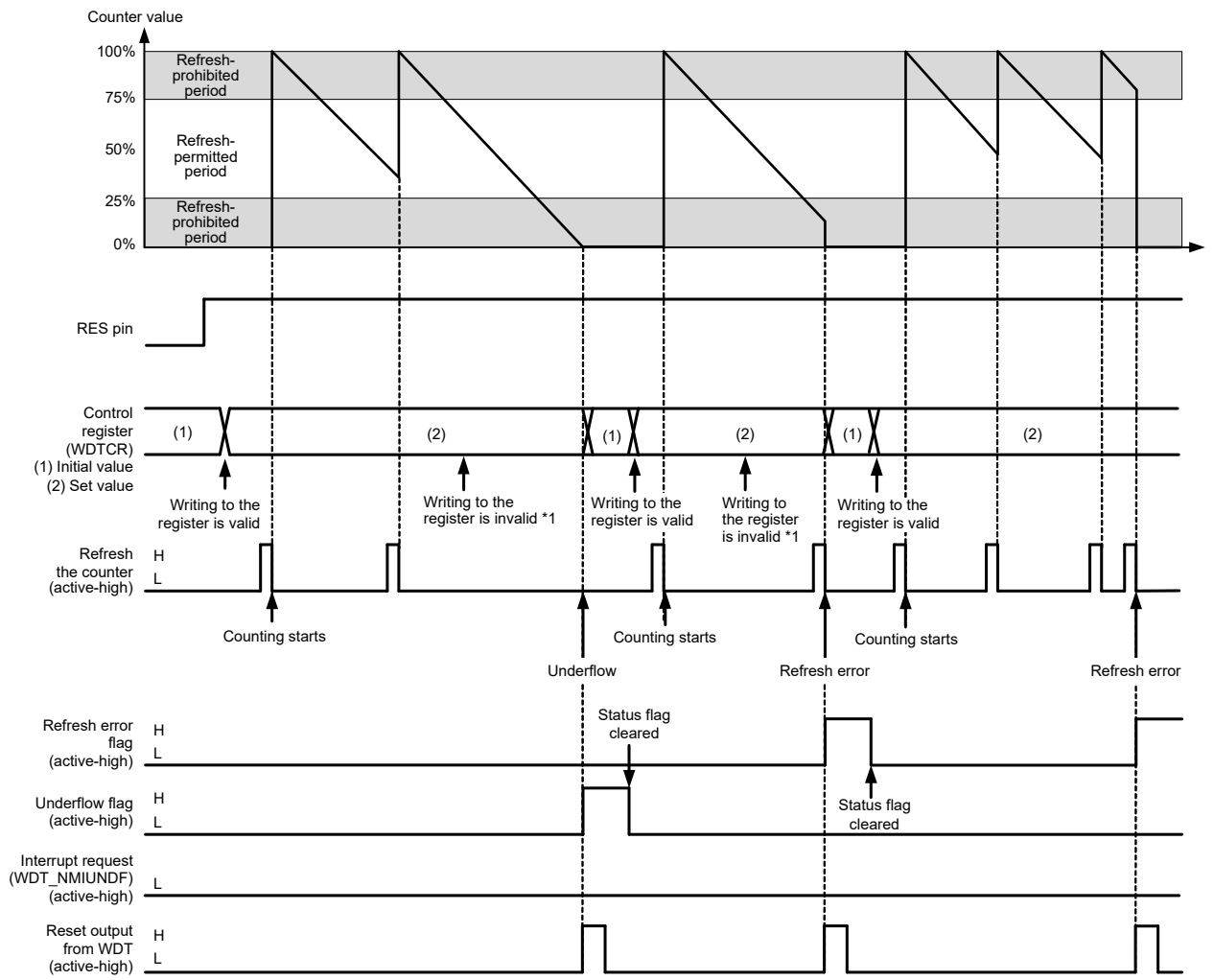
The WDT refresh register (WDTRR) refreshes the down counter.

As a result, the downcount starts at the value set by the timeout period selection bit (WDTCR.TOPS [1: 0]).

After that, as long as the program continues normal operation and the counter is refreshed within the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and down-counting continues. The WDT does not output the reset signal or non-maskable interrupt request/interrupt request as long as the counting continues. However, if the down-counter underflows because the down-counter cannot be refreshed because of a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the WDT outputs the reset signal or non-maskable interrupt request/interrupt request (WDT_NMIUNDF). Reset output or interrupt request output can be selected in the WDT Reset Interrupt Request Select bit (WDTRCR.RSTIRQS). The interrupt enable that initiates NMI can be selected with the WDT Underflow/Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 24.3 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)



Note 1. See section 24.3.2, Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

Figure 24.3 Operation example in register start mode

24.3.1.2 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 on 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]).

After that, as long as the program continues normal operation and the counter is refreshed within the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and down-counting continues. The WDT does not output the reset signal or non-maskable interrupt request/interrupt request as long as the counting continues.

However, if the down-counter underflows because the down-counter cannot be refreshed because of a program runaway, of if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the WDT outputs the reset signal or non-maskable interrupt request/interrupt request (WDT_NMIUNDF).

After the reset signal or non-maskable interrupt request/interrupt request is generated, the counter reloads the timeout period after counting for 1 cycle. The value of the timeout period is set in the down-counter and counting restarts.

Reset output or interrupt request output can be selected in the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS). The interrupt enable that initiates NMI can be selected with the WDT Underflow/Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 24.4 shows an example of operation (non-maskable interrupt) under the following conditions:

- Auto start mode (OFS0.WDTSTRT = 0)
- WDT behavior selection: interrupt (OFS0.WDTRSTIRQS = 0)
- Non-maskable Interrupt: WDT Underflow/Refresh Error Interrupt Enable(NMIER.WDTEN = 1)
- The window start position is 75% (OFS0.WDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.WDTRPES[1:0] = 10b)

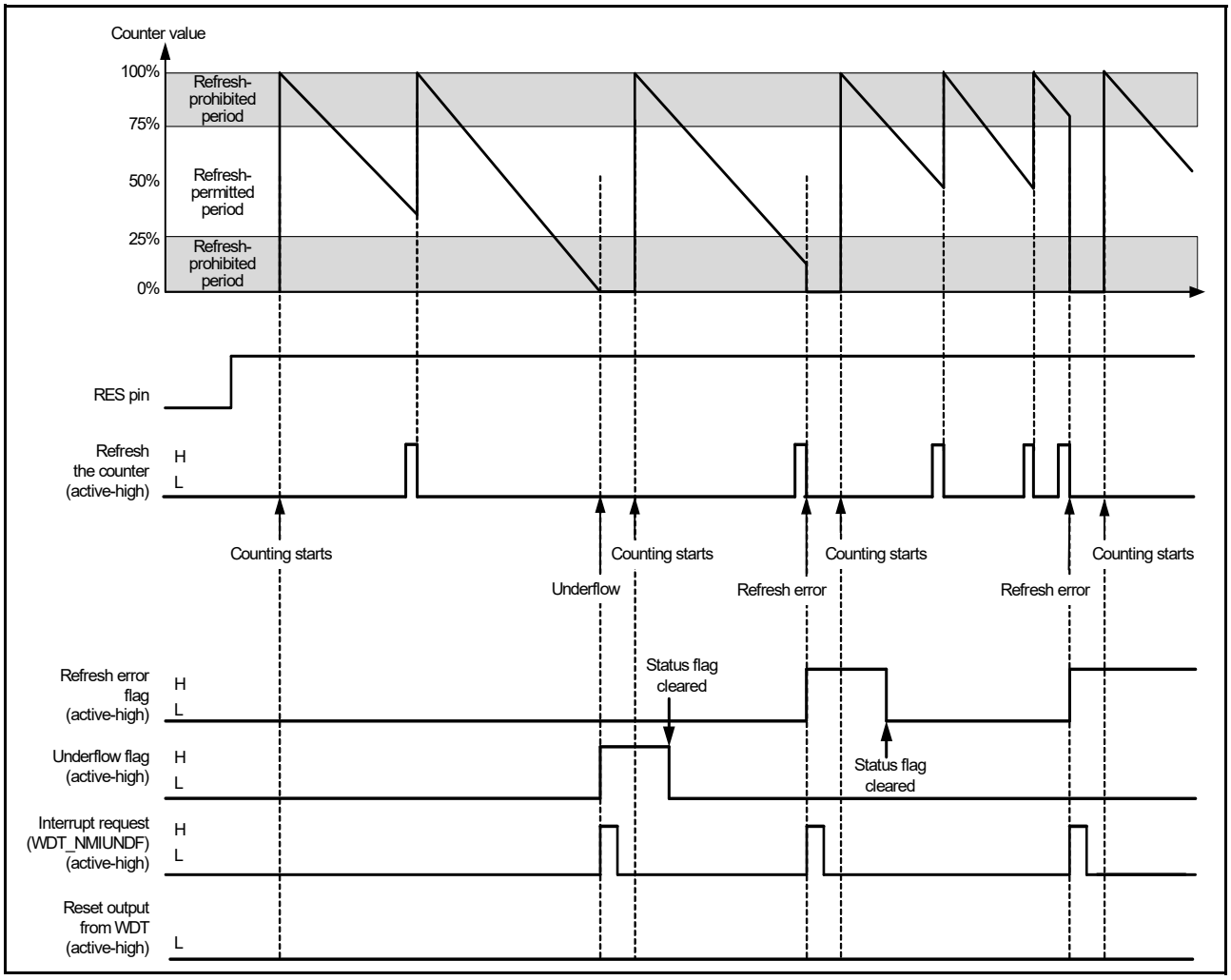


Figure 24.4 Operation example in auto start mode

24.3.2 Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers

Writing to the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), or WDT Count Stop Control Register (WDTCSTPR) is possible once each between the release from the reset state and the first refresh operation.

After a refresh operation (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 24.5 shows control waveforms produced in response to writing to the WDTCR.

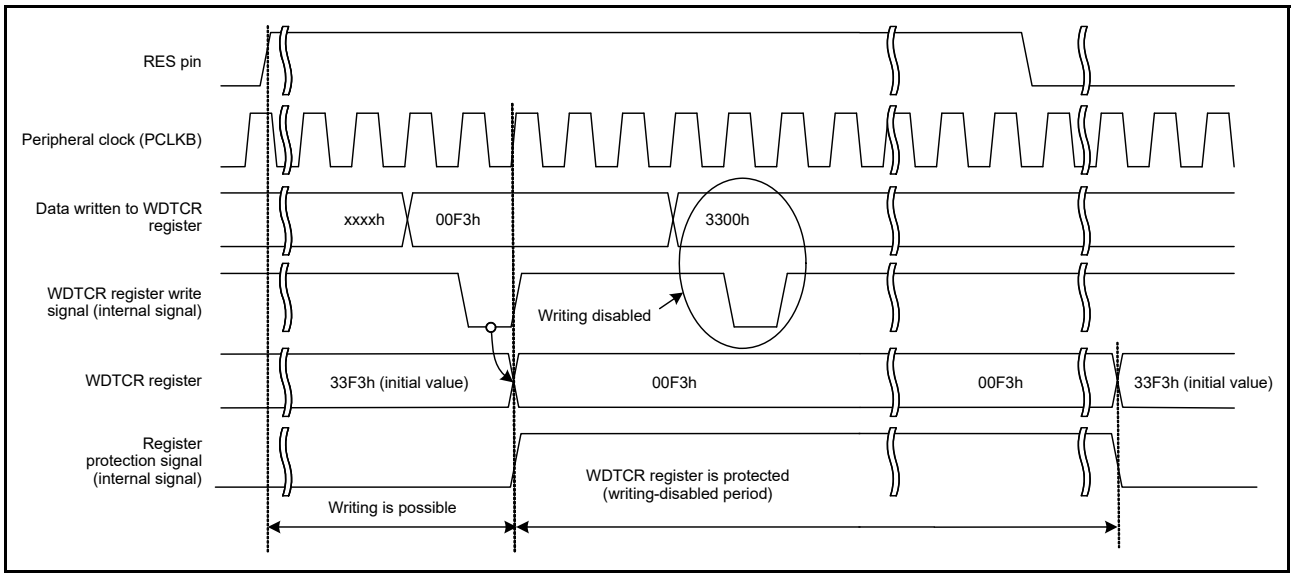


Figure 24.5 Control waveforms produced in response to writes to the WDTCR register

24.3.3 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, the refresh will run successfully 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.

Writing to refresh the counter must be made within the refresh-permitted period, and whether this is done is determined by writing FFh. For this reason, correct refreshing is performed even when 00h is written outside the refresh-permitted period.

[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 not valid when refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h→ AAh (00h and a value other than FFh) → FFh.

After FFh is written to the WDT Refresh Register (WDTRR), refreshing the down-counter requires up to 4 cycles of the signal for counting. To meet this requirement, complete writing FFh to the WDTRR 4 cycle counts before the down-counter underflows.

Figure 24.6 shows the WDT refresh operation waveforms when the clock division ratio is PCLKB/64.

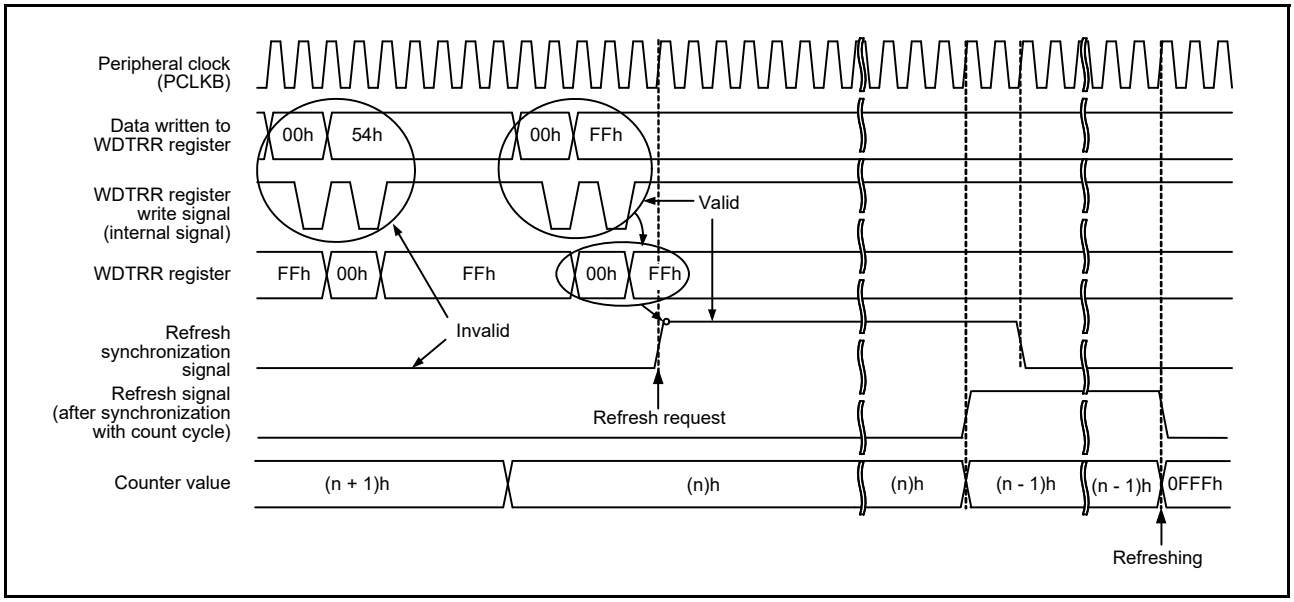


Figure 24.6 WDT refresh operation waveforms when WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b

24.3.4 Reset Output

When the Reset Interrupt Request Select bit (WDTRCR.RSTIRQS) is set to 1 in register start mode, or when the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 1 in auto start mode, a reset signal is output for 1 cycle count when an underflow in the down-counter or a refresh error occurs.

In register start mode, the down-counter is initialized (all bits set to 0) and stopped in that state after output of a reset signal. After the reset state is released and the program is restarted, the counter is set up again and counting down starts again with a refresh. In auto start mode, counting down starts automatically after the reset state is released.

24.3.5 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 signal (WDT_NMIUNDF) 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 13, Interrupt Controller Unit (ICU).

Table 24.4 WDT interrupt sources

Name	Interrupt source	DTC activation
WDT_NMIUNDF	• Down-counter underflow • Refresh error.	Not possible

24.3.6 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 24.7 shows the processing for reading the WDT down-counter value when the clock division ratio is PCLKB/64.

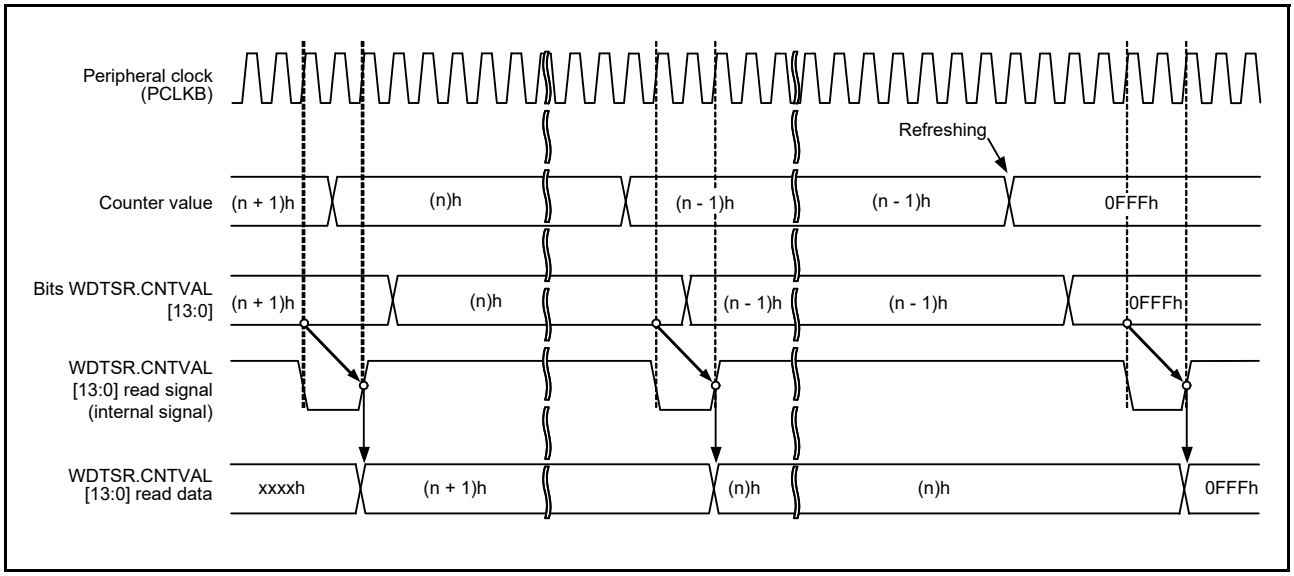


Figure 24.7 Read process for WDT down-counter value when WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b

24.3.7 Association between Option Function Select Register 0 (OFS0) and WDT Registers

Table 24.5 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 24.5 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 interrupt request selection	OFS0.WDTRSTIRQS	WDTRCR.RSTIRQS
Count stop	Sleep mode count stop control	OFS0.WDTSTPCTL	WDTCSTPR.SLCSTP

24.4 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 in the Reset Interrupt Request Select bit (WDTRCR.RSTIRQS) in register start mode or the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in auto start mode. An event signal can also be output when the next interrupt source is generated while the Refresh Error flag (WDTSR.REFEF) or Underflow flag (WDTSR.UNDFF) is 1. For details, see section 17, Event Link Controller (ELC).

24.5 Usage Notes

24.5.1 ICU Event Link Setting Register n (IELSRn) Setting

Setting 18h to ICU Event Link Setting Register n (IELSRn.IELS[7:0]) is prohibited when WDT reset interrupt request selection resets (OFS0.WDTRSTIRQS = 1 or WDTRCR.RSTIRQS = 1), or when enabling the event link operation (ELSRn.ELS[7:0] = 18h).

25. Independent Watchdog Timer (IWDT)

25.1 Overview

The Independent Watchdog Timer (IWDT) is a 14-bit down counter that must be serviced periodically to prevent counter underflow. The IWDT can be used to reset the MCU or to generate a non-maskable interrupt or an underflow interrupt. 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 functions of the IWDT are different from those of the WDT in the following ways:

- The divided IWDT-dedicated clock (IWDTCLK) is used as the count source (not affected by PCLKB)
- IWDT does not support 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 25.1 lists the IWDT specifications and Figure 25.1 shows a block diagram.

Table 25.1 IWDT specifications

Parameter	Specifications
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	Configurable to the following triggers: • Clock frequency division ratio after a reset (OFS0.IWDTCKS[3:0] bits) • Timeout period of the IWDT (OFS0.IWDTTOPS[1:0] bits) • Window start position in the IWDT (OFS0.IWDTRPSS[1:0] bits) • Window end position in the IWDT (OFS0.IWDTRPES[1:0] bits) • Reset output or interrupt request output (OFS0.IWDTRSTIRQS bit) • 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, you must 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.

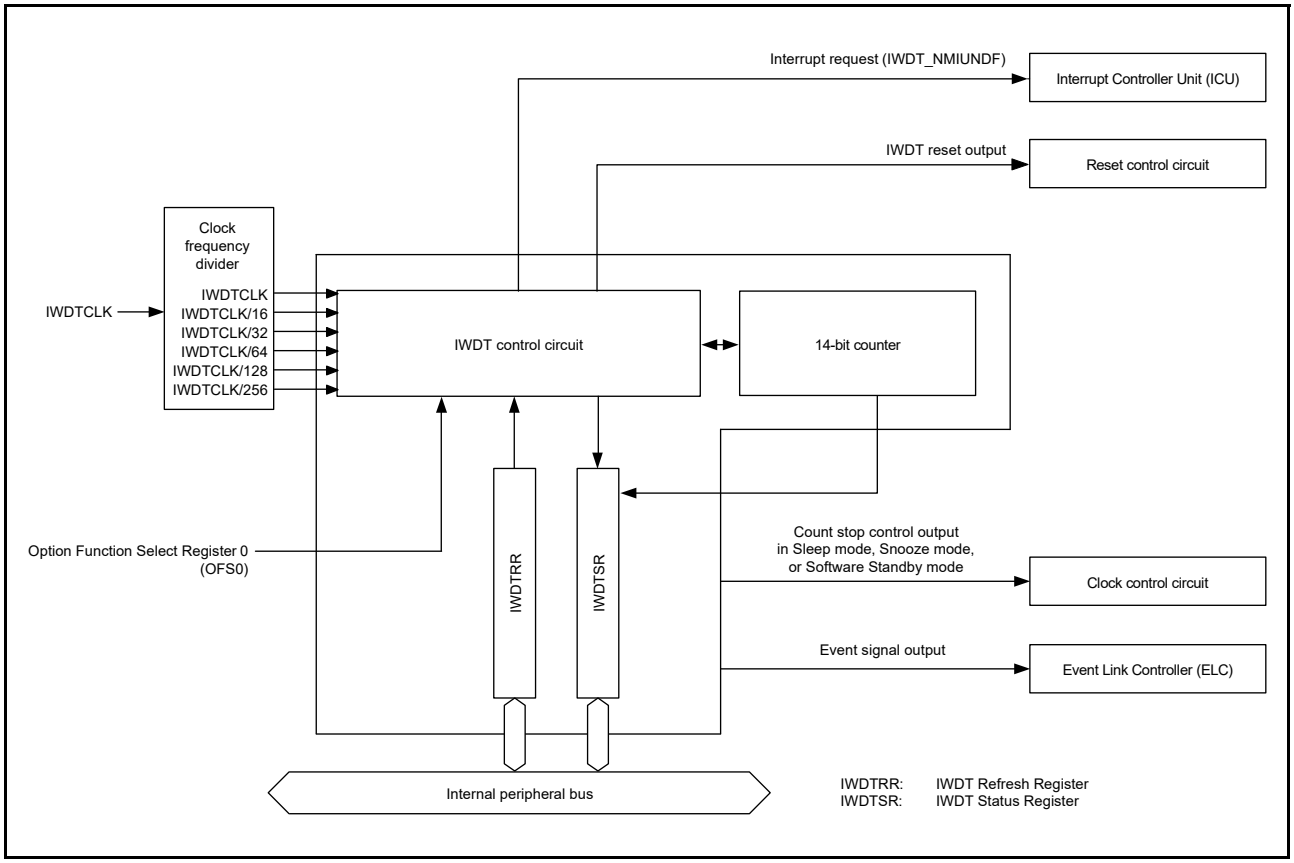


Figure 25.1 IWDT block diagram

25.2 Register Descriptions

25.2.1 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 down-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 in 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 25.3.2, Refresh Operation.

25.2.2 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: 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 (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. 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 + 2) IWDTCLK cycles and 2 PCLKB cycles. In addition, clearing of this flag is ignored for (N + 2) IWDTCLK cycles after an underflow. N is specified in the IWDTCKS[3:0] bits as follows:

- When IWDTCKS[3:0] = 0000b, N = 1
- When IWDTCKS[3:0] = 0010b, N = 16
- When IWDTCKS[3:0] = 0011b, N = 32
- When IWDTCKS[3:0] = 0100b, N = 64
- When IWDTCKS[3:0] = 1111b, N = 128
- When IWDTCKS[3:0] = 0101b, N = 256.

REFEF flag (Refresh Error Flag)

Read the REFEF flag to confirm whether a refresh error occurred. This indicates that a refresh operation was performed during a prohibited period. 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 + 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.

25.2.3 Option Function Select Register 0 (OFS0)

For information on the Option Function Select Register 0 (OFS0), see section 7.2.1, Option Function Select Register 0 (OFS0).

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

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 25.2 lists the relationship between the IWDTCKS[3:0] and IWDTTOPS[1:0] bit settings, the timeout period, and the number of IWDTCLK cycles.

Table 25.2 Timeout period settings

IWDTCKS[3:0] bits				IWDTTOPS[1:0] bits				
b7	b6	b5	b4	b1	b0	Clock division ratio	Timeout period (number of cycles)	IWDTCLK cycles
0	0	0	0	0	0	IWDTCLK	128	128
				0	1		512	512
				1	0		1024	1024
				1	1		2048	2048
0	0	1	0	0	0	IWDTCLK/16	128	2048
				0	1		512	8192
				1	0		1024	16384
				1	1		2048	32768
0	0	1	1	0	0	IWDTCLK/32	128	4096
				0	1		512	16384
				1	0		1024	32768
				1	1		2048	65536
0	1	0	0	0	0	IWDTCLK/64	128	8192
				0	1		512	32768
				1	0		1024	65536
				1	1		2048	131072
1	1	1	1	0	0	IWDTCLK/128	128	16384
				0	1		512	65536
				1	0		1024	131072
				1	1		2048	262144
0	1	0	1	0	0	IWDTCLK/256	128	32768
				0	1		512	131072
				1	0		1024	262144
				1	1		2048	524288

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

The IWDTCKS[3:0] bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from the IWDT-dedicated clock (IWDTCLK) divided by 1, 16, 32, 64, 128, and 256. Combined with the IWDTTOPS[1:0] bit setting, the IWDT can be configured to a count period between 128 and 524288 IWDTCLK cycles.

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

The IWDTRPES[1:0] bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. Set the window end position to a value less than

the window start position (window start position > window end position). If the window end position is set to a value greater than or equal to the window start position, the window start position setting is enabled and the window end position is set to 0%.

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

The IWDTRPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. Set the window start position to a value greater than the window end position. If the window start position is set to a value less than or equal to the window end position, the window start position setting is enable and the window end position is set to 0%.

Table 25.3 lists the counter values for the window start and end positions, and Figure 25.2 shows the refresh-permitted period set in the IWDTRPSS[1:0], IWDTRPES[1:0], and IWDTTOPS[1:0] bits.

Table 25.3 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 position setting ≥ window start position setting, the window end position setting is set to 0%.

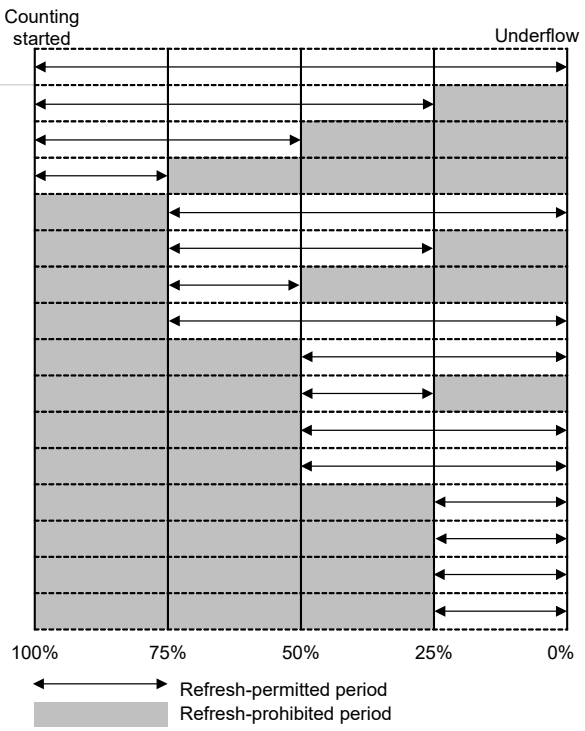


Figure 25.2 IWDTRPSS[1:0] and [IWDTRPES[1:0] bit settings and refresh-permitted period

IWDTRSTIRQS bit (IWDT Reset Interrupt Request Select)

The IWDTRSTIRQS bit specifies the behavior when an underflow or a refresh error occurred. Set this bit to 1 to select reset output. Set this bit to 0 to select interrupt.

IWDTSTPCTL bit (IWDT Stop Control)

The IWDTSTPCTL bit selects whether to stop counting at transition to Sleep, Snooze, or Software Standby mode.

25.3 Operation

25.3.1 Auto Start Mode

When the IWDT Start Mode Select bit (OFS0.IWDTSTRT) in the Option Function Select Register 0 is 0, auto start mode is selected, otherwise the IWDT is disabled.

Within the reset state, the following values 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 set in the IWDT Timeout Period Select bits (OFS0.IWDTTOPS[1:0]).

After that, as long as the program continues normal operation and the counter is refreshed within the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and down-counting continues. The IWDT does not output the reset signal or non-maskable interrupt requests/interrupt request as long as the counting continues. However, if the down-counter underflows because the down-counter cannot be refreshed because of a program runaway, of if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs the reset signal or non-maskable interrupt request/interrupt request (IWDT_NMIUNDF).

After the reset signal or non-maskable interrupt request/interrupt request is generated, the counter reloads the timeout period after counting for 1 cycle. The value of the timeout period is set in the down-counter and counting restarts. Reset output or interrupt request output can be selected in the IWDT Reset Interrupt Request Select bit (OFS0.IWDTRSTIRQS). The Interrupt enable that initiates NMI can be selected with the IWDT Underflow/Refresh Error Interrupt Enable bit (NMIER.IWDTEN).

Figure 25.3 shows an example of operation under the following conditions:

- Auto start mode (OFS0.IWDTSTRT = 0)
- IWDT behavior selection : interrupt (OFS0.IWDTRSTIRQS = 0)
- Non-maskable Interrupt: IWDT Underflow / Refresh Error Interrupt Enabled (NMIER.IWDTEN = 1)
- The window start position is 75% (OFS0.IWDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.IWDTRPES[1:0] = 10b).

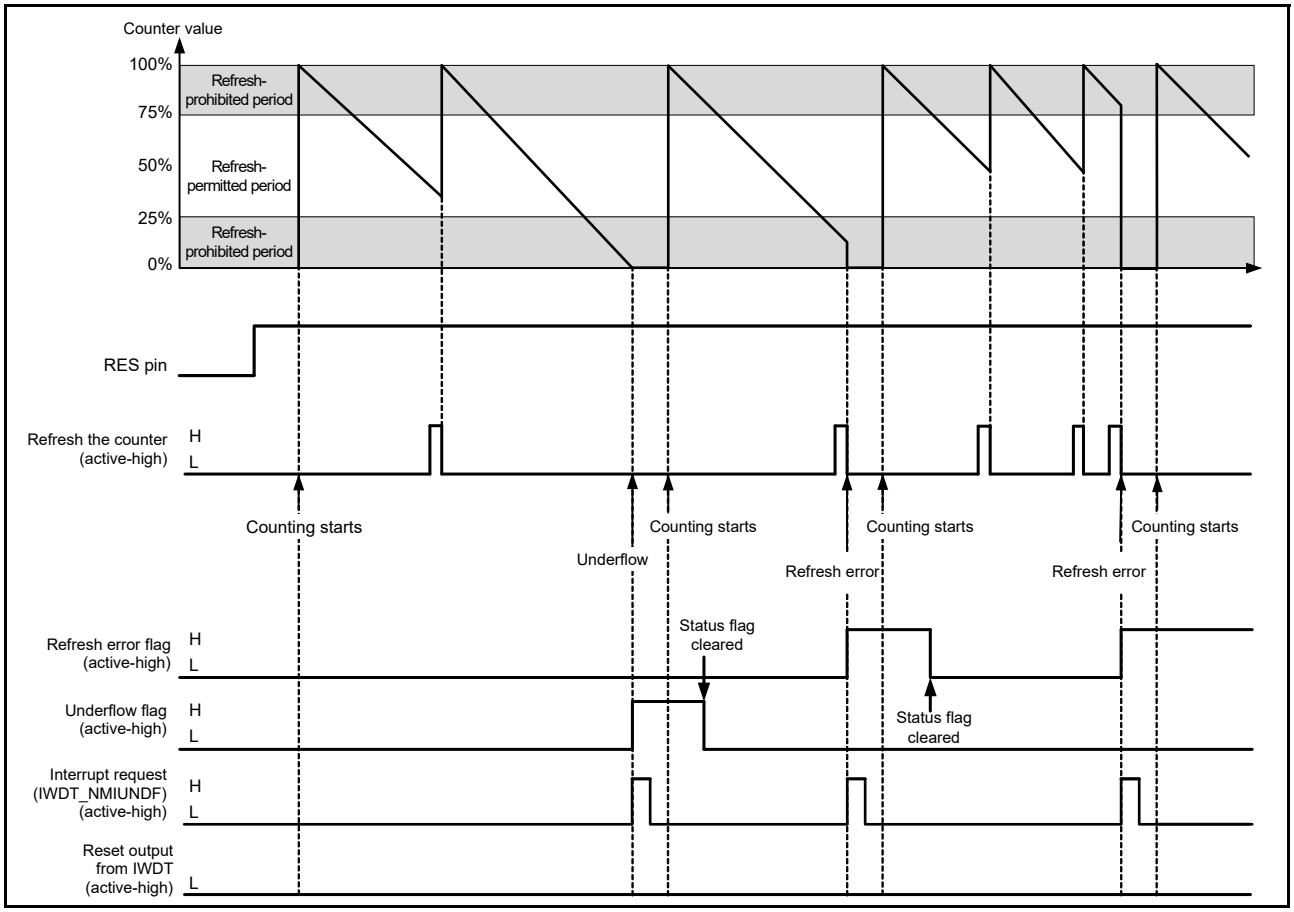


Figure 25.3 Operation example in auto start mode

25.3.2 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, the refresh will run successfully by writing 00h and FFh to the IWDTRR register.

When writing is done in the order of 00h (first time) → 00h (second time), and if FFh is written after that, the writing order 00h → FFh is satisfied. Writing 00h ((n - 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 00h → FFh.

Correct refreshing is also performed regardless of whether 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 for refreshing the counter]

- 00h → FFh
- 00h ((n - 1)th time) → 00h (nth time) → FFh
- 00h → access to another register or read from IWDTRR → FFh.

[Example write sequences that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h → AAh (00h and a value other than FFh) → FFh.

When 00h is written to IWDTRR outside the refresh-permitted period, if FFh is written to IWDTRR in the refresh-permitted period, the writing sequence is valid and refreshing completes.

After FFh is written to the IWDTRR, refreshing the counter requires up to 4 cycles of the signal for counting (the IWDT-Dedicated Clock Frequency Division Ratio Select bits (OFS0.IWDTCKS[3:0]) to determine how many cycles of the IWDT-dedicated clock (IWDTCLK) make up 1 cycle for counting. To meet this requirement, writing 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 in the counter bits (IWDTSR.CNTVAL[13:0]).

[Example refreshing timings]

- When the window start position is set to 01FFh, even if 00h is written to IWDTRR before 01FFh is reached (at 0202h, for example), refreshing occurs if FFh is written to IWDTRR after the value of the IWDTSR.CNTVAL[13:0] bits reaches 01FFh
- When the window end position is set to 01FFh, refreshing occurs if 0203h (4 count cycles before 01FFh) 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 25.4 shows the IWDT refresh-operation waveforms when PCLKB > IWDTCLK and the clock division ratio is IWDTCLK.

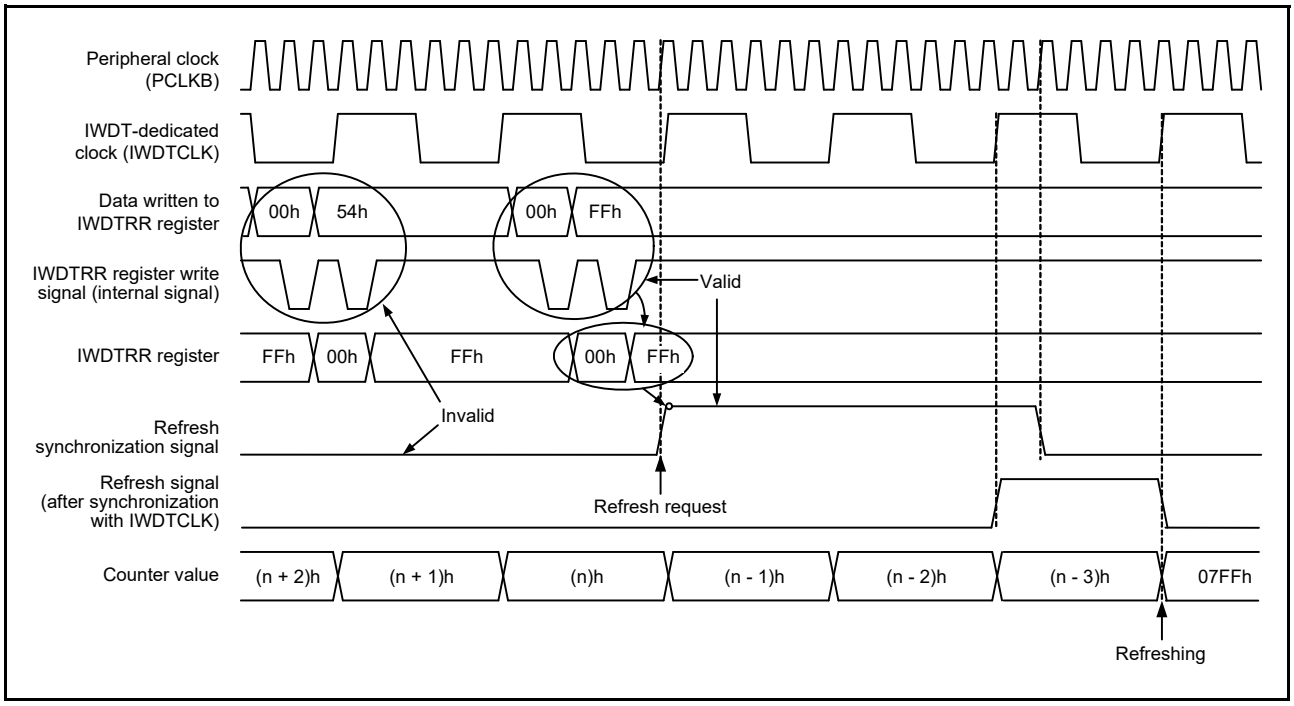


Figure 25.4 IWDT refresh operation waveforms when OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOPS[1:0] = 11b

25.3.3 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. 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 on 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.

25.3.4 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.

25.3.5 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 13, Interrupt Controller Unit (ICU).

Table 25.4 IWDT interrupt source

Name	Interrupt source	DTC activation
IWDT_NMIUNDF	• Down-counter underflow • Refresh error	Not possible

25.3.6 Reading the Down-counter Value

As the counter is a IWDT-dedicated clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral clock (PCLKB) and stores it in the down-counter value bits (IWDTSR.CNTVAL[13:0]) in the IWDT Status Register. Check these bits to obtain the counter value indirectly.

Reading the counter value requires multiple PCLKB clock cycles (up to 4 clock cycles), and the read counter value might differ from the actual counter value by a value of one count.

Figure 25.5 shows the processing for reading the IWDT counter value when PCLKB > IWDTCLK and the clock division ratio is IWDTCLK.

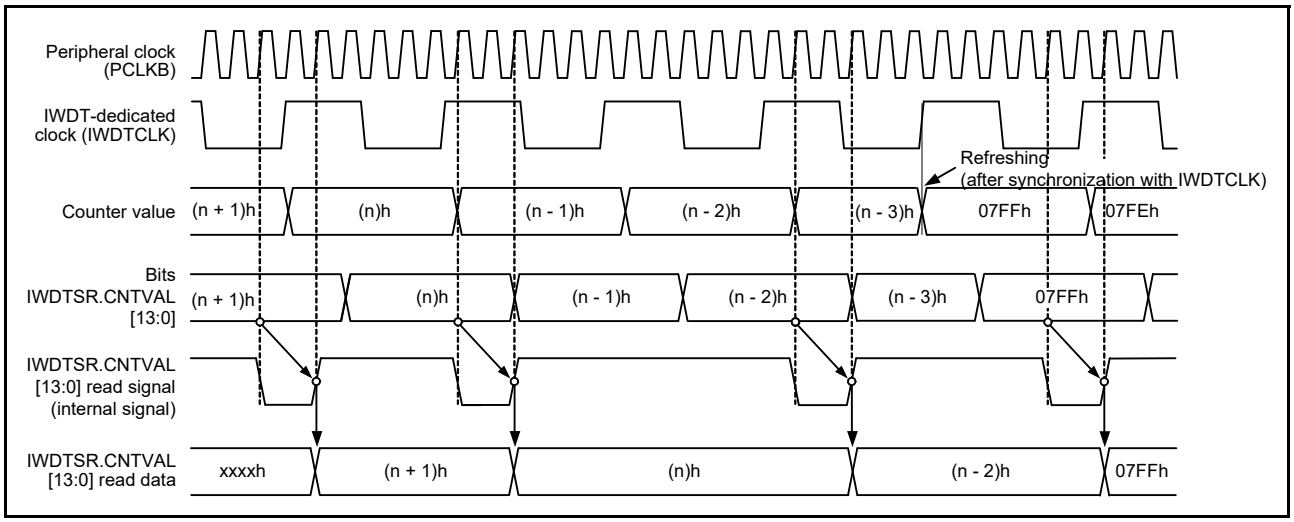


Figure 25.5 Processing for reading IWDT counter value when OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOPS[1:0] = 11b

25.4 Link Operation by the 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 ELC. The event signal is output by the counter underflow or refresh error.

An event signal is output regardless of the setting in the OFS0.WDTRSTIRQS bit. An event signal can also be output when the next interrupt source is generated while the Refresh Error flag (IWDTSR.REFEF) or Underflow flag (IWDTSR.UNDFF) is 1. For details, see section 17, Event Link Controller (ELC).

25.5 Usage Notes

25.5.1 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.

25.5.2 Restrictions on the Clock Division Ratio Setting

Satisfy the following required frequency of the peripheral module clock (PCLKB):

PCLKB ≥ 4 × (the frequency of the count clock source after division).

26. USB 2.0 Full-Speed Module (USBFS)

26.1 Overview

The MCU provides a USB 2.0 Full-Speed module (USBFS) that operates as a device controller compliant with the Universal Serial Bus (USB) specification revision 2.0. The module supports full-speed and low-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 a FIFO buffer for data transfers, providing a maximum of five pipes. Any endpoint number can be assigned to pipes 4 to 7, based on the peripheral devices or the communication requirements for your system.

The MCU supports revision 1.2 of the Battery Charging specification. Because the MCU can be powered at 5 V, the USB LDO regulator provides the internal USB transceiver power supply at 3.3 V.

Table 26.1 lists the USBFS specifications, Figure 26.1 shows a block diagram, and Table 26.2 lists the I/O pins.

Table 26.1 USBFS specifications

Parameter	Specifications
Features	• USB Device Controller (UDC) and USB 2.0 transceiver supporting device controller (one channel) • Self-power mode or bus power mode can be selected • Revision 1.2 of Battery Charging specification is supported • The USB LDO regulator is used to power the internal USB transceiver.
	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 function.
Communication data transfer type	• Control transfer • Bulk transfer • Interrupt transfer.
Pipe configuration	• FIFO buffer for USB communication • Up to five pipes can be selected, including the Default Control Pipe (DCP) • Pipes 4 to 7 can be assigned to any endpoint number.
	Transfer conditions that can be set for each pipe: • Pipe 0: Control transfer with 64-byte single buffer • Pipes 4 and 5: Bulk transfer with 64-byte double buffer • Pipes 6 and 7: Interrupt transfer with 64-byte single buffer.
Other features	• Reception end function using transaction count • Function that changes the BRDY interrupt event notification timing (BFRE) • NAK setting function for response PID generated on transfer end (SHTNAK) • On-chip pull-up and pull-down resistors of 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

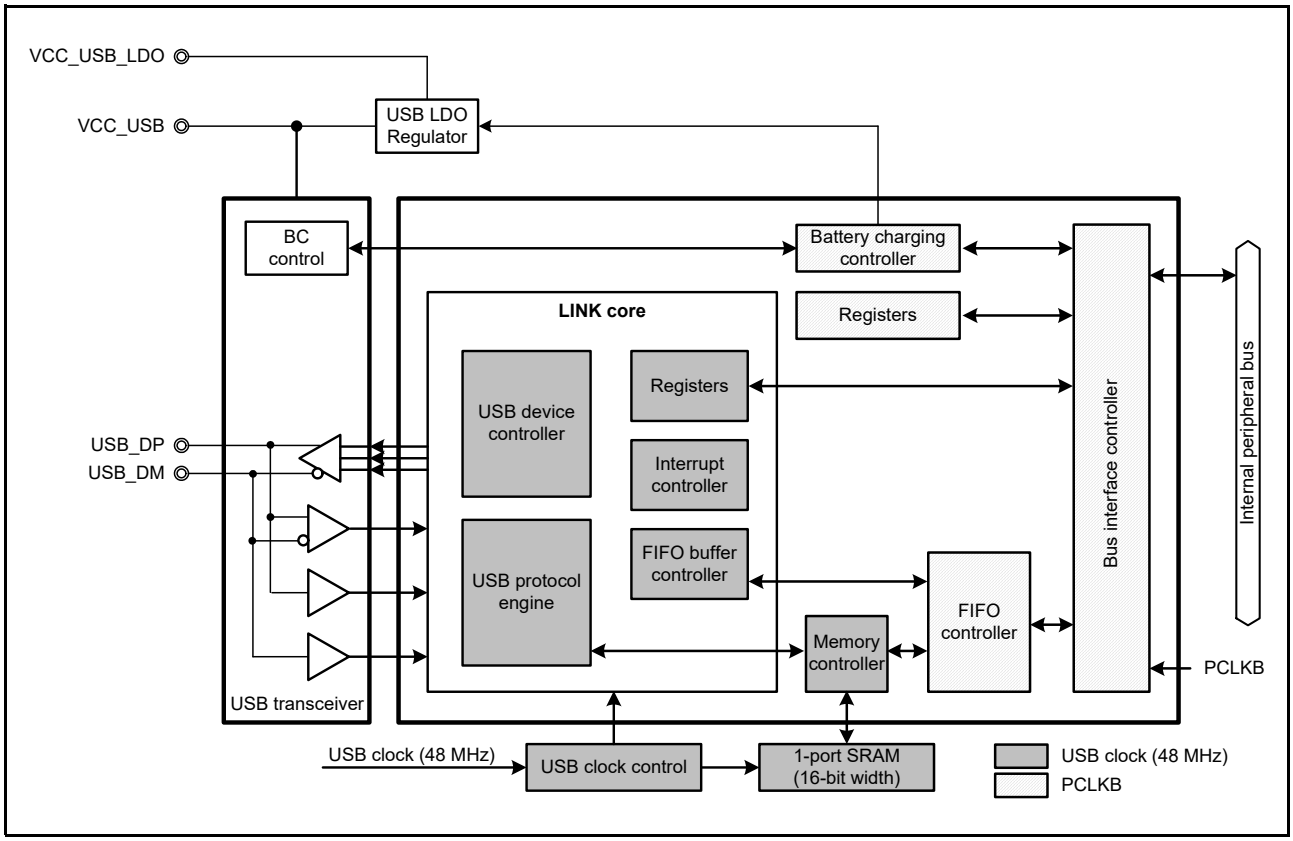


Figure 26.1 USBFS block diagram

Table 26.2 USBFS pin configuration

Port	Pin name	I/O	Function
USBFS	USB_DP	I/O	D+ I/O pin for the on-chip USB transceiver. Must be connected to the D+ data line of the USB bus.
	USB_DM	I/O	D- I/O pin for the USB on-chip USB transceiver. Must be connected to the D- data line of the USB bus.
	USB_VBUS	Input	USB cable connection monitor pin. Must be connected to VBUS signal on the USB bus. The VBUS pin status (connected or disconnected) can be detected.*1
Common	VCC_USB	I/O	Input: Power supply for USB transceiver. Output: USB LDO regulator output pin. This pin must be connected to an external capacitor.
	VCC_USB_LDO	Input	Power supply pin for USB LDO regulator
	VSS_USB	Input	USB ground pin

Note 1. P407 is 5-V tolerant.

26.2 Register Descriptions

26.2.1 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	—	—	—	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
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	CNEN	CNEN Single-Ended Receiver Enable	0: Single-ended receiver operation disabled 1: Single-ended receiver operation 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 26.3. Only change this bit while the SCKE bit is 1.

Table 26.3 Registers initialized by writing 0 to SYSCFG.USBE bit

Register	Bit
SYSSTS0	LNST[1:0]
DVSTCTR0	RHST[2:0]
INTSTS0	DVSQ[2:0]
USBREQ	BREQUEST[7:0], BMREQUESTTYPE[7:0]
USBVAL	WVALUE[15:0]
USBINDX	WINDEX[15:0]
USBLENG	WLENTUH[15:0]

DMRPU bit (D- Line Resistor Control)

The DMRPU bit enables or disables pulling up the D- line.

When the DMRPU bit is set to 1, the bit forces a pull-up of 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.

DPRPU bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line.

When the DPRPU bit is set to 1, the bit forces a pull-up of 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.

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 the 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 the 48-MHz clock supply to the USBFS.

When this bit is 0, only SYSCFG can be read from and written to. No other USBFS-related registers can be read from or written to.

26.2.2 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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LNST[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	LNST[1:0]	USB Data Line Status Monitor	Indicates the status of the USB data lines, see Table 26.4.	R
b15 to b2	—	Reserved	These bits are read as 0	R

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 26.4.

Read the LNST[1:0] bits after connection processing (SYSCFG.DPRPU bit = 1).

Table 26.4 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

26.2.3 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
	—	—	—	—	—	—	—	WKUP	—	—	—	—	—	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	b2 b1 b0 0 0 0: Communication speed indeterminate 0 0 1: USB bus reset in progress 0 1 0: USB bus reset in progress or full-speed connection.	R

Bit	Symbol	Bit name	Description	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	WKUP	Wakeup Output	0: Remote wakeup signal not output 1: Remote wakeup signal output.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RHST[2:0] bits (USB Bus Reset Status)

The RHST[2:0] bits indicate the status of the USB bus reset.

When 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 bit is 1, and a DVST interrupt is generated.

WKUP bit (Wakeup Output)

The WKUP bit enables or disables remote wakeup signals (resume signals) to the USB bus.

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 USBFS 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] = 1xxb) and the USB host enables the remote wakeup signal. Do not stop the internal clock when this bit is 1, even in the suspended state (SYSCFG.SCKE bit is 1).

26.2.4 CFIFO Port Register (CFIFO/CFIFOL)

(1) When the MBW bit is 1

Address(es): USBFS.CFIFO 4009 0014h

	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

	b7	b6	b5	b4	b3	b2	b1	b0
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 setting (CFIFOSEL.MBW) and BIGEND setting (CFIFOSEL.BIGEND). See Table 26.5 and Table 26.6.

CFIFO is configured with:

- A port register (CFIFO) that handles reading of data from the FIFO buffer and writing of data to the FIFO buffer
- A port select register (CFIFOSEL) that selects the pipe assigned to the FIFO port
- A port control register (CFIFOCTR).

CFIFO has the following constraints:

- Access to the FIFO buffer for DCP control transfers is through the CFIFO port
- 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.
- The FIFO buffer cannot be accessed by the DTC.

FIFOPORT[15:0] bits (FIFO Port)

When the FIFOPORT[15:0] bit is accessed, the USBFS reads the receive data from the FIFO buffer or writes the transmit data to the FIFO buffer.

The CFIFO Port Register can be accessed only when the FRDY bit in the CFIFO Port Control Register (CFIFOCTR) is 1. The valid bits in the CFIFO Port Register depend on the MBW and BIGEND settings in the CFIFO Port Select Register (CFIFOSEL). See Table 26.5 and Table 26.6.

Table 26.5 Endian operation in 16-bit access

CFIFOSEL.BIGEND bit	Bits [15:8]	Bits [7:0]
0	N + 1 data	N + 0 data
1	N + 0 data	N + 1 data

Table 26.6 Endian operation in 8-bit access

CFIFOSEL.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.

26.2.5 CFIFO Port Select Register (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 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7. 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 selected 1: Writing to the buffer memory 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

Bit	Symbol	Bit name	Description	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 are cleared when all receive data is read from the CFIFO. In double buffer mode, the DTLN[8:0] bit value is cleared when all data is read from only a single plane. 1: The DTLN[8:0] bits are decremented each time the receive data is read from the CFIFO.	R/W

Note 1. Only 0 can be read.

CURPIPE[3:0] bits (CFIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number to use for reading or writing data through the CFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process.

During FIFO buffer access, the current access setting 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 a selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process. Set this bit and the CURPIPE[3:0] bits simultaneously.

MBW bit (CFIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the CFIFO port.

When the selected pipe is receiving, set the CURPIPE[3:0] 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 that is set in MBW.

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 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 mode, 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 unit 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[8:0] bits.

26.2.6 CFIFO Port Control Register (CFIFOCTR)

Address(es): USBFS.CFIFOCTR 4009 0022h

	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 receive data length. These bits indicate different values depending on the RCNT bit setting in the CFIFO Port Select Register. For details, see the description of the DTLN[8:0] bits.	R
b12 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FRDY	FIFO Port Ready	0: FIFO port access disabled 1: FIFO port access enabled.	R
b14	BCLR	CPU Buffer Clear	0: No operation 1: FIFO buffer cleared in the CPU.	R/W*1
b15	BVAL	Buffer Memory Valid Flag	0: Invalid (writing 0 has no effect) 1: Writing ended.	R/W

Note 1. Only 0 can be read.

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 CFIFOSEL.RCNT bit value as follows:

- RCNT = 0:
  The USBFS sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU has read all the received data from a single FIFO buffer plane.
  While the PIPECFG.BFRE = 1, the USBFS retains the length of the receive data until the BCLR bit is set to 1, even after all the data is read.
- RCNT = 1:
  The USBFS decrements the value indicated in the DTLN[8:0] bits each time data is read from the FIFO buffer. The value is decremented by 1 when the MBW = 0, and by 2 when the MBW 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 of the data is read from the other plane, the USBFS sets these bits to indicate the length of the receive data in the former plane when all of the data is read from the latter plane.

FRDY bit (FIFO Port Ready)

The FRDY bit indicates whether the FIFO port can be accessed by the CPU.

In the following cases, the USBFS sets the FRDY bit to 1 but data cannot be read by the FIFO port because there is no data to be read:

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty
- A short packet is received and the data is completely read while the PIPECFG.BFRE = 1.

In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

BCLR bit (CPU Buffer Clear)

Set the BCLR bit to 1 to clear the FIFO buffer 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 CFIFO 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 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.

26.2.7 Interrupt Enable Register 0 (INTENB0)

Address(es): USBFS.INTENB0 4009 0030h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BRDYE	Buffer Ready Interrupt Enable	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b9	NRDYE	Buffer Not Ready Response Interrupt Enable	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b10	BEMPE	Buffer Empty Interrupt Enable	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b11	CTRE	Control Transfer Stage Transition Interrupt Enable	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b12	DVSE	Device State Transition Interrupt Enable	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b13	SOFE	Frame Number Update Interrupt Enable	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b14	RSME	Resume Interrupt Enable	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b15	VBSE	VBUS Interrupt Enable	0: Interrupt request disabled 1: Interrupt request enabled.	R/W

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.

26.2.8 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
	—	—	—	—	—	—	—	—	PIPE7BRDYE	PIPE6BRDYE	PIPE5BRDYE	PIPE4BRDYE	—	—	—	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 request disabled 1: Interrupt request enabled.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PIPE4BRDYE	BRDY Interrupt Enable for Pipe 4	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b5	PIPE5BRDYE	BRDY Interrupt Enable for Pipe 5	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b6	PIPE6BRDYE	BRDY Interrupt Enable for Pipe 6	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b7	PIPE7BRDYE	BRDY Interrupt Enable for Pipe 7	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b15 to b8	—	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 is set to 1 and the associated PIPEnBRDYE bit (n = 0, 4 to 7) 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.

26.2.9 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
	—	—	—	—	—	—	—	—	PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	—	—	—	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 request disabled 1: Interrupt request enabled.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PIPE4NRDYE	NRDY Interrupt Enable for Pipe 4	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b5	PIPE5NRDYE	NRDY Interrupt Enable for Pipe 5	0: Interrupt request disabled 1: Interrupt request enabled.	R/W

Bit	Symbol	Bit name	Description	R/W
b6	PIPE6NRDYE	NRDY Interrupt Enable for Pipe 6	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b7	PIPE7NRDYE	NRDY Interrupt Enable for Pipe 7	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b15 to b8	—	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 a 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, 4 to 7) 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.

26.2.10 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
	—	—	—	—	—	—	—	—	PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	—	—	—	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 request disabled 1: Interrupt request enabled.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PIPE4BEMPE	BEMP Interrupt Enable for Pipe 4	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b5	PIPE5BEMPE	BEMP Interrupt Enable for Pipe 5	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b6	PIPE6BEMPE	BEMP Interrupt Enable for Pipe 6	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b7	PIPE7BEMPE	BEMP Interrupt Enable for Pipe 7	0: Interrupt request disabled 1: Interrupt request enabled.	R/W
b15 to b8	—	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 a 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, 4 to 7) 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.

26.2.11 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
	—	—	—	—	—	—	—	—	—	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
b15 to b7	—	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 how the BRDY interrupt status flag for each pipe is cleared.

26.2.12 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 not received 1: Setup packet 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

Bit	Symbol	Bit name	Description	R/W
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: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R
b9	NRDY	Buffer Not Ready Interrupt Status	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R
b10	BEMP	Buffer Empty Interrupt Status	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R
b11	CTRT	Control Transfer Stage Transition Interrupt Status	0: No control transfer stage transition interrupt occurred 1: Control transfer stage transition interrupt occurred.	R/W*4
b12	DVST	Device State Transition Interrupt Status	0: No device state transition interrupt occurred 1: Device state transition interrupt occurred.	R/W*4
b13	SOFR	Frame Number Refresh Interrupt Status	0: No SOF interrupt occurred 1: SOF interrupt occurred.	R/W*4
b14	RESM	Resume Interrupt Status*5	0: No resume interrupt occurred 1: Resume interrupt occurred.	R/W*4
b15	VBINT	VBUS Interrupt Status*5	0: No VBUS interrupt occurred 1: VBUS interrupt occurred.	R/W*4

x: Don't care

Note 1. The value is 0 when the MCU is reset and 1 after a USB bus reset.
Note 2. The value is 1 when the USB_VBUS pin is high and 0 when the USB_VBUS pin is low.
Note 3. The value is 000b when the MCU is reset and 001b after a USB bus reset.
Note 4. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID bit, write 0 only to the bits to be cleared. Write 1 to the other bits. Do not write 0 to the status bits indicating 0.
Note 5. The USBFS detects a change in the status indicated by the VBINT and RESM bits even while the clock supply is stopped (SYSCFG.SCKE bit is 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply before clearing the status through software.

DVSQ[2:0] bits (Device State)

The DVSQ[2:0] bits are initialized by a USB bus reset.

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, 4 to 7) on at least one for which BRDY interrupts are enabled (BRDYENB.PIPEnBRDYE = 1).

For the conditions that cause the PIPEnBRDY status to be asserted, see section 26.3.3.1, BRDY interrupt.

The USBFS sets the BRDY bit to 0 when software writes 0 to all the PIPEnBRDY bits associated with the PIPEnBRDYE bits that are set to 1. Writing 0 to the BRDY bit in software does not clear the bit.

NRDY bit (Buffer Not Ready Interrupt Status)

The NRDY bit indicates the NRDY interrupt status.

The USBFS sets the NRDY bit to 1 when it detects a NRDY interrupt status (PIPEnNRDY = 1, n = 0, 4 to 7) on at least one pipe for which NRDY interrupts are enabled (NRDYENB.PIPEnNRDYE = 1).

For the conditions that cause the PIPEnNRDY status to be asserted, see section 26.3.3.2, NRDY interrupt.

The USBFS sets the NRDY bit to 0 when software writes 0 to all the PIPEnNRDY bits associated with the PIPEnNRDYE bits that are set to 1. Writing 0 to the NRDY bit in software does not clear the bit.

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, 4 to 7) on at least one pipe for which the BEMP interrupts are enabled (BEMPENB.PIPEnBEMPE = 1).

For the conditions that cause the PIPEnBEMP status to be asserted, see section 26.3.3.3, BEMP interrupt.

The USBFS sets the BEMP bit to 0 when software writes 0 to all 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)

The USBFS updates the value of the CTSQ[2:0] bits and sets the CTRT bit to 1 on detecting a transition in the control transfer stage. When a control transfer stage transition interrupt occurs, clear the CTRT bit before the USBFS detects the next control transfer stage transition.

DVST bit (Device State Transition Interrupt Status)

The USBFS updates the value of the DVSQ[2:0] bits and sets the DVST bit to 1 on detecting a change in the device state. When a device state transition interrupt occurs, clear the DVST bit before the USBFS detects the next device state transition.

SOFR bit (Frame Number Refresh Interrupt Status)

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*5)

The USBFS sets the RESM bit to 1 on detecting the falling edge of the signal on the USB_DP pin in the suspended state (DVSQ[2:0] = 1xxb).

VBINT bit (VBUS Interrupt Status*5)

The USBFS sets the VBINT bit to 1 on detecting a level change (high to low or low to high) in the USB_VBUS pin input value. The USBFS sets the VBSTS bit to indicate the USB_VBUS pin input value. When a VBUS interrupt occurs, eliminate transient elements by reading the VBSTS bit at least three times through software processing and check that the values read are the same.

26.2.13 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
	—	—	—	—	—	—	—	—	PIPE7BRDY	PIPE6BRDY	PIPE5BRDY	PIPE4BRDY	—	—	—	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: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/W*1
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PIPE4BRDY	BRDY Interrupt Status for Pipe 4*2	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/W*1
b5	PIPE5BRDY	BRDY Interrupt Status for Pipe 5*2	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/W*1
b6	PIPE6BRDY	BRDY Interrupt Status for Pipe 6*2	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/W*1
b7	PIPE7BRDY	BRDY Interrupt Status for Pipe 7*2	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/W*1
b15 to b8	—	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.

26.2.14 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
	—	—	—	—	—	—	—	—	PIPE7NRDY	PIPE6NRDY	PIPE5NRDY	PIPE4NRDY	—	—	—	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: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W*1
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PIPE4NRDY	NRDY Interrupt Status for Pipe 4	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W*1
b5	PIPE5NRDY	NRDY Interrupt Status for Pipe 5	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W*1
b6	PIPE6NRDY	NRDY Interrupt Status for Pipe 6	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W*1
b7	PIPE7NRDY	NRDY Interrupt Status for Pipe 7	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W*1
b15 to b8	—	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.

26.2.15 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
	—	—	—	—	—	—	—	—	PIPE7BEMP	PIPE6BEMP	PIPE5BEMP	PIPE4BEMP	—	—	—	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: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W*1
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PIPE4BEMP	BEMP Interrupt Status for Pipe 4	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W*1
b5	PIPE5BEMP	BEMP Interrupt Status for Pipe 5	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W*1
b6	PIPE6BEMP	BEMP Interrupt Status for Pipe 6	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W*1
b7	PIPE7BEMP	BEMP Interrupt Status for Pipe 7	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W*1
b15 to b8	—	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.

26.2.16 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
	—	—	—	—	—	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
b15 to b11	—	Reserved	These bits are read as 0	R

FRNM[10:0] bits (Frame Number)

The USBFS sets the FRNM[10:0] bits to indicate the latest frame number, which is updated every 1 ms, when an SOF packet is issued or received.

26.2.17 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
b15 to b8	BREQUEST[7:0]	Request	These bits store the USB request bRequest value	R

USBREQ stores setup requests for control transfers. The USBREQ stores the received values of bRequest and bmRequestType.

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.

These bits indicate the value of 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.

These bits indicate the value of USB request data in the setup transactions for reception. Writing to these bits has no effect.

26.2.18 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

USBVAL stores the received value of wValue. 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.

These bits indicate the wValue value of USB requests in the setup transactions for reception. Writing to the WVALUE[15:0] bits has no effect.

26.2.19 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

USBINDX stores setup requests for control transfers. The USBINDX stores the received wIndex value.

USBINDX is initialized by a USB bus reset.

WINDEX[15:0] bits (Index)

The WINDEX[15:0] bits hold the wIndex value of a USB request.

These bits indicate the wIndex value of USB requests in the setup transactions for reception. Writing to the WINDEX[15:0] bits has no effect.

26.2.20 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

USBLENG stores setup requests for control transfers. The received value of wLength is stored.

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.

These bits indicate the wLength value of USB requests in the setup transactions for reception. Writing to the WLENTUH[15:0] bits has no effect.

26.2.21 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	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	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	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe kept open after transfer ends 1: Pipe disabled after transfer ends.	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 after changing DCPCTR.PID[1:0] bits for the DCP from BUF to NAK, check that the DCPCTR.PBUSY bit is 0. However, if the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through software is not required.

SHTNAK bit (Pipe Disabled at End of Transfer*1)

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.

26.2.22 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
	—	—	—	—	—	—	—	—	—	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 data payload specification (maximum packet size) 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
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only set the MXPS[6:0] bits while PID is NAK. Before setting these bits after changing the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK, check that the DCPCTR.PBUSY bit is 0. However, if the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through software is not required. After modifying the MXPS[6:0] bits and setting the DCP to the CURPIPE[3:0] bits in the CFIFO Port Select Register, clear the buffer by setting the BCLR bit in the CFIFO Port Control Register to 1.

MXPS[6:0] bits (Maximum Packet Size)

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.

26.2.23 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	—	—	—	—	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depends on the buffer state) 1 0: STALL response 1 1: STALL response.	R/W
b2	CCPL	Control Transfer End Enable	0: Control transfer completion disabled 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 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: Clears the expected value for the next transaction to DATA0.	R/W*1
b14 to b9	—	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. 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 after changing the PID[1:0] bits from 01b (BUF) to 00b (NAK), check that the PBUSY bit is 0. However, 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 control the USB response type during control transfers.

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 by software and the USBFS receive data that exceeds 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 USB 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)

Setting the CCPL bit to 1 enables the status stage of the control transfer to be completed.

When this 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.

PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether DCP is used for the transaction when the 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 USB 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 26.3.4.1, 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.

The USBFS sets the SQMON bit to 1 (specifies DATA1 as the expected value) on successful reception of the setup packet.

The USBFS does not reference the SQMON 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.

BSTS bit (Buffer Status)

The BSTS bit indicates whether DCP FIFO buffer access is enabled or disabled.

The meaning of the BSTS bit varies as follows depending on the CFIFOSEL.ISEL setting:

- When the ISEL = 0, the BSTS bit indicates whether receive data can be read from the buffer
- When the ISEL = 1, the BSTS bit indicates whether transmitted data can be written to the buffer.

26.2.24 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 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7. 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 4 to 7 using the PIPESEL, PIPECFG, PIPEMAXP, PIPEnCTR, PIPEnTRE, and PIPEnTRN registers (n = 4 to 7).

After selecting the pipe with the PIPESEL register, set the pipe functions using PIPECFG and PIPEMAXP. 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 and PIPEMAXP registers used for data writing and reading. Selecting a pipe number in the PIPESEL[3:0] bits allows writing to and reading from PIPECFG and PIPEMAXP associated with the selected pipe number.

When PIPESEL[3:0] = 0000b, 0 is read from all of the bits in PIPECFG and PIPEMAXP. Writing to these bits is invalid.

26.2.25 Pipe Configuration Register (PIPECFG)

Address(es): USBFS.PIPECFG 4009 0068h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TYPE[1:0]		—	—	—	BFRE	DBLB	—	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	EPNUM[3:0]	Endpoint Number*1	Specifies the endpoint number for the selected pipe. Setting 0000b indicates that 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 4 and 5 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Setting prohibited. • Pipes 6 and 7 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 after changing the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through 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 CFIFO Port Select Register. Before setting these bits after changing the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through software is not required.

Note 3. To change the BFRE, DBLB, and 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 4 to 7. 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 SHTNAK bit is valid for pipes 4 and 5 in the receiving direction.

When software sets this bit to 1 for a receiving pipe, the USBFS changes the PIPEnCTR.PID[1:0] bits associated with the selected pipe to 00b (NAK) on determining the transfer end. The USBFS determines that the transfer has ended on the following conditions:

- A short packet data (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. This bit is valid for pipes 4 and 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 the BRDY interrupt is generated with this setting, write 1 to the BCLR bit in the CFIFO 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 26.3.3.1, 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.

26.2.26 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
	—	—	—	—	—	—	—	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 4 and 5: 8 bytes (008h), 16 bytes (010h), 32 bytes (020h), 64 bytes (040h) Bits MXPS[8:7] and MXPS[2:0] are not supported. • Pipes 6 and 7: 1 byte (001h) to 64 bytes (040h) Bits MXPS[8:7] are not supported.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	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 CFIFO Port Select Register. Before setting these bits after changing the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through software is not required.

PIPEMAXP specifies the maximum packet size for pipes 4 to 7.

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] = 000h, do not write to the FIFO buffer or set PID to BUF. These writes have no effect.

26.2.27 PIPEn Control Registers (PIPEnCTR) (n = 4 to 7)

PIPEnCTR (n = 4 and 5)

Address(es): 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

Bit	Symbol	Bit name	Description	R/W
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 are initialized).	R/W
b10	ATREPM	Auto Response Mode*2	0: Auto response mode disabled 1: Auto response mode enabled.	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	INBUFM	Transmit Buffer Monitor	0: No data to be transmitted is in the FIFO buffer 1: Data to be transmitted is in the FIFO buffer.	R
b15	BSTS	Buffer Status	0: Buffer access by the CPU disabled 1: Buffer access by the CPU enabled.	R

Note 1. Only 0 can be read.

Note 2. Only set the ATREPM bit or write 1 to the SQCLR or SQSET bit while PID is NAK. Before setting these bits after changing the PID[1:0] bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through software is not required.

Note 3. Only set the ACLRM bit while PID[1:0] is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the CFIFO Port Select Register. Before setting this bit after changing the PID[1:0] bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits are changed to NAK by the USBFS, 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 26.7 shows 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.

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 26.7 USBFS operation based on PID[1:0] bit setting

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
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 associated FIFO buffer is ready for transmission. Returns NAK if not ready.
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

PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether the selected pipe is being used for the current transaction.

The USBFS changes the PBUSY bit from 0 to 1 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 settings is possible. For details, see section 26.3.4.1, 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. The USBFS does not toggle the SQMON bit when a DATA-PID mismatch occurs during the transfer in the receiving direction.

SQSET bit (Sequence Toggle Bit Set)

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)

Setting the SQCLR bit to 1 through software allows the USBFS to set DATA0 as the expected value of the sequence toggle bit for the next transaction of the selected pipe. 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 consecutively.

Table 26.8 shows the data cleared by writing 1 and 0 to the ACLRM bit consecutively and the cases in which this processing is required.

Table 26.8 Data cleared by the 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 (two FIFO buffers in double buffer mode)	When initializing the selected pipe
2	Internal flags of the PIPECFG.BFRE bit	When changing the PIPECFG.BFRE setting
3	FIFO buffer toggle control	When changing the PIPECFG.DBLB setting
4	Internal flags of the transaction count	When forcing the transaction count function to terminate

ATREPM bit (Auto Response Mode)

The ATREPM bit enables or disables auto response mode for the selected pipe.

This bit can be set to 1 when the selected pipe is for bulk transfer. When this 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):
  - When ATREPM = 1 and PID = BUF, the USBFS transmits a zero-length packet in response to the IN token
  - The USBFS updates 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 ACK is received. The USBFS does not generate the BRDY or BEMP interrupt.
- When the selected pipe is for bulk OUT transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 0):
  - When ATREPM = 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.

INBUFM bit (Transmit Buffer Monitor)

The INBUFM 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 completes writing data to at least one FIFO buffer plane.

The USBFS sets the INBUFM bit to 0 when it completes transmission of 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 it completes transmission of the data from the two FIFO buffer planes before the CPU 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 and PIPECFG.BFRE settings, as shown in Table 26.9.

Table 26.9 BSTS bit operation

DIR value	BFRE value	BSTS bit function
0	0	The receive data can be read from the FIFO buffer. The receive data is completely read from the FIFO buffer.
	1	The receive data that can be read from the FIFO buffer is set to 1 by software after the receive data is completely read from the FIFO buffer
1	0	The transmit data can be written to the FIFO buffer. The transmit data is completely written to the FIFO buffer.
	1	Setting prohibited

PIPEnCTR (n = 6 and 7)

Address(es): USBFS.PIPE6CTR 4009 007Ah, USBFS.PIPE7CTR 4009 007Ch

	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 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 1: Clear the expected value for the next transaction to DATA0.	R/W*1
b9	ACLRM	Auto Buffer Clear Mode*2,*3	0: Auto buffer clear mode disabled 1: Auto buffer clear mode 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 after changing the PID[1:0] bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits are changed to 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 CFIFO Port Select Register. Before setting this bit after changing the PID[1:0] bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits are changed to 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 USBFS transfer. Table 26.7 shows the basic operation of the USBFS (when there are no errors in the transmitted and received packets) based on the PID[1:0] 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 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 NAK on 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.

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, transition to NAK and then BUF.

PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether the selected pipe is being used for the current transaction.

The USBFS changes the PBUSY bit from 0 to 1 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 set DATA0 as the expected value of the sequence toggle bit for the next transaction of the selected pipe. 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 26.10 shows the data cleared by writing 1 and 0 continuously to the ACLRM bit and the cases in which this processing is required.

Table 26.10 Data cleared by the 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	Internal flags of the PIPECFG.BFRE bit	When changing the PIPECFG.BFRE setting
3	Internal flags of 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 and PIPECFG.BFRE settings, as shown in Table 26.9.

26.2.28 PIPEn Transaction Counter Enable Register (PIPEnTRE) (n = 4 and 5)

Address(es): 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 1: The current counter value is cleared.	R/W
b9	TRENB	Transaction Counter Enable	0: Transaction counter disabled 1: Transaction counter enabled.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set each bit in PIPEnTRE while PID is NAK. Before setting these bits after changing the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through 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 receive data.

For transmitting pipes, set the TRENB bit to 0. When the transaction counter is not used, set the TRENB bit to 0. When the transaction counter is used, set the TRNCNT[15:0] bits before setting the TRENB bit to 1. Set this bit to 1 before receiving the first packet to be counted by the transaction counter.

26.2.29 PIPEn Transaction Counter Register (PIPEnTRN) (n = 4 and 5)

Address(es): 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, this bit specifies the total packets (number of transactions) to be received by the selected pipe. When read from with the PIPEnTRE.TRENB bit at 0, this bit indicates the specified number of transactions. When PIPEnTRE.TRENB bit is 1, this bit indicates the current transaction count.	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 is 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.

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 is 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.

Both of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit is 1
- The USBFS received a short packet.

Both of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit is 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.

26.2.30 USB Module Control Register (USBMC)

Address(es): USBFS.USBMC 4009 00CCh

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

Bit	Symbol	Bit name	Description	R/W
b0	VDDUSBE	USB Reference Power Supply Circuit On/Off Control	0: USB reference power supply circuit off 1: USB reference power supply circuit on.	R/W
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	VDCEN	USB Regulator On/Off Control	0: USB regulator off 1: USB regulator on.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

VDDUSBE bit (USB Reference Power Supply Circuit On/Off Control)

The USB reference power supply circuit generates the reference voltage for battery charging. Set this bit to 1 when using the battery charging function.

VDCEN bit (USB Regulator On/Off Control)

The VDCEN bit controls the USB regulator circuit. Set this bit to 1 when using the USB regulator circuit.

26.2.31 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 Flag*1	0: Not detected 1: Detected.	R

Bit	Symbol	Bit name	Description	R/W
b9	PDDETSTS0	D+ Pin 0.6 V Input Detection Status Flag*2	0: Not detected 1: Detected.	R
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Valid when IDMSINKE0 = 1.
Note 2. Valid when IDPSINKE0 = 1.

RPDME0 bit (D- Pin Pull-Down Control)

When using the battery charging function, set this bit to 1 to control the pull-down resistor of the D- pin.

IDPSRCE0 bit (D+ Pin IDPSRC Output Control)

When the IDPSRCE0 bit is set to 1, 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, 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 set to 1, 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, the USBFS detects whether VDMSRC (0.6 V), output from the device to D-, is connected to D+ (DCP) by the host.

VDMSRCE0 bit (D- Pin VDMSRC (0.6 V) Output Control)

When the VDMSRCE0 bit set to 1, output is enabled on secondary detection and VDMSRC (0.6 V) is applied to D-.

CHGDETSTS0 flag (D- Pin 0.6 V Input Detection Status Flag)

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 Flag)

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.

26.2.32 USB Clock Selection Register (UCKSEL)

Address(es): USBFS.UCKSEL 4009 00C4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UCKSELC
Value after 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	UCKSELC	USB Clock Selection*1	0: High-speed On-Chip Oscillator clock (HOCO) not selected as USB clock 1: High-speed On-Chip Oscillator clock (HOCO) selected as USB clock.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When UCKSELC = 1, the user trimming function cannot be used. For information on the user trimming function, see section 9, Clock Generation Circuit.

26.3 Operation

26.3.1 System Control

This section describes the register settings required for initializing the USBFS and controlling power consumption.

26.3.1.1 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 is 1) enables and starts USBFS operation.

26.3.1.2 Controlling the USBFS data bus 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 and SYSCFG.DMRPU bits.

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.

Table 26.11 Control settings for the USBFS data bus resistors

SYSCFG register settings				
DPRPU bit	**DMRPU bit**	**D-**	**D+**	**Function**
0	0	Open	Open	Not in use
1	0	Open	Pull-up	Full-speed
0	1	Pull-up	Open	Low-speed
Other settings		—	—	Setting prohibited

26.3.1.3 Example of USBFS power supply connection

Figure 26.2 shows an example of power supply connection when the USB regulator is not used. Figure 26.3 and Figure 26.4 show examples of power supply connection when the USB regulator is used.

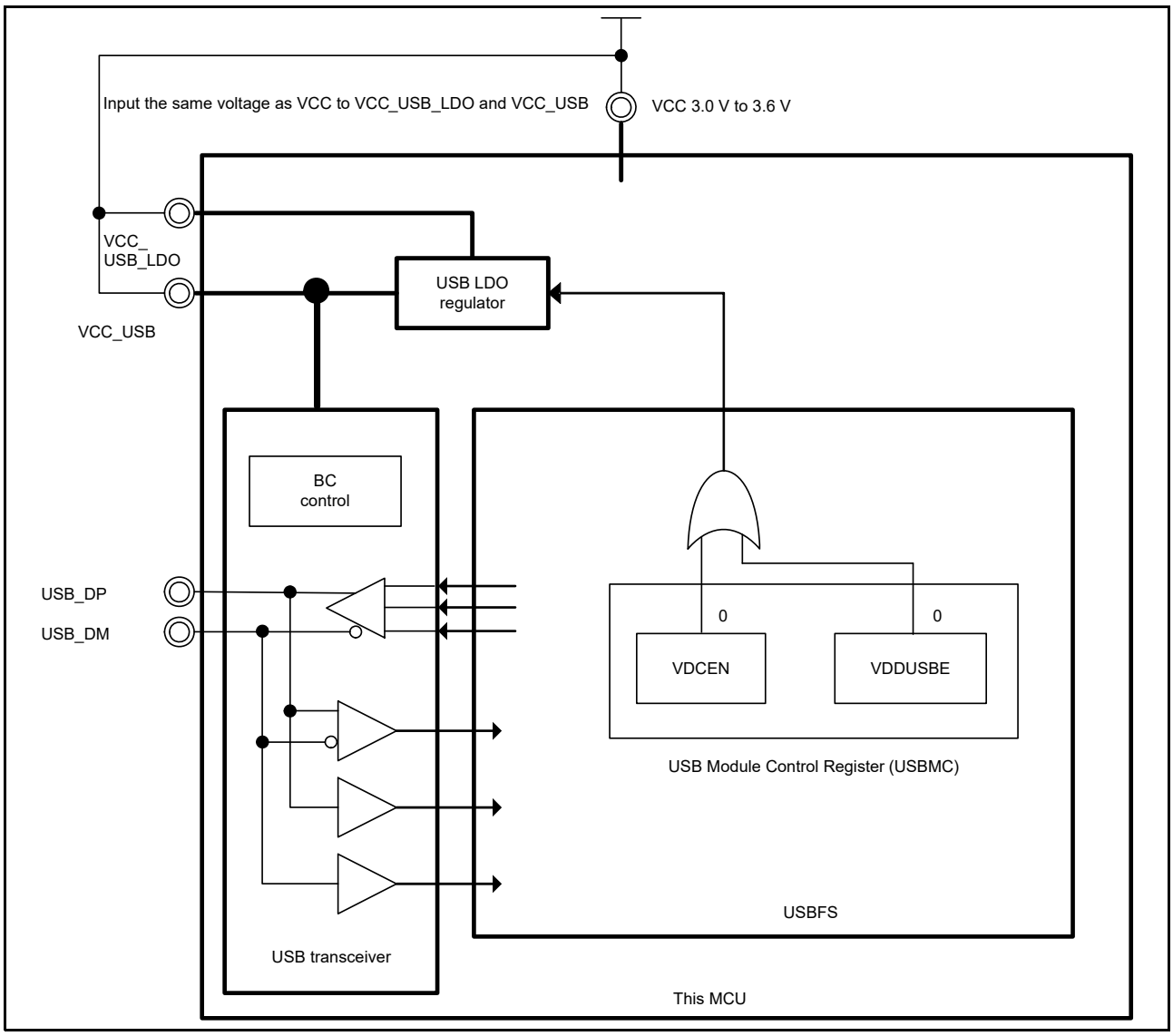
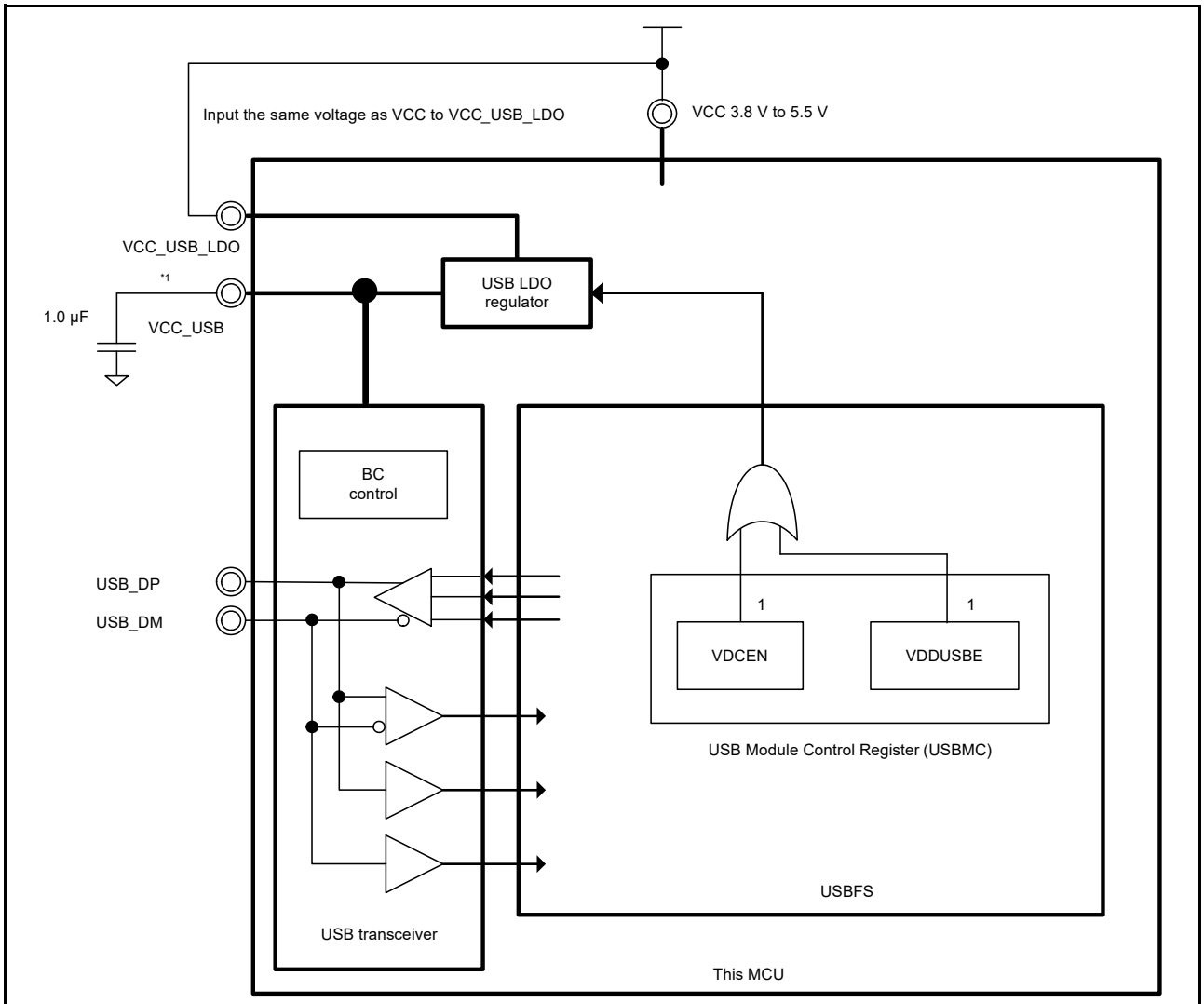


Figure 26.2 Example of power supply connection when the USB LDO regulator is not used



Note 1. Make sure that the resistance of the connected wiring is 0.5 Ω or less to connect a capacitor of ESR ≤ 1 Ω.

Figure 26.3 Example of power supply connection when the USB LDO regulator is used (BC used)

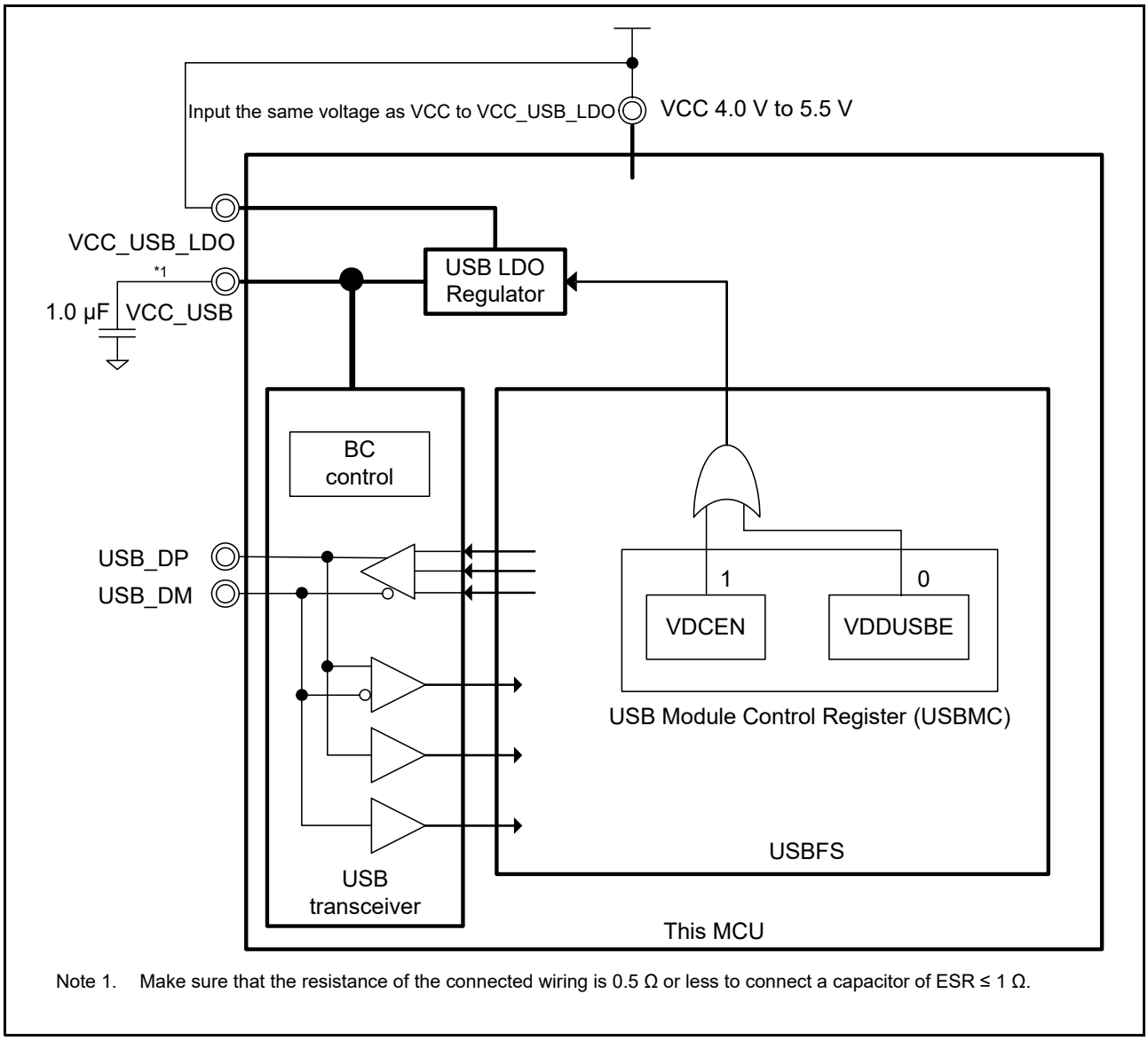


Figure 26.4 Example of power supply connection when the USB LDO regulator is used (BC not used)

26.3.1.4 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 transceiver. Figure 26.6 and Figure 26.7 show examples of external circuits for USB connection.

Figure 26.5 shows an example of functional connection of the USB connector in the self-powered state.

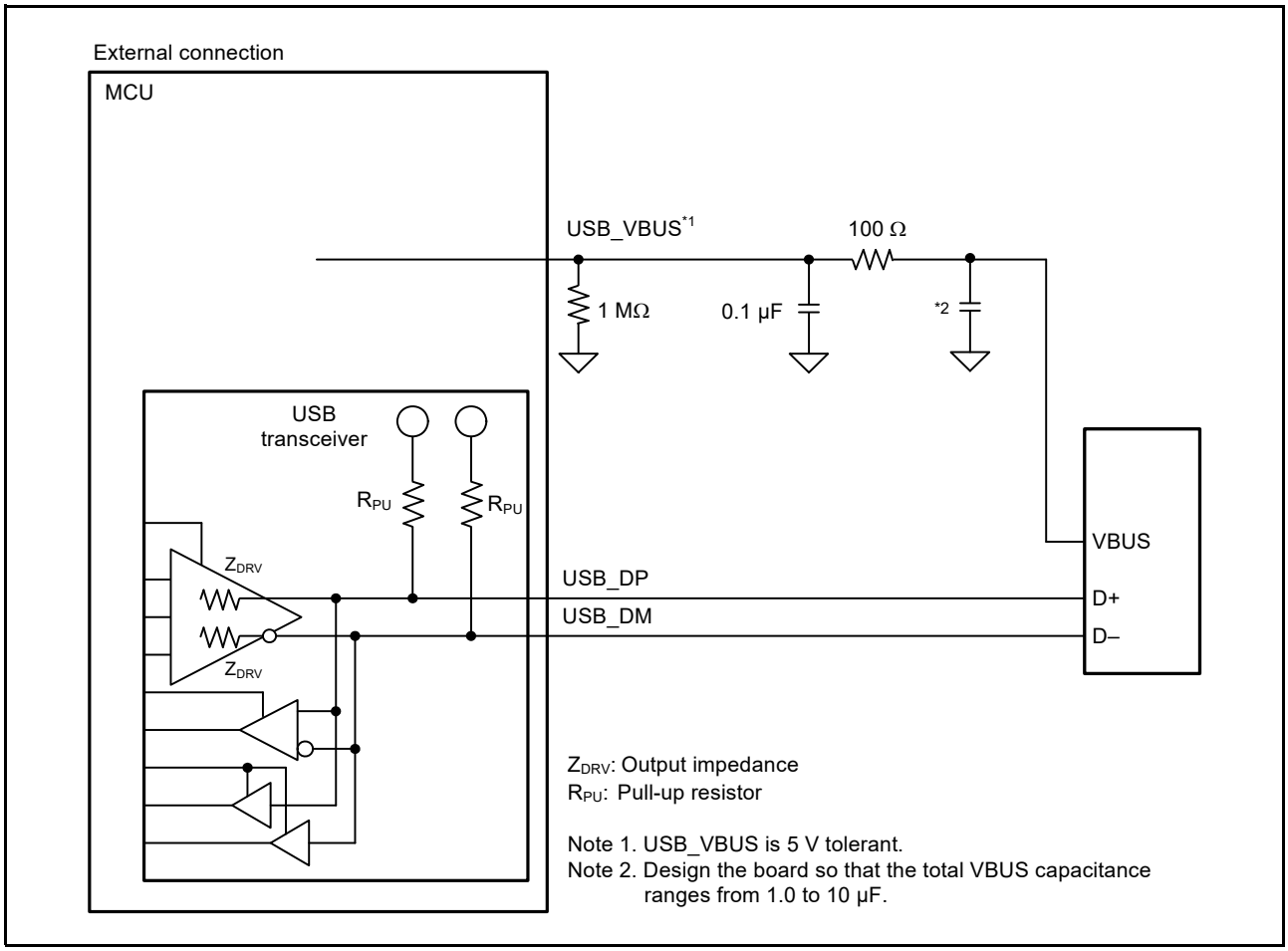


Figure 26.5 Example device connection in self-powered state

Figure 26.6 shows an example of functional connection of the USB connector in bus-powered state 1.

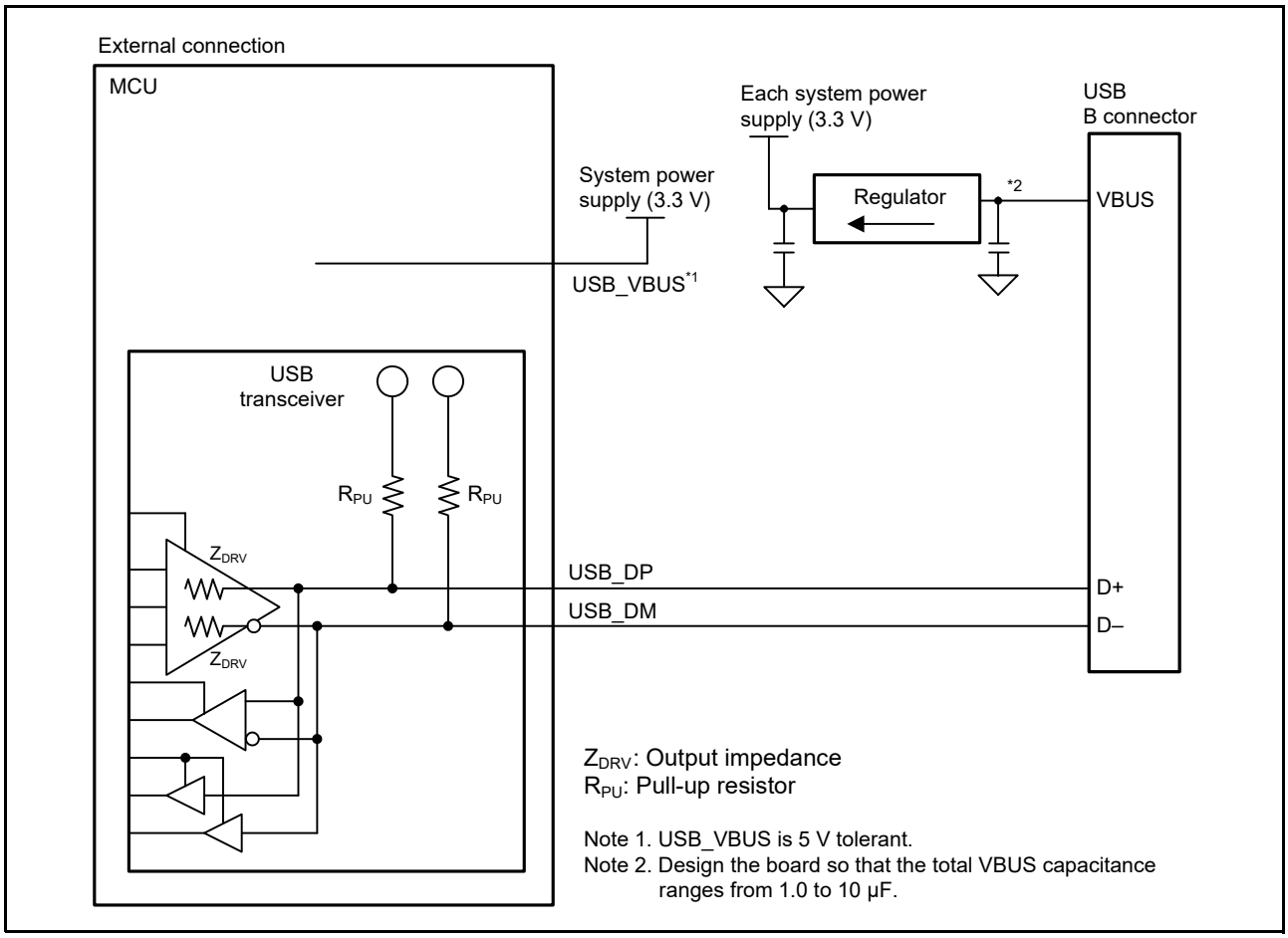


Figure 26.6 Example device connection in bus-powered state 1

Figure 26.7 shows an example of functional connection of the USB connector in bus-powered state 2.

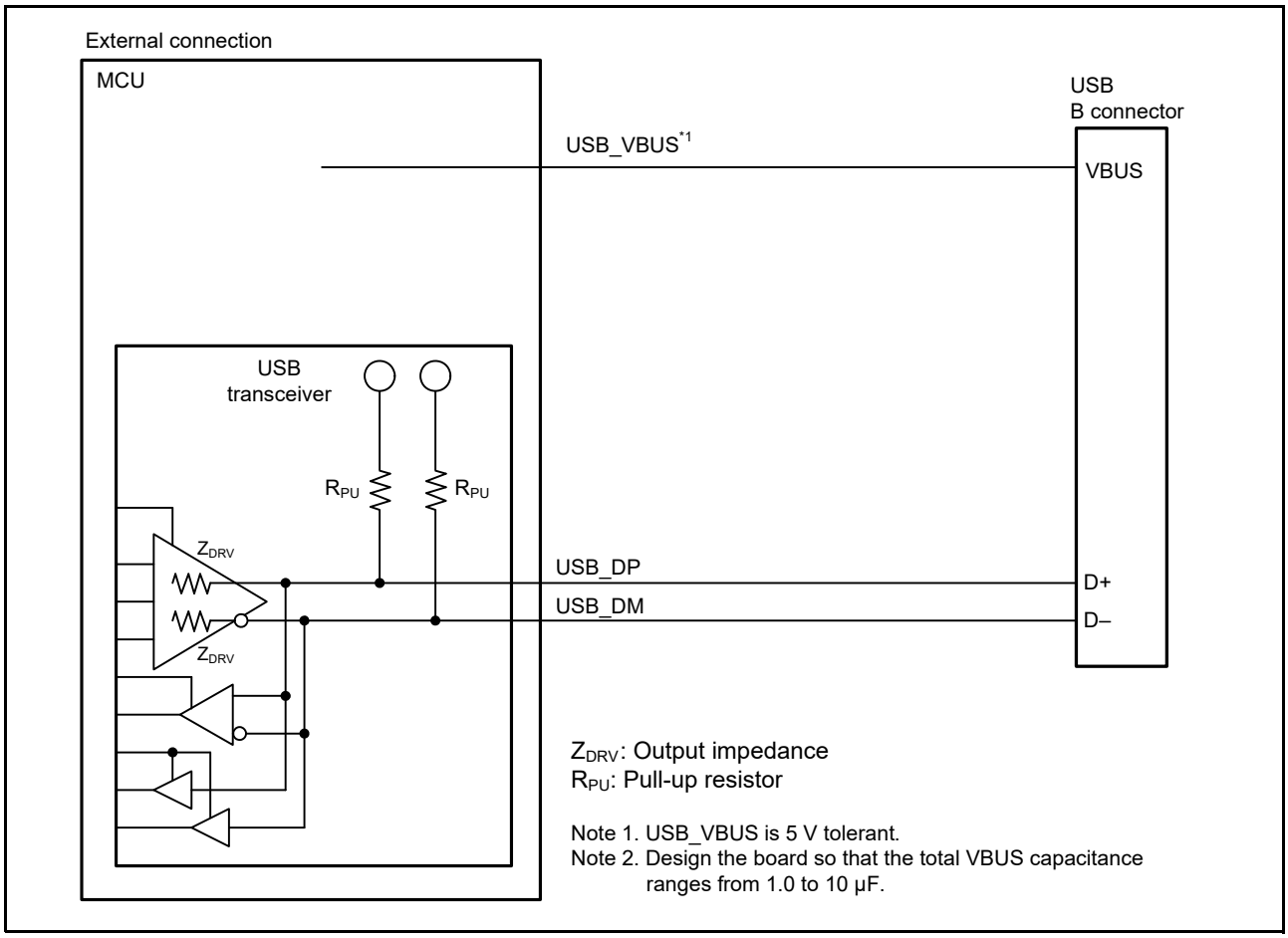


Figure 26.7 Example device connection in bus-powered state 2

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

Figure 26.8 shows an example of functional connection of the USB connector with Battery Charging v1.2 supported.

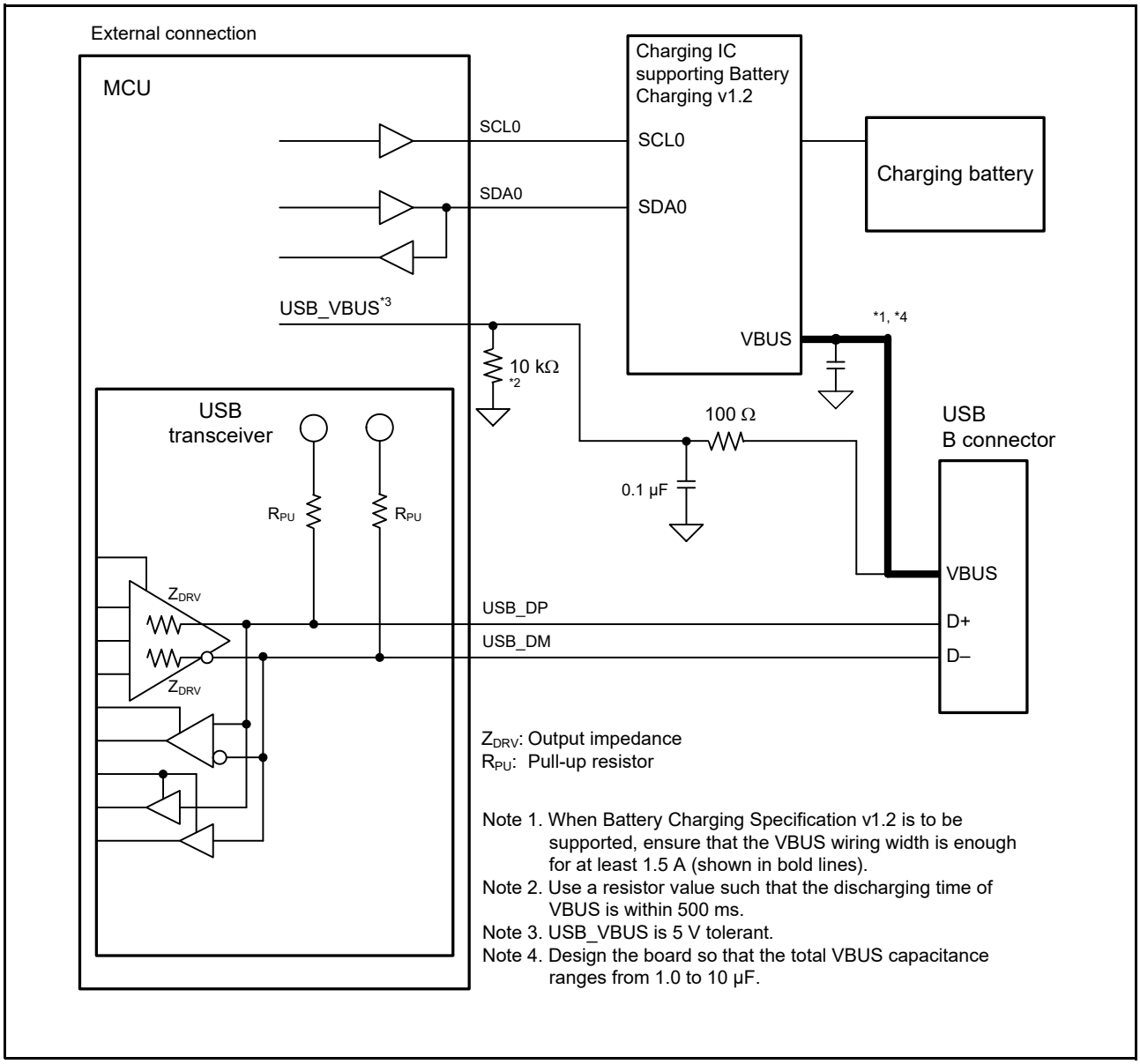


Figure 26.8 Functional connection sample of USB connector with Battery Charging v1.2 supported

26.3.2 Interrupts

Table 26.12 lists the interrupt sources in the USBFS. When an interrupt generation condition is satisfied and the interrupt output is enabled using the associated interrupt enable register, a USBFS interrupt request is issued to the Interrupt Controller Unit (ICU) and a USBFS interrupt is generated. See section 13, Interrupt Controller Unit (ICU).

Table 26.12 Interrupt sources

Bit to be set	Name	Interrupt source	Status flag
VBINT	VBUS interrupt	• When a change in the state of the USB_VBUS input pin is detected (low to high or high to low).	INTSTS0.VBSTS
RESM	Resume interrupt	• When a change in the state of the USB bus is detected in the suspended state (J-state to K-state or J-state to SE0).	—
SOFR	Frame number update interrupt	• When an SOF packet with a different frame number is received.	—
DVST	Device state transition interrupt	• When a device state transition is detected with any of the following conditions: - USB bus reset detected - Suspended state detected - SET_ADDRESS request received - SET_CONFIGURATION request received.	INTSTS0.DVSQ[2:0]
CTRT	Control transfer stage transition interrupt	• When a stage transition is detected in control transfer with any of the following conditions: - 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.	INTSTS0.CTSQ[2:0]
BEMP	Buffer empty interrupt	• When transmission of all data in the buffer memory is complete and the buffer becomes empty • When a packet larger than the maximum packet size is received.	BEMPSTS.PIPEnBEMP
NRDY	Buffer not ready interrupt	• When NAK is returned for an IN or OUT token while PID = BUF.	NRDYSTS.PIPEnNRDY
BRDY	Buffer ready interrupt	• When the buffer becomes ready (reading or writing is enabled).	BRDYSTS.PIPEnBRDY

Figure 26.9 shows the circuits related to the USBFS interrupts.

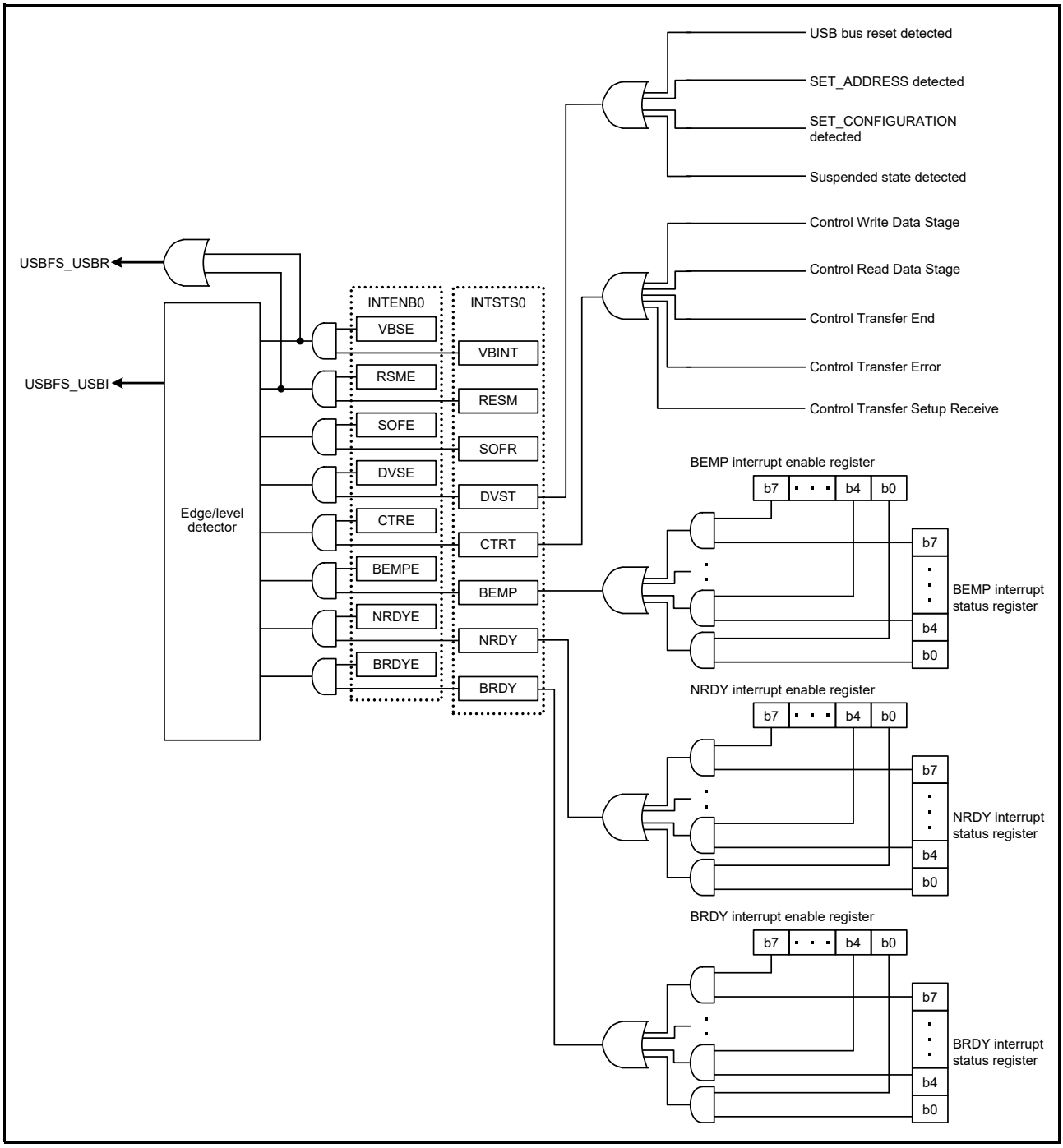


Figure 26.9 USBFS interrupt-related circuits

Table 26.13 shows the interrupts generated by the USBFS.

Table 26.13 USBFS Interrupts

Interrupt name	Interrupt status flag	DTC activation
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	Not possible
USBFS_USBR	VBUS interrupt, resume interrupt	Not possible

26.3.3 Interrupt Descriptions

26.3.3.1 BRDY interrupt

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.PIPEnBRDYE 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 the 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 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 is successfully complete, 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 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 when reception by the other FIFO buffer completes.

The BRDY interrupt is not generated in the status stage of control transfers. The BRDY 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 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 by the PIPEnTRN.TRNCNT[15:0] bits are completely received.

When the pertinent data is completely read after any of the specified 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 CFIFO Port Control Register is 1 and the DTLN[8:0] bits are 000h. In this case, to start the next transfer, write 1 to the BCLR bit in the associated CFIFO Port Control Register through software. With these settings, the USBFS does not detect a BRDY interrupt for the transmitting pipe.

The BRDY interrupt status of the selected pipe can be set to 0 by writing 0 to the corresponding BRDYSTS.PIPEnBRDY bit through software. In this case, the other PIPEnBRDY bits should be set to 1.

In this mode, do not modify the PIPECFG.BFRE bit setting until all data for a single transfer is processed. When modification to the PIPECFG.BFRE bit is required before processing completes, clear all FIFO buffers for the pipe with the PIPEnCTR.ACLRM bit.

(3) When the SOFCFG.BRDYM = 1 and the PIPECFG.BFRE = 0

With these settings, the BRDYSTS.PIPEnBRDY values are linked to the BSTS bit setting for each pipe, that is, the BRDY interrupt status bits (PIPEnBRDY) are set to 1 or 0 by the USBFS depending on the FIFO buffer status.

(a) For transmitting pipes

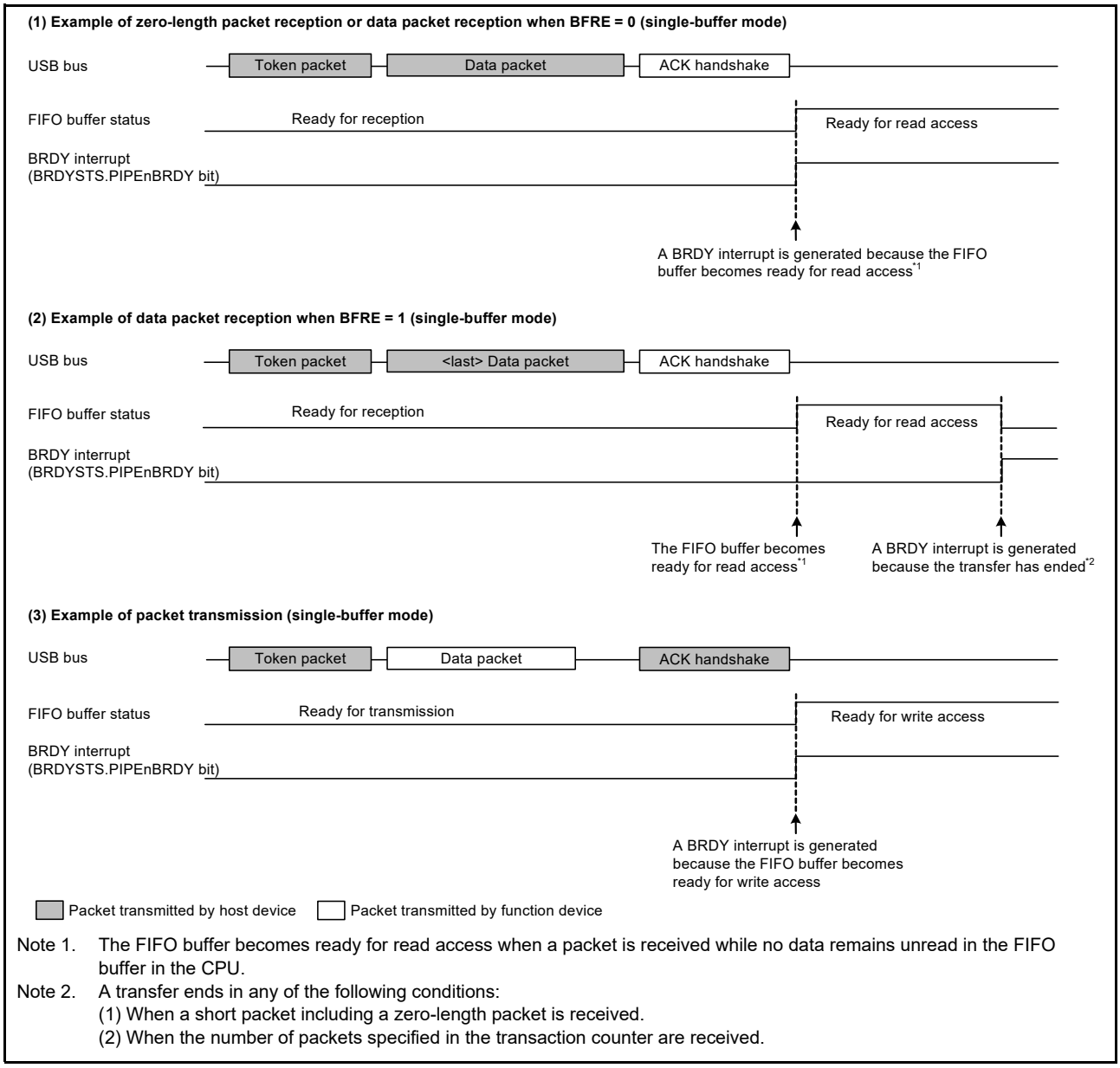
The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready. The BRDY interrupt is not generated for the DCP in the transmitting direction even when it is ready for write access.

(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 26.10 shows the timing of BRDY interrupt generation.



Note 1. The FIFO buffer becomes ready for read access when a packet is received while no data remains unread in the FIFO buffer in the CPU.

Note 2. A transfer ends in any 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 26.10 Timing of BRDY interrupt generation

The condition for clearing the INTSTS0.BRDY bit depends on the SOFCFG.BRDYM bit setting as shown in Table 26.14.

Table 26.14 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

26.3.3.2 NRDY interrupt

On generating an internal NRDY interrupt request for the pipe whose PID[1:0] 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 status stage execution of the control transfer.

(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.

(b) For receiving pipes

- When an OUT token is received but there is no space available in the FIFO buffer. For transfer pipes in which an interrupt is generated, the USBFS generates an 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.

Figure 26.11 shows the timing of NRDY interrupt generation.

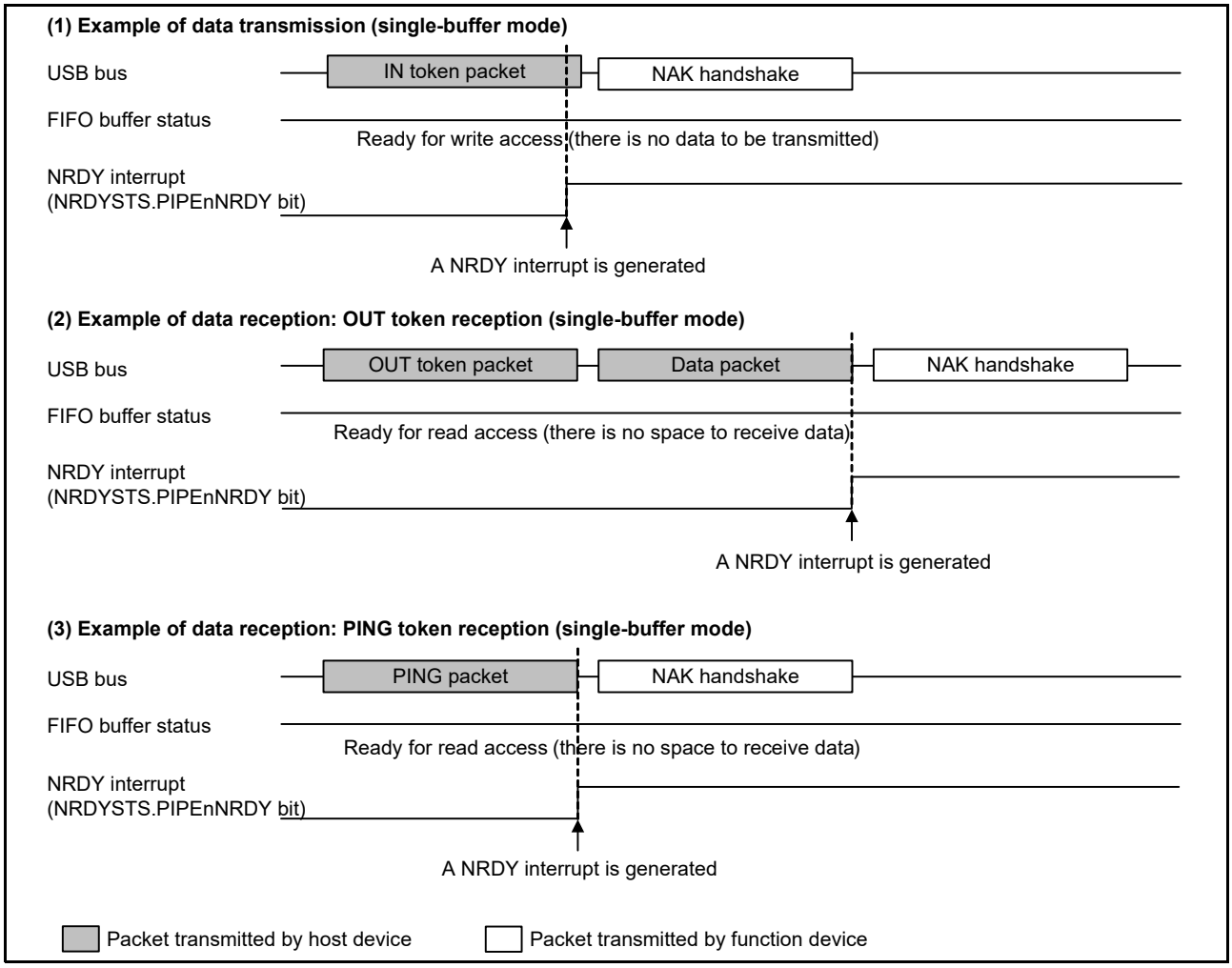


Figure 26.11 Timing of NRDY interrupt generation

26.3.3.3 BEMP interrupt

On detecting a BEMP interrupt for the pipe whose PID[1:0] 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 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 CFIFO Port Control Register to 1
- When an IN transfer (zero-length packet transmission) is performed during the control transfer status stage.

(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 receive data, and changes the associated PID[1:0] setting for the pipe to STALL (11b). The USBFS returns STALL response.

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 receive 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 26.12 shows the timing of BEMP interrupt generation.

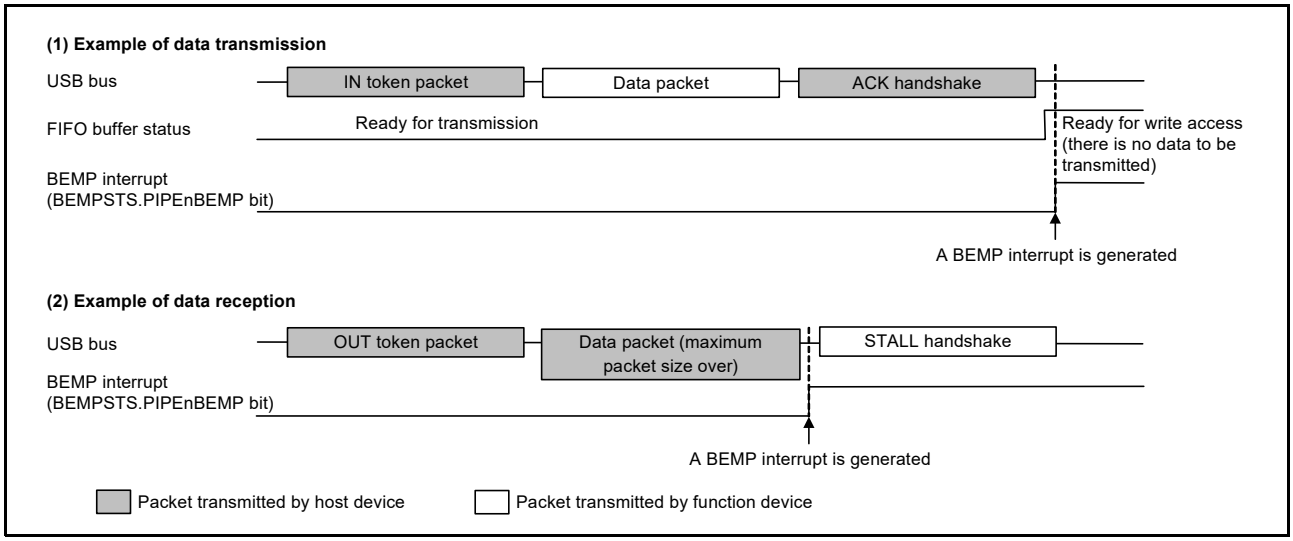
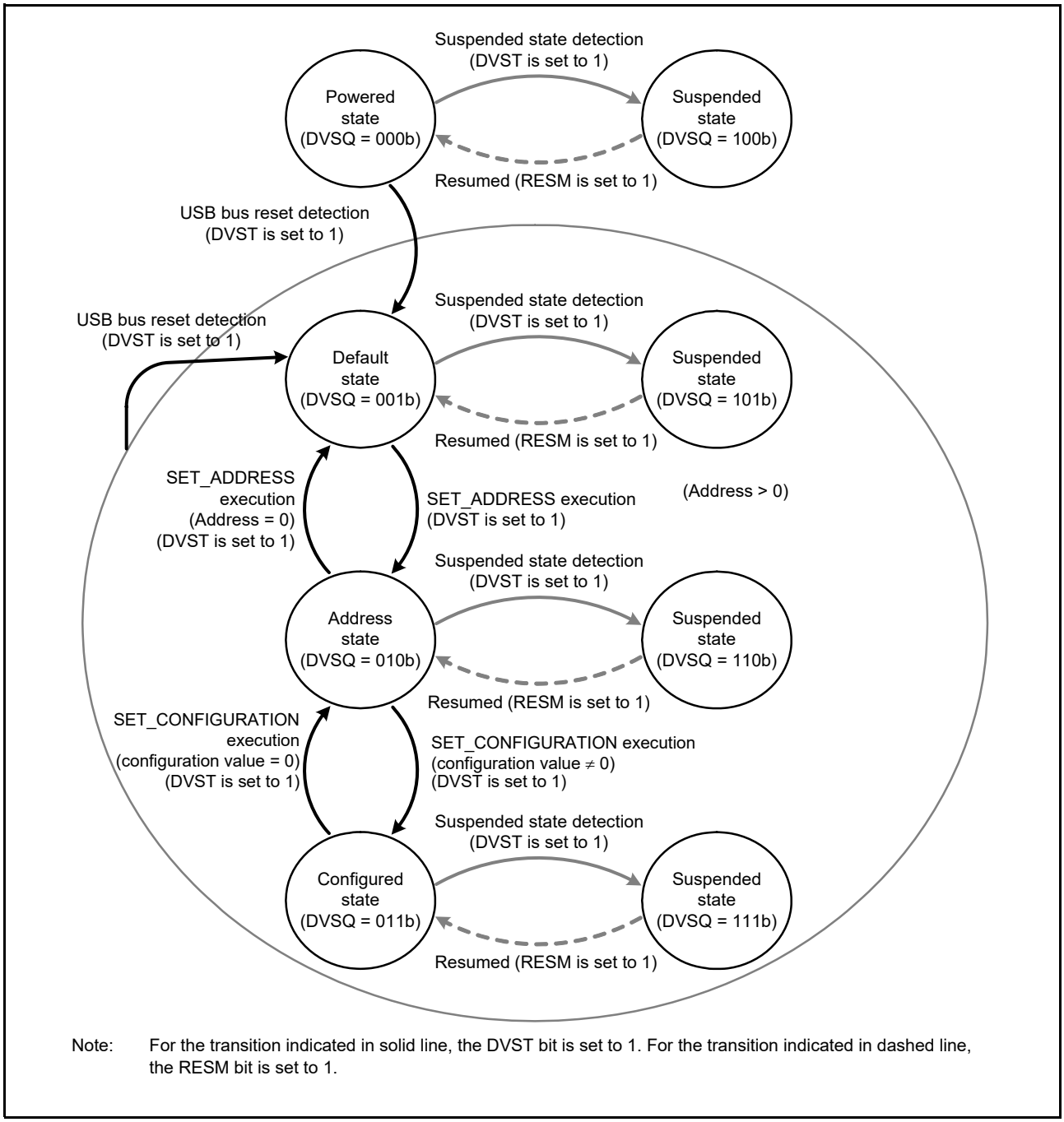


Figure 26.12 Timing of BEMP interrupt generation

26.3.3.4 Device state transition interrupt

Figure 26.13 shows a diagram of device state transitions in the USBFS. The USBFS controls device state and generates device state transition interrupts. However, recovery from the suspended state (resumed 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.



Note: For the transition indicated in solid line, the DVST bit is set to 1. For the transition indicated in dashed line, the RESM bit is set to 1.

Figure 26.13 Device state transitions

26.3.3.5 Control transfer stage transition interrupt

Figure 26.14 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.

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.

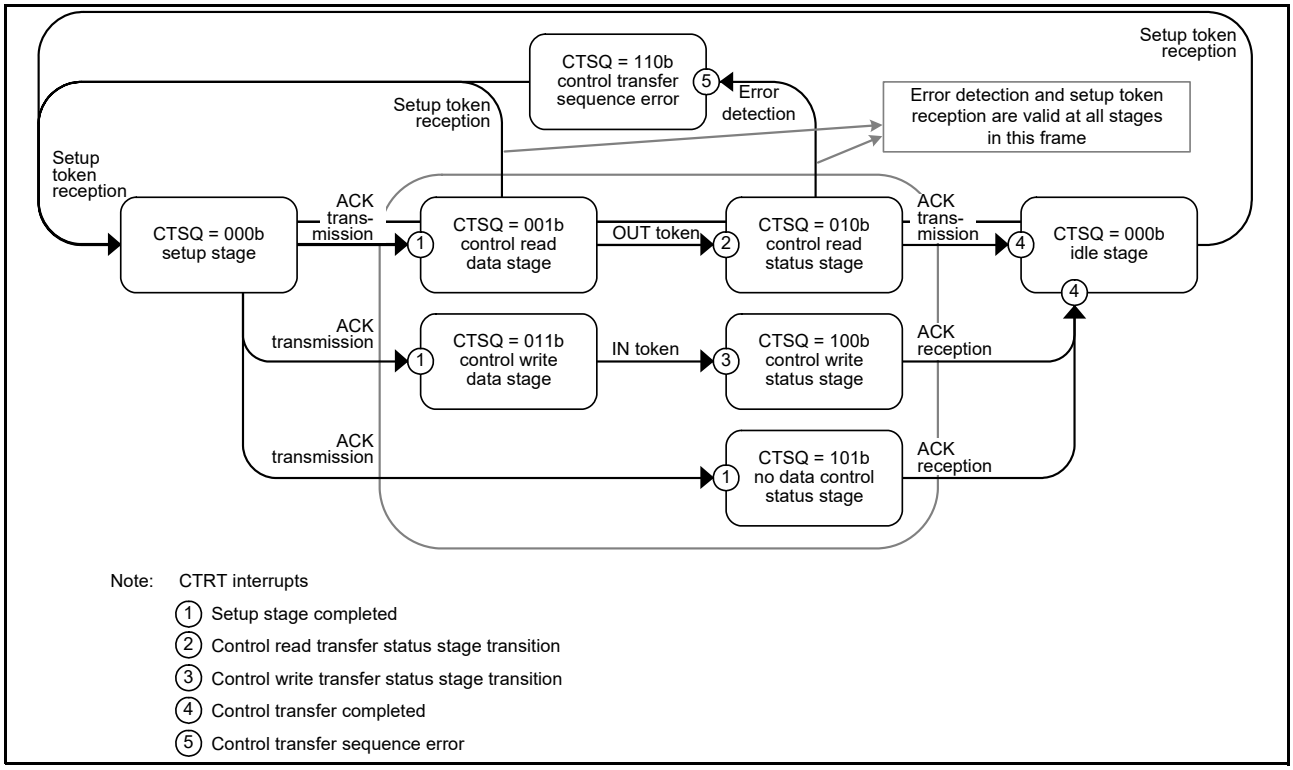


Figure 26.14 Control transfer stage transitions

26.3.3.6 Frame update interrupt

The USBFS updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

26.3.3.7 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.

26.3.3.8 Resume interrupt

A resume interrupt is generated when the device state is the suspended state, and the USB bus state has changed from J-state to K-state, or from J-state to SE0. Recovery from the suspended state is detected by means of the resume interrupt.

26.3.4 Pipe Control

Table 26.15 lists the pipe settings for the USBFS. USB data transfer is performed through logical pipes that software associates with endpoints. The USBFS has five pipes for data transfer. Set up the pipes based on your system specifications.

Table 26.15 Pipe settings

Register name	Bit name	Setting	Remarks
DCPCFG PIPECFG	TYPE	Transfer type	Pipes 4 to 7: Can be set
	BFRE	BRDY interrupt mode	Pipes 4 and 5: Can be set
	DBLB	Double buffer select	Pipes 4 and 5: Can be set
	DIR	Transfer direction select	IN or OUT can be set
	EPNUM	Endpoint number	Pipes 4 to 7: Can be set Set a value other than 0000b when the pipe is used.
	SHTNAK	Disabled state select for pipe when transfer ends	Pipes 4 and 5: Can be set
DCPMAXP PIPEMAXP	MXPS	Maximum packet size	Compliant with the USB 2.0 specification
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	Pipes 4 and 5: Can be set
	ATREPM	Auto response mode	Pipes 4 and 5: Can be set
	ACLRM	Auto buffer clear	Pipes 4 to 7: Can be set
	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 26.3.4.6, Response PID
PIPEnTRE	TRENB	Transaction counter enable	Pipes 4 and 5: Can be set
	TRCLR	Transaction counter clear	Pipes 4 and 5: Can be set
PIPEnTRN	TRNCNT	Transaction counter	Pipes 4 and 5: Can be set

26.3.4.1 Pipe control register switching procedures

Do not change the following bits in the pipe control registers when USB communication is enabled (PID = BUF). These bits in the pipe control registers can be changed only when USB communication is prohibited (PID = NAK):

- Bits in DCPCFG and DCPMAXP
- SQCLR and SQSET bits in DCPCTR
- Bits in PIPECFG and PIPEMAXP
- 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 is not set in the CURPIPE[3:0] bits in CFIFOSEL.

- Bits in DCPCFG and DCPMAXP
- Bits in PIPECFG and PIPEMAXP.

To change pipe information, set the CURPIPE[3:0] bits in the CFIFO Port Select Register to a pipe other than the one to be changed. For the DCP, clear the buffer with the BCLR bit in the CFIFO Port Control Register after the pipe information is changed.

26.3.4.2 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 4 and 5 — Set to bulk transfer
- Pipes 6 and 7 — Set to interrupt transfer.

26.3.4.3 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 4 to 7 — 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.

26.3.4.4 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 4 and 5 can be set to any of the maximum pipe sizes defined in the USB 2.0 specification. For pipes 6 and 7, 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 4 and 5 — Set to 8, 16, 32, or 64 for bulk transfers
- Pipes 6 and 7 — Set to a value between 1 and 64.

26.3.4.5 Transaction counter for pipes 4 and 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 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:

- TRENB = 0: Specified transaction counter value can be read
- TRENB = 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.

26.3.4.6 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

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.

The USBFS can write to the PID[1:0] bits because of specific transaction results as described in the following section.

(2) Hardware response PID settings

- 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 transfer.
- BUF setting — There is no BUF writing by the USBFS
- STALL setting — PID = STALL is set in the following cases, and a STALL is returned to transactions:
  - When a receive data packet exceeds the maximum packet size
  - When a control transfer sequence error is detected (DCP only).

26.3.4.7 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 the DCPCTR and PIPEnCTR registers can be used to change the data PID sequence bit.

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.

For ClearFeature requests for transmission or reception, software must set the data PID sequence bit.

26.3.4.8 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 buffer memory, 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.

26.3.4.9 Auto response mode

For bulk transfer pipes 4 and 5, when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (PIPECFG.DIR bit is 0), OUT-NAK mode is invoked, and during an IN transfer (DIR bit is 1), null auto response mode is invoked.

26.3.4.10 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.

26.3.4.11 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 only be set when the buffer is empty. If the INBUFM bit is 1, empty the buffer with the PIPEnCTR.ACLRM bit. Do not write data to the FIFO port while a transition to null auto response mode is made.

To transition from null auto response mode to normal mode, keep the pipe operation disabled (response PID = NAK) for the period of 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).

26.3.5 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 memory has two states depending on whether the access right is assigned to the system (CPU) or the USBFS (SIE).

(1) Buffer status

Table 26.16 and Table 26.17 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 4 and 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 and the INBUFM bit to monitor the FIFO buffer status on the SIE. When write access to the FIFO port by the CPU is slow and the buffer empty status cannot be determined using the BEMP interrupt, software can use the INBUFM bit to confirm the end of transmission.

Table 26.16 Buffer status indicated by BSTS bit

ISEL or DIR	BSTS	FIFO buffer 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 26.17 Buffer status indicated by INBUFM bit

DIR	INBUFM	FIFO buffer 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.

26.3.6 FIFO Buffer Clearing

Table 26.18 shows the methods for clearing the FIFO buffer. The FIFO buffer can be cleared using the BCLR bit in the port control register, or the PIPEnCTR.ACLRM bit.

Single or double buffering can be selected for pipes 4 and 5 in the PIPECFG.DBLB bit.

Table 26.18 Buffer clearing methods

FIFO buffer clearing mode	Clearing FIFO buffer on the CPU	Auto buffer clear mode for discarding all received packets
Register used	CFIFOCTR	PIPEnCTR
Bit used	BCLR	ACLRM
Clearing condition	Cleared by writing 1	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.

26.3.7 FIFO Port Functions

Table 26.19 shows the settings for the FIFO port functions of the USBFS. 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 CFIFO Port Control Register to end writing. To send a zero-length packet, use the BCLR bit to clear the buffer and 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 (the DTLN[8:0] = 000h), so use the BCLR bit to clear the buffer. The length of the receive data can be confirmed with the DTLN[8:0] bits in the CFIFO Port Control Register.

Table 26.19 FIFO port function settings (1 of 2)

Register name	Bit name	Description
CFIFOSEL	RCNT	Selects DTLN[8:0] read mode
	REW	FIFO buffer rewind (re-read, rewrite)
	MBW	FIFO port access bit width
	BIGEND	Selects FIFO port endian
	ISEL	FIFO port access direction (only for DCP)
	CURPIPE	Selects the current pipe

Table 26.19 FIFO port function settings (2 of 2)

Register name	Bit name	Description
CFIFOCTR	BVAL	Ends writing to the FIFO buffer
	BCLR	Clears the FIFO buffer on the CPU
	DTLN	Checks the length of receive data

(1) FIFO port selection

Table 26.20 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 CFIFO Port Select Register. After the pipe is selected, software must check whether the written value can be correctly read 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 using the MBW bit in the CFIFO Port Select Register. The FIFO buffer access direction conforms to the PIPECFG.DIR setting. Only for the DCP that the ISEL bit in the CFIFO Port Select Register determines the direction.

Table 26.20 FIFO port access by pipe

Pipe	Access method	Port that can be used
DCP	CPU access	CFIFO port register
Pipes 4 to 7	CPU access	CFIFO 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 to process the first pipe again. Use the REW bit in the CFIFO Port Select Register for this processing.

If a pipe is selected in the CURPIPE[3:0] bits in the CFIFO 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 CFIFO Port Control Register is 1 after selecting a pipe.

26.3.8 Control Transfers Using DCP

The Default Control Pipe (DCP) is used for data transfers in the control transfer data stage. The FIFO buffer of the DCP is a 64-byte single buffer with a fixed area for both control reads and control writes. The FIFO buffer can only be accessed through the CFIFO port.

26.3.8.1 Control Transfers

(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:

- INTSTS0.VALID bit to 1
- DCPCTR.PID[1:0] bits to NAK
- 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 bit to 0. When the VALID bit is 1, PID = BUF cannot be set, and the data stage cannot be terminated.

Using the VALID bit function, the USBFS can suspend the current processing request when receiving a new USB request during a control transfer and return a response to the latest request.

In addition, the USBFS automatically detects the direction bit, (bmRequestType bit [8]), and the request data length (wLength) of the received USB request. The USBFS distinguishes between control read transfers, control write transfers, and no-data control transfers, and it 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 26.14.

(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 with the CFIFOSEL.ISEL bit. If the transfer data is larger than the size of the DCP FIFO buffer, execute the data transfer using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

(3) Status Stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to BUF.

After this setting is made, the USBFS automatically executes the status stage based on the data transfer direction determined at the setup stage. The 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.

26.3.9 Bulk Transfers (Pipes 4 and 5)

The FIFO buffer usage (single or 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 26.3.3.1, (2) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 1
- Transaction count function (PIPEnTRE.TRENB, TRCLR, and PIPEnTRN.TRNCNT[15:0] bits), see section 26.3.4.5, Transaction counter for pipes 4 and 5 in the receiving direction
- Response PID = NAK function (PIPECFG.SHTNAK bit), see section 26.3.4.8, Response PID = NAK function
- Auto response mode (PIPEnCTR.ATREPM bit), see section 26.3.4.9, Auto response mode.

26.3.10 Interrupt Transfers (Pipes 6 and 7)

The USBFS performs interrupt transfers based on the timing dictated by the host controller.

26.3.11 Pipe Schedule

26.3.11.1 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 6 → pipe 7, and if there is a pipe for which an 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 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.

26.3.12 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.

26.3.12.1 Processing

The following processing is required when operating the USBFS module 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. You can also 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 26.15 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.

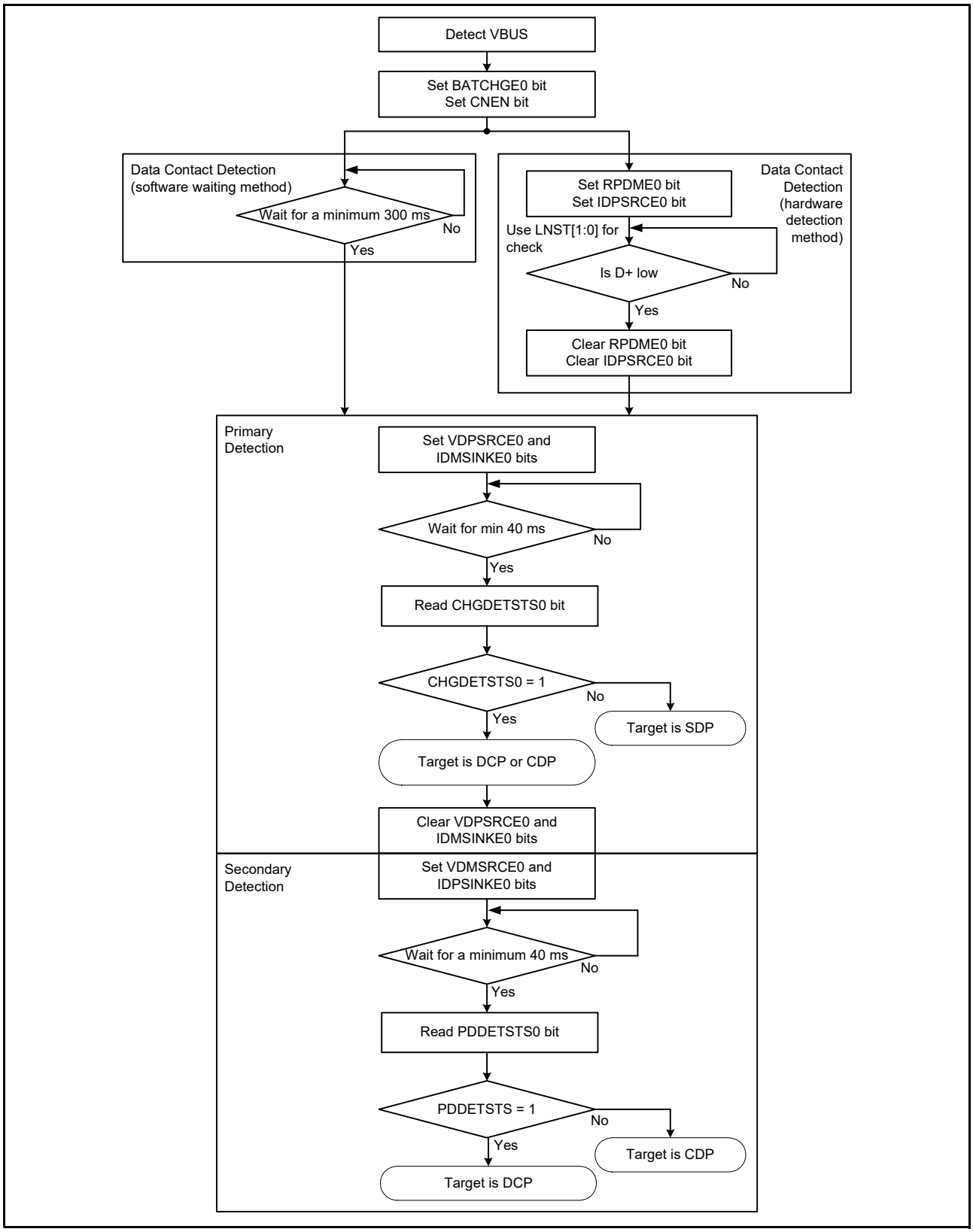


Figure 26.15 Process flow for operating as portable device

26.4 Usage Notes

26.4.1 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 reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

26.4.2 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 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 register in the canceling sequence.

26.4.3 Clearing the Interrupt Status Register after Setting the Port Function

The input buffer is invalid before setting by the PmnPFS.PSEL[4:0] and PmnPFS.PMR ports are set up, so the internal signal is fixed to high or low. The input buffer is enabled after the port is set so that the external pin state is propagated to the MCU. An unexpected interrupt might occur at this time, causing the VBINT bit in INTSTS0 to set to 1. To avoid a malfunction, always clear the INTSTS0 register after setting up the ports.

26.4.4 Notes on 32-pin Products

USBFS-related registers in 32-pin products are reserved bits, therefore do not access these registers.

27. Serial Communications Interface (SCI)

27.1 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. SCI0 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 27.1 lists the SCI specifications, Figure 27.1 shows a block diagram, and Table 27.2 lists the I/O pins by mode.

Table 27.1 SCI specifications (1 of 2)

Parameter		Specifications
Serial communication modes		• Asynchronous • Clock synchronous • Smart card interface • Simple IIC • Simple SPI.
Transfer speed		Bit rate specifiable with the on-chip baud rate generator
Full-duplex communications		• Transmitter: Continuous transmission possible using double-buffering • Receiver: Continuous reception possible using double-buffering.
I/O pins		See Table 27.2
Data transfer		Selectable as LSB-first or MSB-first transfer
Interrupt sources		• Transmit end, transmit data empty, receive data full, receive error, receive data ready, and address match • Completion of generation of a start condition, restart condition, or stop condition (for simple IIC mode).
Module-stop function		Module-stop state can be set for each channel
Snooze end request		SCI0 address mismatch (SCI0_DCUF)
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn*1 pins
	Transmission/Reception	Selectable to 1-stage register or 16-stage FIFO (only SCI0 supports 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*1 pin inputs

Table 27.1 SCI specifications (2 of 2)

Parameter		Specifications
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*1 pins
	Transmission/reception	Selectable to 1-stage register or 16-stage FIFO (only SCI0 supports 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^2C bus format (MSB-first only)
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 400 kbps
	Noise cancellation	The signal paths from input on the SCLn*1 and SDAn*1 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*1 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, 9).
Note 2. Using this event link function is prohibited when the FIFO operation is selected in asynchronous mode.

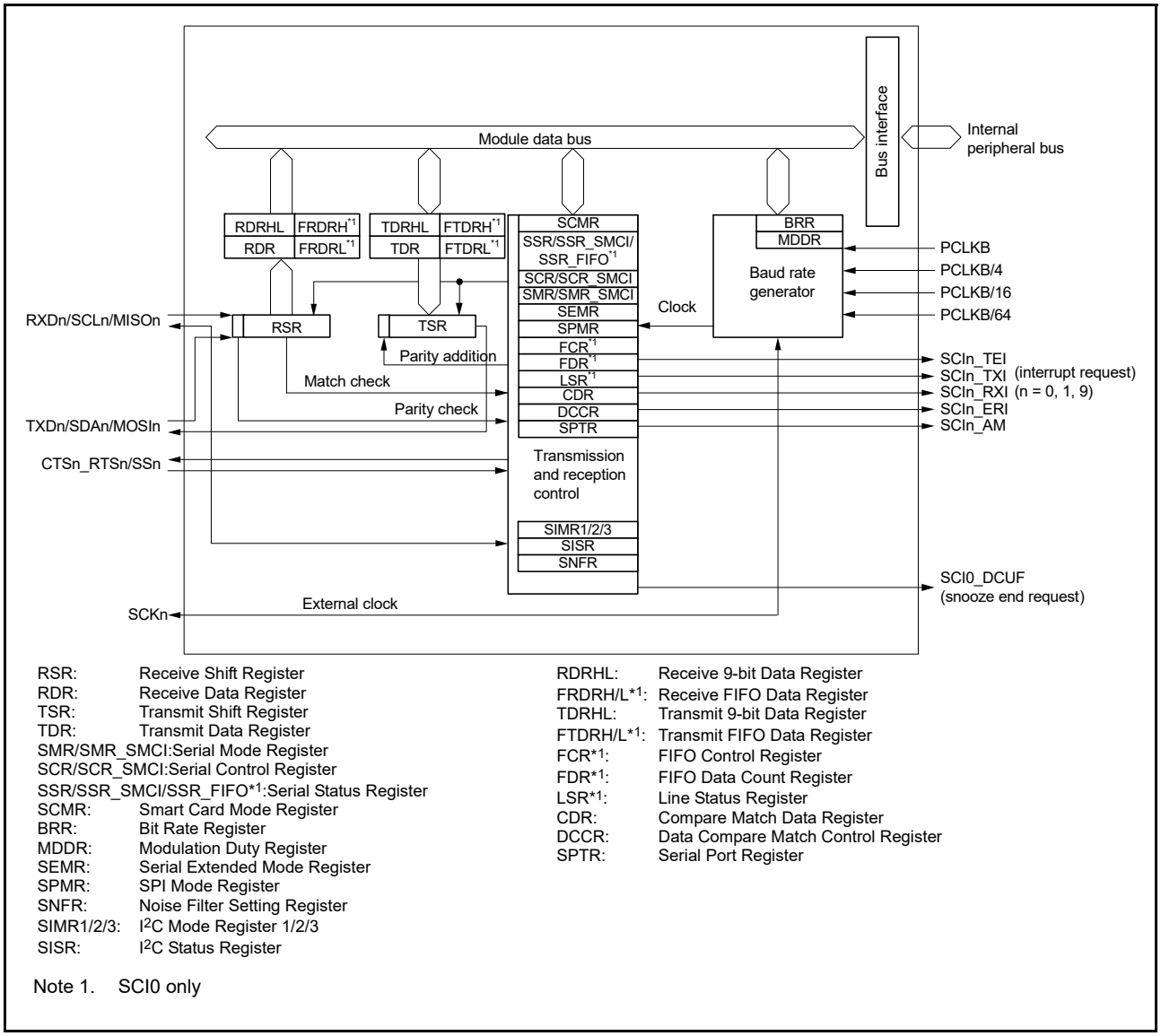


Figure 27.1 SCI block diagram

Table 27.2 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 I2C clock input/output SCI0 slave transmit data input/output
	TXD0/SDA0/MOSI0	Input/Output	SCI0 transmit data output SCI0 I2C data input/output SCI0 master transmit data input/output
	SS0/CTS0_RTS0	Input/Output	SCI0 chip select input, active-low SCI0 transfer start control input/output, active-low

Table 27.2 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^{2}C clock input/output SCI1 slave transmit data input/output
	TXD1/SDA1/ MOSI1	Input/Output	SCI1 transmit data output SCI1 I^{2}C 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
SCI9	SCK9	Input/Output	SCI9 clock input/output
	RXD9/SCL9/ MISO9	Input/Output	SCI9 receive data input SCI9 I^{2}C clock input/output SCI9 slave transmit data input/output
	TXD9/SDA9/ MOSI9	Input/Output	SCI9 transmit data output SCI9 I^{2}C 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

27.2 Register Descriptions

27.2.1 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.

27.2.2 Receive Data Register (RDR)

Address(es): SCI0.RDR 4007 0005h, SCI1.RDR 4007 0025h, 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 only once after a receive data full interrupt (SCIn_RXI) occurs.

Note: If the next frame of data is received before reading the received data from RDR, an overrun error occurs. The CPU cannot write to the RDR.

27.2.3 Receive 9-Bit Data Register (RDRHL)

Address(es): SCI0.RDRHL 4007 0010h, SCI1.RDRHL 4007 0030h, 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 receive data. Use this register when asynchronous mode and 9-bit data length are selected.

The lower 8 bits of RDRHL are the shadow register of RDR, so access to RDRHL affects the RDR register. Access to the RDRHL register is prohibited if 7-bit or 8-bit data length is selected.

After one frame of data is received, the received data is transferred from RSR to the RDR or RDRHL register, allowing the RSR register to receive more data.

The RSR and RDRHL registers form a double-buffered structure to enable continuous reception. RDRHL should be read only when a receive data full interrupt (SCIn_RXI) request is issued. An overrun error occurs when the next frame of data is received before the received data is read from RDRHL. The CPU cannot write to the RDRHL register. Bits [15:9] of the RDRHL register are fixed to 0. These bits are read as 0. The write value should be 0.

27.2.4 Receive FIFO Data Register H, L, HL (FRDRH, FRDRL, FRDRHL)

Receive FIFO Data Register H (FRDRH)

Address(es): SCI0.FRDRH 4007 0010h

Receive FIFO Data Register L (FRDRL)

Address(es): SCI0.FRDRL 4007 0011h

Receive FIFO Data Register HL (FRDRHL)

Address(es): SCI0.FRDRHL 4007 0010h

	SCI0.FRDRHL															
	SCI0.FRDRH								SCI0.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 mode or clock synchronous mode, 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	This flag is the same as SSR_FIFO.DR: 0: Receiving is in progress, or no received data remains in FRDRH and FRDRL after successfully completed reception 1: Next receive data is not received for a period after successfully completed reception.	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	This flag is the same as SSR_FIFO.ORER: 0: No overrun error occurred 1: An overrun error occurred.	R*1
b14	RDF	Receive FIFO Data Full Flag	This flag is the same as SSR_FIFO.RDF: 0: The amount of receive data written in FRDRH and FRDRL is less than the specified receive triggering number 1: The amount of receive data written in FRDRH and FRDRL is equal to or greater than the specified receive triggering number.	R*1
b15	—	Reserved	This bit is read as 0	R

Note 1. If this flag is read, it indicates the same value as that read from the SSR_FIFO register. Write 0 to the SSR_FIFO register to clear the flag.

FRDRHL is a 16-bit register that consists of the 8-bit FRDRH and FRDRL registers.

FRDRH and FRDRL constitute a 16-stage FIFO register that stores serial receive data and related status information readable by software. This register is only valid in asynchronous mode, including multi-processor mode or clock synchronous mode.

The SCI completes reception of one frame of serial data by transferring the received data from the 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 of receive data, subsequent serial receive data is lost. The CPU can read from FRDRH and FRDRL but cannot write to them.

Reading 1 from the RDF, ORER, or DR flag of the FRDRH register is the same as reading from those bits in the SSR_FIFO register. When writing 0 to clear a flag in the SSR_FIFO register after reading the FRDRH register, write 0 only to the flag that is to be cleared and write 1 to the other flags.

When reading both the FRDRH and FRDRL registers, read in the order from FRDRH to FRDRL. The FRDRHL register can be accessed in 16-bit units.

27.2.5 Transmit Data Register (TDR)

Address(es): SCI0.TDR 4007 0003h, SCI1.TDR 4007 0023h, 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).

27.2.6 Transmit 9-Bit Data Register (TDRHL)

Address(es): SCI0.TDRHL 4007 000Eh, SCI1.TDRHL 4007 002Eh, SCI9.TDRHL 4007 012Eh

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

TDRHL is a 16-bit register that stores transmit data. Use this register when asynchronous mode and 9-bit data length are selected.

The lower 8 bits of TDRHL are the shadow register of TDR , so access to TDRHL affects the TDR register. Access to the TDRHL register is prohibited if 7-bit or 8-bit data length is selected. When empty space is detected in the TSR register, the transmit data stored in the TDRHL registers is transferred to TSR and transmission starts.

The TSR and TDRHL registers have a double-buffered structure to support continuous transmission. When the next data to be transmitted is stored in TDRHL after one frame of data is transmitted, the transmitting operation continues by transferring the data from the TDRHL register to the TSR register.

The CPU can read and write to the TDRHL register. Bits [15:9] in TDRHL are fixed to 1. These bits are read as 1. The write value should be 1.

Write transmit data to the TDRHL register only once when a transmit data empty interrupt (SCIn_TXI) request is issued.

27.2.7 Transmit FIFO Data Register H, L, HL (FTDRH, FTDRL, FTDRHL)

Transmit FIFO Data Register H (FTDRH)

Address(es): SCI0.FTDRH 4007 000Eh

Transmit FIFO Data Register L (FTDRL)

Address(es): SCI0.FTDRL 4007 000Fh

Transmit FIFO Data Register HL (FTDRHL)

Address(es): SCI0.FTDRHL 4007 000Eh

	SCI0.FTDRHL															
	SCI0.FTDRH								SCI0.FTDRL							
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MPBT	TDAT[8:0]								
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b8 to b0	TDAT[8:0]	Serial Transmit Data	Serial transmit data, valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode, with FIFO selected	W
b9	MPBT	Multi-Processor Transfer Bit Flag	Specifies the multi-processor bit in the transmission frame: 0: Data transmission cycle 1: ID transmission cycle. 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 8-bit FTDRH and FTDRL registers.

FTDRH and FTDRL constitute a 16-stage FIFO register that stores data for serial transmission and multi-processor transfer bit. This register is valid only in asynchronous mode, including multi-processor mode or clock synchronous

mode.

When the SCI detects that the Transmit Shift Register (TSR) is empty, it transmits data written in the FTDRH and FTDRL registers to the TSR register and starts serial transmission. Continuous serial transmission is executed until no transmit data is left in FTDRH and FTDRL. When FTDRHL is full of transmit data, no more data can be written. If writing new data is attempted, the data is ignored. The CPU can write to the FTDRH and FTDRL registers but cannot read them.

When writing to both the FTDRH and FTDRL registers, write in the order from FTDRH to FTDRL.

MPBT flag (Multi-Processor Transfer Bit Flag)

The MPBT flag specifies the value of the multi-processor bit of the transmit frame. When FCR.FM = 1, SSR.MPBT is not valid.

27.2.8 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 the TSR.

27.2.9 Serial Mode Register (SMR) for Non-Smart Card Interface mode (SCMR.SMIF = 0)

Address(es): SCI0.SMR 4007 0000h, SCI1.SMR 4007 0020h, 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: PCLKB clock (n = 0)*1 0 1: PCLKB/4 clock (n = 1)*1 1 0: PCLKB/16 clock (n = 2)*1 1 1: PCLKB/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: Even parity is selected 1: Odd parity is selected.	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 the transmit/receive character length in combination with the SCMR.CHR1 bit: 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 27.2.17, 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 SCR.TE = 0 and SCR.RE = 0 (both serial transmission and reception are disabled).

The SMR register sets the communication format and clock source for the on-chip baud rate generator.

CKS[1:0] bits (Clock Select)

The CKS[1:0] bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of these bits and the baud rate, see section 27.2.17, 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 PM bit setting 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 at reception.

Regardless of the PE bit setting, the parity bit is not added or checked in multi-processor format.

CHR bit (Character Length)

The CHR bit selects the data length for transmission and reception in combination with the SCMR.CHR1 bit.

In modes other than asynchronous 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.

27.2.10 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, 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: PCLKB clock (n = 0)*1 0 1: PCLKB/4 clock (n = 1)*1 1 0: PCLKB/16 clock (n = 2)*1 1 1: PCLKB/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 SCMR.BCP2 bit. Table 27.3 lists the combinations of the SCMR.BCP2 and SMR.BCP[1:0] bits.	R/W*2
b4	PM	Parity Mode	Valid only when the PE bit is 1: 0: Even parity is selected 1: Odd parity is selected.	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 27.2.17, 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 27.2.17, 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 27.6.4, Receive Data Sampling Timing and Reception Margin.

Table 27.3 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	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (initial value)

Table 27.3 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	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in BRR, see section 27.2.17, 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 27.6.2, Data Format (Except in Block Transfer Mode).

PE bit (Parity Enable)

Set the PE bit to 1. The parity bit is added to the transmit data before transmission, and the parity bit is checked at reception.

BLK bit (Block Transfer Mode)

Set the BLK bit to 1 to enable block transfer mode operation. For details, see section 27.6.3, Block Transfer Mode.

GM bit (GSM Mode)

Set the GM bit to 1 to enable GSM mode operation.

In GSM mode, the SSR_SMCI.TEND flag set timing is moved forward to 11.0 ETUs (elementary time unit = 1-bit transfer time) from the start bit, and the clock output control function is enabled. For details, see section 27.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 27.6.8, Clock Output Control.

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

Address(es): SCI0.SCR 4007 0002h, SCI1.SCR 4007 0022h, 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 based on 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 the SEMR.ABCS bit is 0. Input a clock signal with a frequency eight times the bit rate when the SEMR.ABCS bit is 1. Clock synchronous mode: b1 b0 0 x: Internal clock. The SCKn pin functions as the clock output pin 1 x: External clock. The SCKn pin functions as the clock input pin.	R/W*1
b2	TEIE	Transmit End Interrupt Enable	0: An SCIn_TEI interrupt request is disabled 1: An SCIn_TEI interrupt request is enabled.	R/W

Bit	Symbol	Bit name	Description	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: An SCIn_TXI interrupt request is disabled 1: An SCIn_TXI interrupt request is enabled.	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0, and the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written to TE and RE. When the SMR.CM bit is 0 and the SIMR1.IICM bit is 0, writing is enabled under any condition.

Note 3. When writing a new value to a bit other than the MPIE bit of this register during multi-processor mode (SMR.MP = 1), write 0 to MPIE bit using the store instruction to avoid accidentally setting the MPIE bit to 1 by a read-modify-write operation when using a bit manipulation instruction.

The SCR 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 SCIn_TEI interrupt request.

In simple IIC mode, SCIn_TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STIn). In this case, the TEIE bit can be used to enable or disable the STI.

MPIE bit (Multi-Processor Interrupt Enable)

When the MPIE bit is set to 1 and data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags RDRF, ORER, 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 27.4, Multi-Processor Communications Function.

When the MPB bit in the SSR is 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 MPB bit is 1, the MPIE bit is automatically set to 0, SCIn_RXI and SCIn_ERI interrupt requests are enabled (if the RIE bit in SCR is set to 1), and setting of the ORER and FER flags to 1 is enabled.

Set MPIE to 0 if the multi-processor communications function is not used.

RE bit (Receive Enable)

The RE bit enables or disables serial reception.

When this bit is set to 1, serial reception starts by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Set the reception format in the SMR register 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 the SSR register 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 this 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.

Note: To switch the TIE bit value from 0 to 1 in FIFO mode, set the TIE and TE bits to 1 simultaneously or set the TIE bit to 1 when TE = 1. When TE = 0 in FIFO mode, setting the TIE bit to 1 is prohibited.

27.2.12 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, 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: Output clock 1 x: Setting prohibited. • When GM in SMR_SMCI = 1: b1 b0 0 0: Output fixed low x 1: Output clock 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.

The SCR_SMCI sets transmission and reception control, interrupt control, and clock source selection for transmission and reception.

For details on interrupt requests, see section 27.10, 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 27.6.8, Clock Output Control.

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 value is 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, 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.

27.2.13 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, 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	Value of 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 being 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 in RDR register 1: Received data in RDR register.	R/W*1
b7	TDRE	Transmit Data Empty Flag	0: Transmit data in TDR register 1: No transmit data in TDR register.	R/W*1

Note 1. Only 0 can be written to clear the flag after reading 1.

The SSR register provides the SCI status flags and transmission/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 on transmission of the tail-end bit of a character.

[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 1 is read when the SCR.TE bit is 1.

PER flag (Parity Error Flag)

The PER flag indicates that a parity error occurred during reception in asynchronous mode and the reception ended abnormally.

[Setting condition]

- When a parity error is detected during reception in asynchronous mode and the address match function is disabled (DCCR.DCME = 0).

  Although receive data is transferred to the RDR register when the parity error occurs, no SCIn_RXI interrupt request occurs. When the PER flag is set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to the PER flag after 1 is read. After writing 0 to this flag, read it to verify that its value is 0.

When the RE bit in SCR 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 occurred during reception in asynchronous mode and the reception ended abnormally.

[Setting condition]

- When 0 is sampled as the stop bit during reception in asynchronous mode and the address match function is disabled (DCCR.DCME = 0).

  In 2-stop-bit mode, only the first stop bit is checked, but the second stop bit is not checked. Although receive 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 receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to the FER flag after 1 is read. After writing 0 to this flag, read it to verify that its value is 0.

When the RE bit in SCR 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 occurred during reception and the reception ended abnormally.

[Setting condition]

- When the next data is received before receive data that does not have a parity error and a framing error is read from RDR.

  In RDR, data received prior to an overrun error occurrence is saved, but data received after the overrun error is lost. When the ORER flag is set to 1, data received is not forwarded to the RDR register. In clock synchronous mode, serial transmission and reception are stopped.

[Clearing condition]

- When 0 is written to the ORER flag after 1 is read. After writing 0 to this flag, read it to verify that its value is 0.

When the RE bit in SCR is set to 0, the ORER flag is not affected and retains its previous value.

RDRF flag (Receive Data Full Flag)

The RDRF flag indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing conditions]

- When 0 is written to RDRF after 1 is read
- When data is read from the RDR register.

Note: Do not clear the RDRF flag by accessing RDRF 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.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 flag after 1 is read
- When the SCR.TE bit is 1, and data is written to the TDR register.

Note: Do not clear the TDRE flag by accessing TDRE in the SSR register unless communication is aborted.

27.2.14 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

	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: Reception is in progress, or no received data remains in FRDRHL after successfully completed reception (receive FIFO is empty) 1: Next receive data is not received for a period after successfully completed reception, 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 being 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 less than the specified receive triggering number 1: The amount of receive data written in FRDRHL is equal to or greater than the specified receive triggering number.	R/W*1
b7	TDFE	Transmit FIFO Data Empty Flag	0: The amount of transmit data written in FTDRHL exceeds the specified transmit triggering number 1: The amount of transmit data written in FTDRHL is equal to or less than the specified transmit triggering number.	R/W*1

Note 1. Only 0 can be written to clear the flag after reading 1.

The SSR_FIFO register provides SCI with FIFO mode status flags.

DR flag (Receive Data Ready Flag)

The DR flag indicates that the amount of data stored in the Receive FIFO Data Register (FRDRHL) falls below the

specified receive triggering number, and that no next data is received after 15 ETUs (element 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 operation 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 next data is received after 15 ETUs*1 from the last stop bit, and the SSR_FIFO.FER and SSR_FIFO.PER flags are 0.

[Clearing conditions]

- When 1 is read from DR and 0 is written after all received data are read
- When the FCR.FM bit is changed from 0 to 1.

Note 1. This is equivalent to 1.5 frames in the 8-bit format with one stop bit (ETU).
The DR flag is only set to 1 when FIFO is selected in asynchronous mode, including multi-processor mode. It is not set to 1 in other operation modes.

TEND flag (Transmit End Flag)

The TEND flag indicates that FTDRHL does not contain valid data when transmitting the last bit of a serial character, so 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, when the SCR.TE bit is 1
- When the FCR.FM bit is changed 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 1 is read.

The receive operation is continuous when receive data is stored in the FRDRHL register even when a parity error occurs during reception.

When the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and keeps its previous value.

FER flag (Framing Error Flag)

The FER flag indicates whether there is a framing error in the data read from the FRDRHL register in asynchronous mode when the address match function is disabled (DCCR.DCME = 0).

[Setting condition]

- When 0 is sampled as the stop bit during reception and the address match function is disabled (DCCR.DCME = 0).

[Clearing condition]

- When 0 is written to FER after 1 is read.

The receive operation is continuous when receive data is stored to the FRDRHL register even when a framing error occurs during reception.

When the SCR.RE bit is set to 0 (serial reception is disabled), the FER flag is not affected and keeps its previous value.

ORER flag (Overrun Error Flag)

The ORER flag indicates that the receive operation stopped abnormally because an overrun error occurred.

[Setting condition]

- When the next serial reception completes while the receive FIFO is full with 16-byte receive data.

[Clearing condition]

- When 0 is written after 1 is read.

When the SCR.RE bit is set to 0 (serial reception is disabled), the ORER flag is not affected and keeps its previous value.

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 equals 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
- When FRDRHL is read by the DTC, but only when block transfer is the last transmission
- When the setting and clearing conditions occur at the same time, the RDF flag is 0. After that, when the amount of data stored in the FRDRHL register is equal to or greater than the RTRG value, RDF is set to 1 after 1 PCLKB.

Note: Do not clear the RDF flag by accessing RDF 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 the FTDRHL register into the TSR register, the amount of data in FTDRHL is less than the specified transmit triggering number, and writing of transmit data to FTDRHL is enabled.

[Setting conditions]

- When the TE bit in SCR is 0
- When the amount of transmit data written in FTDRHL is equal to or less than the specified transmit triggering number*1.

[Clearing conditions]

- When writing to FTDRHL is executed on the last transmission while the DTC is activated
- When 0 is written to the TDFE flag after 1 is read.
  The setting conditions are given priority when TE = 0. When the setting and clearing conditions occur at the same time, the TDFE flag is 0. After that, when the amount of data stored in the FTDRHL register is equal to or greater than the TTRG value, TDFE is set to 1 after 1 PCLKB.

Note: Do not clear the TDFE flag by accessing TDFE 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.

27.2.15 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, 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 being transmitted 1: Character transfer is complete.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error occurred.	R/W*1
b4	ERS	Error Signal Status Flag	0: Low error signal is not sampled 1: Low error signal is sampled.	R/W*1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error occurred.	R/W*1
b6	RDRF	Receive Data Full Flag	0: No received data in RDR register 1: Received data in RDR register.	R/W*1
b7	TDRE	Transmit Data Empty Flag	0: Transmit data in TDR register 1: No transmit data in TDR register.	R/W*1

Note 1. Only 0 can be written to clear the flag after 1 is read.

The SSR_SMCI register provides SCI with smart card interface mode status flags.

TEND flag (Transmit End Flag)

When there is no error signal from the receiving side, the TEND flag is set to 1 when more data is ready to be transferred to the TDR register.

[Setting conditions]

- When the SCR_SMCI.TE bit = 0 (serial transmission is disabled). When the SCR_SMCI.TE bit changes from 0 to 1, the TEND flag is not affected and 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 when the SCR_SMCI.TE bit is 1
- When 0 is written to TDRE after 1 is read when the SCR_SMCI.TE bit is 1.

PER flag (Parity Error Flag)

The PER flag indicates that a parity error occurred during reception in asynchronous mode and the reception ended abnormally.

[Setting condition]

- When a parity error is detected during reception. Although receive data is transferred to RDR when a parity error occurs, no SCIn_RXI interrupt request occurs. After the PER flag is set to 1, the next receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after 1 is read. After writing 0 to the PER flag, read it to verify that its value is 0.

When the RE bit in SCR_SMCI is set to 0 (serial reception is disabled), the PER flag is not affected and keeps its previous value.

ERS flag (Error Signal Status Flag)

[Setting condition]

- When a low error signal is sampled.

[Clearing condition]

- When 0 is written to ERS after 1 is read.

ORER flag (Overrun Error Flag)

The ORER flag indicates that an overrun error occurred during reception and the reception ended abnormally.

[Setting condition]

- When the next data is received before receive data that does not have a parity error is read from the RDR register. The data received before an overrun error occurred is saved in the RDR, but data received after the overrun error is lost. When the ORER flag is set to 1, receive data is not forwarded to the RDR register.

[Clearing condition]

- When 0 is written to ORER after 1 is read. 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 the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing conditions]

- When 0 is written to RDRF after 1 is read
- When data is read from the RDR register.

Note: Do not clear the RDRF flag by accessing RDRF 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 TDRE after 1 is read
- When the SCR_SMCI.TE bit is 1 and data is written to the TDR register.

Note: Do not clear the TDRE flag by accessing TDRE in the SSR register unless communication is aborted.

27.2.16 Smart Card Mode Register (SCMR)

Address(es): SCI0.SCMR 4007 0006h, SCI1.SCMR 4007 0026h, 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 register contents are transmitted as is. Receive data is stored as received in the RDR 1: TDR register contents are inverted before transmitted. Received data is stored in inverted form in the RDR. This bit can be used in the following modes: • Smart card interface mode • Asynchronous mode including 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 including 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	Valid only in asynchronous mode*2. Selects the transmit/receive character length in combination with the SMR.CHR bit: 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 27.4 lists the combinations of the SCMR.BCP2 and SMR_SMCI.BCP[1:0] bits.	R/W*1

Note 1. Writable only when the TE and RE bits in SCR/SCR_SMCI are 0 (both serial transmission and reception are disabled).
Note 2. The setting is invalid and a fixed data length of 8 bits is used in modes other than asynchronous mode.
Note 3. LSB-first should be selected and the value of the MSB bit [7] in TDR cannot be transmitted.

The SCMR register selects the smart card interface and communication format.

SMIF bit (Smart Card Interface Mode Select)

Setting the SMIF bit to 1 selects the smart card interface mode. Setting it to 0 selects all 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 27.4 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. For S, see section 27.2.17, Bit Rate Register (BRR).

27.2.17 Bit Rate Register (BRR)

Address(es): SCI0.BRR 4007 0001h, SCI1.BRR 4007 0021h, SCI9.BRR 4007 0121h

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

BRR is an 8-bit register that adjusts the bit rate.

As each SCI channel has independent baud rate generator control, different bit rates can be set for each. Table 27.5 shows the relationship between the setting (N) in the BRR and the bit rate (B) for asynchronous mode, multiprocessor transfer, clock synchronous mode, smart card interface mode, simple SPI mode, and simple IIC mode.

The initial value of the BRR register is FFh. The 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 27.5 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	$N = \frac{PCLKB \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{ \frac{PCLKB \times 10^6}{B \times 64 \times 2^{2n-1} \times (N + 1)} - 1 \} \times 100$
	1	0	0	$N = \frac{PCLKB \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{ \frac{PCLKB \times 10^6}{B \times 32 \times 2^{2n-1} \times (N + 1)} - 1 \} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLKB \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{ \frac{PCLKB \times 10^6}{B \times 16 \times 2^{2n-1} \times (N + 1)} - 1 \} \times 100$
	Don't care	Don't care	1	$N = \frac{PCLKB \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{ \frac{PCLKB \times 10^6}{B \times 12 \times 2^{2n-1} \times (N + 1)} - 1 \} \times 100$
Clock synchronous, simple SPI				$N = \frac{PCLKB \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	-
Smart card interface				$N = \frac{PCLKB \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error\ (\%) = \{ \frac{PCLKB \times 10^6}{B \times S \times 2^{2n+1} \times (N + 1)} -1 \} \times 100$
Simple IIC*1				$N = \frac{PCLKB \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	-

B: Bit rate (bps).
N: BRR setting for on-chip baud rate generator ($0 \leq N \leq 255$).
PCLKB: Operating frequency (MHz).
n and S: Determined by the SMR/SMR_SMCI and SCMR register settings as listed in Table 27.7 and Table 27.8.

Note 1. Adjust the bit rate so that the high and low-level widths of the SCLn output in simple IIC mode satisfy the I^2C bus standard.

Table 27.6 Calculating widths at high and low level for SCL

Mode	SCL	Formula (result in seconds)
Simple IIC	Width at high level (minimum value)	$(N + 1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLKB \times 10^6}$
	Width at low level (minimum value)	$(N + 1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLKB \times 10^6}$

Table 27.7 Clock source settings

SMR or SMR_SMCI.CKS[1:0] bit setting		
CKS[1:0] bits	**Clock source**	**n**
0 0	PCLKB clock	0
0 1	PCLKB/4 clock	1
1 0	PCLKB/16 clock	2
1 1	PCLKB/64 clock	3

Table 27.8 Base clock settings in smart card interface mode

SCMR.BCP2 bit setting	SMR_SMCI.BCP[1:0] bit setting	Base clock cycles for 1-bit period	S
0	0 0	93 clock cycles	93
0	0 1	128 clock cycles	128
0	1 0	186 clock cycles	186
0	1 1	512 clock cycles	512
1	0 0	32 clock cycles	32
1	0 1	64 clock cycles	64
1	1 0	372 clock cycles	372
1	1 1	256 clock cycles	256

Table 27.9 and Table 27.10 list examples of BRR (N) settings in asynchronous mode. Table 27.11 lists the maximum bit rate selectable for each operating frequency. Table 27.15 lists examples of BRR (N) settings in smart card interface mode.

Table 27.17 lists examples of BRR (N) settings in simple IIC mode. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, see section 27.6.4, Receive Data Sampling Timing and Reception Margin. Table 27.12 and Table 27.14 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 27.16. When both of those bits are set to 1, the bit rate becomes four times the listed value.

Table 27.9 Examples of BRR settings for different bit rates in asynchronous mode (1)

Bit rate (bps)	Operating frequency PCLKB (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 PCLKB (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

Bit rate (bps)	Operating frequency PCLKB (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
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 27.10 Examples of BRR settings for various bit rates in asynchronous mode (2)

Bit rate (bps)	Operating frequency PCLKB (MHz)		
	20		
	n	N	Error (%)
110	3	88	-0.25
150	3	64	0.16
300	2	129	0.16
600	2	64	0.16
1200	1	129	0.16
2400	1	64	0.16
4800	0	129	0.16
9600	0	64	0.16
19200	0	32	-1.36
31250	0	19	0.00
38400	0	15	1.73

Note: In this example, SEMR.ABCS = 0, SEMR.ABCSE = 0, and SEMR.BGDM = 0.
When either the ABCS or BGDM bit is set to 1, the bit rate doubles.
When both ABCS and BGDM are set to 1, the bit rate quadruples.

Table 27.11 Maximum bit rate for each operating frequency in asynchronous mode (1 of 2)

PCLKB (MHz)	SEMR settings					Maximum bit rate (bps)	PCLKB (MHz)	SEMR settings					Maximum bit rate (bps)
	BGDM bit	ABCS bit	ABCSE bit	n	N			BGDM bit	ABCS bit	ABCSE bit	n	N	
8	0	0	0	0	0	250000	17.2032	0	0	0	0	0	537600
		1	0	0	0	500000			1	0	0	0	1075200
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1000000			1	0	0	0	2150400
	Don't care	Don't care	1	0	0	1333333		Don't care	Don't care	1	0	0	2867200
9.8304	0	0	0	0	0	307200	18	0	0	0	0	0	562500
		1	0	0	0	614400			1	0	0	0	1125000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1228800			1	0	0	0	2250000
	Don't care	Don't care	1	0	0	1638400		Don't care	Don't care	1	0	0	3000000

Table 27.11 Maximum bit rate for each operating frequency in asynchronous mode (2 of 2)

PCLKB (MHz)	SEMR settings BGDM bit	ABCS bit	ABCSE bit	n	N	Maximum bit rate (bps)	PCLKB (MHz)	SEMR settings BGDM bit	ABCS bit	ABCSE bit	n	N	Maximum bit rate (bps)
10	0	0	0	0	0	312500	19.6608	0	0	0	0	0	614400
		1	0	0	0	625000			1	0	0	0	1228800
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1250000			1	0	0	0	2457600
	Don't care	Don't care	1	0	0	1666666		Don't care	Don't care	1	0	0	3276800
12	0	0	0	0	0	375000	20	0	0	0	0	0	625000
		1	0	0	0	750000			1	0	0	0	1250000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1500000			1	0	0	0	2500000
	Don't care	Don't care	1	0	0	2000000		Don't care	Don't care	1	0	0	3333333
12.288	0	0	0	0	0	384000	-	-	-	-	-	-	-
		1	0	0	0	768000							
	1	0	0	0	0								
		1	0	0	0	1536000							
	Don't care	Don't care	1	0	0	2048000							
14	0	0	0	0	0	437500	-	-	-	-	-	-	-
		1	0	0	0	875000							
	1	0	0	0	0								
		1	0	0	0	1750000							
	Don't care	Don't care	1	0	0	2333333							
16	0	0	0	0	0	500000	-	-	-	-	-	-	-
		1	0	0	0	1000000							
	1	0	0	0	0								
		1	0	0	0	2000000							
	Don't care	Don't care	1	0	0	2666666							

Table 27.12 Maximum bit rate with external clock input in asynchronous mode

PCLKB (MHz)	External input clock (MHz)	Maximum bit rate (bps) SEMR.ABCS = 0	SEMR.ABCS = 1
8	2.0000	125000	250000
9.8304	2.4576	153600	307200
10	2.5000	156250	312500
12	3.0000	187500	375000
12.288	3.0720	192000	384000
14	3.5000	218750	437500
16	4.0000	250000	500000
17.2032	4.3008	268800	537600
18	4.5000	281250	562500
19.6608	4.9152	307200	614400
20	5.0000	312500	625000

Table 27.13 BRR settings for different bit rates in clock synchronous and simple SPI modes

	Operating frequency PCLKB (MHz)							
	8		10		16		20	
Bit rate (bps)	n	N	n	N	n	N	n	N
110	x	x	x	x	x	x	x	x
250	3	124	—	—	3	249	x	x
500	2	249	—	—	3	124	—	—
1 k	2	124	—	—	2	249	—	—
2.5 k	1	199	1	249	2	99	2	124
5 k	1	99	1	124	1	199	1	249
10 k	0	199	0	249	1	99	1	124
25 k	0	79	0	99	0	159	0	199
50 k	0	39	0	49	0	79	0	99
100 k	0	19	0	24	0	39	0	49
250 k	0	7	0	9	0	15	0	19
500 k	0	3	0	4	0	7	0	9
1 M	0	1			0	3	0	4
2.5 M	x	x	0	0*1	x	x	0	1
5 M	x	x	x	x	x	x	0	0*1
7.5 M	x	x	x	x	x	x	x	x

x: 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 27.14 Maximum bit rate with external clock input in clock synchronous and simple SPI modes

PCLKB (MHz)	External input clock (MHz)	Maximum bit rate (Mbps)
8	1.3333	1.3333333
10	1.6667	1.6666667
12	2.0000	2.0000000
14	2.3333	2.3333333
16	2.6667	2.6666667
18	3.0000	3.0000000
20	3.3333	3.3333333

Table 27.15 BRR settings for different bit rates in smart card interface mode, n = 0, S = 372

	Operating frequency PCLKB (MHz)											
	7.1424			10.00			10.7136			13.00		
bit rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	0	0.00	0	1	-30	0	1	-25	0	1	-8.99

	Operating frequency PCLKB (MHz)											
	14.2848			16.00			18.00			20.00		
bit rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	0.00	0	1	12.01	0	2	-15.99	0	2	-6.66

Table 27.16 Maximum bit rate for each operating frequency in smart card interface mode, S = 32

PCLKB (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

Table 27.17 BRR settings for different bit rates in simple IIC mode

	Operating frequency PCLKB (MHz)								
	8			10			16		
Bit rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	0	24	0.0	0	30	0.8	1	12	-3.8
25 k	0	9	0.0	0	12	-3.8	1	4	0.0
50 k	0	4	0.0	0	5	4.2	1	2	-16.7
100 k*1	0	2	-16.7	0	3	-21.9	0	4	0.0
250 k	0	0	0.0	0	0	25.0	0	1	0.0
350 k	—	—	—	—	—	—	—	—	—
400 k*1	—	—	—	—	—	—	—	—	—

	Operating frequency PCLKB (MHz)		
	20		
Bit rate (bps)	n	N	Error (%)
10 k	1	15	-2.3
25 k	1	5	4.2
50 k	1	2	4.2
100 k*1	0	6	-10.7
250 k	0	2	-16.7
350 k	0	1	-10.7
400 k*1	0	1	-21.9

Note 1. The bit rate of 100 kbps and 400 kbps indicates the set value at which the error is on the minus side.

Table 27.18 Minimum widths at high and low level for SCL at different bit rates in simple IIC mode

	Operating frequency PCLKB (MHz)								
	8			10			16		
Bit rate (bps)	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
25 k	0	9	17.50/20.00	0	12	18.2/20.80	1	4	17.50/20.00
50 k	0	4	8.75/10.00	0	5	8.40/9.60	1	2	10.50/12.00
100 k	0	2	5.25/6.00	0	3	5.60/6.40	0	4	4.38/5.00
250 k	0	0	1.75/2.00	0	0	1.40/1.60	0	1	1.75/2.00
350 k	—	—	—	—	—	—	—	—	—
400 k	—	—	—	—	—	—	—	—	—

Bit rate (bps)	Operating frequency PCLKB (MHz)		
	20		
	n	N	Min. Widths at High/Low Level for SSL (μs)
10 k	1	15	44.80/51.20
25 k	1	5	16.80/19.20
50 k	1	2	8.40/9.60
100 k	0	6	4.90/5.60
250 k	0	2	2.10/2.40
350 k	0	1	1.40/1.60
400 k	0	1	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 27.17.

27.2.18 Modulation Duty Register (MDDR)

Address(es): SCI0.MDDR 4007 0012h, SCI1.MDDR 4007 0032h, 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 27.19 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 27.19 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	$N = \frac{PCLKB \times 10^6}{64 \times 2^{2n-1} \times (256 / M) \times B} - 1$	$Error (\%) = \{ \frac{PCLKB \times 10^6}{B \times 64 \times 2^{2n-1} \times (256 / M) \times (N + 1)} - 1 \} \times 100$
	1	0	0	$N = \frac{PCLKB \times 10^6}{32 \times 2^{2n-1} \times (256 / M) \times B} - 1$	$Error (\%) = \{ \frac{PCLKB \times 10^6}{B \times 32 \times 2^{2n-1} \times (256 / M) \times (N + 1)} - 1 \} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLKB \times 10^6}{16 \times 2^{2n-1} \times (256 / M) \times B} - 1$	$Error (\%) = \{ \frac{PCLKB \times 10^6}{B \times 16 \times 2^{2n-1} \times (256 / M) \times (N + 1)} - 1 \} \times 100$
	Don't care	Don't care	1	$N = \frac{PCLKB \times 10^6}{12 \times 2^{2n-1} \times (256 / M) \times B} - 1$	$Error (\%) = \{ \frac{PCLKB \times 10^6}{B \times 12 \times 2^{2n-1} \times (256 / M) \times (N + 1)} - 1 \} \times 100$
Clock synchronous, simple SPI*1				$N = \frac{PCLKB \times 10^6}{8 \times 2^{2n-1} \times (256 / M) \times B} - 1$	
Smart card interface				$N = \frac{PCLKB \times 10^6}{S \times 2^{2n+1} \times (256 / M) \times B} - 1$	$Error (\%) = \{ \frac{PCLKB \times 10^6}{B \times S \times 2^{2n+1} \times (256 / M) \times (N + 1)} - 1 \} \times 100$
Simple IIC*2				$N = \frac{PCLKB \times 10^6}{64 \times 2^{2n-1} \times (256 / M) \times B} - 1$	

B: Bit rate (bps).
M: MDDR setting (128 ≤ MDDR ≤ 255).
N: BRR setting for baud rate generator (0 ≤ N ≤ 255).
PCLKB: Operating frequency (MHz).
n and S: Determined by the settings of the SMR/SMR_SMCI and SCMR registers as listed in Table 27.8 and Table 27.9.
Bit Rate Register (BRR).

Note 1. 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).

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 I²C standard.

Table 27.20 and Table 27.21 list examples of N settings in BRR and M settings in MDDR in asynchronous mode.

Table 27.20 Examples of BRR and MDDR settings for different bit rates in asynchronous mode (1)

Bit rate (bps)	Operating frequency PCLKB (MHz)														
	8					9.8304					10				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	5	236	0	0.03	0	7	(256)*1	0	0.00	0	10	173	1	-0.01
57600	0	3	236	0	0.03	0	4	240	0	0.00	0	4	236	0	0.03
115200	0	1	236	0	0.03	0	1	192	0	0.00	0	4	236	1	0.03
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 PCLKB (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 PCLKB (MHz)														
	16					17.2032					18				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	11	236	0	0.03	0	13	(256)*1	0	0.00	0	18	166	1	-0.01
57600	0	7	236	0	0.03	0	6	192	0	0.00	0	18	249	1	-0.01
115200	0	3	236	0	0.03	0	6	192	1	0.00	0	8	236	1	0.03
230400	0	1	236	0	0.03	0	3	219	1	-0.20	0	1	210	0	0.14
460800	0	1	236	1	0.03	0	1	219	1	-0.20	0	0	210	0	0.14

Note 1. In this example, the ABCS and ABCSE in SEMR are 0. SEMR.BRME = 0 (M = 256) disables the bit rate modulation function.

Table 27.21 Examples of BRR and MDDR settings for different bit rates in asynchronous mode (2)

	Operating frequency PCLKB (MHz)									
	19.6608					20				
Bit Rate (bps)	n	N	M	BGDMbit	Error (%)	n	N	M	BGDMbit	Error (%)
38400	0	15	(256)*1	0	0.00	0	10	173	0	-0.01
57600	0	9	240	0	0.00	0	9	236	0	0.03
115200	0	4	240	0	0.00	0	4	236	0	0.03
230400	0	1	192	0	0.00	0	4	236	1	0.03
460800	0	0	192	0	0.00	0	0	189	0	0.14

Note 1. In this example, the ABCS and ABCSE bits in SEMR are 0. SEMR.BRME = 0 (M = 256) disables the bit rate modulation function.

27.2.19 Serial Extended Mode Register (SEMR)

Address(es): SCI0.SEMR 4007 0007h, SCI1.SEMR 4007 0027h, SCI9.SEMR 4007 0127h

	b7	b6	b5	b4	b3	b2	b1	b0
	RXDESEL	BGDM	NFEN	ABCS	ABCSE	BRME	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	BRME	Bit Rate Modulation Enable	0: Bit rate modulation function is disabled 1: Bit rate modulation function is enabled.	R/W*1
b3	ABCSE	Asynchronous Mode Extended Base Clock Select	Valid only in asynchronous mode with SCR.CKE[1] = 0: 0: Clock cycle for 1-bit period is determined with combination of BGDM and ABCS bits 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 and 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: The 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 a 1-bit period in asynchronous mode.

BRME bit (Bit Rate Modulation Enable)

The BRME bit enables or disables the bit rate modulation function. The bit rate generated by the on-chip baud rate

generator is evenly corrected when this function is enabled.

ABCSE bit (Asynchronous Mode Extended Base Clock Select)

The ABCSE bit sets the pulse number for the base clock in a 1-bit period to 6, and the double-frequency clock is output from the baud rate generator. When the bit rate is set to 6 while dividing the bus clock frequency, use the ABCSE bit and set SMR.CKS[1:0] to 00b and BRR to 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 function is disabled, input signals are transferred as received, and as internal signals.

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.

This bit is valid when the on-chip baud rate generator is selected as the clock source (SCR.CKE[1] = 0) in asynchronous mode (SMR.CM = 0). 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.

27.2.20 Noise Filter Setting Register (SNFR)

Address(es): SCI0.SNFR 4007 0008h, SCI1.SNFR 4007 0028h, 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.

27.2.21 I2C Mode Register 1 (SIMR1)

Address(es): SCI0.SIMR1 4007 0009h, SCI1.SIMR1 4007 0029h, 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 0 1: Simple IIC mode 1 0: Smart card interface mode 1 1: Setting prohibited.	R/W*1
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	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 the 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 PCLKB 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.

27.2.22 I2C Mode Register 2 (SIMR2)

Address(es): SCI0.SIMR2 4007 000Ah, SCI1.SIMR2 4007 002Ah, 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	IIC 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 ACK/NACK reception.	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 (IIC Interrupt Mode Select)

The IICINTM bit selects the sources of interrupt requests in simple IIC mode.

IICCSC bit (Clock Synchronization)

Set the IICCSC bit to 1 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 SCL clock signal is not synchronized if the IICCSC bit is 0. The SCLn clock signal is generated according to the rate selected in the BRR register regardless of the level being input on the SCLn pin.

Set the IICCSC bit to 1 except during debugging.

IICACKT bit (ACK Transmission Data)

The IICACKT bit transmits data that contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

27.2.23 I²C Mode Register 3 (SIMR3)

Address(es): SCI0.SIMR3 4007 000Bh, SCI1.SIMR3 4007 002Bh, 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 a start, restart, or stop condition 1 0: Output low level on the 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 the 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 in addition to setting 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 in addition to setting 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 while an interrupt request is enabled by setting the SCR.TEIE bit, an STI interrupt is output.

[Setting condition]

- When generation of a start, restart, or stop condition completes. If the setting condition 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.

27.2.24 I²C Status Register (SISR)

Address(es): SCI0.SISR 4007 000Ch, SCI1.SISR 4007 002Ch, SCI9.SISR 4007 012Ch

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

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	IICACKR	ACK Reception Data Flag	0: ACK received 1: NACK received.	R
b1	—	Reserved	This bit is read as 0	R
b2	—	Reserved	The read value is undefined	R
b3	—	Reserved	This bit is read as 0	R
b5, b4	—	Reserved	The read values are undefined	R
b7, b6	—	Reserved	These bits are read as 0	R

SISR monitors 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 received ACK/NACK bit.

27.2.25 SPI Mode Register (SPMR)

Address(es): SCI0.SPMR 4007 000Dh, SCI1.SPMR 4007 002Dh, 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 the SCR are 0 (both serial transmission and reception are disabled).
Note 2. Only 0 can be written to this bit to clear the flag.

SPMR selects the extension settings in asynchronous and clock synchronous modes.

SSE bit (SSn Pin Function Enable)

Set the SSE bit to 1 to use the SSn pin to control transmission and reception in simple SPI mode. Set this bit to 0 in all other modes. 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 set both the SSE and CTSE bits to 1 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 for controlling 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 enable both the CTSE and SSE bits 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 or slave operation in simple SPI mode. The functions of the TXDn and RXDn pins are reversed when the MSS bit is set to 1, so that data is received through the TXDn pin and transmitted through the RXDn pin. Set this bit to 0 in modes other than simple SPI mode.

MFF flag (Mode Fault Flag)

The MFF flag indicates mode fault errors. In a multi-master configuration, determine the mode fault error occurrence by reading this flag.

[Setting condition]

- When input on the SSn pin is low during master operation in simple SPI mode (SSE = 1 and MSS = 0).

[Clearing condition]

- Writing 0 to the flag after 1 is read.

CKPOL bit (Clock Polarity Select)

The CKPOL bit selects the polarity of the clock signal output through the SCKn pin. See Figure 27.70 for details.

Set this 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 27.70 for details.

Set this bit to 0 in modes other than simple SPI and clock synchronous modes.

27.2.26 FIFO Control Register (FCR)

Address(es): SCI0.FCR 4007 0014h

	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	Selects the interrupt request when detecting a receive data ready: 0: Receive data full interrupt (SCIn_RXI) 1: Receive error interrupt (SCIn_ERI).	R/W
b7 to b4	TTRG[3:0]	Transmit FIFO Data Trigger Number	Valid only in asynchronous mode, including multi-processor or clock synchronous mode: 0 0 0 0: Trigger number 0 : 1 1 1 1: Trigger number 15.	R/W
b11 to b8	RTRG[3:0]	Receive FIFO Data Trigger Number	Valid only in asynchronous mode, including multi-processor or clock synchronous mode 0 0 0 0: Trigger number 0 : 1 1 1 1: 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: 0 0 0 0: Trigger number 0 : 1 1 1 1: Trigger number 15.	R/W

Note 1. Writable only when TE = 0 and RE = 0.

FCR selects the FIFO mode, resets FTDRHL/FRDRHL, selects the FIFO data trigger number of transmission/reception, and selects the RTS output active trigger number.

FM bit (FIFO Mode Select)

When the FM bit is set to 1, FTDRHL and FRDRHL are selected for communication. When the FM bit is set to 0, TDR and RDR, or TDRHL and RDRHL are selected for communication.

RFRST bit (Receive FIFO Data Register Reset)

When the RFRST bit is set to 1, the FRDRHL register is reset, and the receive data count is reset to 0. When 1 is written to the RFRST bit, it is set to 0 after 1 PCLKB.

TFRST bit (Transmit FIFO Data Register Reset)

When the TFRST bit is set to 1, the FTDRHL register is reset, and the transmit data count is reset to 0. When 1 is written to the TFRST bit, it is set to 0 after 1 PCLKB.

DRES bit (Receive Data Ready Error Select)

On detecting a receive data ready error, the DRES bit selects between an SCIn_RXI interrupt request or an SCIn_ERI interrupt request. Set the DRES bit to 1 when starting the 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 FTDRHL is equal to or less than the transmit triggering number specified in TTRG[3:0] bits, and software can write data to FTDRHL. If SCR.TIE = 1, an SCIn_TXI interrupt request occurs.

RTRG[3:0] bits (Receive FIFO Data Trigger Number)

The RDF flag is set to 1 when the amount of receive data in the FRDRHL is equal to or greater than the specified receive triggering number specified in RTRG[3:0] bits, and software can read data from FRDRHL. If SCR.RIE = 1, an SCIn_RXI interrupt request occurs.

When RTRG[3:0] is set to 0, the RDF flag is not set even when the amount of data in the receive FIFO is equal to 0, and an SCIn_RXI interrupt does not occur.

RSTRG[3:0] bits (RTS Output Active Trigger Number Select)

When the amount of receive data stored in the FRDRHL is equal to or greater than the receive triggering number specified in the RSTRG[3:0] bits, the RTSn signal goes high.

When RSTRG[3:0] is set to 0, the RTSn signal is not high even when the amount of data in the receive FIFO is equal to 0.

27.2.27 FIFO Data Count Register (FDR)

Address(es): SCI0.FDR 4007 0016h

	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 indicates 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 indicates no transmit data, and 10h means that all (maximum count) of the data to be transmitted is stored in FTDRHL.

27.2.28 Line Status Register (LSR)

Address(es): SCI0.LSR 4007 0018h

	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 in the receive data stored in the Receive FIFO Data Register (FRDRHL)	R
b15 to b13	—	Reserved	These bits are read as 0	R

Note 1. If this flag is 1, write 0 to SSR_FIFO.ORER to clear the flag.

The LSR register indicates the status of receive error.

ORER flag (Overrun Error Flag)

The ORER flag reflects the value in SSR_FIFO.ORER.

FNUM[4:0] bits (Framing Error Count)

The FNUM[4:0] bit value indicates the amount of data with a framing error stored in the FRDRHL register.

PNUM[4:0] bits (Parity Error Count)

The PNUM[4:0] bit value indicates the amount of data with a parity error stored in the FRDRHL register.

27.2.29 Compare Match Data Register (CDR)

Address(es): SCI0.CDR 4007 001Ah, SCI1.CDR 4007 003Ah, 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	Holds 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 compare data for the address match function.

CMPD[8:0] bits (Compare Match Data)

The CMPD[8:0] bits set the data to be compared to receive data for the address match function when the address match function is enabled (DCCR.DCME = 1).

Three bit lengths are available:

- CMPD[6:0] with 7-bit length
- CMPD[7:0] with 8-bit length
- CMPD[8:0] with 9-bit length.

27.2.30 Data Compare Match Control Register (DCCR)

Address(es): SCI0.DCCR 4007 0013h, SCI1.DCCR 4007 0033h, SCI9.DCCR 4007 0133h

	b7	b6	b5	b4	b3	b2	b1	b0
	DCME	IDSEL	—	DFER	DPER	—	—	DCMF
Value after reset:	0	1	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DCMF	Data Compare Match Flag	0: Not matched 1: Matched.	R/W*1
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	DPER	Data Compare Match Parity Error Flag	0: No parity error occurred 1: A parity error occurred.	R/W*1
b4	DFER	Data Compare Match Framing Error Flag	0: No framing error occurred 1: A framing error occurred.	R/W*1
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	IDSEL	ID Frame Select	Valid only in asynchronous mode, including multi-processor mode: 0: Always compare data regardless of the MPB bit value 1: Only 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 mode: 0: Address match function disabled 1: Address match function enabled.	R/W

Note 1. Only 0 can be written to clear the flag after reading 1.

The DCCR register 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 SCR.RE bit to 0 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 (receive data match detection).

[Setting condition]

- When a parity error is detected in the frame in which an address match is 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 (receive 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 condition]

- When 0 is written after 1 is read from DFER

When the SCR.RE bit is set to 0 (serial reception is disabled), the DFER flag is not affected and keeps its previous value.

IDSEL bit (ID Frame Select)

The IDSEL bit selects whether to compare data regardless of the MPB bit value or to compare data only when MPB = 1 (ID frame), when the address match function is enabled.

DCME bit (Data Compare Match Enable)

The DCME bit enables or disables the address match function (data compare match function).

If the SCI detects a match between the comparison data (CDR.CMPD) and receive data, the DCME bit is cleared automatically and the SCI operates in receive mode without the data compare match function. See section 27.3.6, Address Match (Receive Data Match Detection) Function.

The write value should be 0 for any mode other than asynchronous mode.

27.2.31 Serial Port Register (SPTR)

Address(es): SCI0.SPTR 4007 001Ch, SCI1.SPTR 4007 003Ch, 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	Indicates the state of the RXDn pin: 0: RXDn pin is low 1: RXDn pin is high.	R
b1	SPB2DT	Serial Port Break Data Select	Selects the output level of the 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 on the TXDn pin: 0: The value of SPB2DT bit is not output on TXDn pin 1: The value of SPB2DT bit is output on 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 the serial reception pin (RXDn) status and sets transmission pin (TXDn) 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 27.22.

Table 27.22 TXDn pin status

Value of SCR.TE	Value of SPTR.SPB2IO	Value of SPTR.SPB2DT	TXDn pin status
0	0	x	Hi-Z (initial value)
0	1	0	Low level output
0	1	1	High level output
1	x	x	Serial transmit data is output

x: Don't care.

Note: Use the SPTR register in asynchronous mode only. Using this register in any other mode is not guaranteed.

27.3 Operation in Asynchronous Mode

Figure 27.2 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 receiver have a double-buffered structure in addition to FIFO mode, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

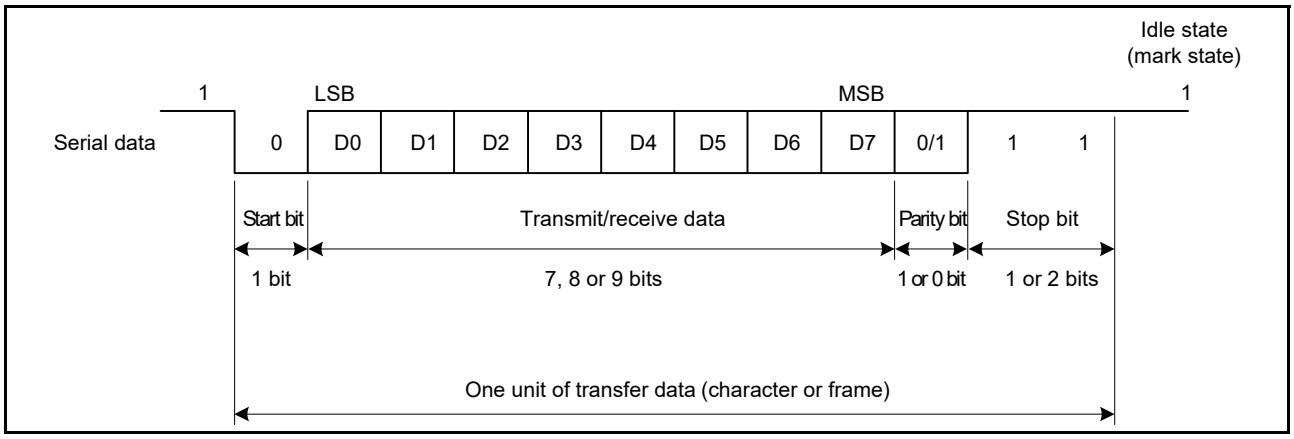


Figure 27.2 Data format in asynchronous serial communications with 8-bit data, parity, and 2 stop bits

27.3.1 Serial Data Transfer Format

Table 27.23 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 27.4, Multi-Processor Communications Function.

Table 27.23 Serial transfer formats in asynchronous mode (1 of 2)

SCMR setting	SMR setting				Serial transfer format and frame length												
CHR1	CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	0	0	0	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		
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		

Table 27.23 Serial transfer formats in asynchronous mode (2 of 2)

SCMR setting	SMR setting				Serial transfer format and frame length												
CHR1	CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	—	1	0	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

27.3.2 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 27.3. The reception margin in asynchronous mode is determined by the following formula (1):

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\ F - \frac{\left| D - 0.5 \right|}{N} (1 + F) \right| \times 100\ [\%] \cdots \text{Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock
N = 16 when SEMR.ABCSE = 0 and SEMR.ABCS = 0
N = 8 when SEMR.ABCS = 1, N = 6 when SEMR.ABCSE = 1
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 13)
F: Absolute value of clock frequency deviation

Assuming the values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the following formula:

$$M = \{0.5 - 1 / (2 \times 16)\} \times 100\ (\%) = 46.875\%$$

This represents the computed value. Renesas recommends that a margin of 20% to 30% should be allowed in system design.

Note 1. In this example, the SEMR.ABCS bit is 0 and SEMR.ABCSE bit 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 receive data is sampled on the rising edge of the 3rd pulse of the base clock.

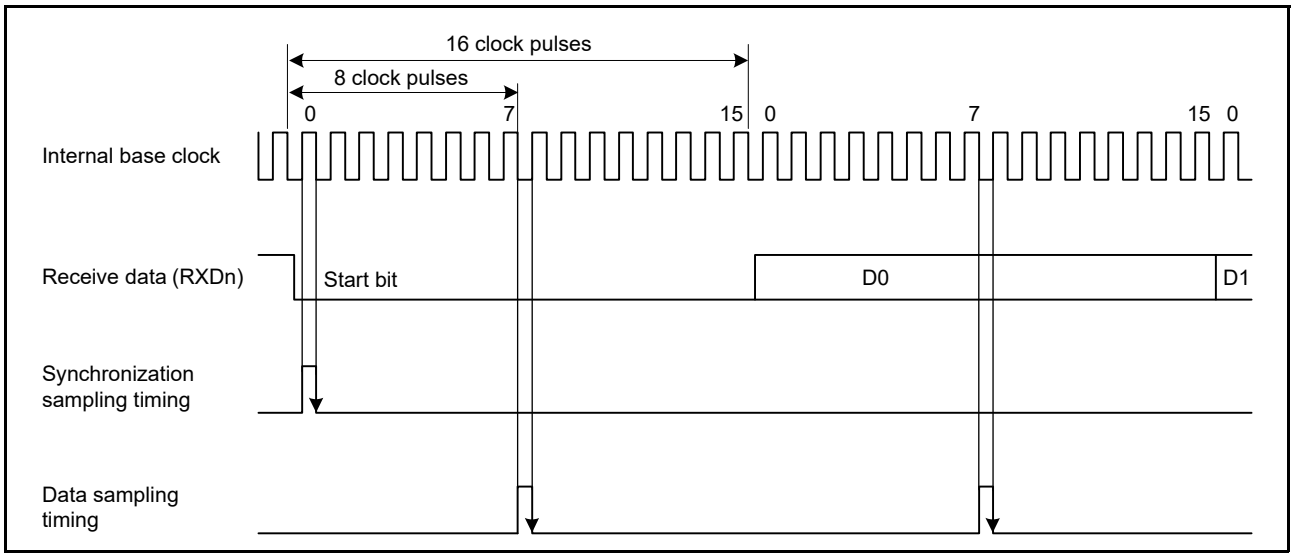


Figure 27.3 Receive data sampling timing in asynchronous mode

27.3.3 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 SMR.CM bit and the SMR.CKE[1:0] bit settings.

When an external clock is input to the SCKn pin, the clock frequency must be 16 times the bit rate (when SEMR.ABCS = 0) or 8 times the bit rate (when SEMR.ABCS = 1).

When the SCI uses its internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as Figure 27.4 shows.

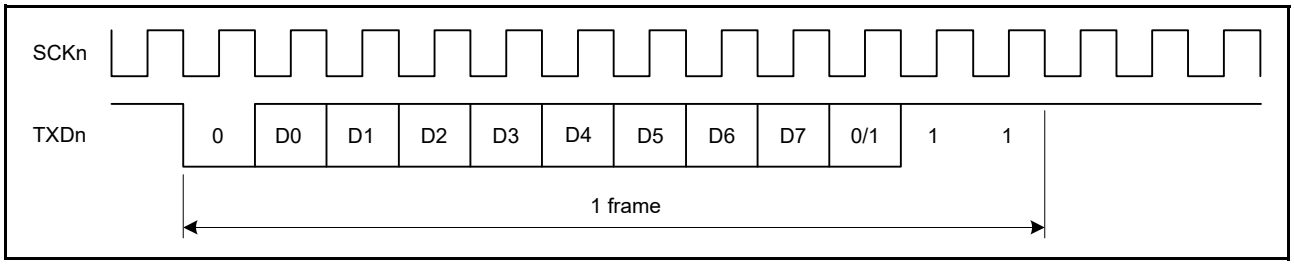


Figure 27.4 Phase relationship between output clock and transmit data in asynchronous mode when SCMR.CHR1 = 1, SMR.CHR = 0, PE = 1, MP = 0, STOP = 1

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

When the SEMR.ABCS bit 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 that is equal to twice the value when ABCS is set to 0. When the SEMR.BGDM bit is set to 1, the cycle of the base clock is half and the bit rate is double the value when BGDM is set to 0. When the SCR.CKE[1] bit is set to 0 and the on-chip baud rate generator is selected, setting the ABCS and BGDM bits to 1 allows the SCI to operate at a bit rate equal to four times the value when the ABCS and BGDM bits are set to 0. When the SEMR.ABCSE bit is set to 1, the number of basic clock pulses is 6 during a period of 1 bit, and the SCI operates at a bit rate that is equal to 16/3 times the value when SEMR.ABCS, SEMR.BGDM, and SMER.ABCSE are 0.

As shown by formula (1) in section 27.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, the reception margin decreases when the ABCS or ABCSE bit in SEMR is set to 1. Therefore, if the target bit rate can be obtained with ABCS or ABCSE set to 0, it is recommended that you use the SCI with ABCS and ABCSE set to 0.

27.3.5 CTS and RTS Functions

The CTS function uses input on the CTSn_RTSn pin in transmission control. Setting the SPMR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, driving the CTSn_RTSn pin low causes transmission to start.

Driving the CTSn_RTSn pin high while transmission is in progress does not affect transmission of the current frame.

In the RTS function that uses output on the CTSn_RTSn pin, a low level is output when reception becomes possible. Conditions for 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 SCR.RE bit is 1
- Reception is not in progress
- There is no receive data yet to be read
- The ORER, FER, and PER flags in the SSR register are all 0.

(b) FIFO selected when all of the following conditions are satisfied

- The value of the SCR.RE bit is 1
- The amount of receive data written in FRDRHL is equal to or less than the specified receive triggering number
- The ORER bit in the SSR_FIFO register (ORER in FRDRH) is 0.

[Conditions 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 values after SCR.RE = 0.

(b) FIFO selected

- The conditions for low-level output are not satisfied.

27.3.6 Address Match (Receive Data Match Detection) Function

The address match function can be used only in asynchronous mode.

If the DCCR.DCME 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 between the comparison data (CDR.CMPD*3) and 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 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 bit is set to 0, the SCI performs address match detection regardless the MPB bit value of the received data. Until the SCI detects a match between the comparison data (CDR.CMPD*3) and receive data, received data is skipped (discarded), and the SCI cannot detect parity error or framing error. When the SCI detects a match, the DCCR.DCME is automatically cleared, and DCCR.DCMF flag is set to 1.

If DCCR.IDSEL bit is set to 1, the SCR.MPIE bit is automatically cleared. If DCCR.IDSEL bit 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, the DCCR.DPER bit is set to 1. The compared receive data is not stored in the RDR register*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.

Examples of the address match function are shown in Figure 27.5 and Figure 27.6.

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.

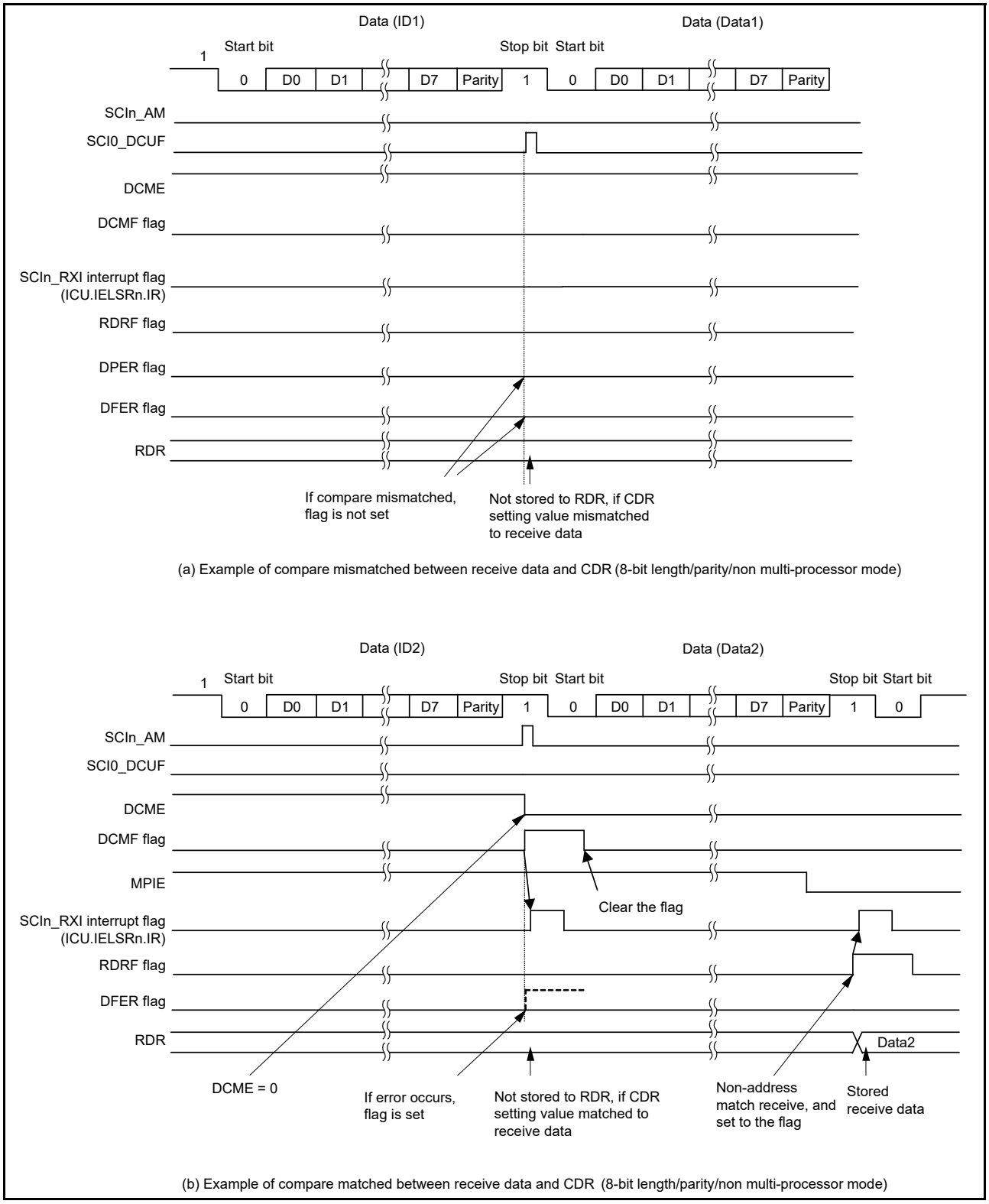


Figure 27.5 Example of address match (1) non multi-processor mode

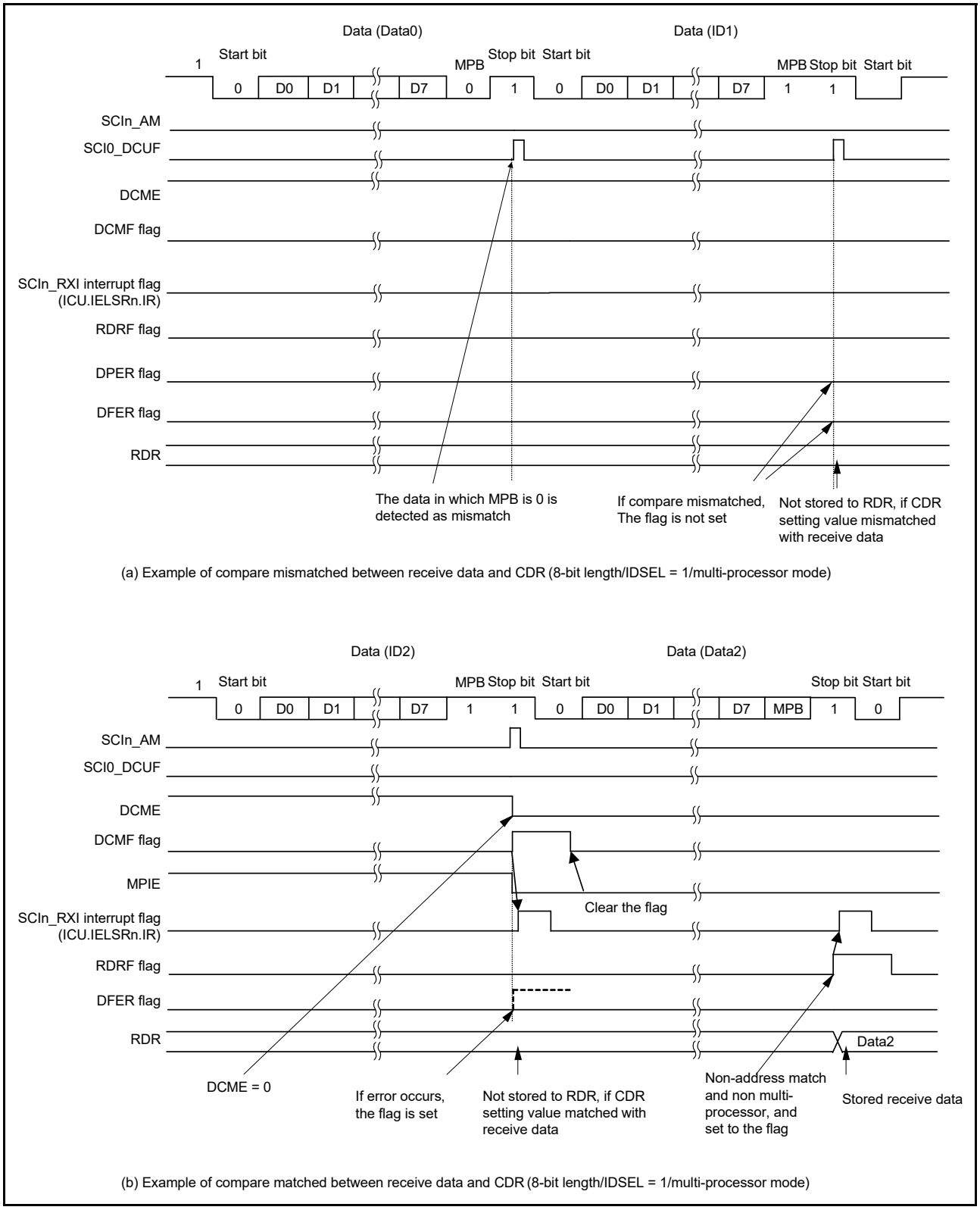


Figure 27.6 Example of address match (2) multi-processor mode

27.3.7 SCI Initialization in Asynchronous Mode

Before transmitting and receiving data, start by writing the initial value 00h to the SCR register, then continue through to the SCI procedure (select non-FIFO or FIFO) shown in Figure 27.7 and Figure 27.8. Whenever the operating mode or transfer format is to be changed, the SCR register must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied during initialization.

Note: 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: In non-FIFO mode, switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of an SCIn_TXI interrupt request.

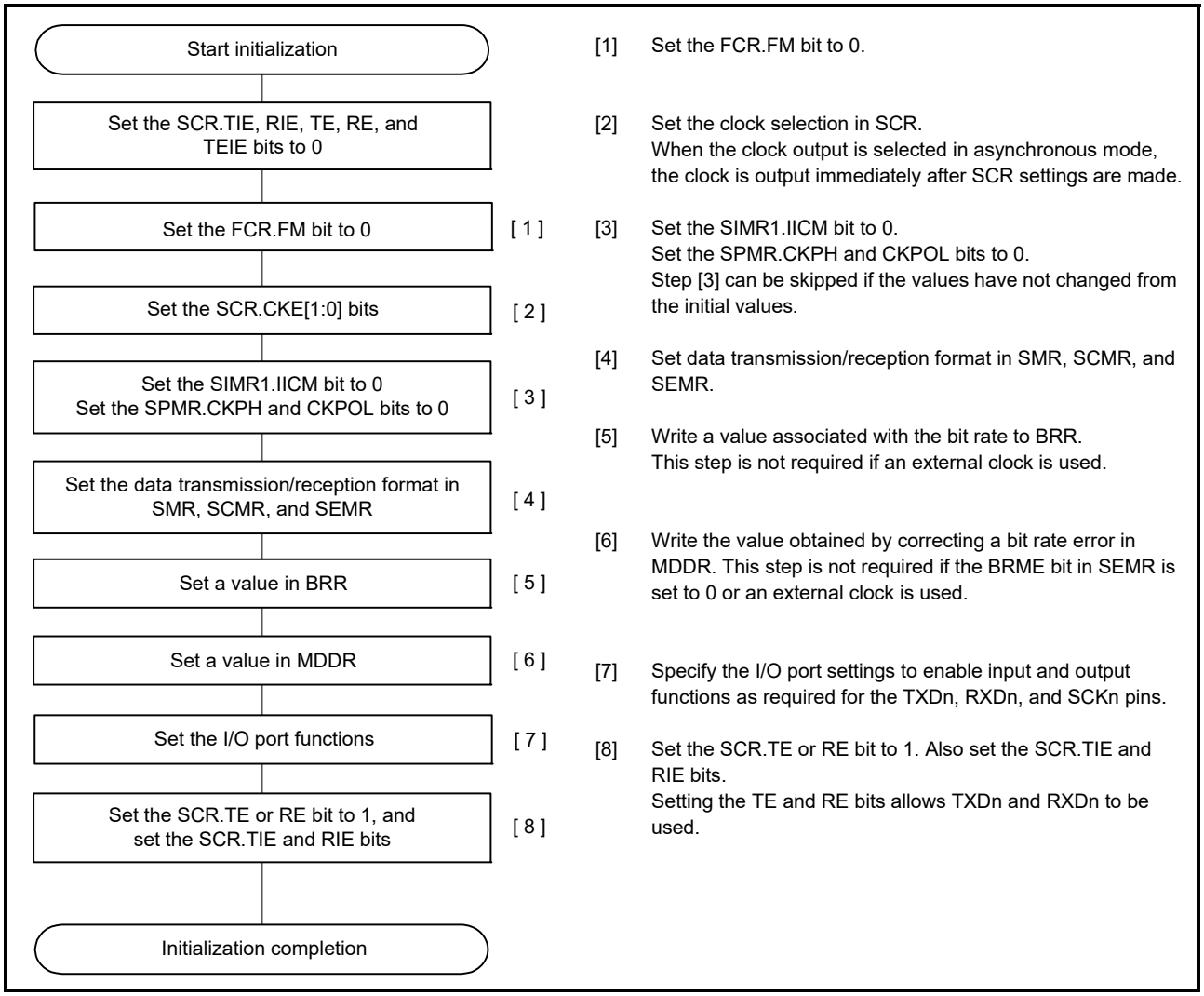
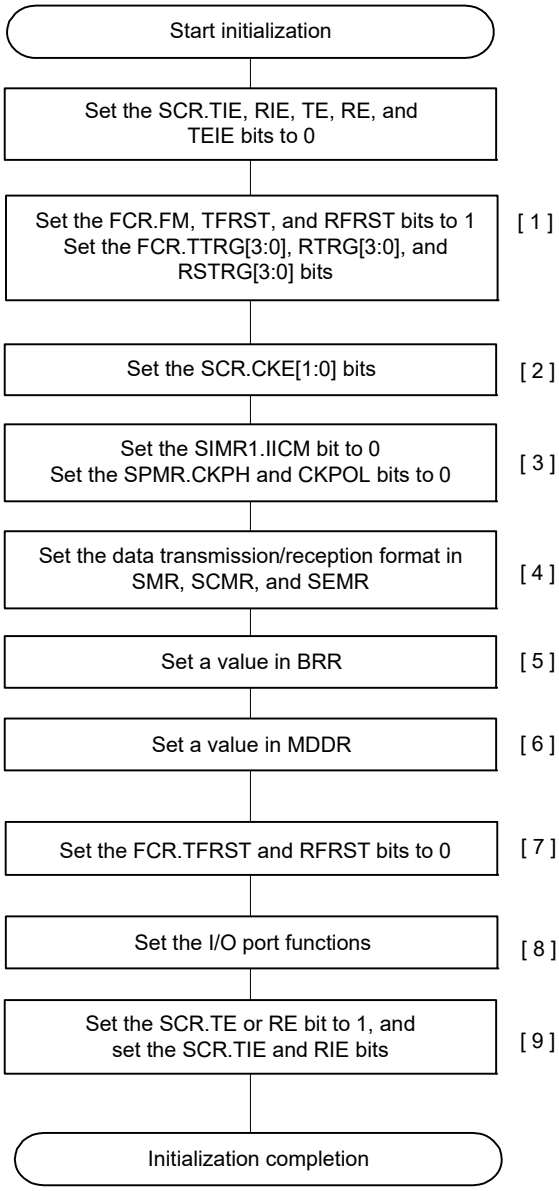


Figure 27.7 Example SCI initialization flow in asynchronous mode with non-FIFO selected



[1] Set the FCR.FM, TFRST, and RFRST bits 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 the SIMR1.IICM bit to 0.
Set the SPMR.CKPH and CKPOL bits 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 the FCR.TFRST and RFRST bits to 0.

[8] Specify the I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[9] Set the SCR.TE or RE bit to 1. Also set the SCR.TIE and RIE bits.
Setting the TE and RE bits allow TXDn and RXDn to be used.

Figure 27.8 Example SCI initialization flow in asynchronous mode with FIFO selected

27.3.8 Serial Data Transmission in Asynchronous Mode

(1) Non-FIFO selected

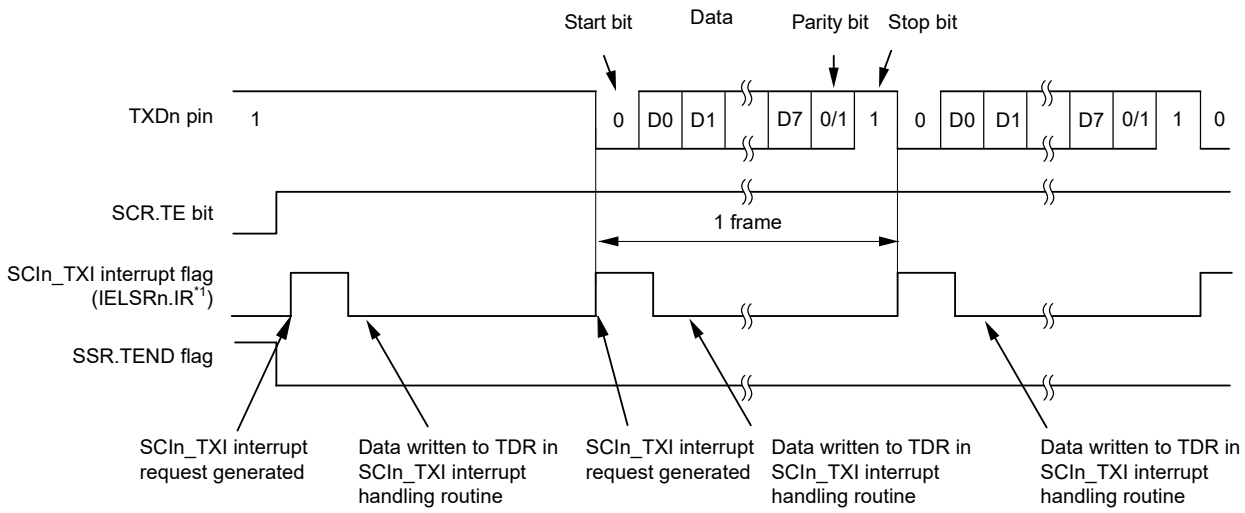
Figure 27.9, Figure 27.10, and Figure 27.11 show examples of serial transmission in asynchronous mode.

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 TXDn.

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 the SCR are set to 1 simultaneously by a single instruction.
2. Transmission starts after the SPMR.CTSE bit is set to 0 (CTS function is disabled) or a low level on the CTSn_RTSn pin causes data transfer from TDR*1 to TSR. If the SCR.TIE bit 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 the SCR.TIE bit to 0 (an SCIn_TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (an SCIn_TEI interrupt request is enabled) after the last of the data to be transmitted is written to 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 an update of the TDR register on output of the stop bit.
5. When TDR is updated, setting the SPMR.CTSE bit to 0 (CTS function is disabled) or a low-level input on the CTSn_RTSn pin, causes the 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 SSR.TEND flag is set to 1, the stop bit is sent, and the mark state is entered, where 1 is output. If the SCR.TEIE bit is 1, the SSR.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated.

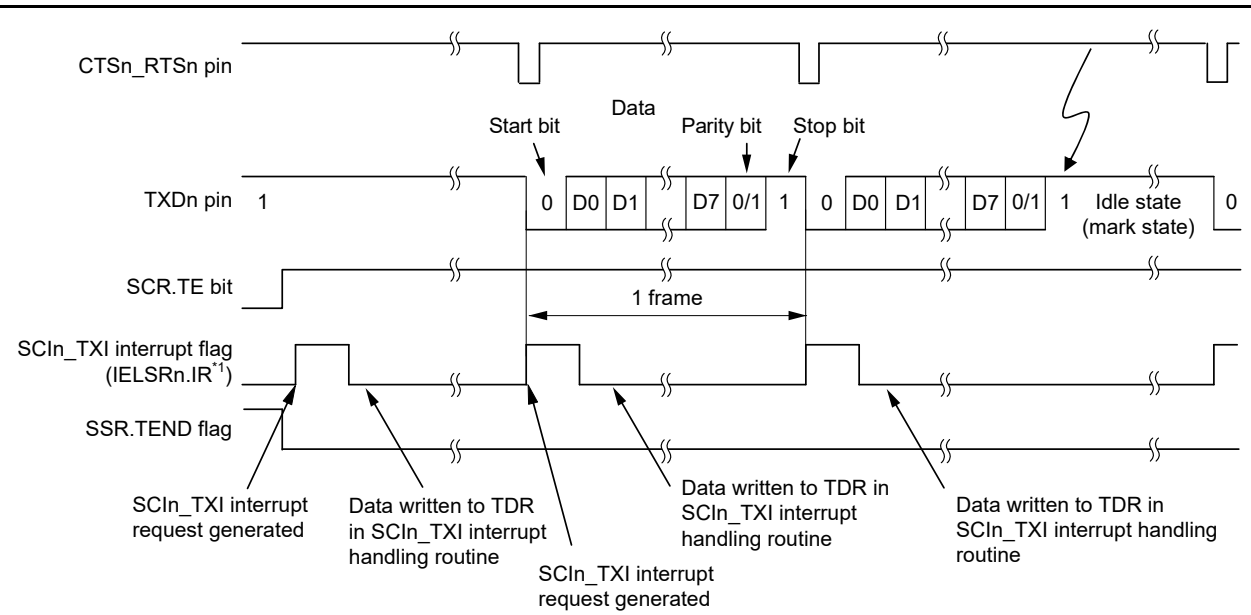
Note 1. Only write data to the TDRHL register when 9-bit data length is selected.

Figure 27.9, Figure 27.10, and Figure 27.11 show an example flow of serial transmission in asynchronous mode.



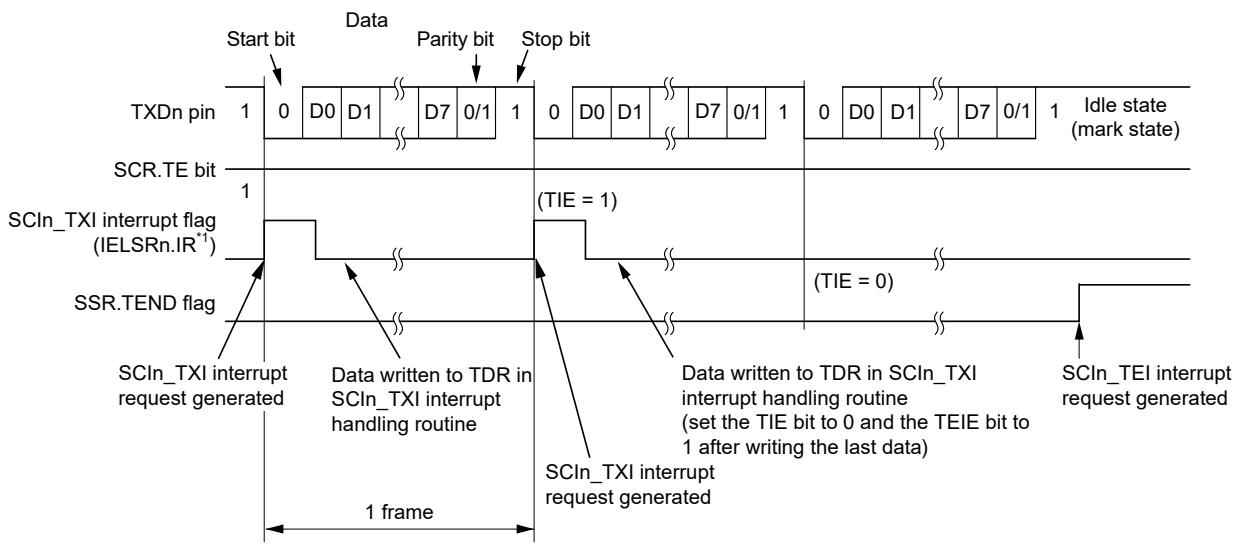
Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 27.9 Example of operation for serial transmission in asynchronous mode (1) with 8-bit data, parity bit, 1 stop bit, CTS function not used, and at the beginning of transmission



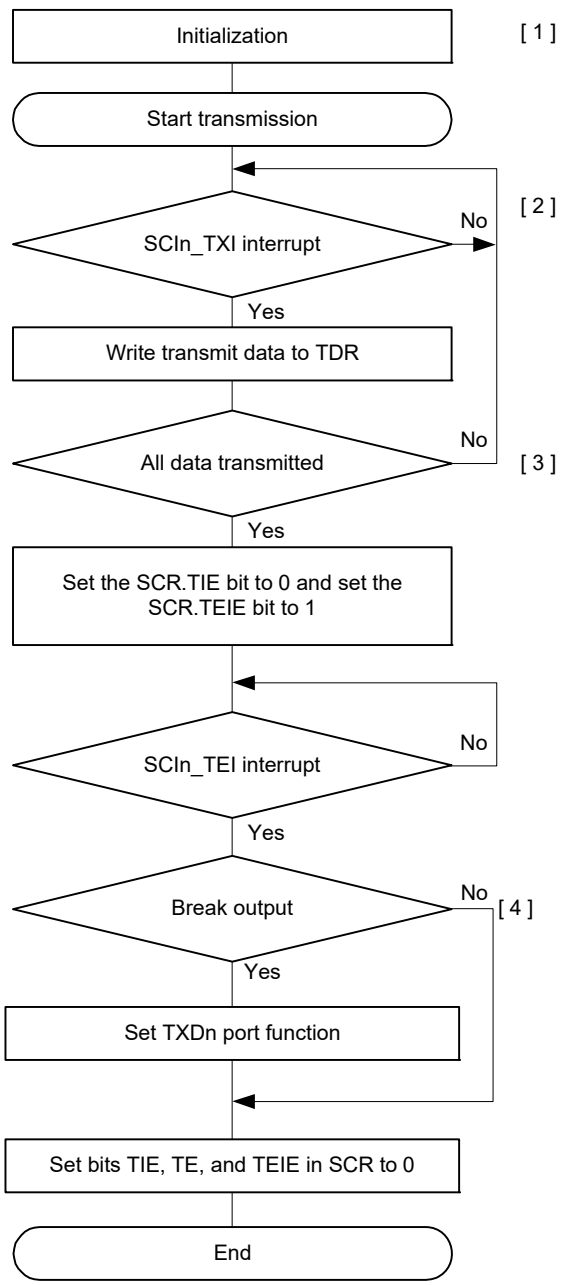
Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 27.10 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



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 27.11 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



[1] SCI Initialization:
Set data transmission.
After the SCR.TE bit 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 DTC.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, after setting the output state (low-level output) of TXDn pin using the SPTR.SPB2IO and SPTR.SPB2DT bits, set the SCR.TE bit to 0.

Note: The TDR register becomes the TDRHL register when 9-bit data length is selected.

Figure 27.12 Example of serial transmission flow in asynchronous mode with non-FIFO selected

(2) FIFO selected

Figure 27.13 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	1	—	—	—	—	—	—	—	—	—	7-bit transmit data						
8 bits	1	0	—	—	—	—	—	—	—	—	8-bit transmit data							
9 bits	0	Don't care	—	—	—	—	—	—	—	9-bit transmit data								

—: Invalid. The write value should be 0.

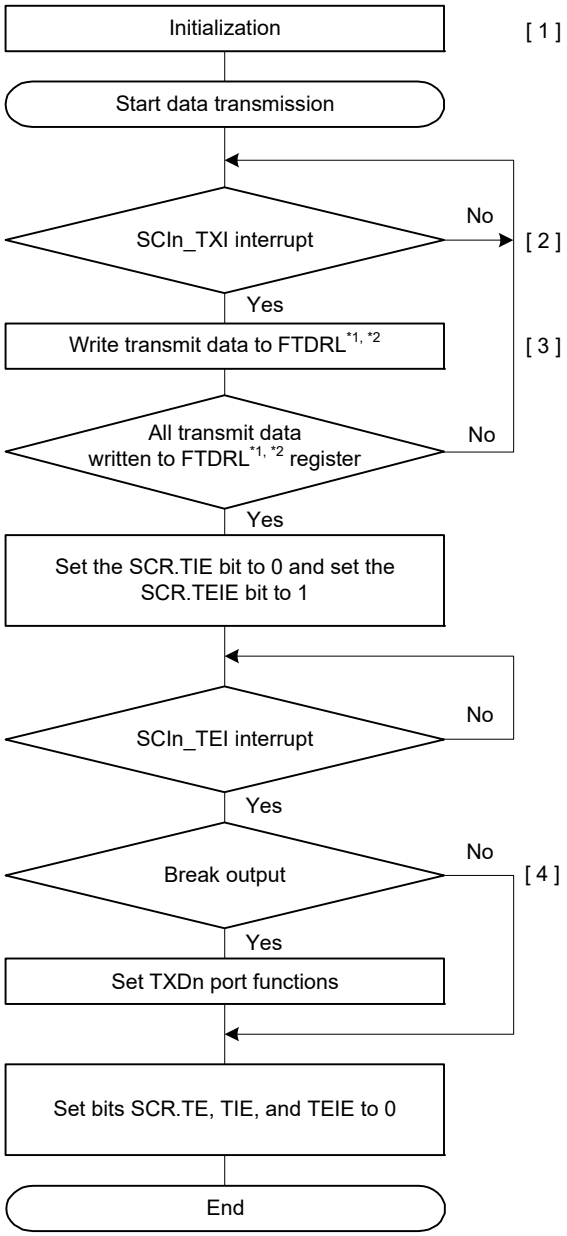
Figure 27.13 Data format written to FTDRH and FTDRL with FIFO selected

In serial transmission, the SCI operates as described in this section. When the TE bit is set to 1, the high level for one frame (preamble) is output to TXDn.

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 SPMR.CTSE bit is set to 0 (CTS function is disabled) and a low level on the CTSn_RTSn pin causes data transfer from FTDRL*1 to TSR. When the amount of transmit data written in FTDRL is equal to or less than the specified transmit triggering number, SSR_FIFO.TDFE is set to 1. If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to FTDRL*1 in the SCIn_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 (an SCIn_TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (an SCIn_TEI interrupt request is enabled) after the last of the data to be transmitted is written to FTDRL*1 *2 from the handling routine for SCIn_TXI requests.
3. Data is sent from the TXDn pin in the following order:
   a. Start bit
   b. Transmit data
   c. Parity bit or multi-processor bit (can be omitted depending on the format)
   d. Stop bit.
4. On output of the stop bit, the SCI checks whether non-transmitted data remains in FTDRL*3.
5. When data is set to FTDRL*3, setting the SPMR.CTSE bit to 0 (CTS function is disabled) or a low-level input on the CTSn_RTSn pin causes transfer of the next transmit data from FTDRL*1 to TSR and transmission of the stop bit, after which serial transmission of the next frame starts.
6. If data is not set in FTDRL*3, the TEND flag in SSR_FIFO is set to 1, the stop bit is sent, and the mark state is entered where 1 is output. If the SCR.TEIE bit 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 an update to the FTDRL register and not the FTDRH register when 9-bit data length is selected.

Figure 27.14 shows an example flow of serial transmission in asynchronous mode with FIFO selected.



[1] SCI initialization:
Set data transmission.
After the TE bit in SCR 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 clear the SSR_FIFO.TDFE flag to 0. It is possible to write16 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 DTC. When data is written to FTDRL by the DTC, the TDFE flag is cleared automatically. Therefore, do not write to the TDFE flag.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the FTDRL.

[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 the SCR.TE bit 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 27.14 Example of serial transmission flow in asynchronous mode with FIFO selected

27.3.9 Serial Data Reception in Asynchronous Mode

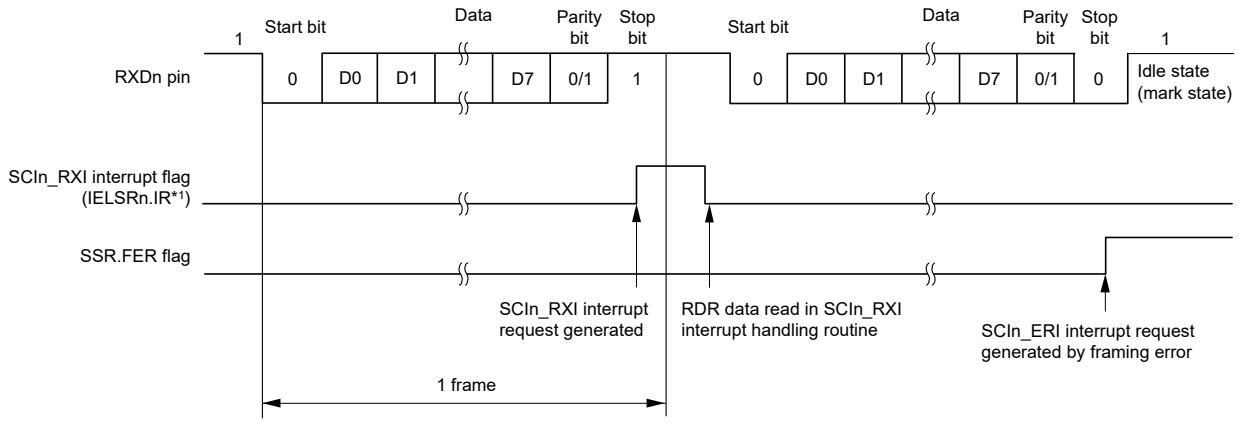
(1) Non-FIFO selected

Figure 27.15 and Figure 27.16 show an example of the operation for serial data reception in asynchronous mode.

In serial data reception, the SCI operates as follows:

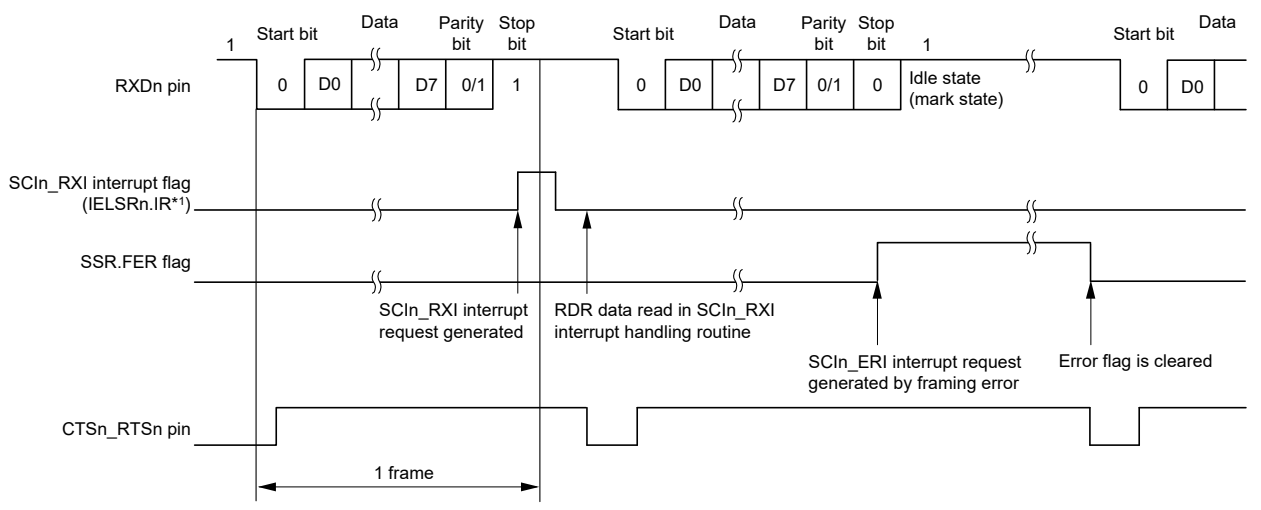
1. When the value of the SCR.RE bit becomes 1, the output signal on the CTSn_RTSn pin goes low.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to RDR*1.
4. If a parity error is detected, the SSR.PER flag 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 SSR.FER flag 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 SCR.RIE bit 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 the RDRHL register when 9-bit data length is selected.



Note 1. See section 13, Interrupt Controller Unit (ICU) for details on the associated interrupt event number.

Figure 27.15 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



Note 1. See section 13, Interrupt Controller Unit (ICU) for details on the associated interrupt event number.

Figure 27.16 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 27.24 lists the states of the flags in the SSR register and receive data handling when a receive error is detected.

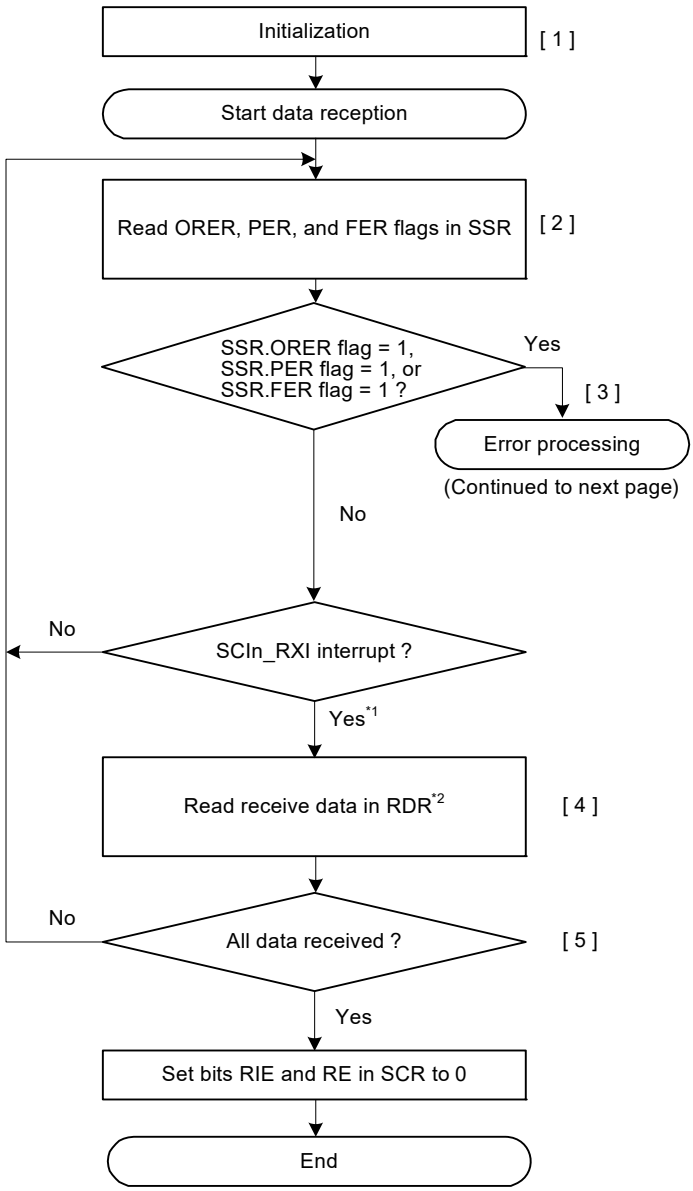
If a receive error is detected, an SCIn_ERI interrupt request is generated but an SCIn_RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Also, set the ORER, FER, and PER flags to 0 before resuming reception. In addition, be sure to read the RDR or the RDRHL register during overrun error processing. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR or RDRHL register because the received data that is not yet read might be left in RDR or RDRHL.

Figure 27.17 and Figure 27.18 show example flows for serial data reception.

Table 27.24 Flags in SSR Status Register and receive data handling

Flags in the SSR Status Register				
ORER	**FER**	**PER**	**Receive data**	**Receive error type**
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR*1	Framing error
0	0	1	Transferred to RDR*1	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR*1	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error

Note 1. Only read data in the RDRHL register when 9-bit data length is selected.



[1] SCI initialization:
Set data reception.

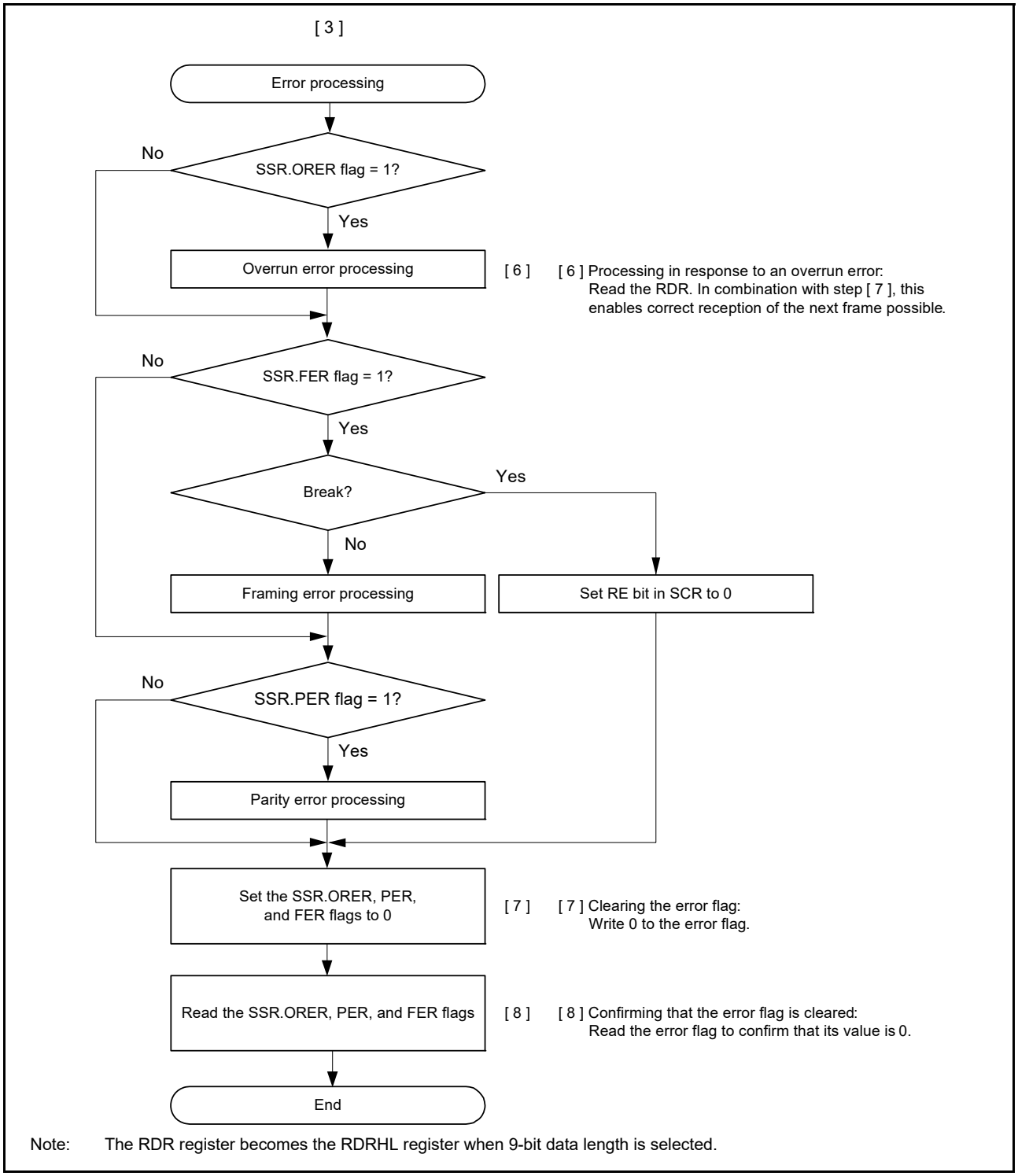
[2] [3] Receive error processing and break detection:
If a receive error occurs, an SCIn_ERI interrupt is generated. The error type is identified by reading the ORER, PER, and FER flags in SSR. After performing the appropriate error processing, be sure to set the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. For a framing error, a break can be detected by reading the value of the input port associated with the RXDn pin.

[4] Read the receive data in RDR once in the SCIn_RXI interrupt handling routine.

[5] 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 DTC.

Note 1. Do not set 0 to RE before reading RDR.
Note 2. The RDR register becomes the RDRHL register when 9-bit data length is selected.

Figure 27.17 Example flow of serial reception in asynchronous mode with non-FIFO selected (1)



Note: The RDR register becomes the RDRHL register when 9-bit data length is selected.

Figure 27.18 Example flow of serial reception in asynchronous mode with non-FIFO selected (2)

(2) FIFO selected

Figure 27.19 shows an example of a data format that is written to FRDRH and FRDRL in asynchronous mode.

In asynchronous mode, 0 is written to the MPB flag in FRDRH. Data that corresponds to the data length is written to FRDRH and FRDRL. Unused bits are written as 0. Read in the order from FRDRH to FRDRL. If software reads FRDRL, the SCI updates FER, PER 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	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	—	RDF	ORER	FER	PER	DR	0	0	0	7-bit receive data						
8 bits	1	0	—	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								

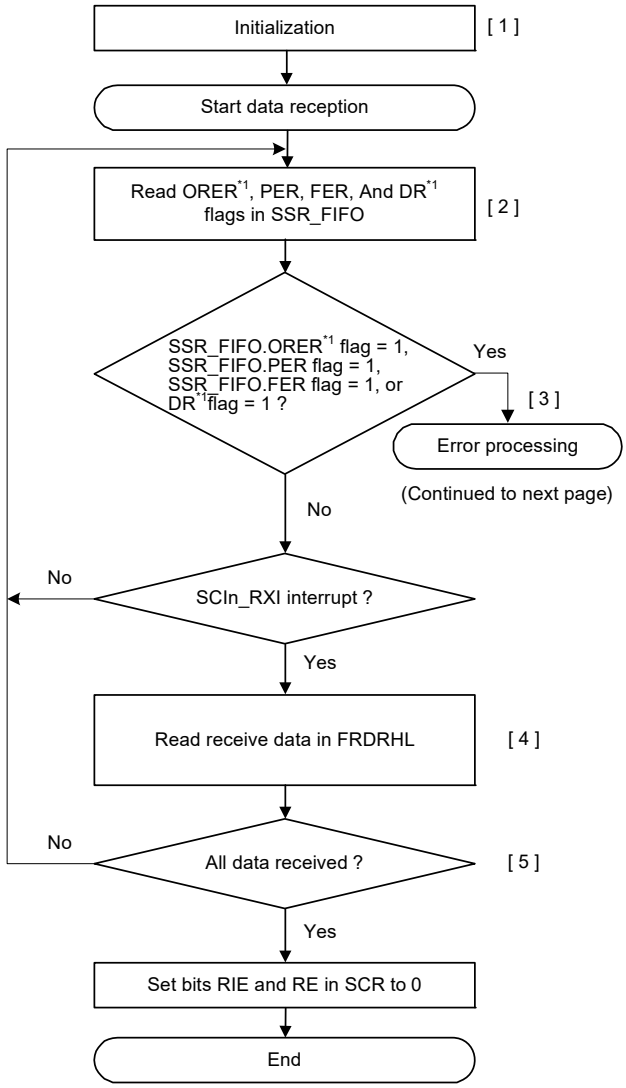
Note: 0 is always read for MPB flag (FRDRHL[9])
When data length is 7 bits, 0 is always read for FRDRHL[8] and FRDRHL[7]
When data length is 8 bits, 0 is always read for FRDRHL[8]
FRDRHL[15] bit is read as an indefinite value.

Figure 27.19 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 SCR.RE bit 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 SCR.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 the 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 a RIE bit 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 FRDRL register falls below the specified receive triggering number, and the next data is not received after 15 ETUs from the last stop bit in asynchronous mode, the SSR_FIFO.DR bit is set to 1. When the RIE bit is 1 and the FCR.DRES bit is 0, SCI generates an SCIn_RXI interrupt request. When the FCR.DRES bit is 1, SCI generates an SCIn_ERI interrupt request.
8. When reception finishes successfully, receive data is transferred to FRDRL*1. The RDF bit is set to 1 when the amount of receive data written to FRDRHL is equal to or greater than the specified receive triggering number. When the SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to FRDRL*2 in the SCIn_RXI interrupt handling routine, before an overrun error occurs. If the received data that is transferred to FRDRL*3 is less than the RTS trigger number, the CTSn_RTSn pin outputs low.

Note 1. Only read data in the FRDRH and FRDRL registers when 9-bit data length is selected.
Note 2. Read data in the order from FRDRH to FRDRL when 9-bit data length is selected.
Note 3. The SCI only checks for an update to the FRDRL register and not to the FRDRH register when 9-bit data length is selected.



Note 1. Can be read by the FRDRHL.ORER and DR flags.

[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 ORER*1 flag is set to 1. The reception operation is continuous, even when FER = 1 or PER = 1 or DR*1 = 1.

[4] Read the receive data in FRDRHL in the SCIn_RXI interrupt handling routine. The receive data stored in the FRDRHL register is read until the number of stored data is below the FCR.RTRG[3:0] value. Confirm the number of receive data bits in the FIFO by reading the FDR.R[4:0] 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 DTC. The RDF flag is cleared automatically in this case. Therefore, do not write to the RDF flag.

Figure 27.20 Example flow of serial reception in asynchronous mode with FIFO selected (1)

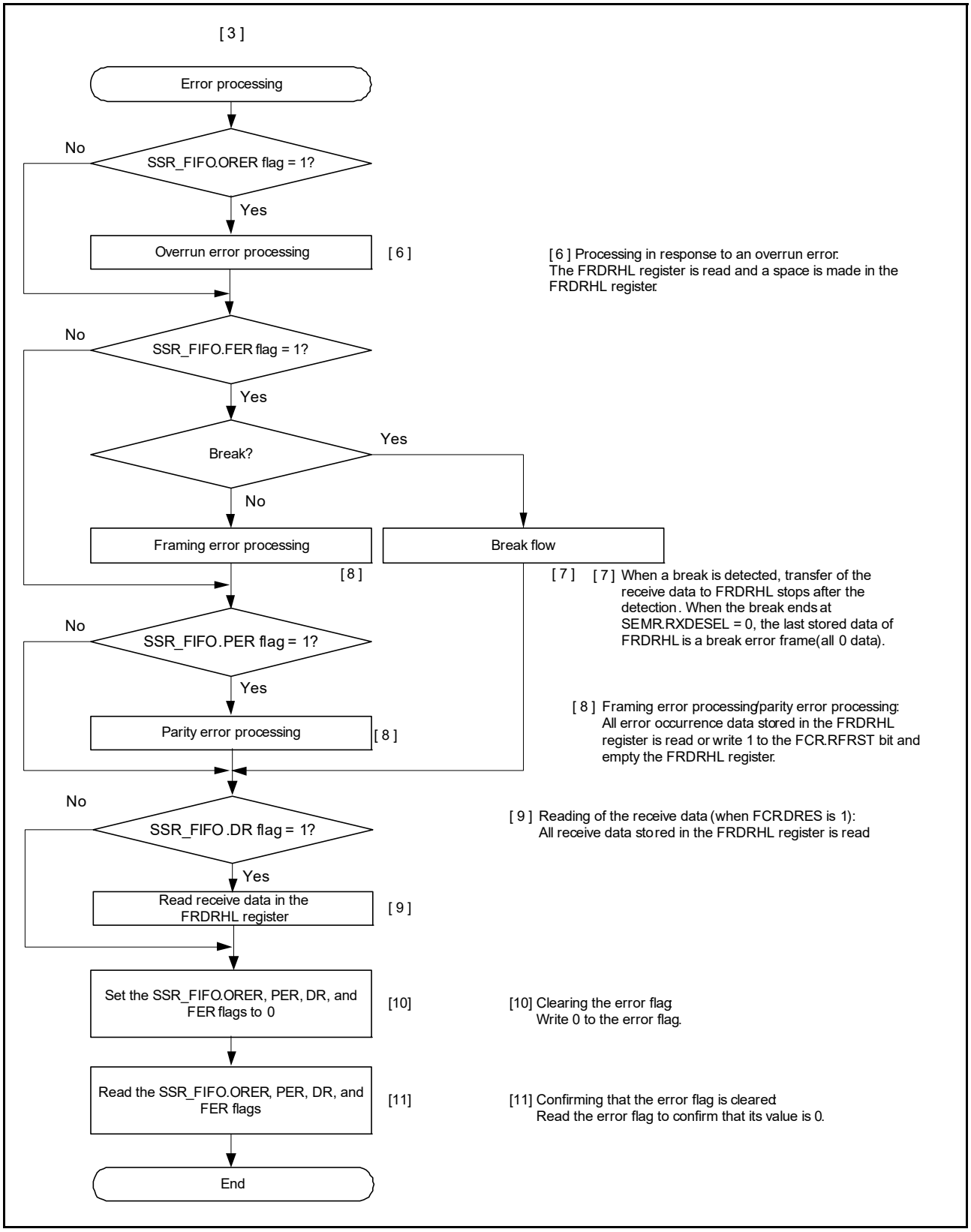


Figure 27.21 Example flow of serial reception in asynchronous mode with FIFO selected (2)

27.4 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 multiple-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 27.22 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 SCR.MPIE bit. When the MPIE bit is set to 1, the following operations are disabled until the reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from the RSR to the RDR (the RDRHL register when 9-bit data length is selected)
- Detection of a receive error
- Setting of the respective RDRF, ORER and FER status flags in the SSR register.

When the SCI receives a character in which the multi-processor bit is set to 1, the SSR.MPBT bit is set to 1 and the SSR.MPIE bit 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.

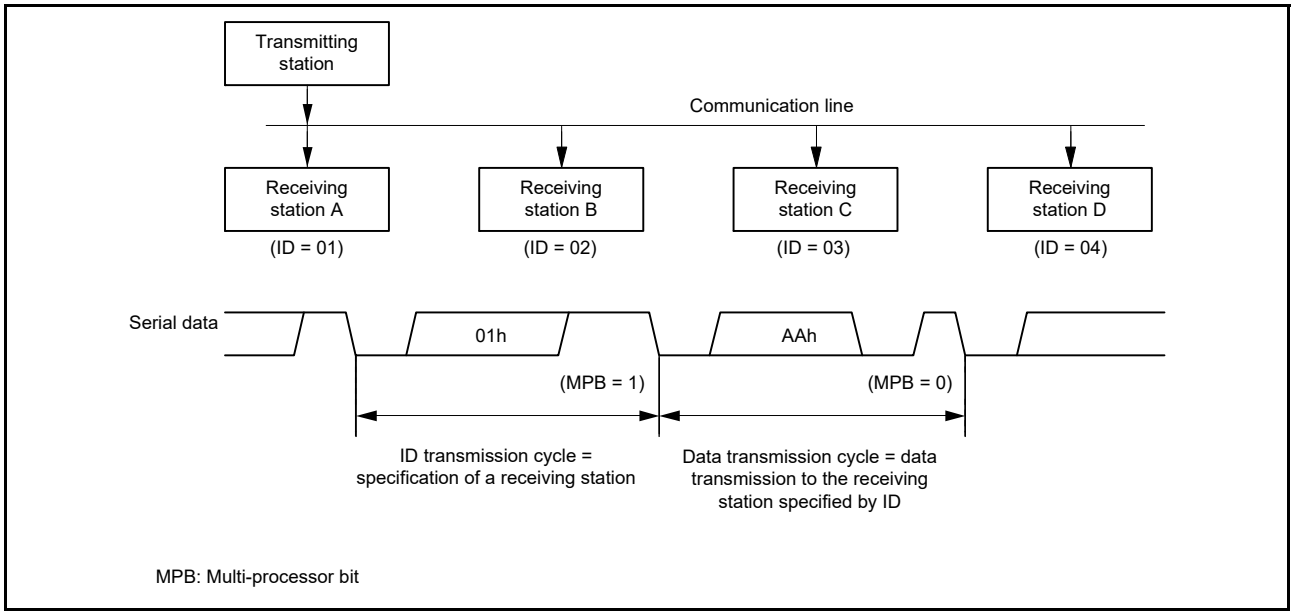


Figure 27.22 Example of communication using multi-processor format with transmission of data AAh to receiving station A

(2) FIFO selected

For data transmission, software must write data to FTDRHL.MPBT that corresponds to transmit data in FTDRHL.TDAT. For data reception, the multi-processor bit that is part of the receive data is written to FRDRHL.MPB and receive data is written to FRDRL.

When the MPIE bit is set to 1, the following operations are disabled until reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from RSR to FRDRHL
- Detection of a receive error
- Break
- Setting of the respective RDF, ORER, and FER status flags in the SSR_FIFO register.

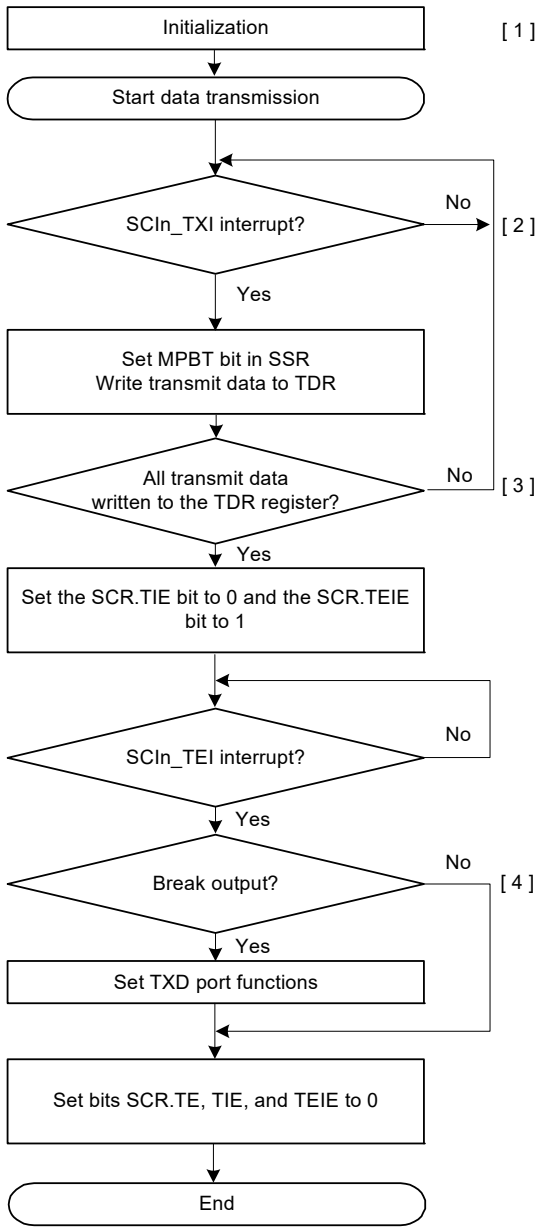
On receiving an 8-bit character in which the multi-processor bit is set to 1, the FRDRHL.MPB bit is set to 1 and receive data is written to FRDRHL.RDAT. The SCR.MPIE bit is automatically cleared, therefore returning the SCI to non multi-processor reception operation. An SCIn_RXI interrupt is generated if the RIE bit in SCR is set.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in non multi-processor asynchronous mode with non-FIFO selected.

27.4.1 Multi-Processor Serial Data Transmission

(1) Non-FIFO selected

Figure 27.23 shows an example flow of multi-processor data transmission. In the ID transmission cycle, the ID must be transmitted with the SSR.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as in asynchronous mode.



[1] SCI initialization:
Set data transmission.
After the TE bit in SCR is set to 1, 1 is output for a frame, and transmission is enabled.

[2] SCIn_TXI interrupt request:
When transmit data is transferred from TDR to TSR, A transmit data empty interrupt (SCIn_TXI) request is generated.
Set the MPBT bit in SSR to 0 or 1, and write transmit data to TDR once in the SCIn_TXI interrupt processing routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR once using an SCIn_TXI interrupt request.
Transmit data can also be written to TDR by activating the DTC.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, after setting the output state (low level output) of TXDn pin using SPTR.SPB2IO and SPTR.SPB2DT bits, set the TE bit in the SCR to 0.

Figure 27.23 Example flow of multi-processor serial transmission with non-FIFO selected

(2) FIFO selected

Figure 27.24 shows an example of data format that is written to FTDRH and FTDRL in multi-processor mode.

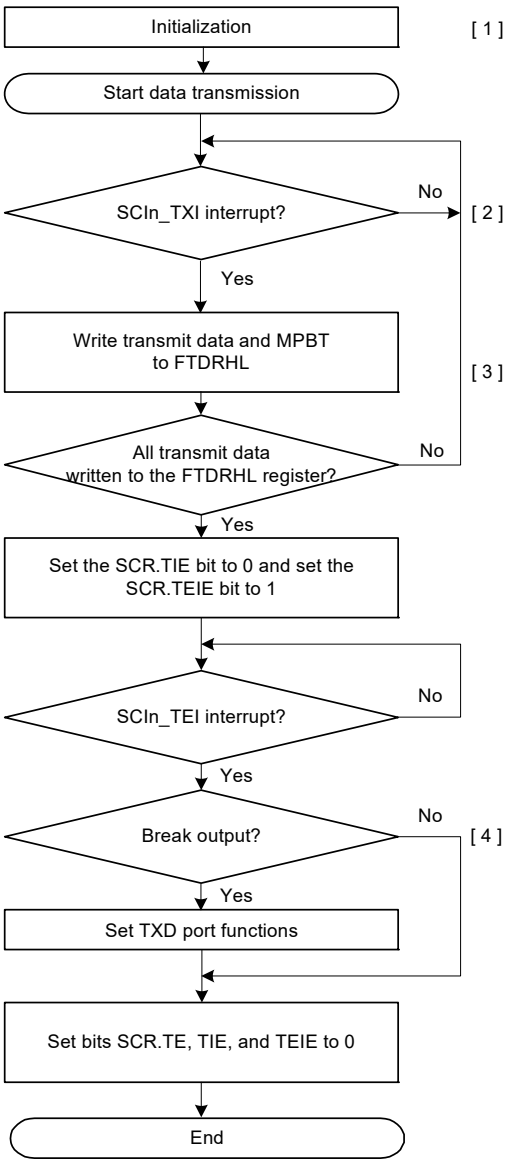
The MPBT is set to 1 in FTDRH. Data is set to FTDRH and FTDRL with the correct data length. Write 0 for unused bits. Write in the order from FTDRH to FTDRL.

Data Length	Register Setting		Transmit data in FTDRH, FTDRL															
			FTDRHL															
			FTDRH								FTDRL							
	SCMR. CHR1	SMR. CHR	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	—	—	—	—	—	—	MPBT	—	—	7-bit transmit data						
8 bits	1	0	—	—	—	—	—	—	MPBT	—	8-bit transmit data							
9 bits	0	Don't care	—	—	—	—	—	—	MPBT	9-bit transmit data								

—: Invalid. The write value should be 0.

Figure 27.24 Data format written to FTDRH and FTDRL in multi-processor mode with FIFO selected

Figure 27.25 shows an example flow of multi-processor data transmission with FIFO selected. In the ID transmission cycle, the ID must be transmitted with the FTDRH.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as in asynchronous mode with non-FIFO selected.



[1] SCI initialization:
Set data transmission.
After the TE bit in SCR is set to 1, 1 is output for a frame, and transmission is enabled.

[2] Transmit data write to 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 DTC.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the FTDRHL.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, after setting the output state (low level output) of TXDn pin using SPTR.SPB2IO and SPTR.SPB2DT bits, set the TE bit in the SCR to 0.

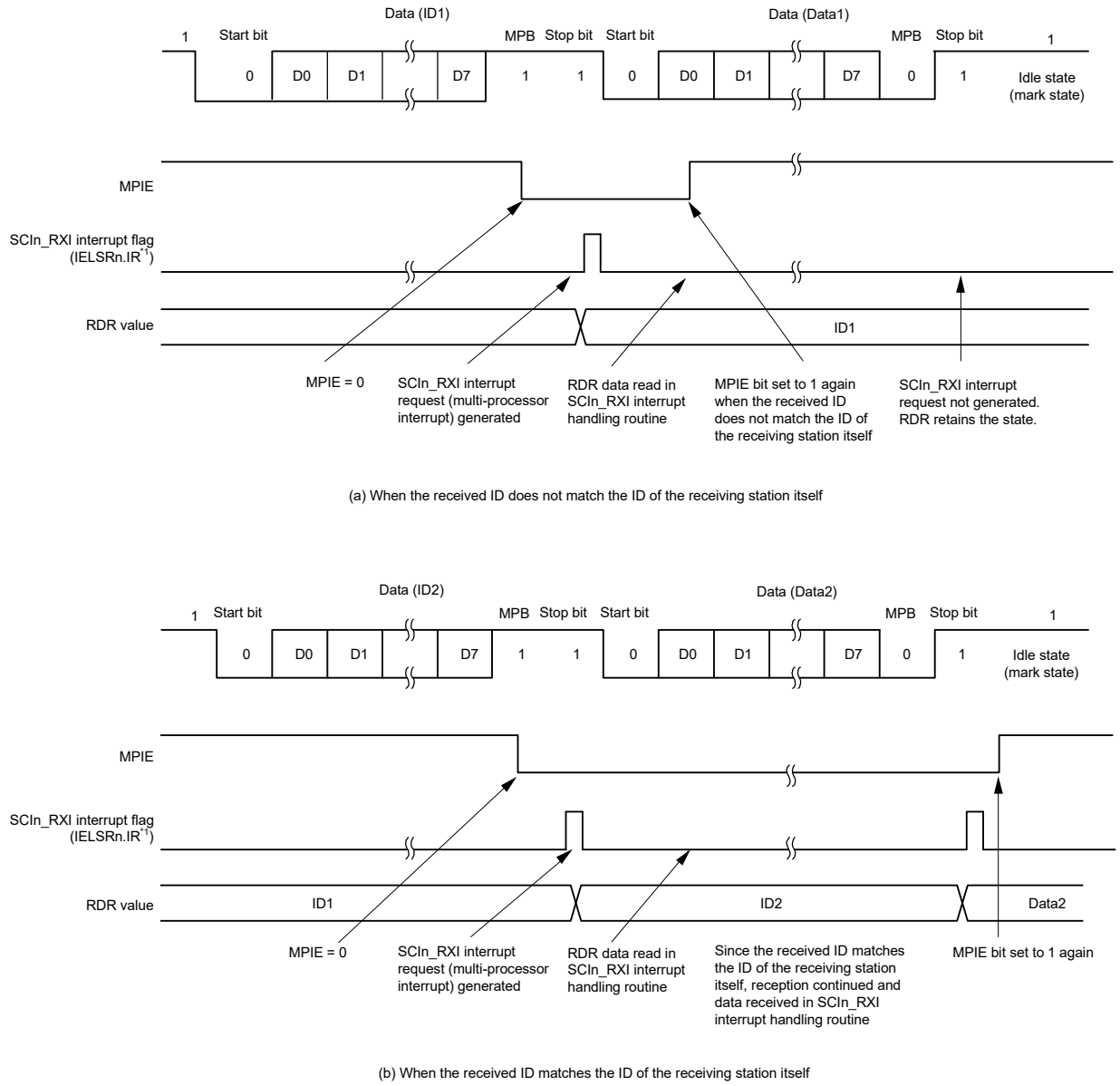
Figure 27.25 Example flow of serial transmission in multi-processor mode with FIFO selected

27.4.2 Multi-Processor Serial Data Reception

(1) Non-FIFO selected

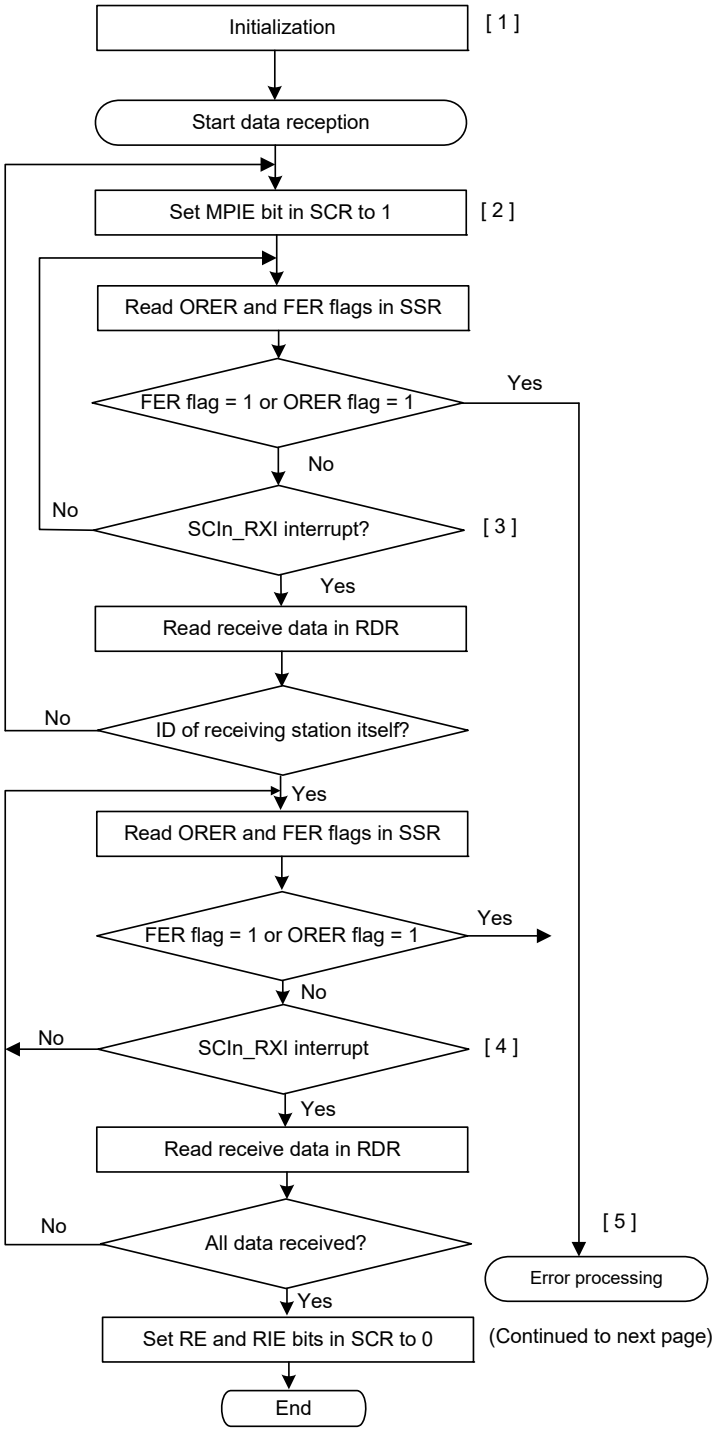
Figure 27.27 and Figure 27.28 show example flows of multi-processor data reception. When the SCR.MPIE bit is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to 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 27.26 shows an example operation for data reception.



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 27.26 Example of SCI reception with 8-bit data, multi-processor bit, and 1 stop bit



[1] SCI initialization:
Set data reception.

[2] ID reception cycle:
Set MPIE bit in SCR to 1 and wait for ID reception.

[3] SCI status confirmation and reception and comparison of ID:
Read data in RDR*1 at the first SCIn_RXI interrupt and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again, and wait for another SCIn_RXI interrupt request.

[4] Data reception at an SCIn_RXI interrupt:
Read data in RDR*1 once in the SCIn_RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and FER flags in SSR. After performing the appropriate error processing be sure to set ORER and FER flags to 0. Reception cannot resume if any of these flags is set to 1. For a framing error, a break can be detected by reading SPTR.RXDMON bit.

Note 1. The RDR register becomes the RDRHL register when 9-bit data length is selected.

Figure 27.27 Example flow of multi-processor serial reception with non-FIFO selected (1)

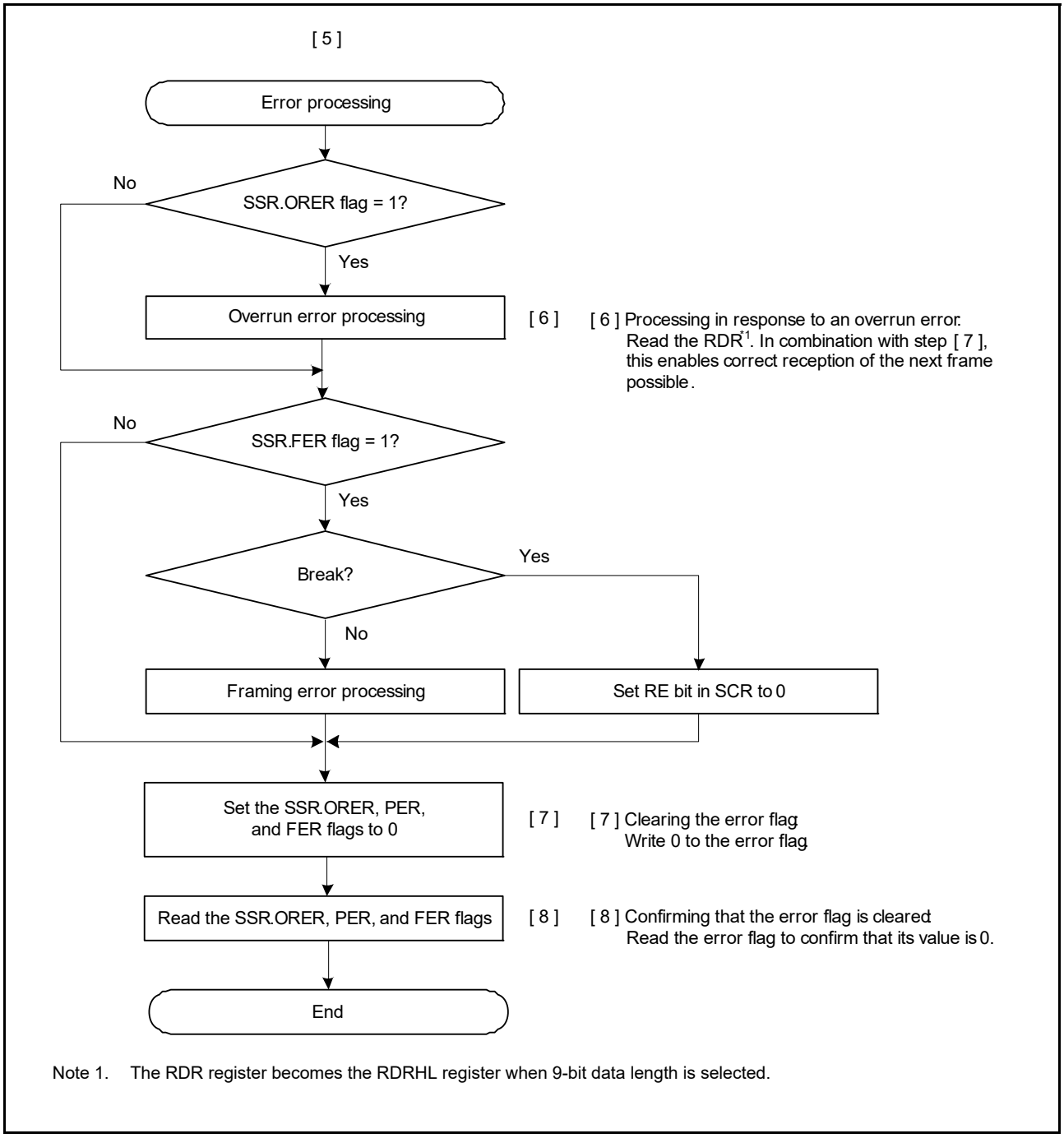


Figure 27.28 Example flow of multi-processor serial reception with non-FIFO selected (2)

(2) FIFO selected

Figure 27.29 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 FRDRH.MPB flag (FRDRHL[9]). A value of 0 is written to the FRDRH.PER flag (FRDRHL[11]). Data is written to FRDRH and FRDRL with the correct data length. Unused bits are written with 0.

Read in the order from FRDRH to FRDRL. When 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	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	—	RDF	ORER	FER	0	DR	MPB	0	0	7-bit receive data						
8 bits	1	0	—	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								

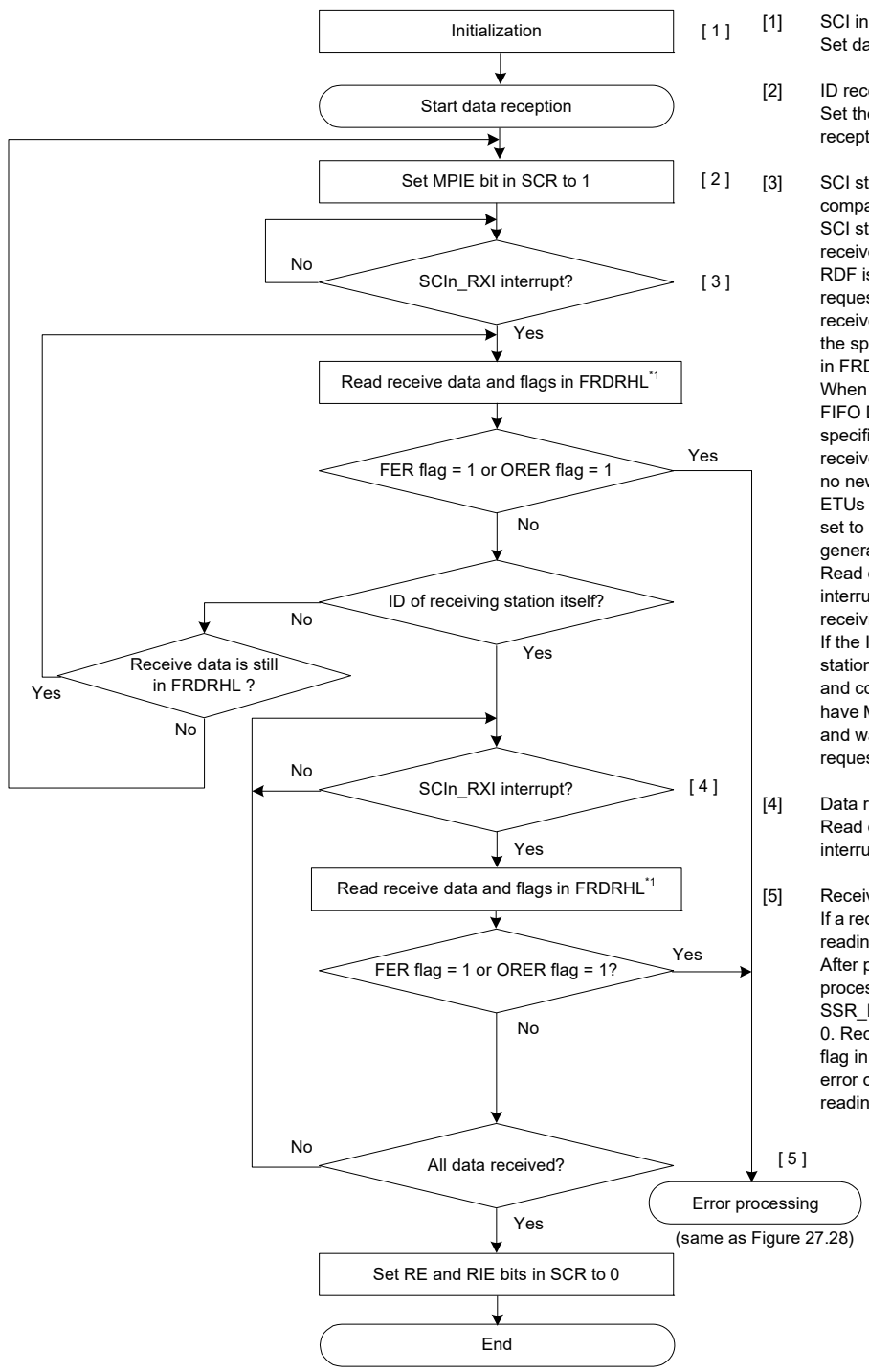
Note: When data length is 7 bits, 0 is always read for FRDRHL[8] and FRDRHL[7].
When data length is 8 bits, 0 is always read for FRDRHL[8].
FRDRHL[15] bit is read as an indefinite value.

Figure 27.29 Data format stored to FRDRH and FRDRL in multi-processor mode with FIFO selected

Figure 27.30 shows an example flow for multi-processor data reception with FIFO selected.

When the SCR.MPIE bit is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data, MPB and associated errors are transferred to FRDRHL. The SCR.MPIE bit is automatically cleared, and non multi-processor reception continues.

If a frame error occurs and the SSR_FIFO.FER flag is set to 1, the SCI continues data reception. The rest of the operations are the same as in asynchronous mode with non-FIFO selected.



[1] SCI initialization:
Set data reception.

[2] ID reception cycle:
Set the MPIE bit in SCR 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 which 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 = 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 once in the SCIn_RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading the ORER and FER flags in SSR_FIFO. After performing the appropriate error processing, be sure to set the SSR_FIFO.ORER and SSR_FIFO.FER flags to 0. Reception cannot be resumed if the ORER flag in SSR_FIFO is set to 1. When a framing error occurs, a break can be detected by reading the SPTR.RXDMON bit.

Note 1. If FRDRH and FRDRL are used instead of FRDRHL, read in the order from FRDRH to FRDRL.

Figure 27.30 Example flow of serial reception in multi-processor mode with FIFO selected

27.5 Operation in Clock Synchronous Mode

Figure 27.31 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 SPMR.CKPH bit is 1 in slave mode, the transmission line holds the first bit as the 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.

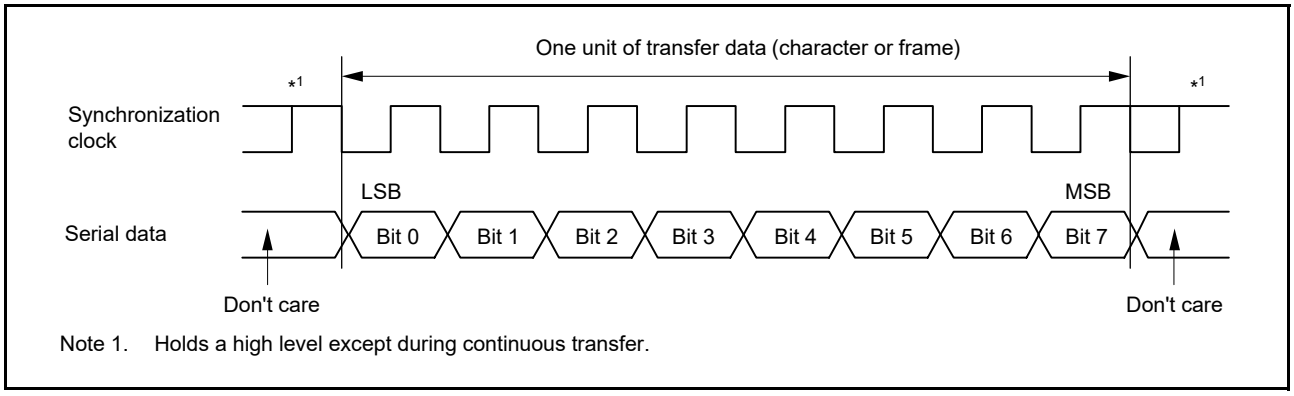


Figure 27.31 Data format in clock synchronous serial communications with LSB-first

27.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCKn pin can be selected based on the SCR.CKE[1:0] setting.

When the SCI operates on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character. When no transfer is performed, the clock is held high. However, when only data reception is performed while the CTS function is disabled, the synchronization clock output starts when the SCR.RE bit is set to 1. The synchronization clock stops when it is held high*1 and when an overrun error occurs, or when the SCR.RE bit is set to 0.

When only data reception occurs and the CTS function is enabled, the clock output does not start when the SCR.RE bit is set to 1 and the CTSn_RTSn input is high. The synchronization clock output starts when the SCR.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 goes high*1 and when an overrun error occurs, or when the SCR.RE bit is set to 0.

Note 1. The signal is held high when SPMR.CKPH = 0 and SPMR.CKPOL = 0, or when SPMR.CKPH = 1 and SPMR.CKPOL = 1.
It is held low when SPMR.CKPH = 0 and SPMR.CKPOL = 1, or when SPMR.CKPH = 1 and SPMR.CKPOL = 0.

27.5.2 CTS and RTS Functions

In the CTS function, the CTSn_RTSn pin input controls the start of data reception or transmission when the clock source is the internal clock. Setting the SPMR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, setting the CTSn_RTSn pin low causes data reception or transmission to start.

Setting the CTSn_RTSn pin high while the data transmission or reception is in progress does not affect transmission or reception of the current frame.

In the RTS function, the CTSn_RTSn pin output is used to request the start of data reception or transmission when the clock source is an external synchronizing clock. The CTSn_RTSn output goes low when serial communication becomes possible. Conditions for output of CTSn_RTSn low and high are as follows:

[Conditions for low-level output]

(a) Non-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
- There is no received data available to be read when the SCR.RE bit is 1
- Transmit data is written when the SCR.TE bit is 1 and SCR.CKE[1] bit is 0
- Data is available for transmission in the TSR register when the SCR.TE bit is 1 and SCR.CKE[1] bit is 1
- The ORER flag in SSR is 0.

(b) FIFO selected when all of the following conditions are satisfied

- The value of the RE or TE bit in the SCR is 1
- When serial communication is enabled
- 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 the SCR.TE bit is 1 and SCR.CKE[1] bit is 0
- Data is available for transmission in TSR when SCR.TE bit is 1 and SCR.CKE[1] bit is 1
- The ORER flag in SSR_FIFO is 0.

[Conditions 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 values after SCR.RE = 0.

(b) FIFO selected

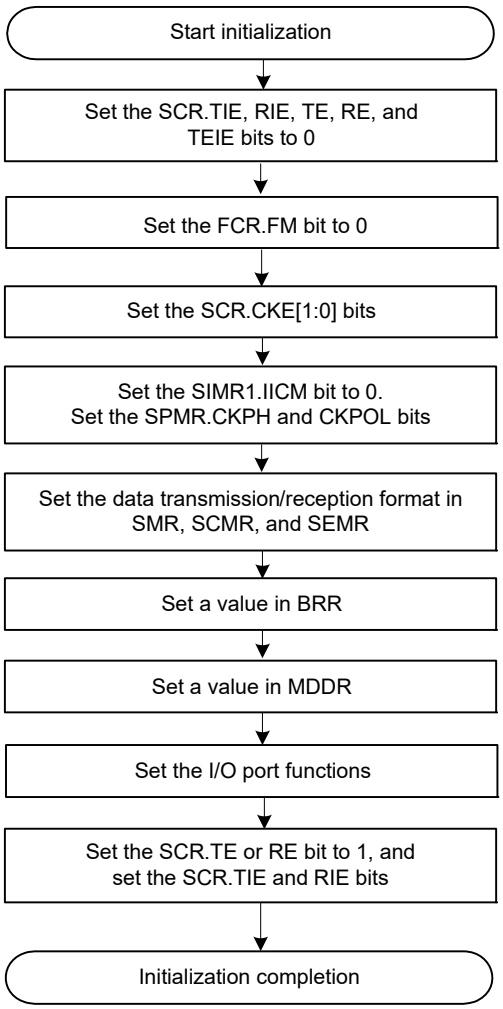
- The conditions for low-level output are not satisfied.

27.5.3 SCI Initialization in Clock Synchronous Mode

Before transmitting and receiving data, start by writing the initial value 00h to the SCR, then continue through the SCI initialization procedure given in the sections describing no-FIFO or FIFO selection in section 27.5.2, CTS and RTS Functions. Anytime the operating mode or transfer format is to be changed, the SCR register must be initialized before the change can be made.

Note: Setting the SCR.RE bit to 0 does not initialize the ORER, FER, RDRF, RDF, PER, and DR flags in SSR/SSR_FIFO or the RDR and RDRHL registers. When the TE bit is set to 0, the TEND flag for the selected FIFO buffer is not initialized.

Note: In non-FIFO mode, Switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 when the SCR.TIE bit is 1 generates an SCIn_TXI interrupt request.



[1] [1] Set the FCR.FM bit to 0.

[2] [2] Set the clock selection in SCR.

[3] [3] Set the SIMR1.IICM bit 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] [4] Set data transmission/reception format in SMR, SCMR, and SEMR.

[5] [5] Write a value associated with the bit rate to BRR.
This step is not required if an external clock is used.

[6] [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] [7] Specify the I/O port settings to enable input and output functions as required for the TXDn, RXDn, and SCKn pins.

[8] [8] Set the SCR.TE or RE bit to 1. Also set the SCR.TIE and RIE bits.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note: In simultaneous transmit and receive operations, the TE and RE bits in SCR must both be set to 0 or set to 1 simultaneously.

Figure 27.32 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected

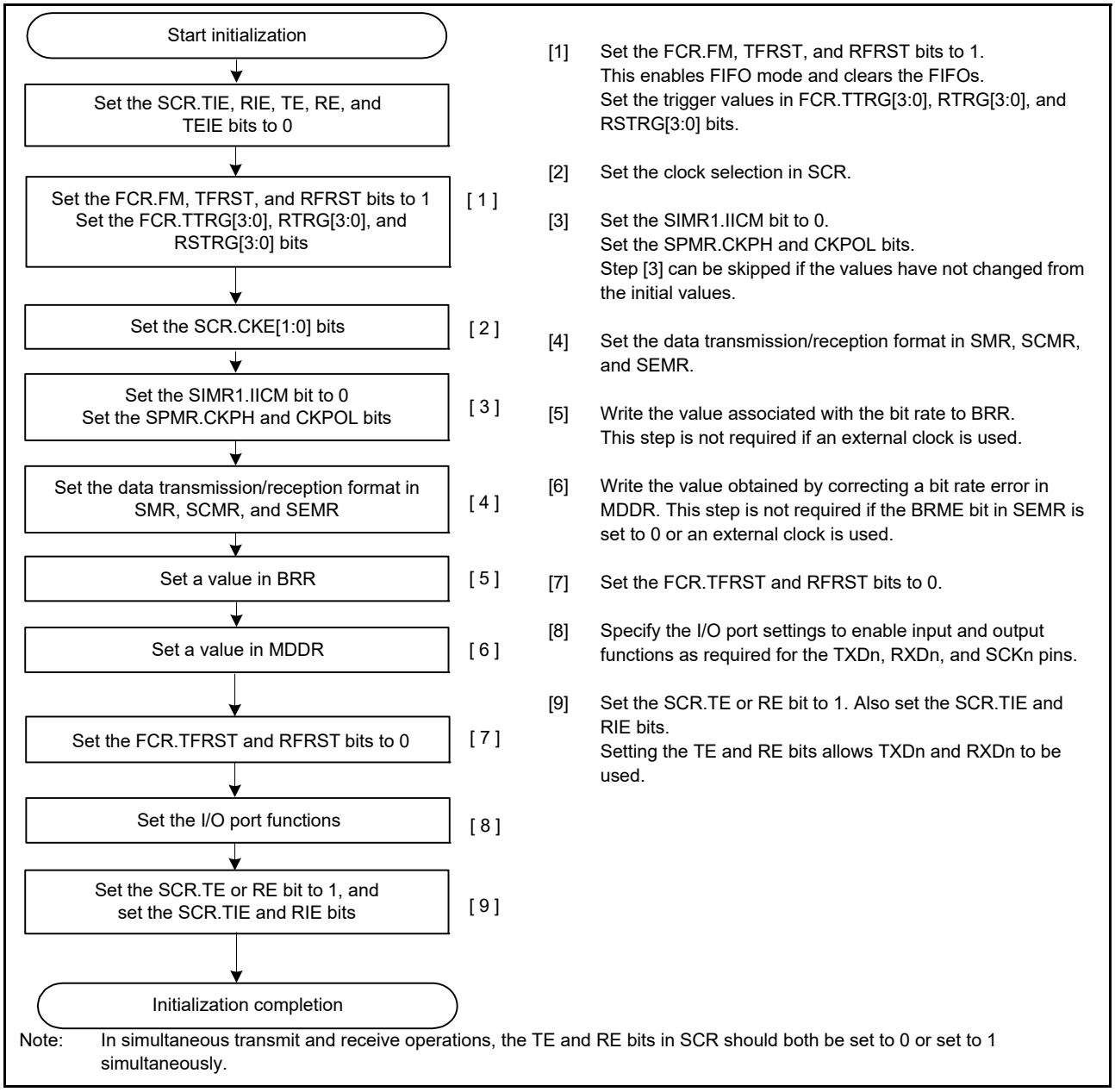


Figure 27.33 Example flow of SCI initialization in clock synchronous mode with FIFO selected

27.5.4 Serial Data Transmission in Clock Synchronous Mode

(1) Non-FIFO selected

Figure 27.34, Figure 27.35, and Figure 27.36 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 SCR.TIE bit is set to 1, an SCIn_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to TDR in the SCIn_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data

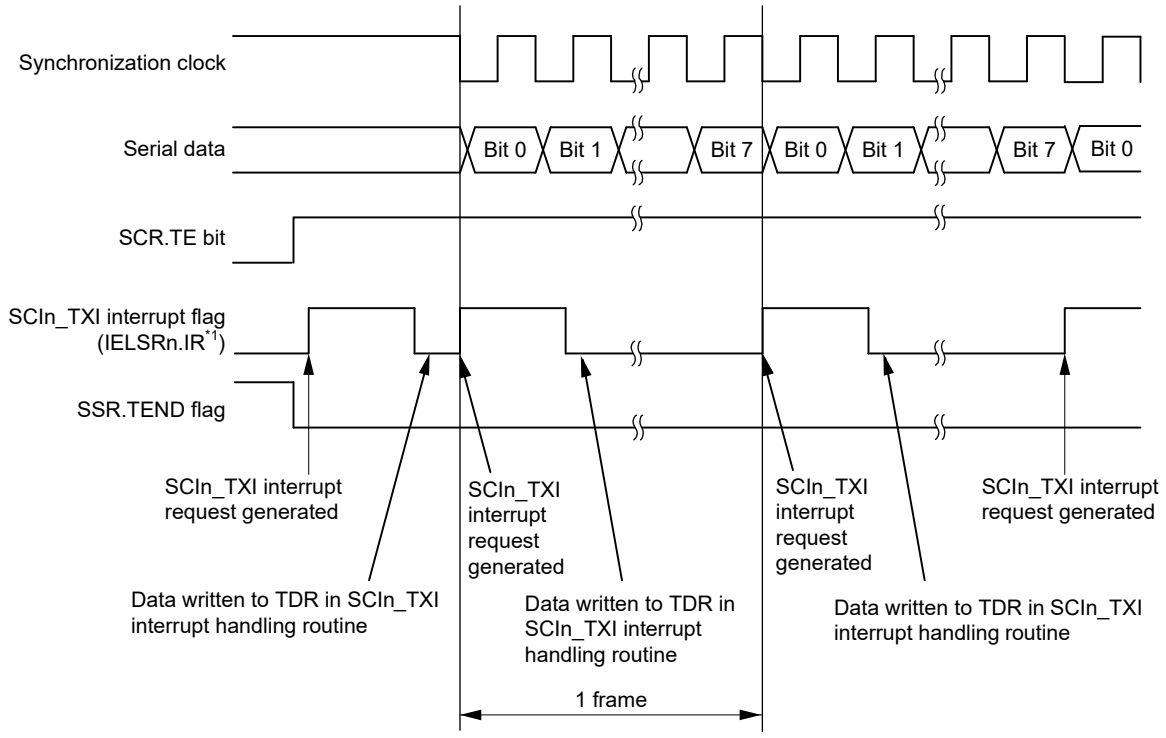
to be transmitted is written to the TDR from the handling routine for SCIn_TXI requests.

3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified, and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the input CTS signal is low and while the CTSE bit in SPMR is 1.
4. The SCI checks for update to the 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, set the SSR.TEND flag to 1. The TXDn pin keeps the output state of the last bit. If the TEIE bit in SCR is 1, an SCIn_TEI interrupt request is generated and the SCKn pin is held high.

Figure 27.34, Figure 27.35, and Figure 27.36 show example flows of serial data transmission.

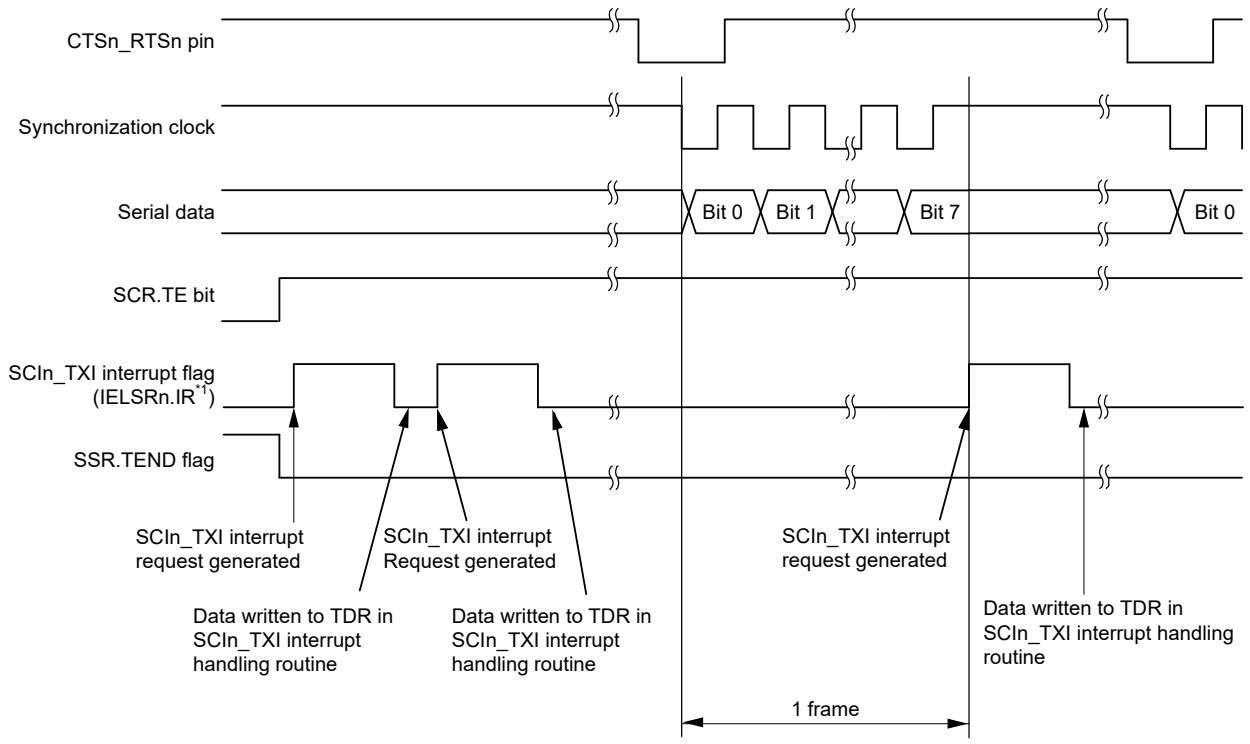
Transmission does not start while a receive error flag (ORER, FER, or PER in SSR) is set to 1. Be sure to set the receive error flags to 0 before starting transmission.

Note: Setting the SCR.RE bit to 0 does not clear the receive error flags.



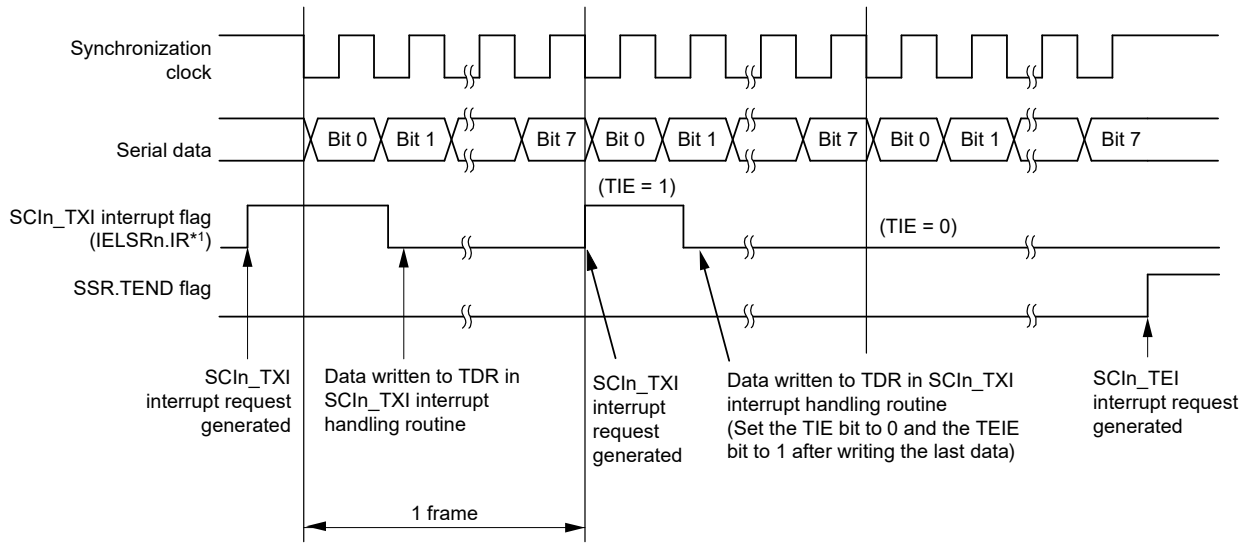
Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 27.34 Example of serial data transmission in clock synchronous mode when the CTS function is not used at the beginning of transmission



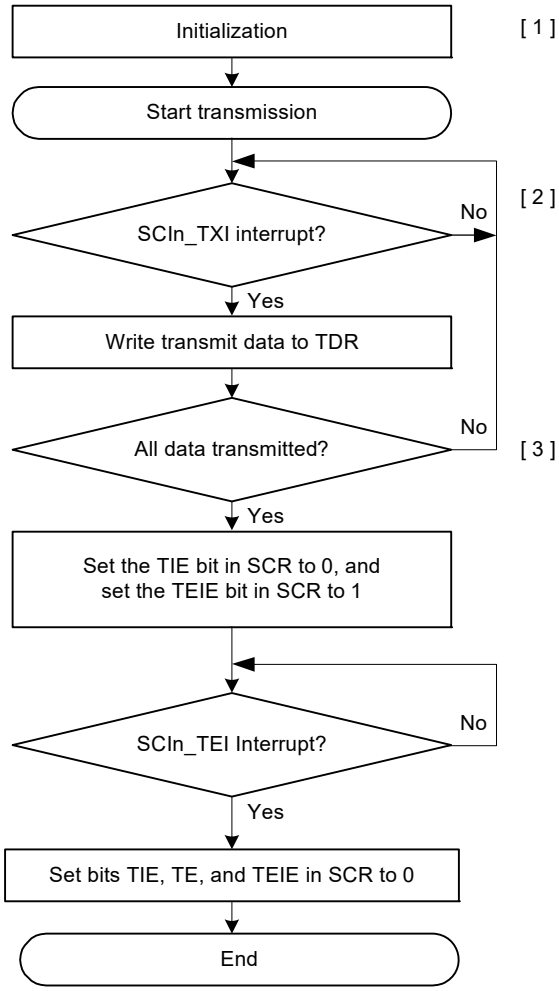
Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 27.35 Example of serial data transmission in clock synchronous mode when the CTS function is used at the beginning of transmission



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 27.36 Example of serial data transmission in clock synchronous mode from the middle of transmission until transmission completion



[1] SCI initialization:
Set data transmission.

[2] Write transmit data to TDR by an SCIn_TXI interrupt request:
When data transmission 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 on accepting a transmit data empty interrupt (SCIn_TXI) request. Transmit data can also be written to TDR by activating the DTC by the SCIn_TXI interrupt request.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the TDR.

Note: When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Setting the SCR.TE bit to 0 immediately after this might lead to insufficient hold time for received data on the receiver side.

Figure 27.37 Example flow of serial transmission in clock synchronous mode with non-FIFO selected

(2) FIFO selected

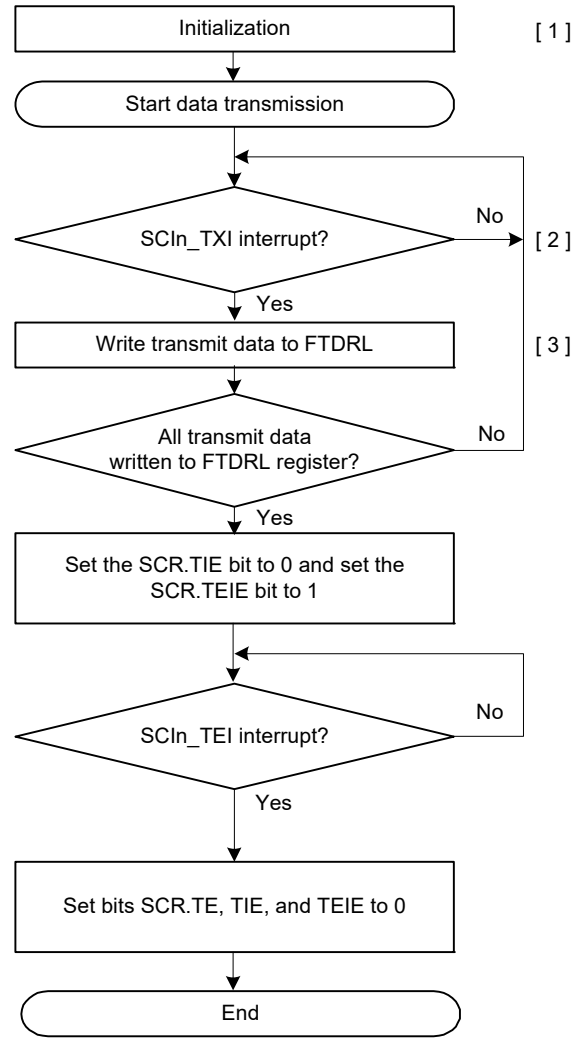
Figure 27.34 shows an example of serial transmission in clock synchronous mode with FIFO selected.

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 the TE bit in SCR is set to 1 but only after the SCR.TIE bit is also set to 1, or when SCR.TE and SCR.TIE are both set to 1 simultaneously by a single instruction.
2. After transferring data from FTDRL to TSR, the SCI starts transmission. When the amount of transmit data written in FTDRL is equal to or less than the specified transmit triggering number, the SSR_FIFO.TDFE is set to 1. When the SCR.TIE bit is set to 1, an SCIn_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to FTDRL in the SCIn_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the FTDRL from the handling routine for SCIn_TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the CTS 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 SSR_FIFO.TEND flag is set to 1. The TXDn pin keeps the output state of the last bit. If the TEIE 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.



[1] SCI initialization:
Set data transmission.
After the TE bit in SCR is set to 1, 1 is output for a frame, and transmission is enabled.

[2] Transmit data write to FTDRL by an SCIn_TXI interrupt request:
When data transmission is transferred from FTDRL to TSR, and when the amount of transmit data written in FTDRL is equal to or less than the specified transmit triggering number, a transmit data FIFO empty interrupt (SCIn_TXI) request is generated. Write transmit data to FTDRL once in the SCIn_TXI interrupt handling routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write the next transmit data to FTDRL in this SCIn_TXI interrupt handling routine and clear the SSR_FIFO.TDFE flag to 0 before transmission of the current transmit data is complete. Transmit data can also be written to FTDRL by activating the DTC. The TDFE flag is cleared automatically in this case. Do not write to the TDFE flag.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the FTDRL.

Note: When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR_FIFO.TEND flag to 1. Setting the SCR.TE bit to 0 immediately after this might lead to insufficient hold time for received data on the receiver side.

Figure 27.38 Example flow of serial transmission in clock synchronous mode with FIFO selected

27.5.5 Serial Data Reception in Clock Synchronous Mode

(1) Non-FIFO selected

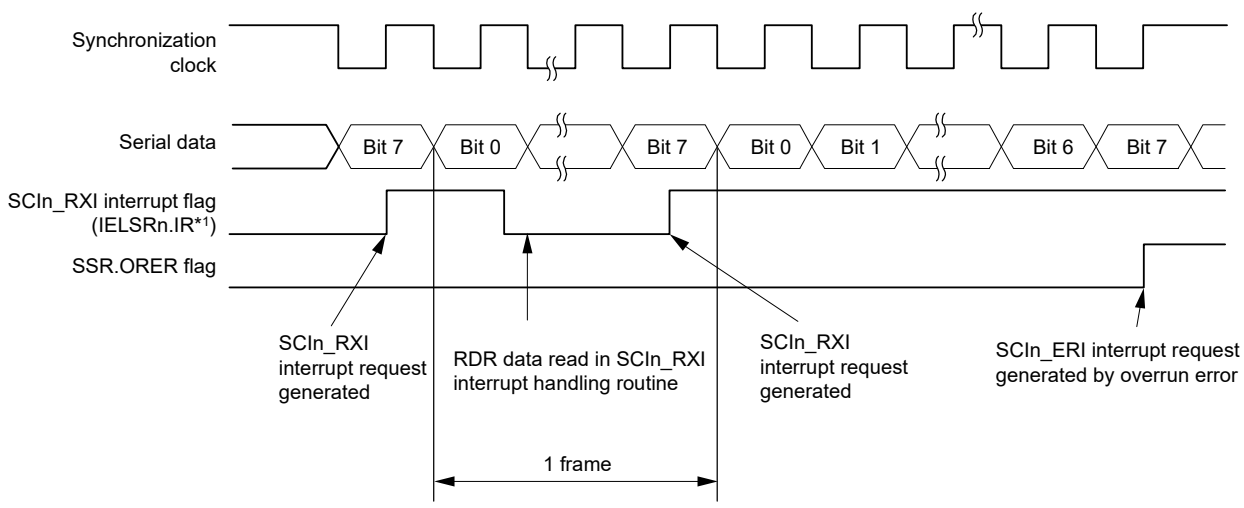
Figure 27.39 and Figure 27.40 show examples of SCI operation for serial reception in clock synchronous mode.

In serial data reception, the SCI operates as follows:

1. When the value of the SCR.RE bit becomes 1, the CTSn_RTSn pin goes low.
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in RSR.
3. If an overrun error occurs, the SSR.ORER flag is set to 1. If the RIE bit in SCR is 1, an SCIn_ERI interrupt request

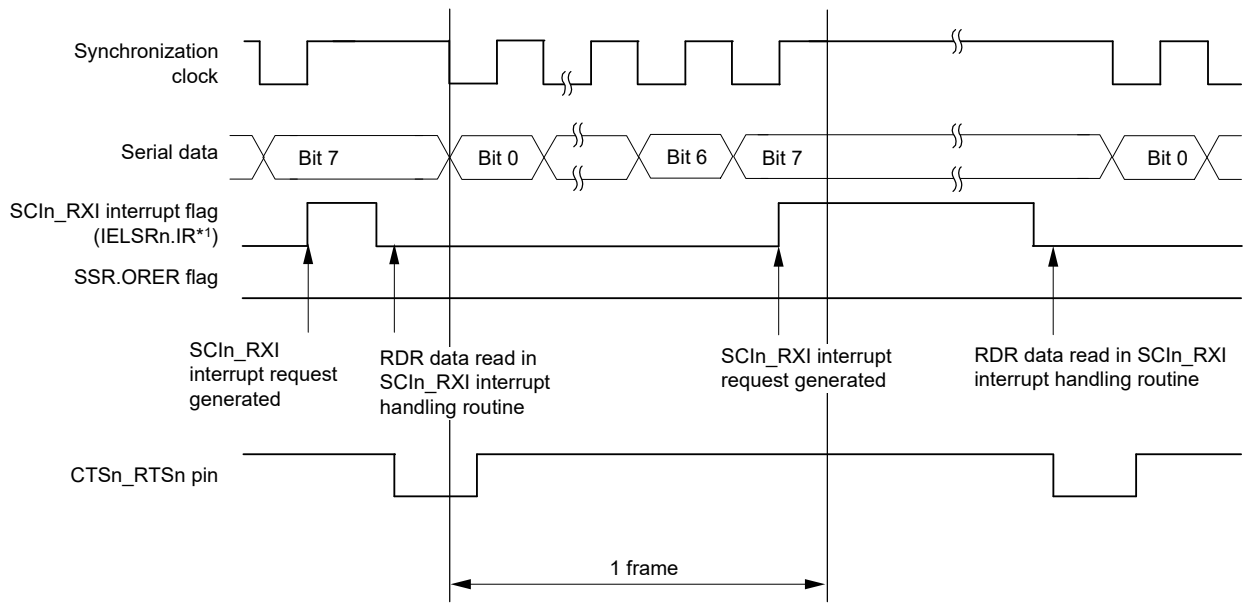
is generated. Receive data is not transferred to RDR.

4. When reception completes successfully, receive data is transferred to RDR. If the SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in the SCIn_RXI interrupt handling routine before reception of the next receive data completes. Reading the received data from RDR causes the CTSn_RTSn pin to output low.



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 27.39 Example operation of serial reception in clock synchronous mode (1) when RTS function is not used

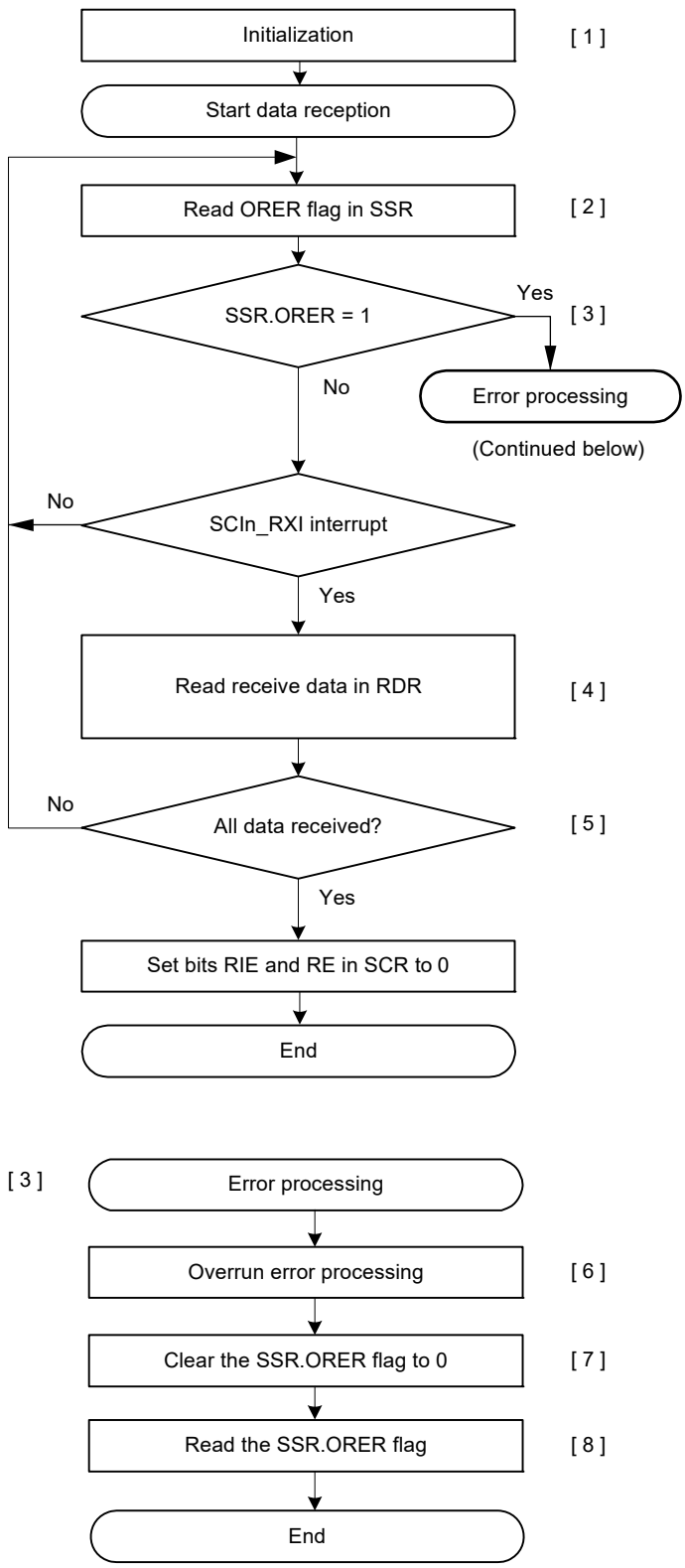


Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 27.40 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 register during overrun error processing. When a data reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR register because received data that is not yet read might be left in the RDR register.

Figure 27.41 shows an example flow of serial data reception.



[1] SCI initialization:
Set the input port for pins to be used as RXDn pins.

[2] [3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, perform the relevant error processing, then set the ORER flag to 0. Data reception cannot resume while the ORER flag is 1.

[4] Read the receive data in RDR once in the receive data full interrupt (SCIn_RXI) request handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB bit [7] of the current frame is received, finish reading the receive data in RDR. The RDR data can also be read by activating the DTC by an SCIn_RXI interrupt request.

[6] Processing in response to an overrun error:
Read the 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 27.41 Example flow of serial reception in clock synchronous mode with non-FIFO selected

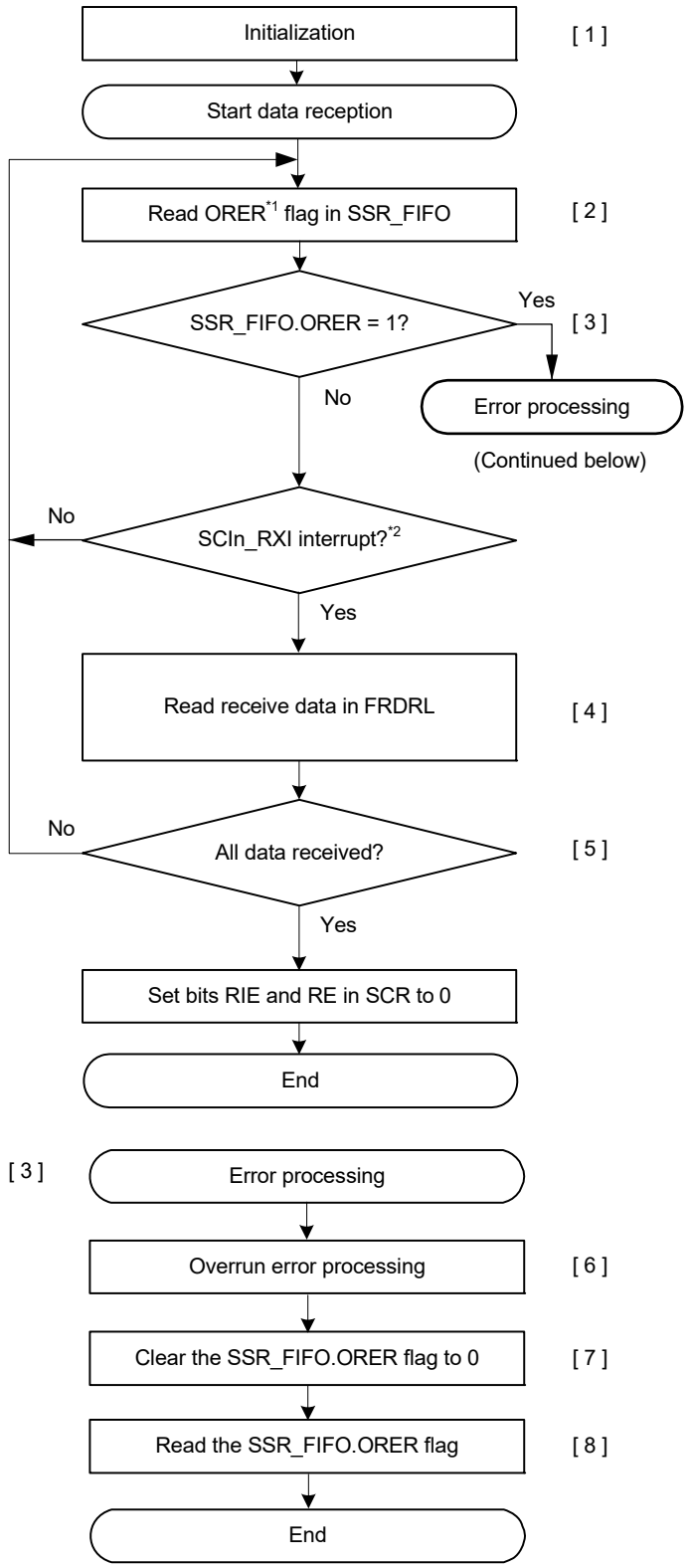
(2) FIFO selected

Figure 27.42 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 the SCR.RE bit becomes 1, the CTSn_RTSn pin goes low.
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in 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 bit 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 SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous data reception is enabled by reading the receive data transferred to FRDRL*2 in the SCIn_RXI interrupt handling routine before an overrun error occurs. If the amount of received data that is transferred to FRDRL is less than the RTS trigger number, the CTSn_RTSn pin goes low.

Note 1. In clock synchronous mode, FTDRH is not used.
Note 2. Read data in the order from FRDRH to FRDRL when RDF and ORER are read with receive data.



[1] SCI initialization:
Set the input port for pins to be used as RXDn pins.

[2] [3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR_FIFO, perform the relevant error processing, then set the SSR_FIFO.ORER flag to 0. Data reception cannot resume while the ORER flag is 1.

[4] The receive data stored in FRDRL register is read until the amount of stored data is less than the FCR.RTRG[3:0] value. Software can check readable data in FDR.R[4:0].

[5] Serial reception continuation procedure:
To continue serial reception before overrun error occurs, finish reading the receive data in FRDRL and clear the SSR_FIFO.RDF flag to 0.The FRDRL data can also be read by activating the DTC by an SCIn_RXI interrupt request. The RDF flag is cleared automatically in this case. Do not write to the RDF flag.

[6] Processing in response to an overrun error:
Read the FRDRL. In combination with step [7], this enables correct reception of the next frame possible.

[7] Clearing the error flag:
Write 0 to the error flag.

[8] Confirming that the error flag is cleared:
Read the error flag to confirm that its value is 0.

Note 1. ORER can also read from FRDRH.ORER. However, to clear the ORER flag, write 0 to the associated bit in the SSR_FIFO register.
Note 2. All receive data is an integer multiple of the FIFO triggering number.

Figure 27.42 Example flow of serial reception in clock synchronous mode with FIFO selected

27.5.6 Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode

(1) Non-FIFO selected

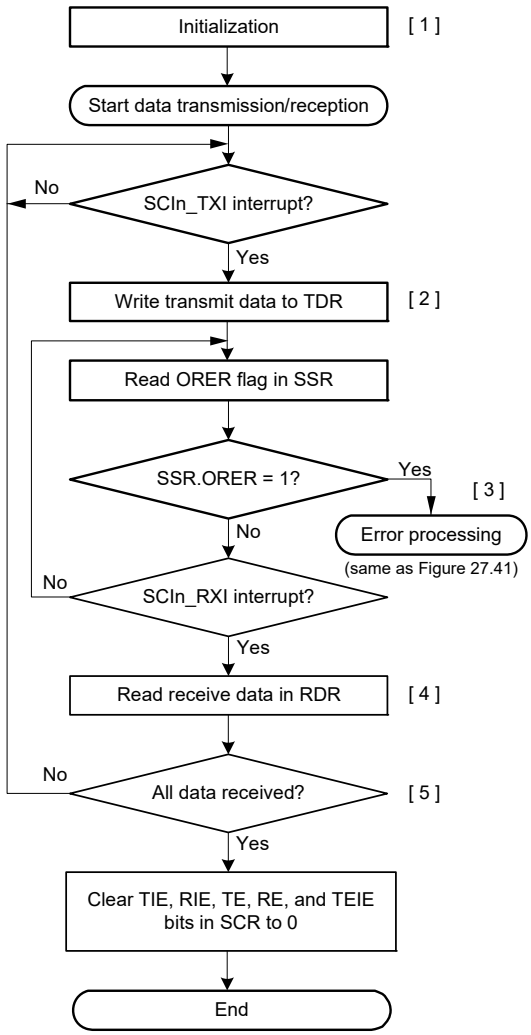
Figure 27.43 shows an example flow of simultaneous serial transmission and reception operations in clock synchronous mode. After initializing the SCI, use the following procedure for simultaneous serial data transmission and reception operations.

To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the data transmission by verifying that the SSR.TEND flag is set to 1.
2. Initialize the SCR register and then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously with 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 in the SCR register to 0, then check that the receive error flag ORER in the SSR register is 0.
3. Set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously with a single instruction.



[1] SCI initialization:
The TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to TDR once in the SCIn_TXI interrupt request handling routine.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, perform the relevant error processing, then set the ORER flag to 0. Data reception cannot resume while the ORER flag is 1.

[4] Reading receive data:
Read the receive data in RDR once in the SCIn_RXI interrupt request handling routine.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission and reception, before the MSB bit [7] of the current frame is received, finish reading the receive data in RDR by the SCIn_RXI interrupt. Also, before the MSB bit [7] of the current frame is transmitted, write data to TDR by the SCIn_TXI interrupt.
Transmit data can also be written to TDR by activating the DTC by a transmit data empty interrupt (SCIn_TXI) request. Similarly, the RDR data can also be read by activating the DTC by a receive data full interrupt (SCIn_RXI) request.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first set the TIE, RIE, TE, RE, and TEIE bits in SCR to 0, then set TIE, RIE, TE, and RE bits to 1, simultaneously.

Figure 27.43 Example flow of simultaneous serial transmission and reception in clock synchronous mode with non-FIFO selected

(2) FIFO selected

Figure 27.44 shows an example flow of simultaneous serial transmit and receive operations in clock synchronous mode with FIFO selected.

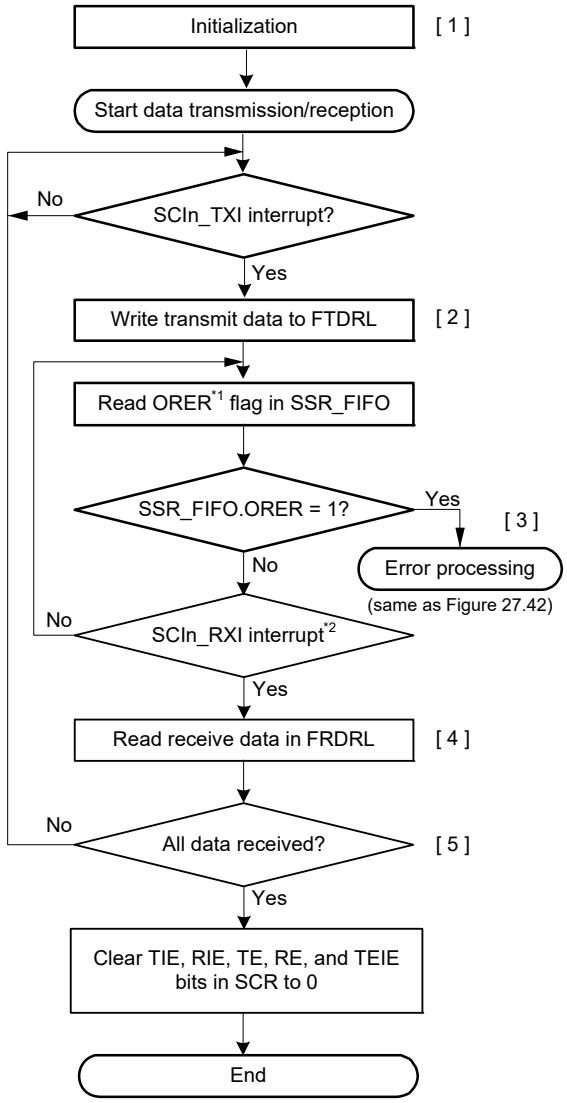
After initializing the SCI, use the following procedure for simultaneous serial data transmit and receive operations.

To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the transmission by verifying that the TEND flag in SSR_FIFO is set to 1.
2. Initialize the SCR register, then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously with a single instruction.

To switch from receive mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the reception.
2. Set the RIE and RE bits to 0, 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 with a single instruction.



[1] SCI initialization:
The TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to FTDRL in the SCIn_TXI interrupt request handling routine. The amount of transmit data that can be written is 16 bits, which is the number of stored transmit data in the FIFO.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR_FIFO, perform the relevant error processing, and then set the ORER flag to 0. Data reception cannot resume while the ORER flag is 1.

[4] Reading receive data:
The receive data stored in FRDRL register is read until the number of stored data is less than the FCR.RTRG value. Software can check readable data in FDR.R[4:0].

[5] Serial transmission/reception continuation procedure:
To continue serial reception, before overrun error occurs, finish reading the receive data in FRDRL and clear the SSR_FIFO.RDF flag to 0. To continue serial transmission, before transmission of the current transmit data is finished, write the next transmit data to FTDRL in the SCIn_TXI interrupt handling routine and clear the SSR_FIFO.TDFE flag to 0.
Transmit data can also be written to FTDRL by activating the DTC by a transmit FIFO data empty interrupt (SCIn_TXI) request. Similarly, the FRDRL data can also be read by activating the DTC by a receive FIFO data full interrupt (SCIn_RXI) request. The RDF and TDFE flags are cleared automatically in this case.
Do not write to the RDF and TDFE flags.

Note 1. 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 27.44 Example flow of simultaneous serial transmission and reception in clock synchronous mode with FIFO selected

27.6 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.

27.6.1 Example Connection

Figure 27.45 shows an example connection between a smart card (IC card) and the MCU.

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, as shown in Figure 27.45.

Setting the TE and RE bits in the SCR_SMCI register 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.

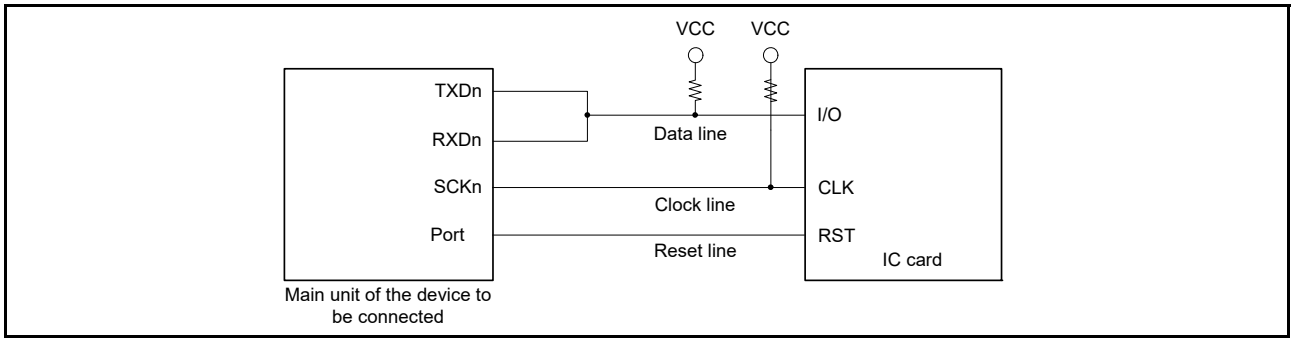


Figure 27.45 Example connection with a smart card (IC Card)

27.6.2 Data Format (Except in Block Transfer Mode)

Figure 27.46 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 value of at least 2 ETUs (elementary time unit, which is the time required to transfer 1 bit) is set as a guard time from the end of the parity bit until the start of the next frame
- If a parity error is detected during reception, a low error signal is output for 1 ETU after 10.5 ETUs elapse from the start bit
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 ETUs.

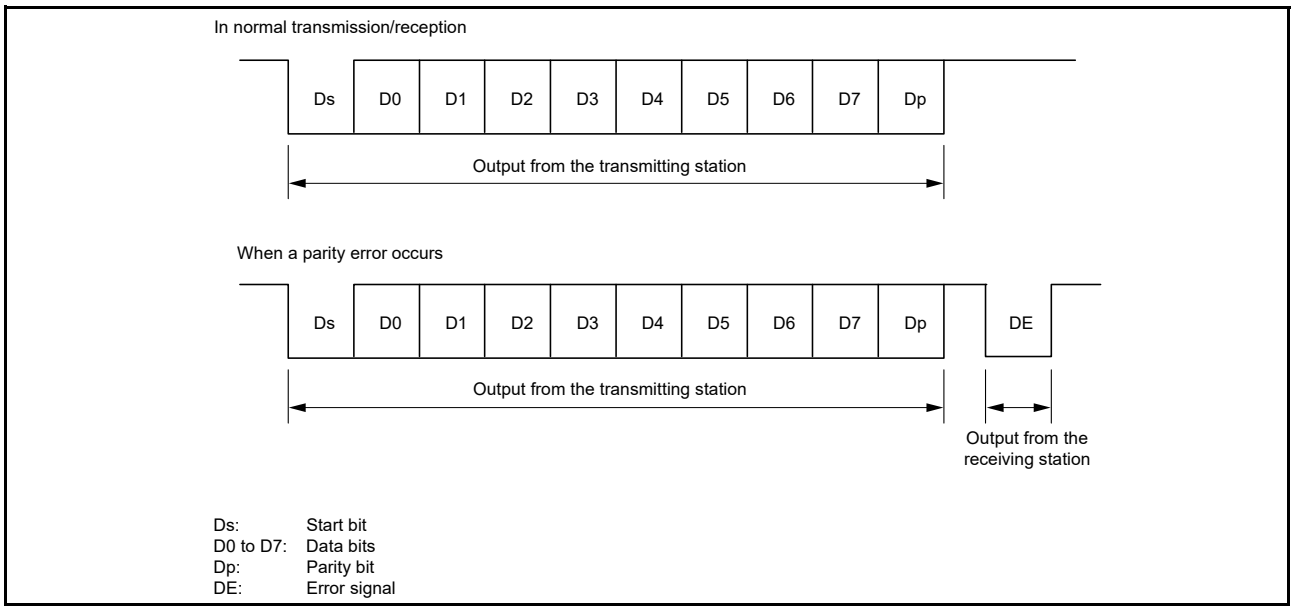


Figure 27.46 Data formats in smart card interface mode

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 27.47. Therefore, data in the start character in Figure 27.47 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.

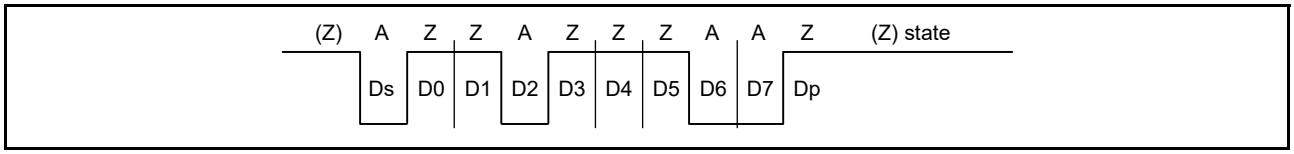


Figure 27.47 Direct convention with SDIR in SCMR = 0, SINV in SCMR = 0, and PM in SMR_SMCI = 0

(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 Figure 27.48 shows. Therefore, data in the start character in the figure is 3Fh.

When using the inverse convention type, write 1 to both the SDIR and SINV bits in SCMR. The parity bit is logic level 0 to produce even parity, which is described by the smart card standard, and corresponds to state Z. Because the SINV bit only inverts data bits D7 to D0, write 1 to the PM bit in SMR_SMCI to invert the parity bit for both transmission and reception.

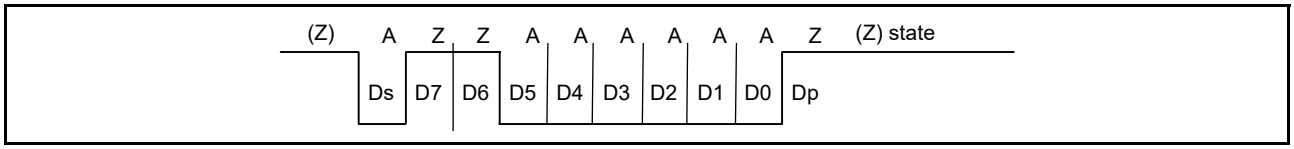


Figure 27.48 Inverse convention with SDIR in SCMR = 1, SINV in SCMR = 1, and PM in SMR_SMCI = 1

27.6.3 Block Transfer Mode

Block transfer mode differs from non-block transfer mode of the smart card interface mode as follows:

- If a parity error is detected during reception, no error signal is output. Because the PER flag in SSR_SMCI is set by error detection, clear the PER flag before receiving the parity bit of the next frame.
- During transmission, at least 1 ETU is set as a guard time from the end of the parity bit until the start of the next frame
- Because the same data is not retransmitted, the TEND flag in SSR_SMCI is set to 11.5 ETUs after transmission starts
- In block transfer mode, the ERS flag in SSR_SMCI indicates the error signal status as in non-block transfer mode of smart card interface mode, but the flag is read as 0 because no error signal is transferred.

27.6.4 Receive Data Sampling Timing and Reception Margin

Only the clock generated by the on-chip baud rate generator can be used as a transfer clock in smart card interface mode.

In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate according to the SCMR.BCP2 and SMR_SMCI.BCP[1:0] bit settings.

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 shown in Figure 27.49. The reception margin is determined by the following formula:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ [\%]$$

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 × 372)} × 100 [%] = 49.866%

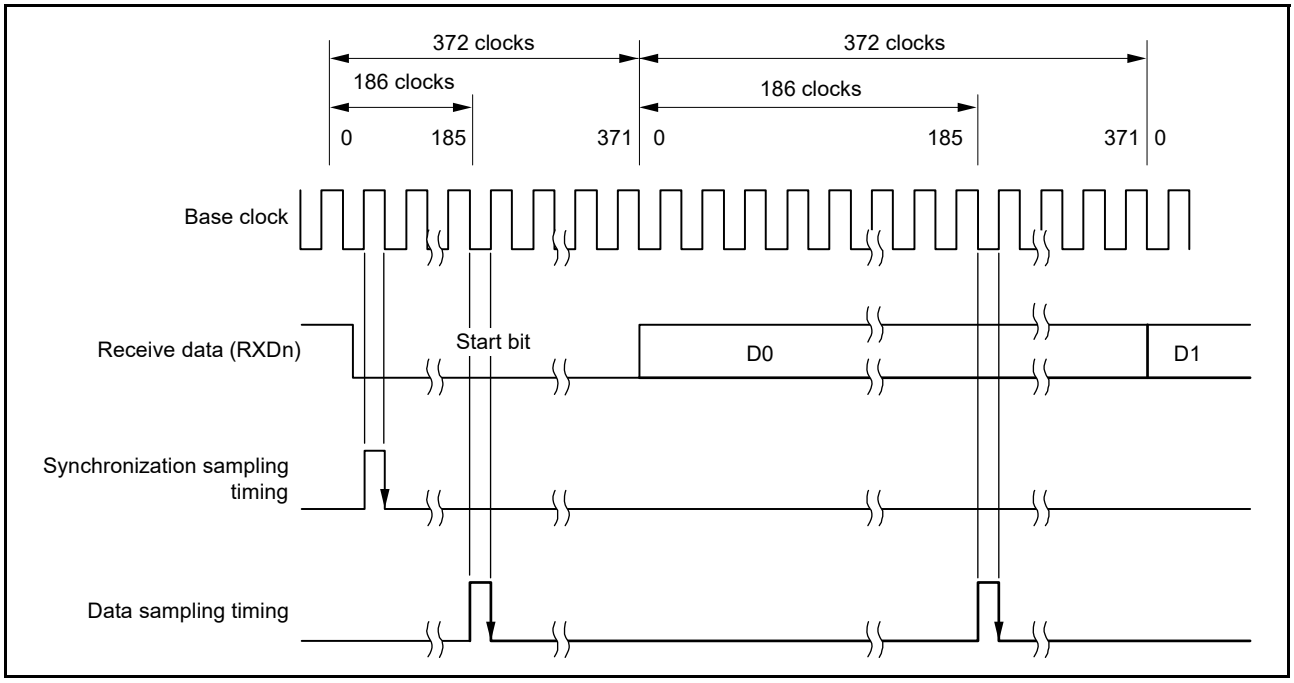


Figure 27.49 Receive data sampling timing in smart card interface mode when clock frequency is 372 times the bit rate

27.6.5 SCI Initialization

Before transmitting and receiving data, write the initial value 00h in the SCR_SMCI register and initialize the SCI using the example flow shown in Figure 27.50.

Be sure to set the initial value in the TIE, RIE, TE, RE, TEIE bits in the SCR_SMCI register before switching from transmission to reception mode, or from reception to transmission mode. When the SCR_SMCI.RE bit is set to 0, the RDR register is not initialized.

To change from reception mode to transmission mode, first check that reception is complete, then initialize the SCI. At the end of initialization, set TE = 1 and RE = 0. Reception completion can be verified by reading the SCIn_RXI request, ORER, or PER flag in SSR_SMCI.

To change from transmission mode to reception mode, first check that transmission is complete, then initialize the SCI. At the end of initialization, set TE = 0 and RE = 1. Transmission completion can be verified by reading the TEND flag in SSR_SMCI.

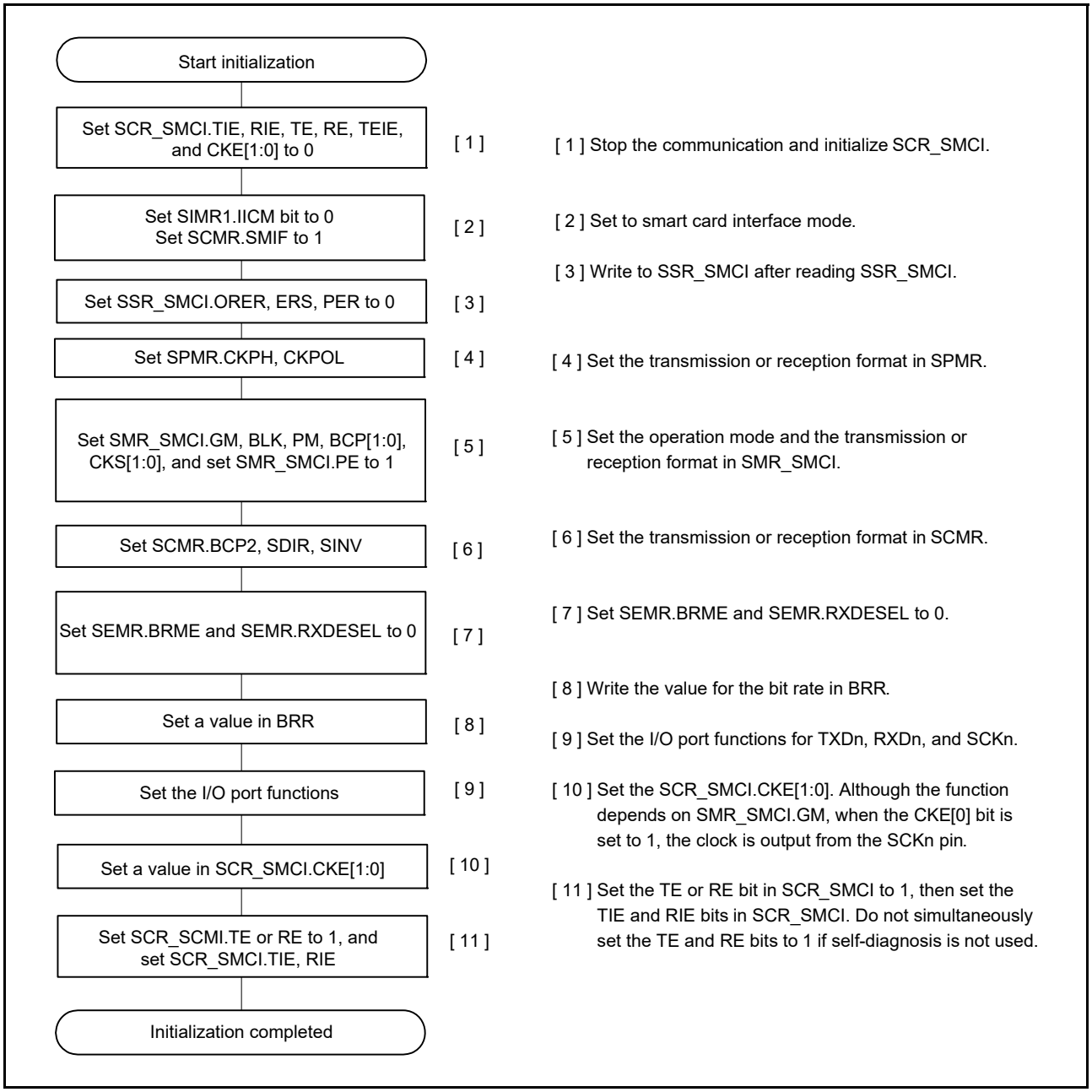


Figure 27.50 Example flow of SCI initialization in smart card interface mode

Figure 27.51 shows a timing diagram when data transmission is performed by transitioning to smart card interface mode according to the flow in Figure 27.50. Figure 27.51 shows when the GM bit in SMR_SMCI is set to 0. The timing in Figure 27.51 shows when the port is connected as SCKn pin and TXDn pin, 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 the 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 the preamble period. Pull-up or pull-down for the SCKn and TXDn pins is required outside the MCU.

In the 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.

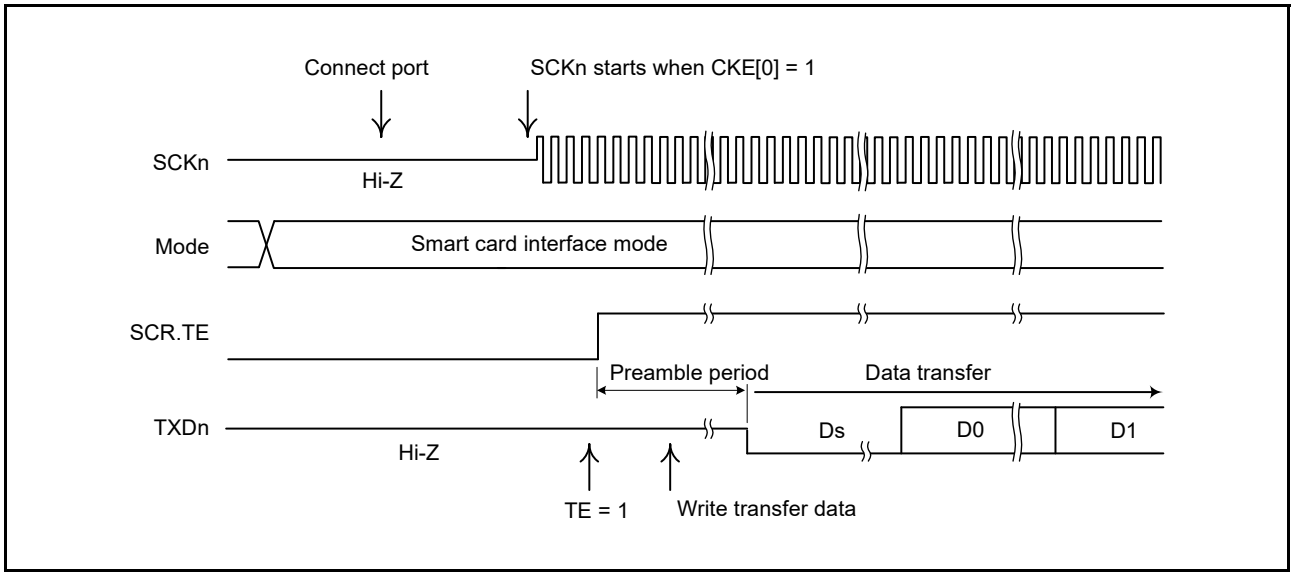


Figure 27.51 Example timing of data transmission in smart card interface mode

27.6.6 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 interface mode. Figure 27.52 shows the data retransfer operation during transmission.

- [1] indicates when an error signal from the receiver end is sampled after 1-frame data is transmitted, the SSR_SMCI.ERS flag is set to 1. If the SCR_SMCI.RIE bit is 1, an SCIn_ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.
- [2] indicates for a frame in which an error signal is received, the SSR_SMCI.TEND flag 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 SCR_SMCI.TIE bit is 1, an SCIn_TXI interrupt request is generated. Write transmit data to the TDR to start transmission of the next data.

Figure 27.54 shows an example flow of serial transmission. All the processing steps are automatically performed using an SCIn_TXI interrupt request to activate the DTC.

When the SSR_SMCI.TEND flag is set to 1 in transmission and when the SCR_SMCI.TIE bit is 1, an SCIn_TXI interrupt request is generated.

The DTC is activated by an SCIn_TXI interrupt request if the SCIn_TXI interrupt request is previously specified as a source of DTC activation, allowing the transfer of transmit data. The TEND flag is automatically set to 0 when the DTC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept at 0 and the DTC is not activated. Therefore, the SCI and DTC automatically transmit the specified number of bytes, including retransmission when an error occurs. Because the ERS flag is not automatically cleared, set the RIE bit to 1 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 DTC, always enable the DTC before setting the SCI.

For DTC settings, see section 16, Data Transfer Controller (DTC).

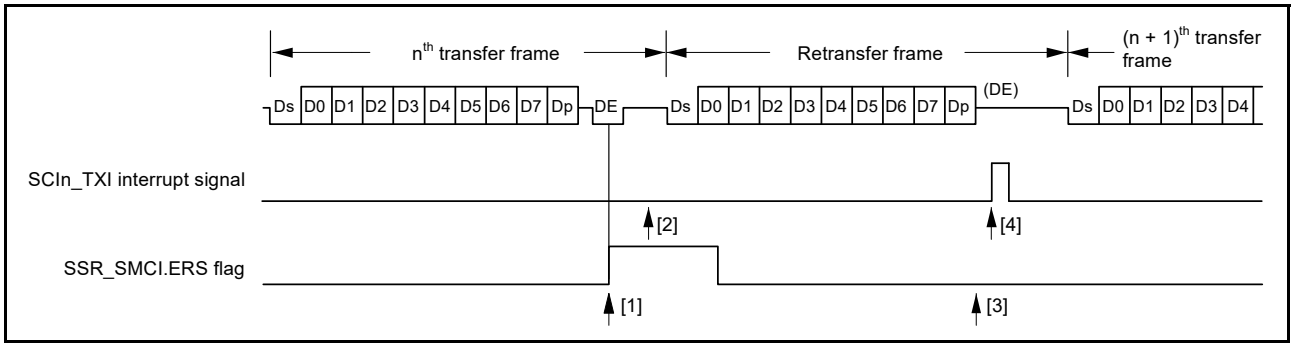


Figure 27.52 Data retransfer operation in smart card interface transmission mode

Note: The SSR_SMCI.TEND flag is set at different timings depending on the SMR_SMCI.GM bit setting.

Figure 27.53 shows the TEND flag generation timing.

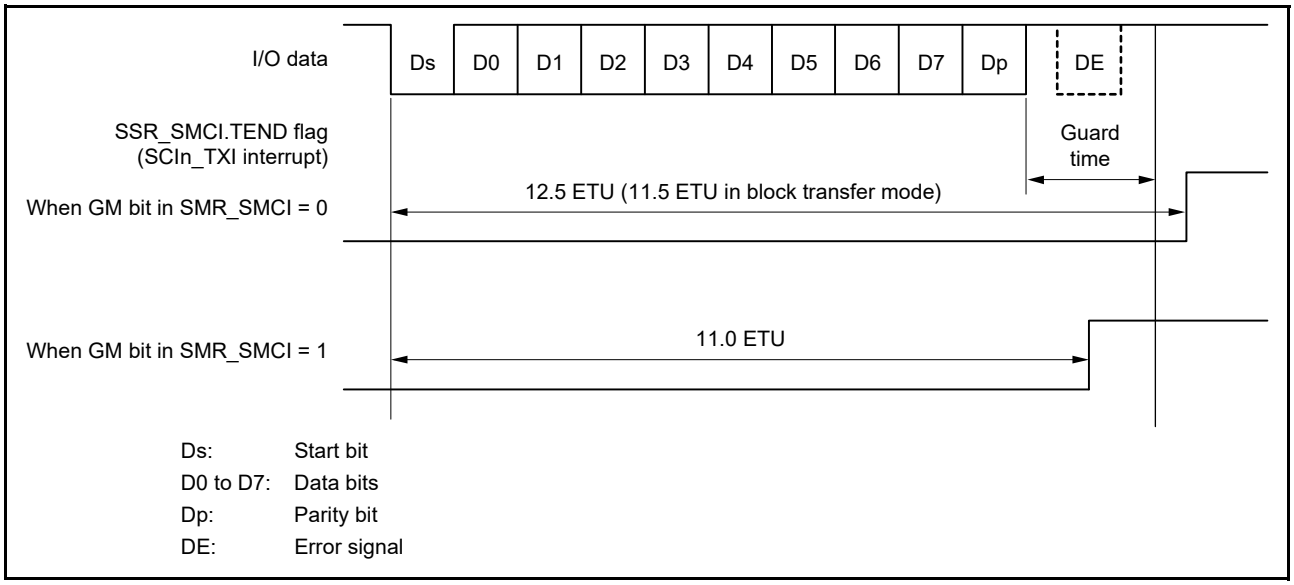


Figure 27.53 SSR.TEND flag generation timing during transmission

Figure 27.54 Example flow of smart card interface transmission

27.6.7 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 27.55 shows the data retransfer operation in reception mode.

- [1] indicates if a parity error is detected in the receive data, the SSR_SMCI.PER flag is set to 1. When the SCR_SMCI.RIE bit is 1, an SCIn_ERI interrupt request is generated. Clear the PER flag to 0 before the next parity bit is sampled.
- [2] 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 SSR_SMCI.PER flag is not set to 1.
- [4] indicates the data is determined to be received successfully. When the SCR_SMCI.RIE bit is 1, an SCIn_RXI interrupt request is generated.

Figure 27.56 shows an example flow of serial data reception. All the processing steps are automatically performed using an SCIn_RXI interrupt request to activate the DTC.

In reception, setting the RIE bit to 1 allows an SCIn_RXI interrupt request to be generated. The DTC is activated by an SCIn_RXI interrupt request if the SCIn_RXI interrupt request is previously specified as a source of DTC activation, allowing the transfer of receive data.

If an error occurs during reception and either the ORER or PER flag in SSR_SMCI is set to 1, a receive error interrupt (SCIn_ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DTC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC is transferred.

If a parity error occurs and the PER flag is set to 1 during reception, the receive data is transferred to RDR, therefore allowing the data to be read.

When a reception is forced to terminate by setting the SCR_SMCI.RE bit 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 27.3.9, Serial Data Reception in Asynchronous Mode.

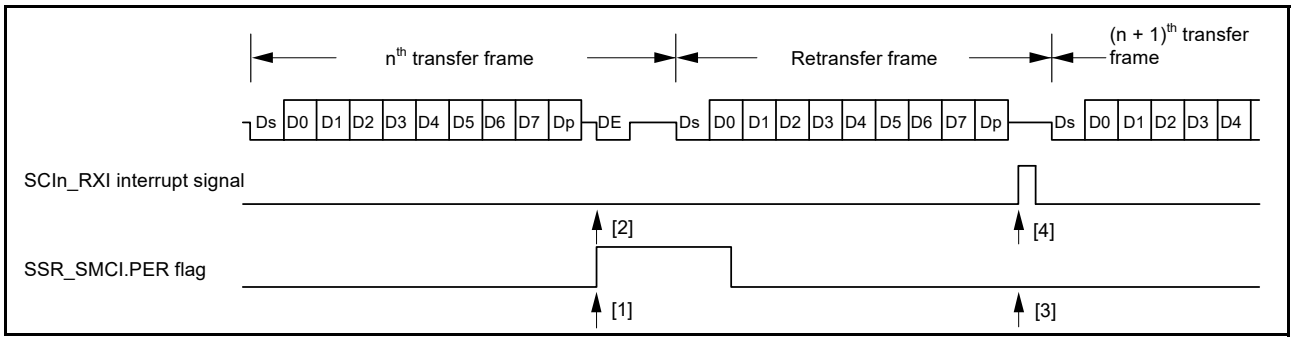


Figure 27.55 Data retransfer operation in smart card interface reception mode

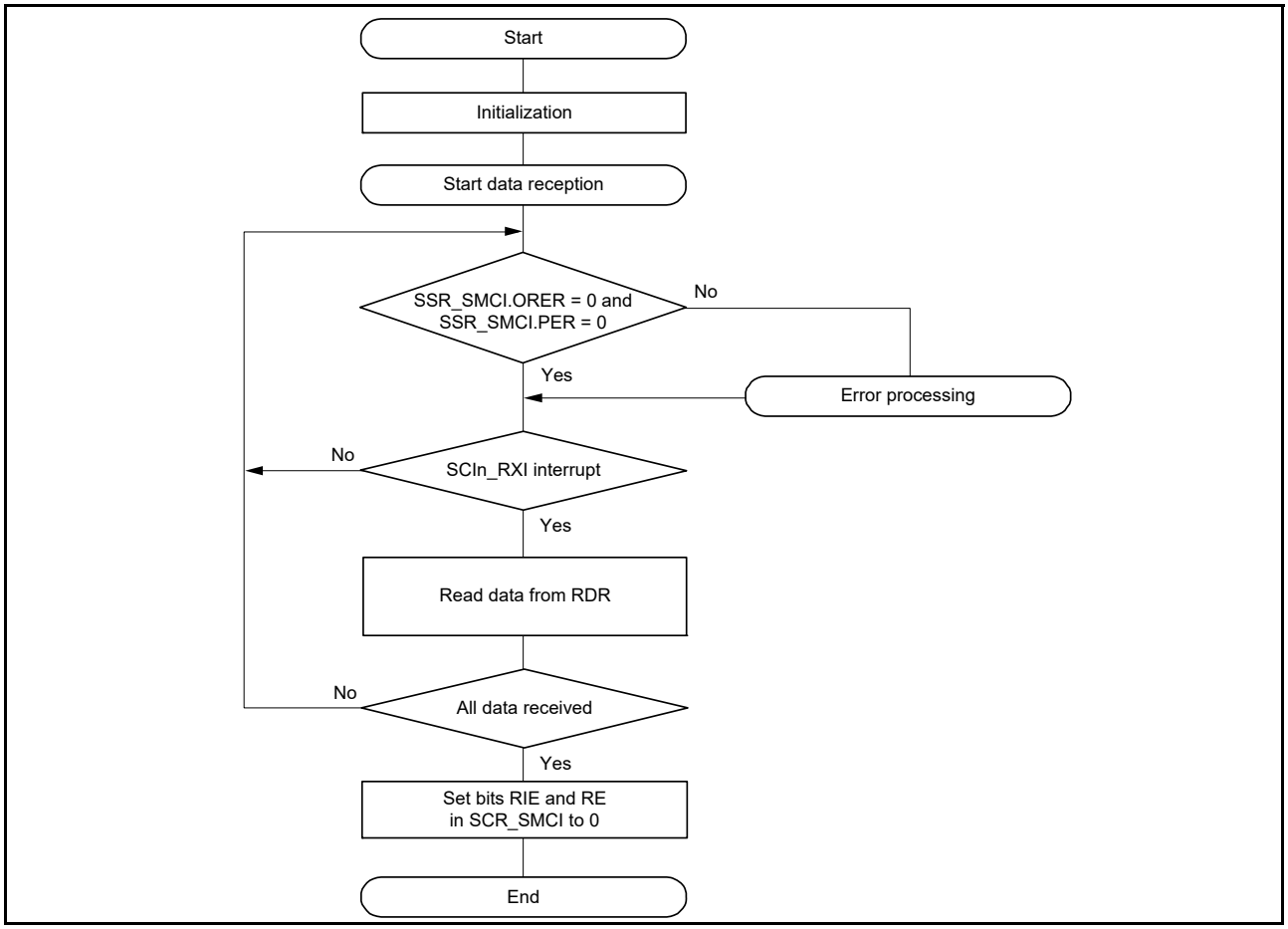


Figure 27.56 Example flow of smart card interface reception

27.6.8 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 27.2.12, Serial Control Register for Smart Card Interface Mode (SCR_SMCI) (SCMR.SMIF = 1). When setting the clock output, the base clock described in section 27.6.4, Receive Data Sampling Timing and Reception Margin is output.

Figure 27.57 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 in 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.

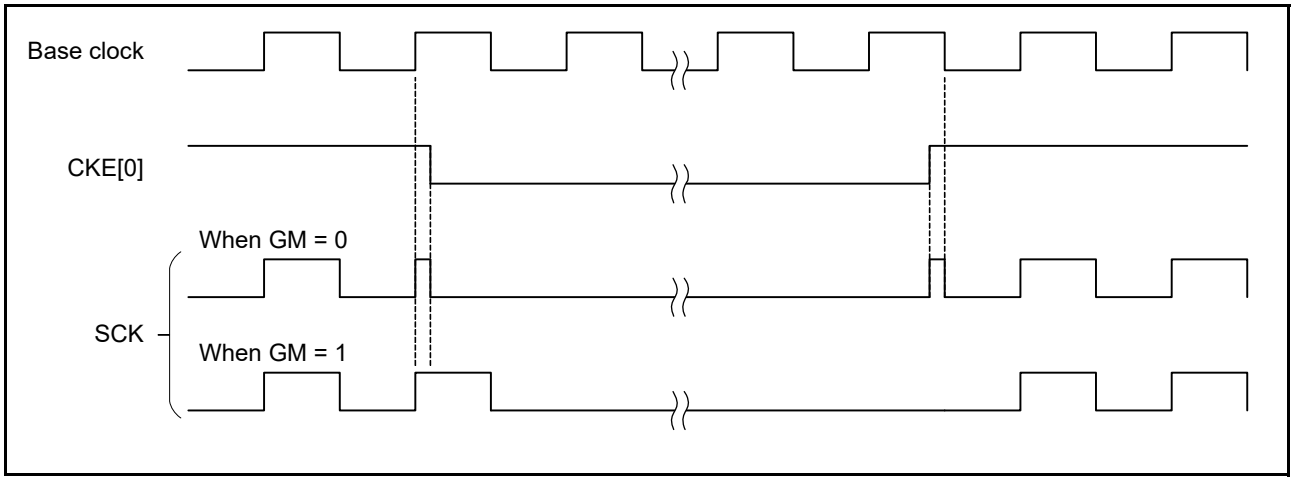


Figure 27.57 Clock output control

27.7 Operation in Simple IIC Mode

Simple I^{2}C 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 I^{2}C bus format and timing are shown in Figure 27.58 and Figure 27.59.

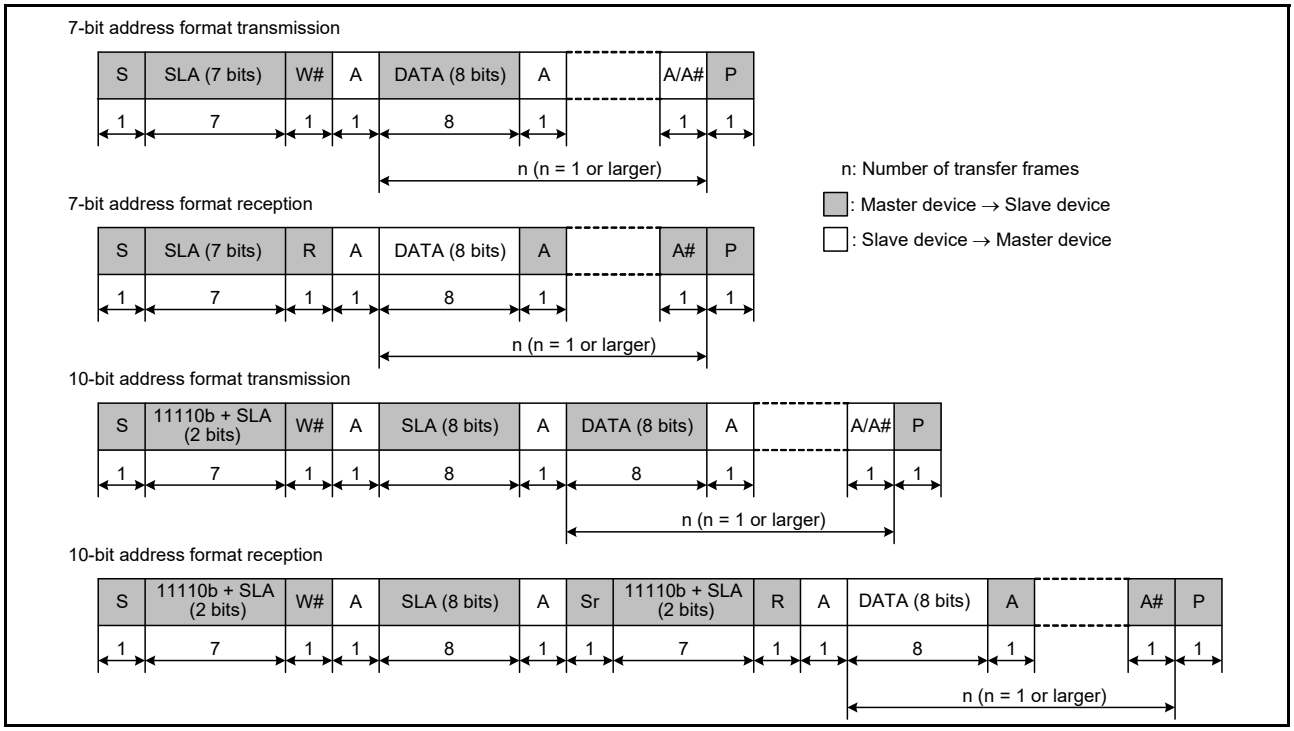


Figure 27.58 I²C bus format

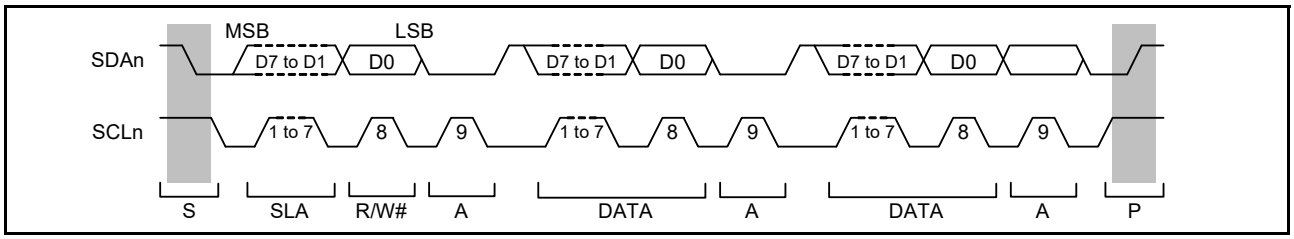


Figure 27.59 I²C bus timing when SLA is 7 bits

S: Indicates a start condition, when the master device changes the level on the SDAn line from high to low while the SCLn line is high.

SLA: Indicates a slave address, by which the master device selects a slave device.

R/W#: Indicates the direction of transfer (reception or transmission). The value 1 indicates transfer from the slave device to the master device and 0 indicates transfer from the master device to the slave device.

A/A#: Indicates an acknowledge bit. This is returned by the slave device for master transmission and by the master device for master reception. Return low indicates ACK and return high indicates NACK.

Sr: Indicates a restart condition, when the master device changes the level on the SDAn line from high to low while the SCLn line is high and after the setup time elapses.

DATA: Indicates the data being received or transmitted.

P: Indicates a stop condition, when the master device changes the level on the SDAn line from low to high while the SCLn line is high.

27.7.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the SIMR3.IICSTAREQ bit causes the generation of a start condition. The generation of a start condition proceeds through the following operations.

- The level on the SDAn line falls (from 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 in SIMR3 is set to 0, and a start-condition generated interrupt is output.

Writing 1 to the IICRSTAREQ bit in SIMR3 causes the generation of a restart condition. The generation of a restart condition proceeds through the following operations:

- The SDAn line is released and the SCLn line is kept at 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 level to high level)
- When a high level is detected on the SCLn line, the setup time for the restart condition is set as half of a bit period at the bit rate determined by the BRR setting
- The level on the SDAn line falls (from 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 a 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 level to high level)
- When a high level is detected on the SCLn line, the setup time for the stop condition is set as half of a bit period at the bit rate determined by the BRR setting
- The SDAn 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 27.60 shows the timing of operations in the generation of start, restart, and stop conditions.

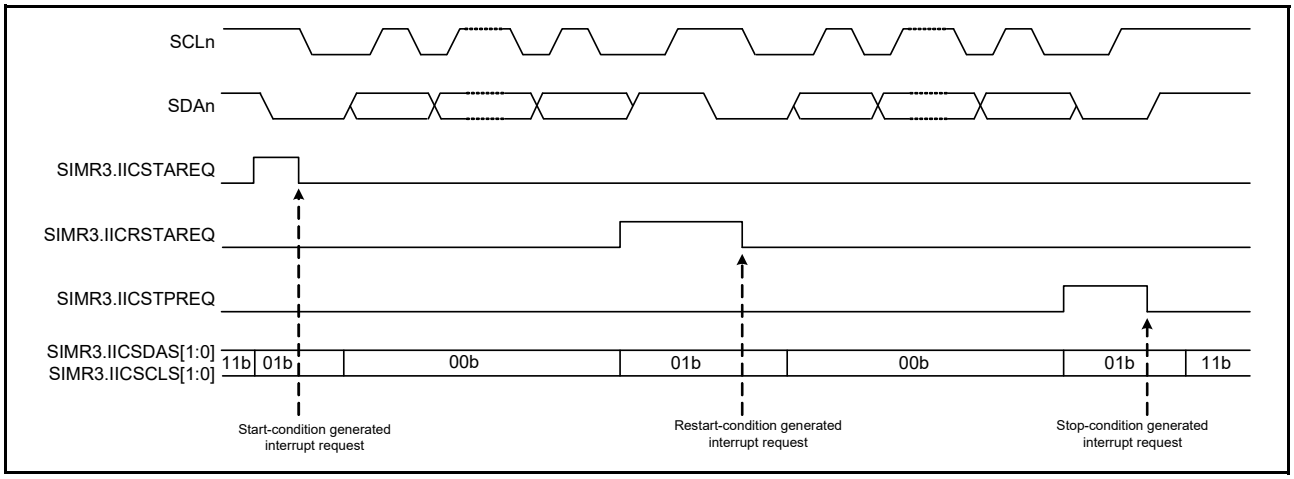


Figure 27.60 Timing of operations to generate start, restart, and stop conditions

27.7.2 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 SIMR2.IICCSC bit is set to 1, the level of the internal SCLn clock signal changes from low to high. Counting to determine the period at a high level stops while the low level is 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 PCLKB 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 SIMR2.IICCSC bit is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SCLn pin and the internal SCLn clock. If the SIMR2.IICCSC bit is 0, synchronization with the internal SCLn clock is obtained for the transmission and reception of data.

If a slave device inserts a wait period into the interval until the transition of the internal SCLn clock signal from the low to the high level after a request for the generation of a start, restart, or stop condition is issued, the time until generation is prolonged by that period.

If a slave device inserts a wait period after the transition of the internal SCLn clock signal from the low to the high level, although the generation-completed interrupt is issued without stopping the waiting period, generation of the condition itself is not guaranteed. Figure 27.61 shows an example operation to synchronize the clocks.

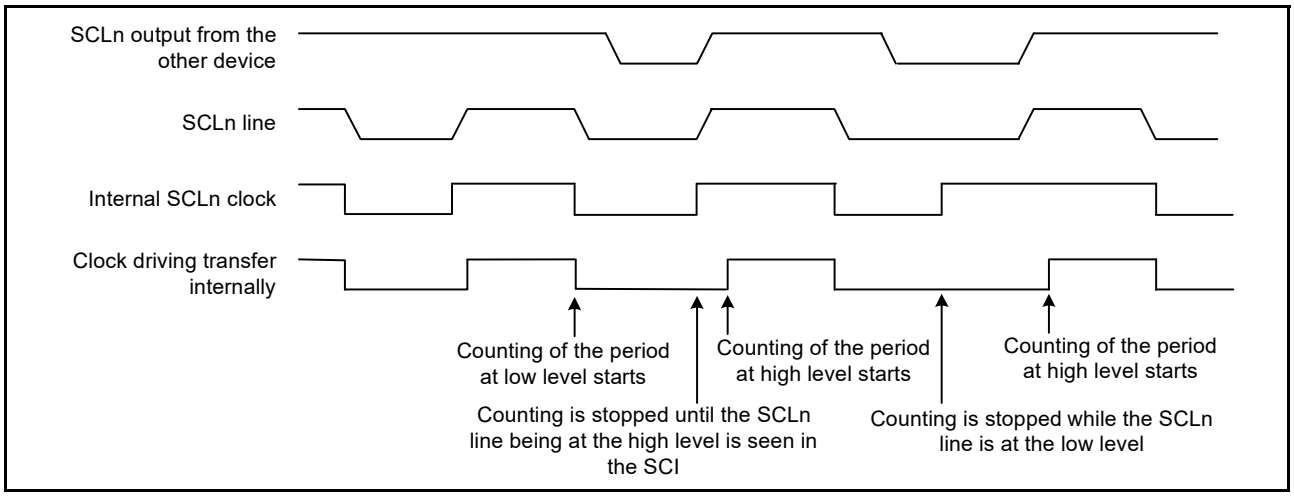


Figure 27.61 Example operation for clock synchronization

27.7.3 SDAn Output Delay

The SIMR1.IICDL[4:0] bits can be used to set a delay for the 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, PCLKB, by the divisor selected in the SMR.CKS[1:0] bits). A delay for output on the SDAn pin applies to the start condition/restart condition/stop condition signal, 8-bit transmit data, and 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 27.62 shows the timing of delays in SDAn output.

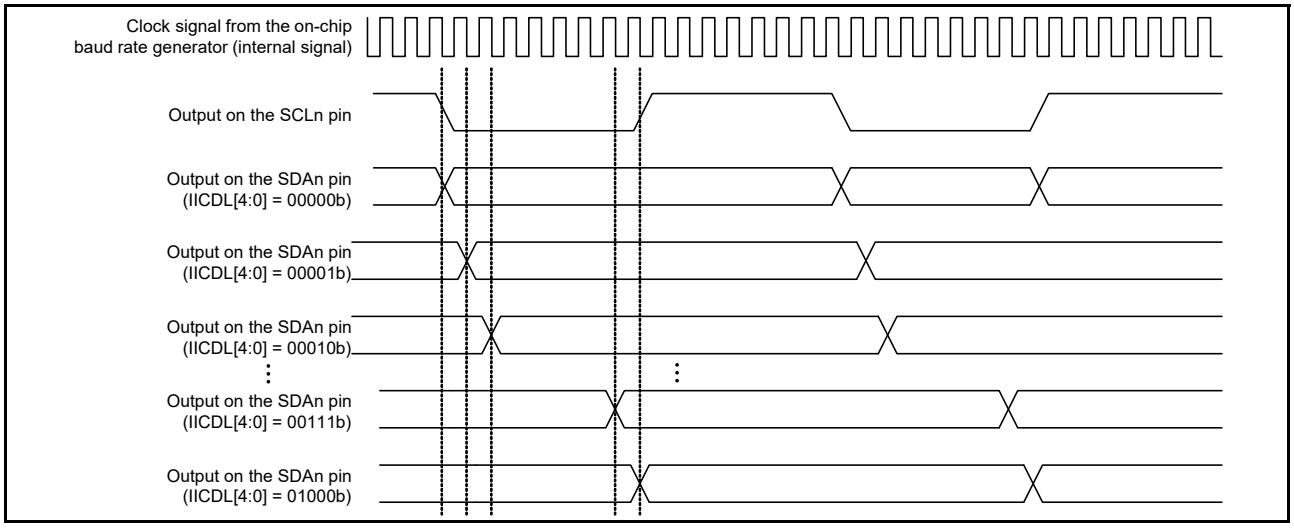


Figure 27.62 Timing of delays in SDAn output

27.7.4 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 shown in Figure 27.63. Before making any changes to the operating mode or transfer format, be sure to set SCR to its initial value. In simple IIC mode, the open-drain setting for the communication ports should be made on the port side.

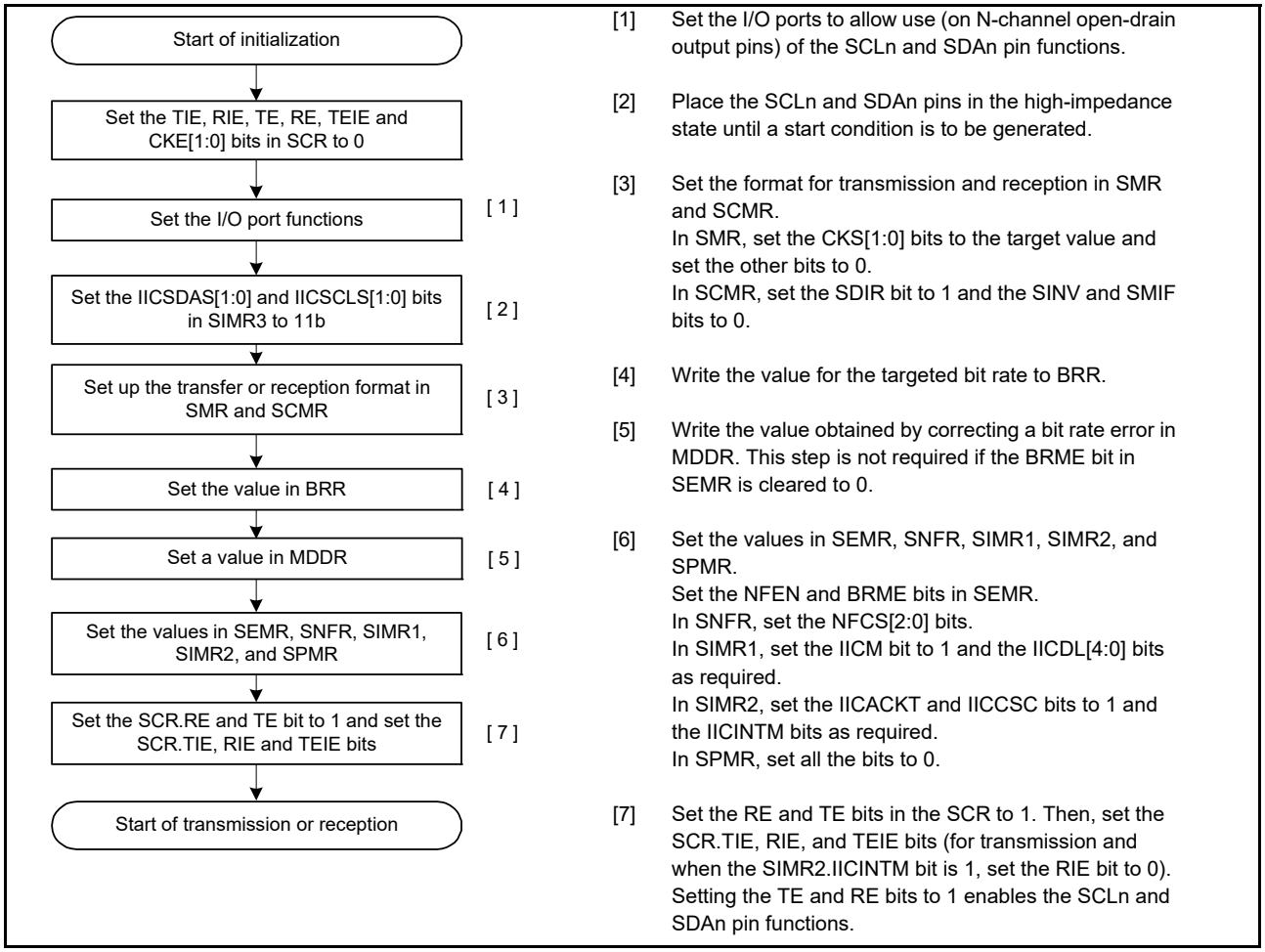


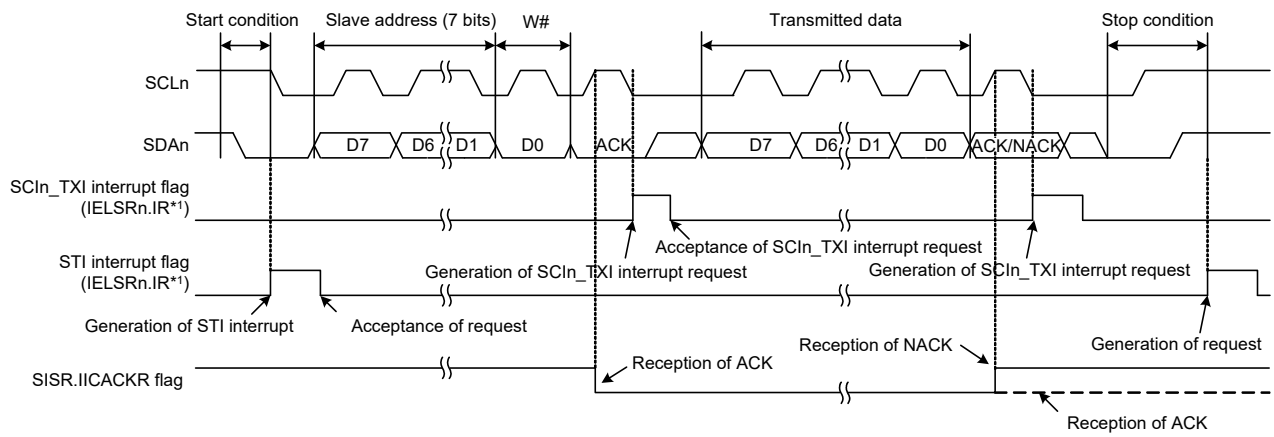
Figure 27.63 Example flow of SCI initialization in simple IIC mode

27.7.5 Operation in Master Transmission in Simple IIC Mode

Figure 27.64 and Figure 27.65 show examples of master transmission and Figure 27.66 shows an example flow of data transmission. The value of the SIMR2.IICINTM bit is assumed to be 1 (use reception and transmission interrupts) and the value of the SCR.RIE bit is assumed to be 0 (SCIn_RXI and SCIn_ERI interrupt requests are disabled). See Table 27.29 for more information on the STI interrupt.

When 10-bit slave addresses are in use, steps [3] and [4] in Figure 27.66 are repeated twice.

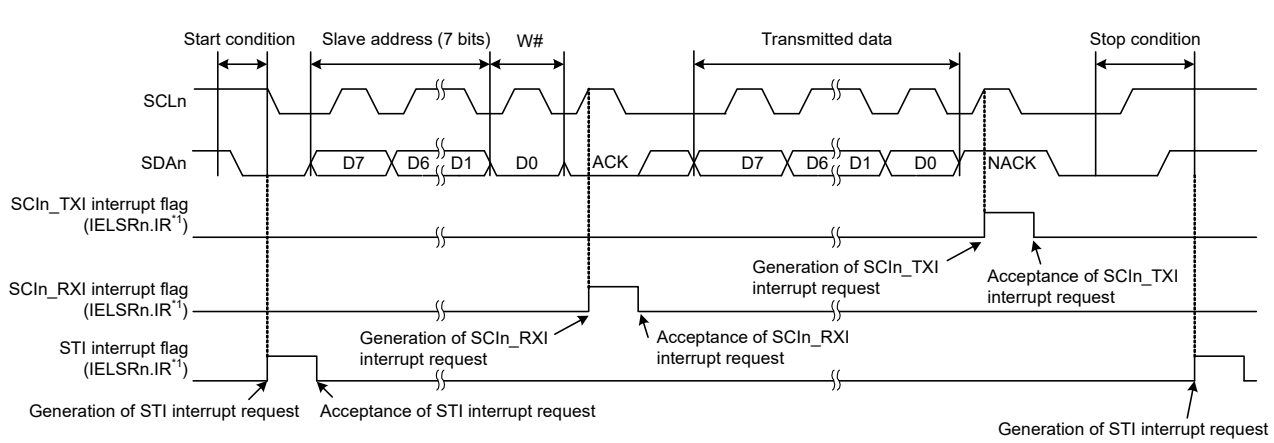
In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the timing of the SCIn_TXI interrupt request generation during clock synchronous transmission.



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

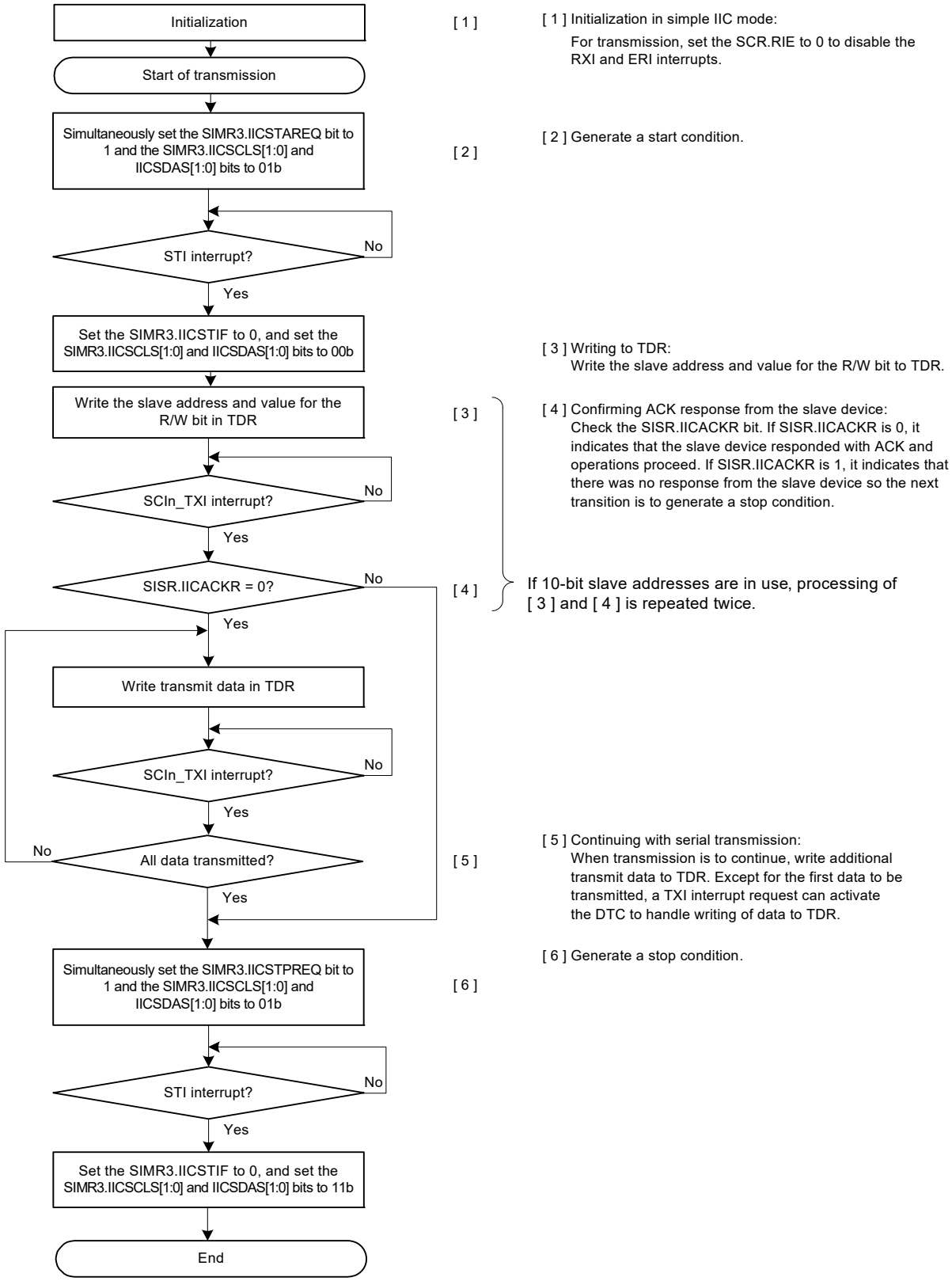
Figure 27.64 Example 1 operation for master transmission in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts

When the SIMR2.IICINTM bit is set to 0, using ACK/NACK interrupts during master transmission, the 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.



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 27.65 Example 2 operation for master transmission in simple IIC mode with 7-bit slave addresses, ACK interrupts, and NACK interrupts



Note: In simple IIC mode, the SCIn_TXI interrupt is generated when communication completes.

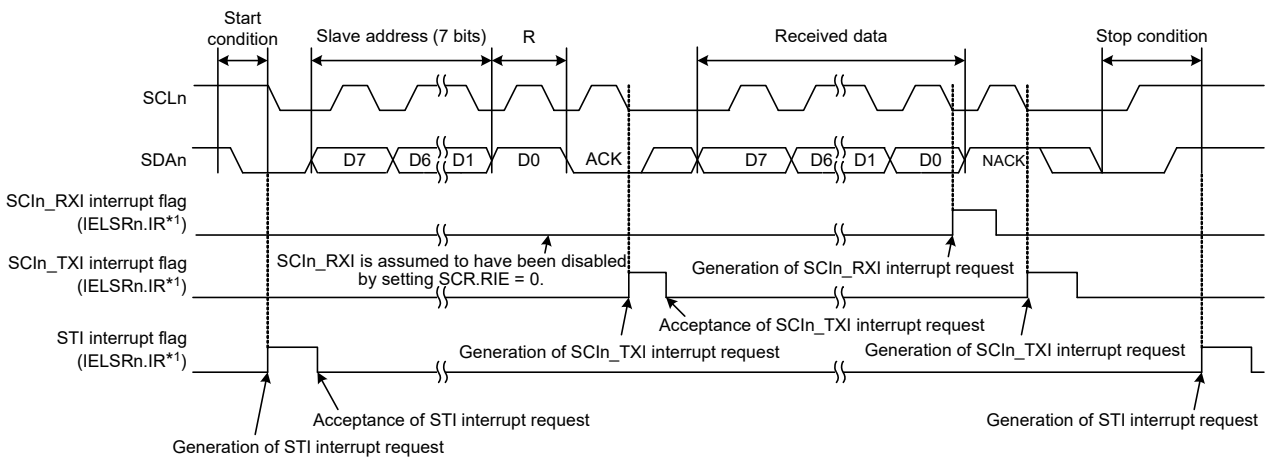
Figure 27.66 Example procedure for master transmission in simple IIC mode with transmission interrupts and reception interrupts

27.7.6 Master Reception in Simple IIC Mode

Figure 27.67 shows an example of master reception operation in simple IIC mode and Figure 27.68 shows an example flow of master reception.

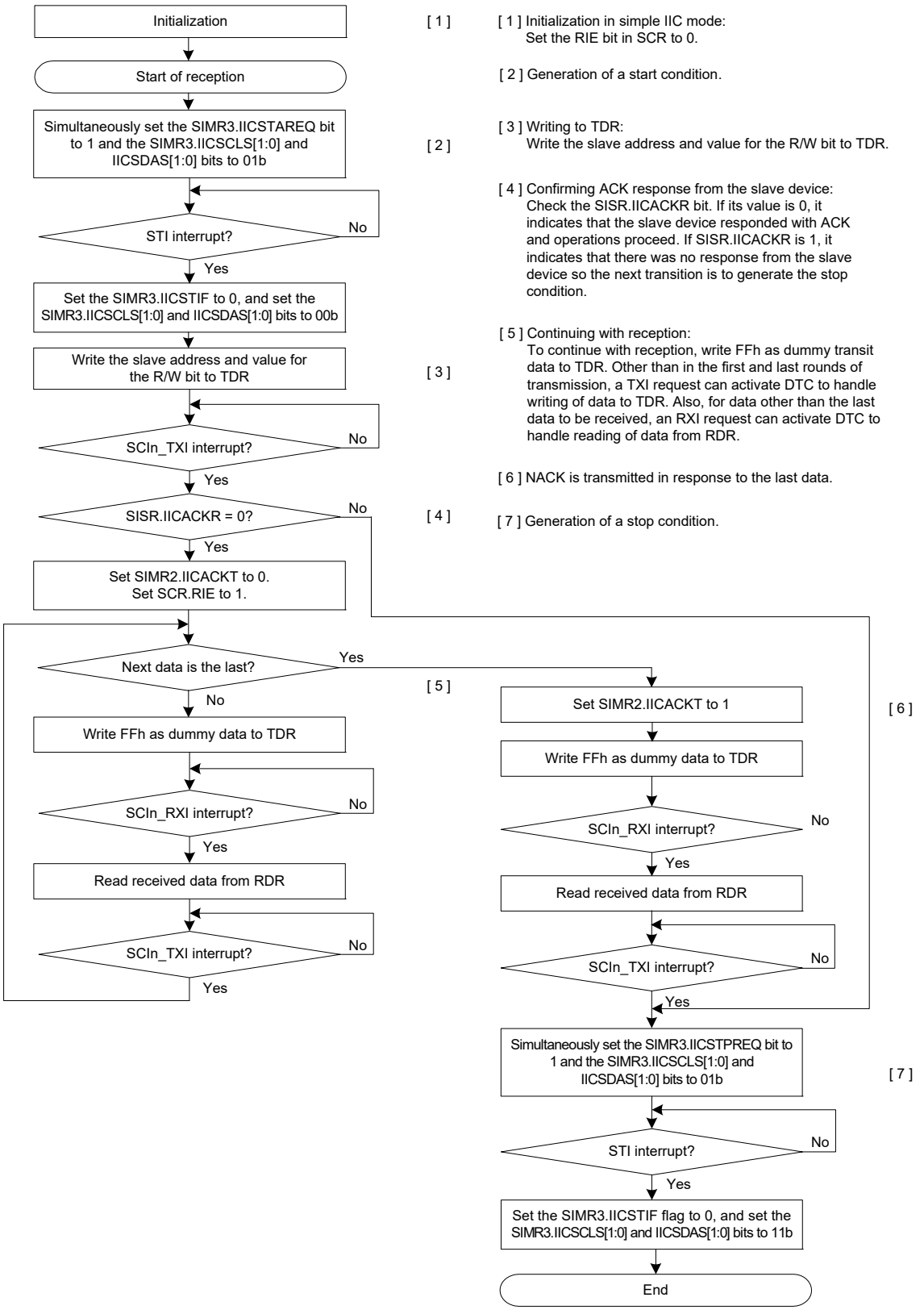
The value of the SIMR2.IICINTM bit is assumed to be 1 using reception and transmission interrupts.

In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the timing of the SCIn_TXI interrupt request generation during clock synchronous transmission.



Note 1. See section 13, Interrupt Controller Unit (ICU) for information on the associated interrupt event number.

Figure 27.67 Example operation for master reception in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts



Note: In simple IIC mode, the TXI interrupt request is generated when communication is complete.

Figure 27.68 Example flow of master reception in simple IIC mode with transmission interrupts and reception interrupts

27.8 Operation in Simple SPI Mode

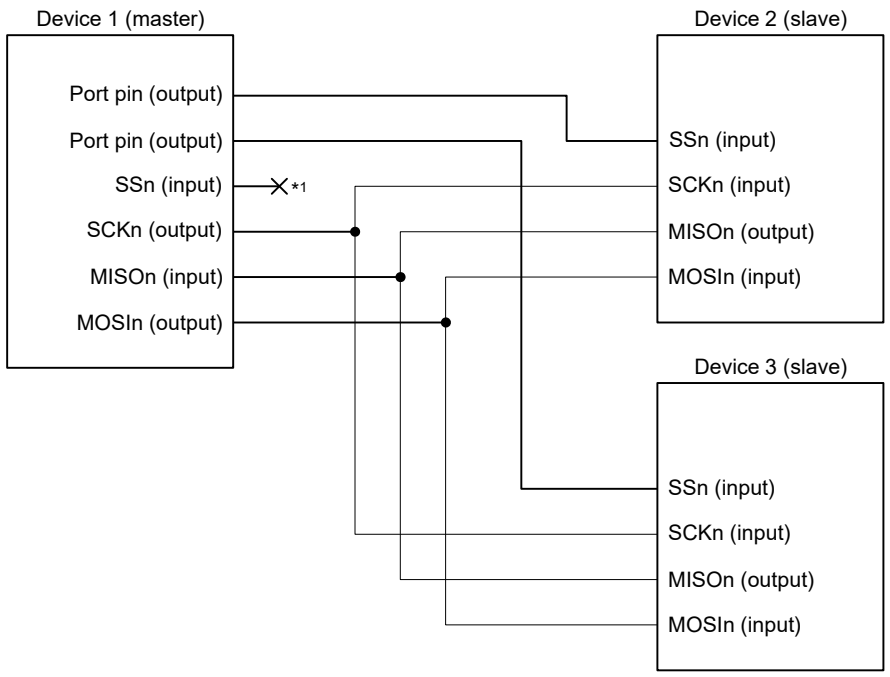
As an extended function, the SCI supports a simple SPI mode that handles transfer in one or multiple master devices and multiple slave devices.

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 SPMR.SSE bit to 1. When the configuration only has a single master, the SSn pin function on the master side is not required to connect the device used as the master in simple SPI mode. Therefore, set the SPMR.SSE bit to 0 in this case.

Figure 27.69 shows an example of connections for 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 SCMR.SINV bit to 1.

Because the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a common clock signal. Additionally, because both the transmitter and receiver have a buffered structure, it is possible to write the next transmit data while transmission is in progress and also 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 27.69 Example connections using simple SPI mode in single master mode with SPMR.SSE = 0

27.8.1 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 27.25 lists the relationship between the pin states, mode, and the input level on the SSn pin.

Table 27.25 Pin states by mode and input level on the 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 the 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 a high-impedance state when serial transmission is disabled (SCR.TE = 0).

Note 3. The SCKn pin output is in a high-impedance state when serial transmission is disabled (SCR.TE and RE = 00b) in a multi-master configuration (SPMR.SSE = 1).

27.8.2 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 a high-impedance state and starting transmission or reception is not possible. In addition, the value of the SPMR.MFF bit is 1, indicating a mode fault error. In a multi-master configuration, start error processing by reading the SPMR.MFF flag. If a mode fault error occurs while transmission or reception is in progress, transmission or reception does not stop, but the MOSIn and SCKn outputs are in the high-impedance state after completion of the transfer. Use a general port pin to produce the SS output signal from the master.

27.8.3 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 the 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, and SCIn_TEI) is then generated.

27.8.4 Relationship between Clock and Transmit/Receive Data

The CKPOL and CKPH bits in the 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 27.70. The relation is the same for both master and slave operation. This is the same as when the level on the SSn pin is high.

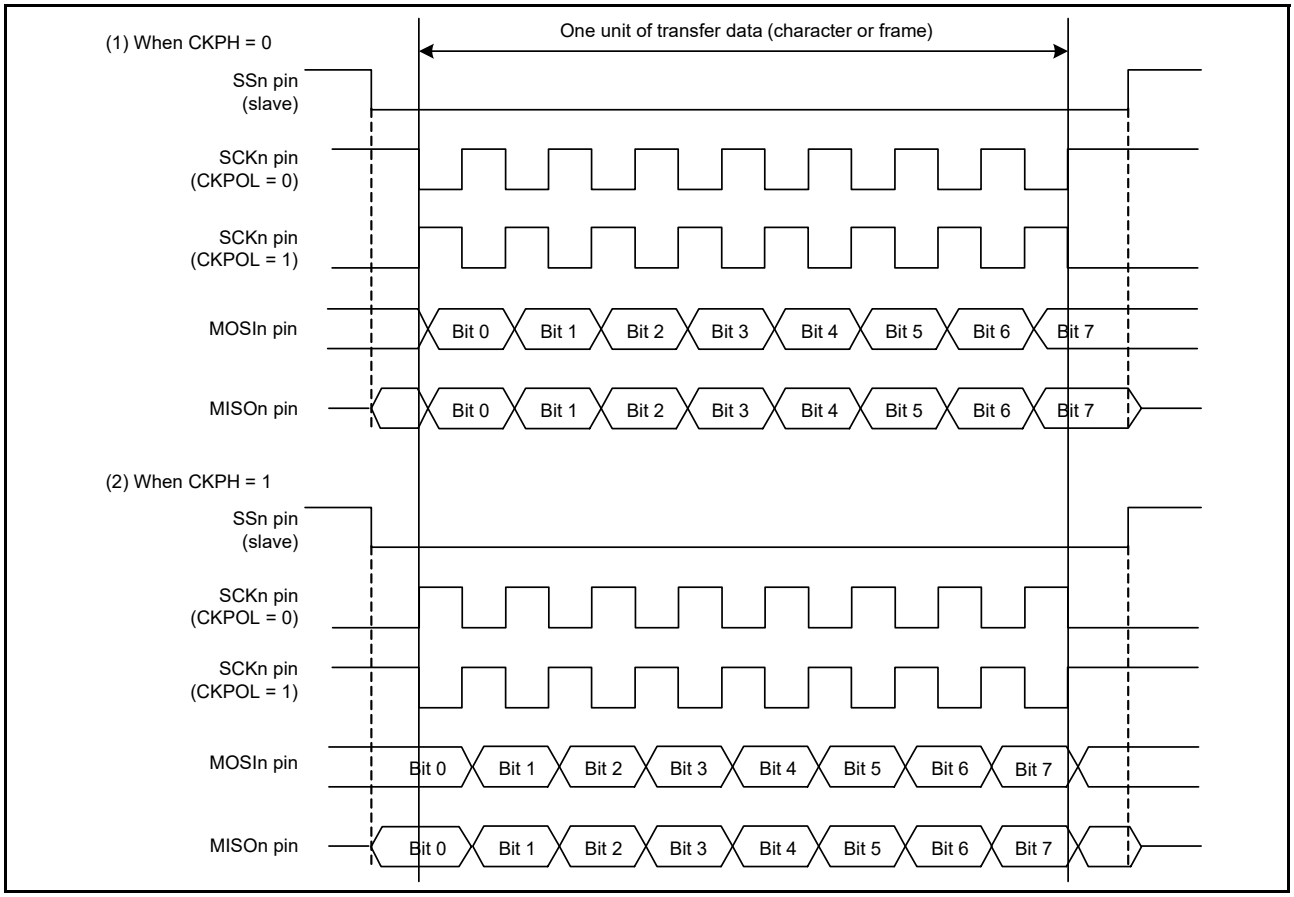


Figure 27.70 Relation between clock signal and transmit or receive data in simple SPI mode

27.8.5 SCI Initialization in Simple SPI Mode

Initialization in simple SPI mode is the same as in clock synchronous mode. See Figure 27.32 for an example initialization flow. The CKPOL and CKPH bits in the 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 RE bit is set to 0. The SSR.ORER, FER, PER, and RDR flags are not initialized.

Changing the value of the TE bit from 1 to 0 or from 0 to 1 when the TIE bit in the SCR register is 1 at the same time, leads to the generation of a transmit data empty interrupt (SCIn_TXI).

27.8.6 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.

27.9 Bit Rate Modulation Function

Using the bit rate modulation function, the bit rate can be evenly corrected using the number specified in the MDDR register when PCLKB is selected with the CKS[1:0] bits in SMR/SMR_SMCI.

Figure 27.71 shows an example where PCLKB is selected in the CKS[1:0] bits in SMR/SMR_SMCI, the BRR bit is set to 0, and the MDDR is set to 160 in asynchronous mode. In this example, the cycle of the base clock is evenly corrected (256/160) and the bit rate is also corrected (160/256).

Note: Enabling an internal clock causes bias and expansion. Contraction is generated in the pulse width of the internal base clock.

Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).

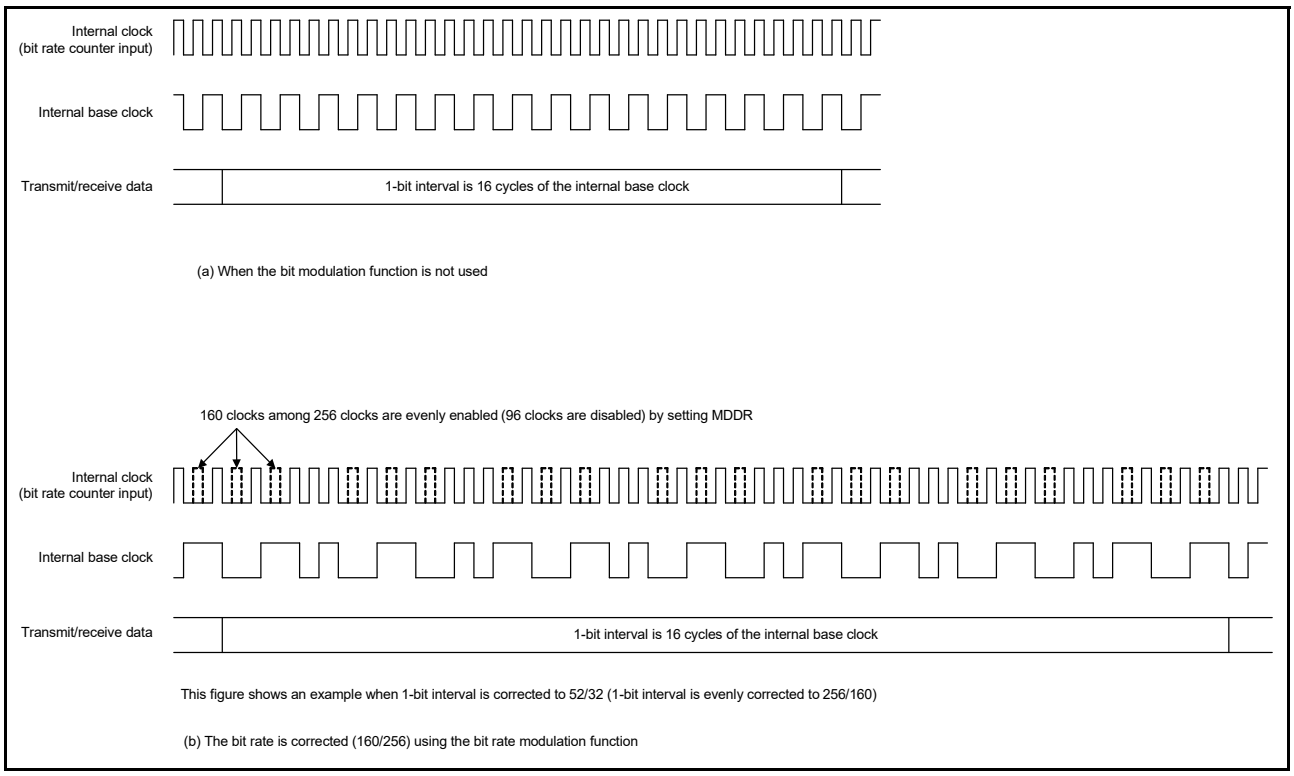


Figure 27.71 Example of internal base clock using bit rate modulation function

27.10 Interrupt Sources

27.10.1 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 saving one request per source.

When 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.

27.10.2 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 SCIn_TXI and SCIn_RXI interrupts are satisfied, an interrupt request is generated.

27.10.3 Interrupts in Asynchronous, Clock Synchronous, and Simple SPI Modes

(1) Non-FIFO selected

Table 27.26 lists interrupt sources in asynchronous, clock synchronous, and simple SPI modes. A different interrupt vector can be assigned to each interrupt source. Individual interrupt sources can be enabled or disabled with the enable bits in the SCR register.

If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated when transmit data is transferred from the TDR or TDRHL register*1 to the TSR register. An SCIn_TXI interrupt request can also be generated using a single instruction to set the SCR.TE and SCR.TIE bits to 1 simultaneously. An SCIn_TXI interrupt request can activate the DTC to handle data transfer.

An SCIn_TXI interrupt request is not generated by setting SCR.TE to 1 when SCR.TIE is 0 or by setting SCR.TIE to 1 when SCR.TE is 1.*2

When new data is not written by the time of transmission of the last bit of the current transmit data and SCR.TEIE is 1, the SSR.TEND flag becomes 1 and an SCIn_TEI interrupt request is generated. Additionally, when SCR.TE is 1, the SSR.TEND flag saves the value 1 until additional transmit data are written to the TDR or TDRHL register*1, and setting the SCR.TEIE bit to 1 leads to the generation of an SCIn_TEI interrupt request.

Writing data to the TDR or TDRHL register*1 leads to clearing of the SSR.TEND flag and, after a certain time, discarding of the SCIn_TEI interrupt request.

If the SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated when received data is stored in the RDR. An SCIn_RXI interrupt request can activate the DTC to handle data transfer.

Setting any of the ORER, FER, and PER flags in the SSR to 1 when the SCR.RIE bit is 1 leads to the generation of an SCIn_ERI interrupt request. An SCIn_RXI interrupt request is not generated at this time. Clearing all three flags (ORER, FER, and PER) leads to discarding of the SCIn_ERI interrupt request.

(2) FIFO selected

Table 27.27 lists the interrupt sources in FIFO selected mode.

If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated when the stored number of data in FTDRL register becomes the threshold value indicated in FCR.TTRG[3:0] or below. An SCIn_TXI interrupt request can also be generated using a single instruction to set the SCR.TIE and SCR.TE bits to 1 simultaneously or by setting SCR.TIE to 1 when SCR.TE is 1.

An SCIn_TXI interrupt request is not generated by setting SCR.TE to 1 while SCR.TIE is 0.

If SCR.TEIE bit is 1 and if the next data is not written to the FTDRL register by the time the last bit of the transmit data is sent, the SSR_FIFO.TEND flag is set to 1 and the SCIn_TEI interrupt request is generated.

If SCR.RIE bit is 1, the SCIn_RXI interrupt request is generated when the amount of stored data in the FRDRL register is equal to or greater than the threshold value indicated in FCR.RTRG[3:0] bits. When RTRG[3:0] bits are set to 0000b, an SCIn_RXI interrupt does not occur even when the amount of data in the receive FIFO is equal to 0.

If the SCR.RIE bit is 1, when the SSR_FIFO.ORER flag is set to 1 or data with a framing error or a parity error is stored in the FRDRL register, the SCIn_ERI interrupt request is generated. When the amount of data stored in the FRDRL register is at the threshold value or above, the SCIn_RXI interrupt request is also generated. The SCIn_ERI interrupt request can be canceled in which case SSR_FIFO.ORER, FER, and PER flags are all cleared.

Note 1. When asynchronous mode and 9-bit data length are selected.
Note 2. To temporarily prohibit SCIn_TXI interrupts on transmission of the last of the data when a new round of transmission is to be started, after handling the transmission-completed interrupt, control activation of the interrupt by using the Interrupt Request Enable bit in the ICU rather than using the SCR.TIE bit. This approach can prevent the suppression of SCIn_TXI interrupt requests in the transfer of new data.

Table 27.26 SCI interrupt sources with non-FIFO selected

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation
SCIn_ERI	Receive error*1	ORER, FER, PER, DFER, DPER	RIE	Not possible
SCIn_RXI	Receive data full	RDRF	RIE	Possible
	Address match	DCMF	RIE	Possible
SCIn_AM	Address match	DCMF	-	Possible
SCIn_TXI	Transmit data empty	TDRE	TIE	Possible
SCIn_TEI	Transmit end	TEND	TEIE	Not possible

Note 1. The interrupt flag is only ORER when in clock synchronous mode and simple SPI mode.

Table 27.27 SCI interrupt sources with FIFO selected

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation
SCIn_ERI	Receive error*1	ORER, FER, PER, DFER, DPER	RIE	Not possible
		DR (when FCR.DRES = 1)	RIE	Not possible
SCIn_RXI	Receive data full	RDF	RIE	Possible
	Receive data ready	DR (when FCR.DRES = 0)	RIE	Possible
	Address match	DCMF	RIE	Possible
SCIn_AM	Address match	DCMF	—	Possible
SCIn_TXI	Transmit data empty	TDFE	TIE	Possible
SCIn_TEI	Transmit end	TEND	TEIE	Not possible

Note 1. The interrupt flag is only ORER when in clock synchronous mode and simple SPI mode.

27.10.4 Interrupts in Smart Card Interface Mode

Table 27.28 lists the 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 27.28 SCI interrupt sources in smart card interface mode

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation
SCIn_ERI	Receive error or error signal detection	ORER, FER, ERS	RIE	Not possible
SCIn_RXI	Receive data full	RDRF	RIE	Possible
SCIn_TXI	Transmit end	TEND	TIE	Possible

Data transmission or reception using the DTC is also possible in smart card interface mode. In transmission, when the SSR_SMCI.TEND flag is set to 1, an SCIn_TXI interrupt request is generated. The SCIn_TXI interrupt request activates the DTC allowing the transfer of transmit data if the SCIn_TXI request is previously specified as a source of DTC activation. The TEND flag is automatically set to 0 when the DTC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept at 0 and the DTC is not activated. Therefore, the SCI and DTC automatically transmit the specified number of bytes, including retransmission when errors occur. However, the SSR_SMCI.ERS flag is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the SCR_SMCI.RIE bit to 1 to enable an SCIn_ERI interrupt request to be generated at error occurrence.

When transmitting or receiving data using the DTC, always enable the DTC before setting the SCI. For DTC settings, see section 16, Data Transfer Controller (DTC).

In reception, an SCIn_RXI interrupt request is generated when receive data is set to RDR. The SCIn_RXI interrupt request activates the DTC allowing the transfer of receive data if the SCIn_RXI request is previously specified as a source of DTC activation. If an error occurs, the error flag is set. Therefore, the DTC is not activated and an SCIn_ERI interrupt request is issued to the CPU instead. The error flag must be cleared.

27.10.5 Interrupts in Simple IIC Mode

Table 27.29 lists the interrupt sources in simple IIC mode. The STI interrupt is allocated to the transmit end interrupt (SCIn_TEI) request. The receive error interrupt (SCIn_ERI) and the address match (SCIn_AM) request cannot be used.

The DTC can also be used to handle transfer in simple IIC mode.

When the SIMR2.IICINTM bit is 1:

- An SCIn_RXI request is generated on the falling edge of the SCLn signal for the 8th bit. If SCIn_RXI is previously set up as an activation source for the DTC, the SCIn_RXI request activates the DTC to handle transfer of the received data.
- An SCIn_TXI request is generated on the falling edge of the SCLn signal for the 9th bit (acknowledge bit). If SCIn_TXI is previously set up as an activation source for the DTC, the SCIn_TXI request activates the DTC to

handle transfer of the transmit data.

When the SIMR2.IICINTM bit is 0:

- An SCIn_RXI request (ACK detection) is generated if the input on the SDAn pin is low on the rising edge of the SCLn signal for the 9th bit (acknowledge bit)
- An SCIn_TXI request (NACK detection) is generated if the input on the SDAn pin is high on the rising edge of the SCLn signal for the 9th bit (acknowledge bit)
- If the SCIn_RXI is previously set up as an activation source for the DTC, the SCIn_RXI request activates the DTC to handle transfer of the received data.

If the DTC is used for data transfer in reception or transmission, always set up and enable the DTC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in SIMR3 are used to generate a start condition, restart condition, or stop condition, the STI interrupt is issued when generation is complete.

Table 27.29 SCI interrupt sources in simple IIC mode

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation
SCIn_RXI	Reception, ACK detection	-	RIE	Possible
SCIn_TXI	Transmission, NACK detection	-	TIE	Possible
STIn	Completion of generation of a start, restart, or stop condition	IICSTIF	TEIE	Not possible

Note: Activation of the DTC is only possible when the SIMR2.IICINTM bit is 1 (use reception and transmission interrupts).

27.11 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 the 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 the FCR.DRES bit is 1.

(2) Receive data full event output

- Indicates that ACK is detected if the SIMR2.IICINTM bit is 0 in simple IIC mode
- Indicates that the 8th bit SCLn falling edge is detected if the SIMR2.IICINTM bit is 1 in simple IIC mode
- When the SIMR2.IICINTM bit is 1 during master transmission in simple IIC mode, set the ELC so that receive data full events are not used.

(a) Non-FIFO selected

- Indicates that received data is set 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 SIMR2.IICINTM bit is 0 in simple IIC mode
- Indicates that the 9th bit SCLn falling edge is detected if the SIMR2.IICINTM bit is 1 in simple IIC mode.

(a) Non-FIFO selected

- Indicates that transmit data is transferred from the Transmit Data Register (TDR or TDRHL) to the Transmit Shift Register (TSR).

(b) FIFO selected

- Using this event output is prohibited.

(4) Transmit end event output

- Indicates the completion of transmission
- Indicates that the starting condition, resumption condition, or termination condition is generated in simple IIC mode

Note: When FIFO is selected, using this event output is prohibited.

(5) Address match event output

- Indicates a match of the comparison data (CDR.CMPD) with one frame of receive data when DCCR.DCME is set to 1 in asynchronous mode, including multi-processor mode.

27.12 Address Mismatch Event Output (SCI0_DCUF)

The address mismatch event output, SCI0_DCUF, indicates a mismatch of the comparison data (CDR.CMPD) with one frame of receive data when DCCR.DCME is set to 1 in asynchronous mode, including multi-processor mode. This event can be used for snooze end request only.

27.13 Noise Cancellation Function

Figure 27.72 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 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 and 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 to the RXDn pin. The receive level of the RXDn is sampled from the flip-flop circuit of the noise filter on the base clock of the asynchronous mode.

In simple IIC mode, the noise cancellation function can be used for each input on SDAn and SCLn. The sampling clock for the noise cancellation function is selected in the SNFR.NFCS bit by dividing the baud rate generator source clock by 1, 2, 4, or 8.

If the base clock is stopped once with the noise filter enabled and then the base clock input is restarted again, the noise filter operation resumes from the state where the clock was stopped. When SCR.TE and SCR.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, the function determines that a level match is detected and the result is 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.

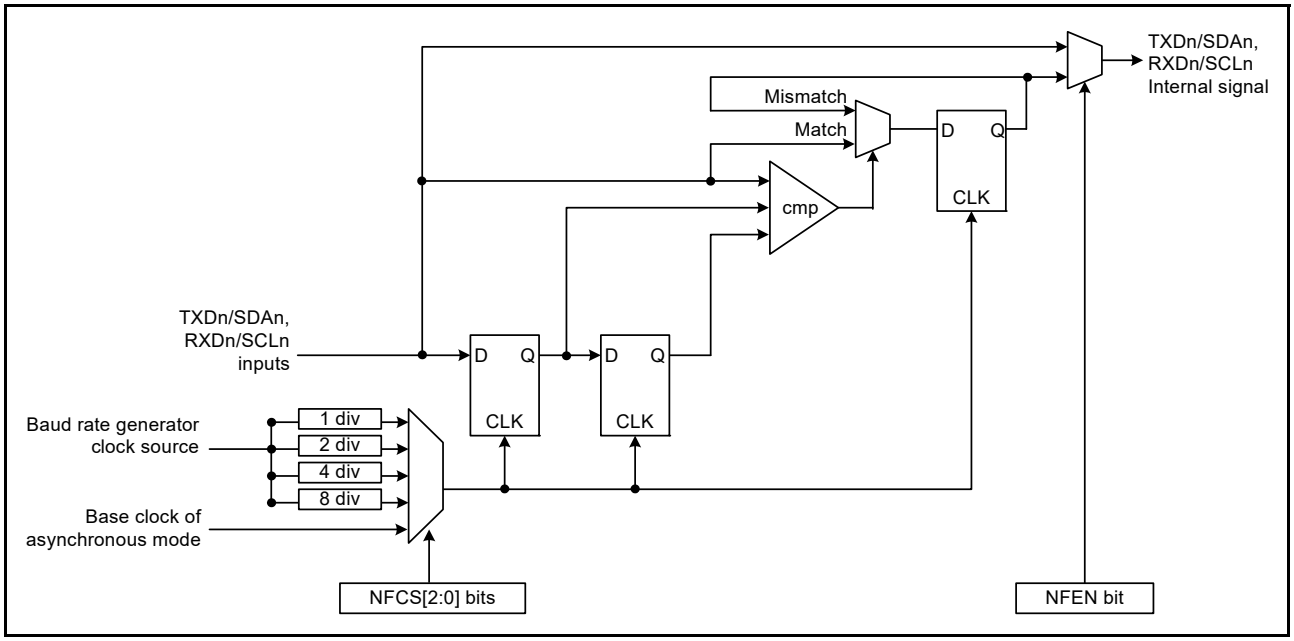


Figure 27.72 Digital noise filter circuit block diagram

27.14 Usage Notes

27.14.1 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.

27.14.2 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 the TSR register. 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 27.73 shows an example flow of transition to Software Standby mode during transmission. Figure 27.74 and Figure 27.75 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 a wakeup condition

Before specifying the module-stop state or making a transition to Software Standby mode, stop the receive operations (RE = 0 in SCR/SCR_SMCI). If transition is made during data reception, the received data is invalid.

Figure 27.76 shows an example flow for transition to Software Standby mode during reception.

(b) When address match function is used as a wake-up condition

Before specifying the module-stop state or transitioning to Software Standby mode:

1. Set the operations after cancellation of the low power state.
2. Set CDR.CMPD and DCCR.DCME to 1.
3. Set the receive operations (RE = 1 in SCR/SCR_SMCI).
4. Set the module-stop state or Software Standby mode.

When the SCI transfers to the low power mode, if the receive data pin (RXDn) is at the low level, set SEMR.RXDESEL to 0. If SEMR.RXDESEL is set to 1, there is a possibility that a start bit (falling edge of RXDn pin) cannot be detected on release of the low power mode.

Figure 27.77 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.

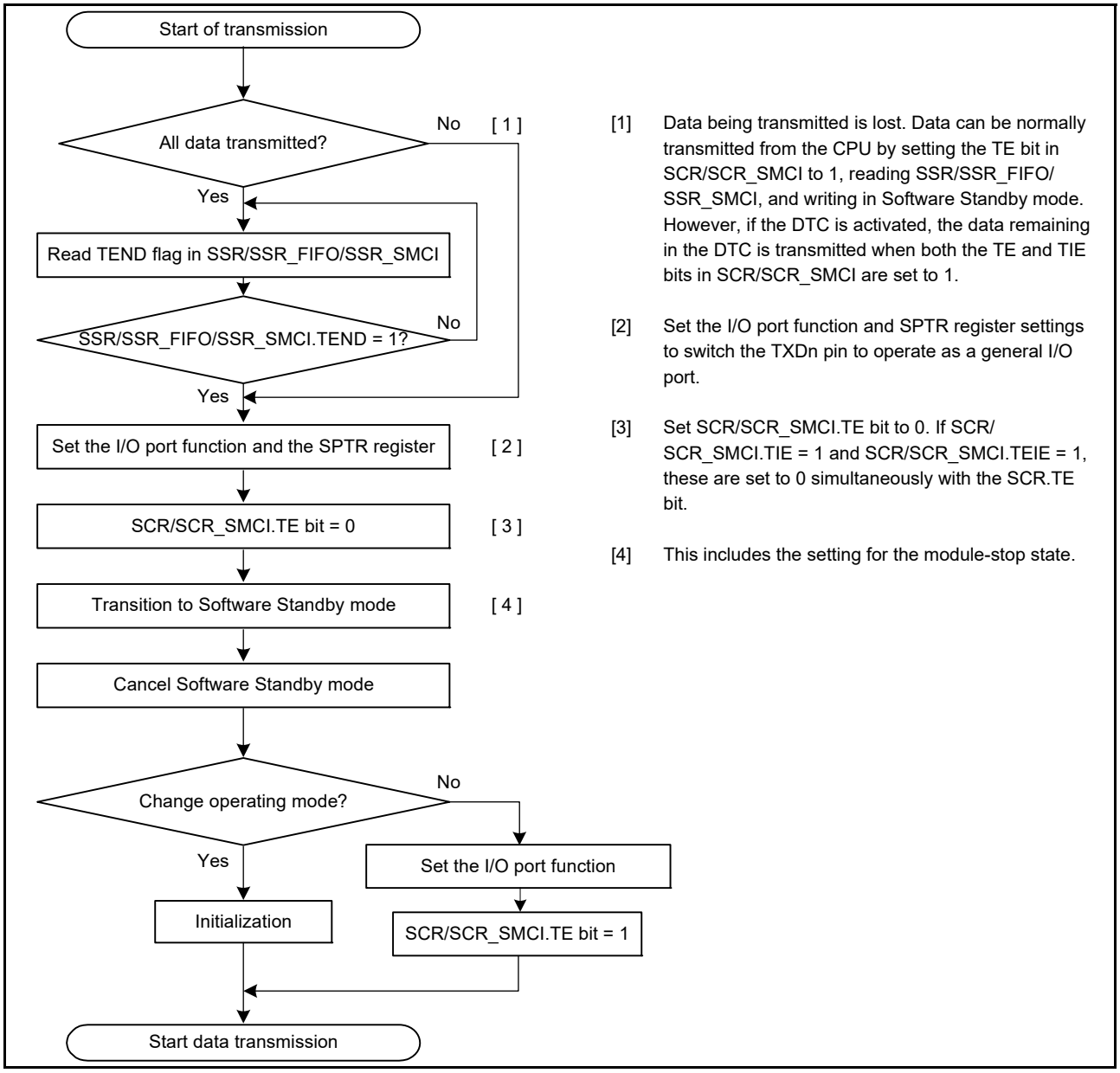


Figure 27.73 Example flow of transition to Software Standby mode during transmission

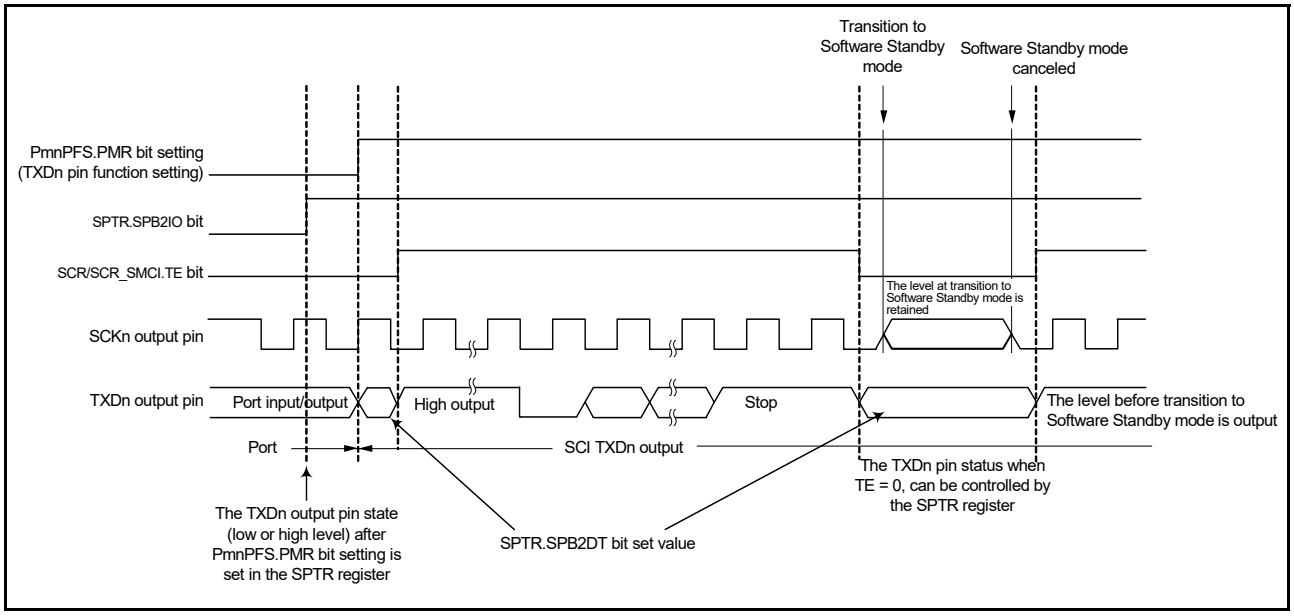


Figure 27.74 Port pin states during transition to Software Standby mode with internal clock and asynchronous transmission

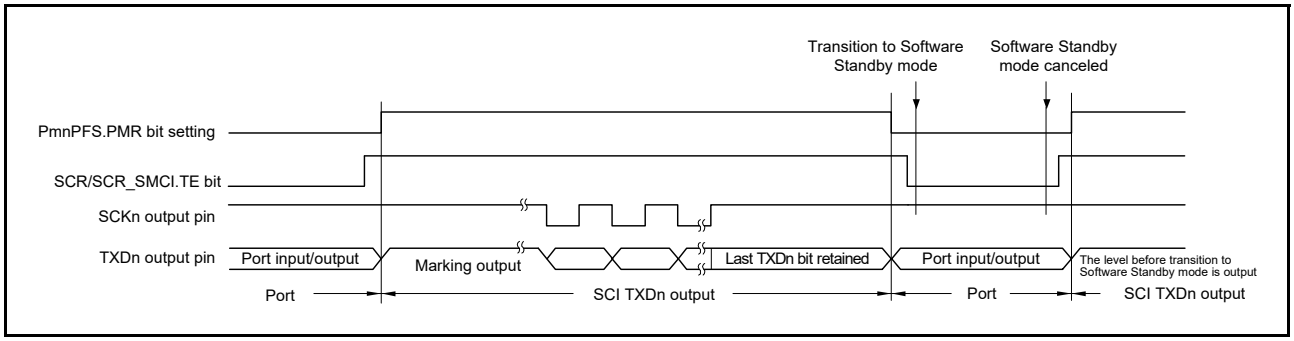


Figure 27.75 Port pin states during transition to Software Standby mode with internal clock and clock synchronous transmission

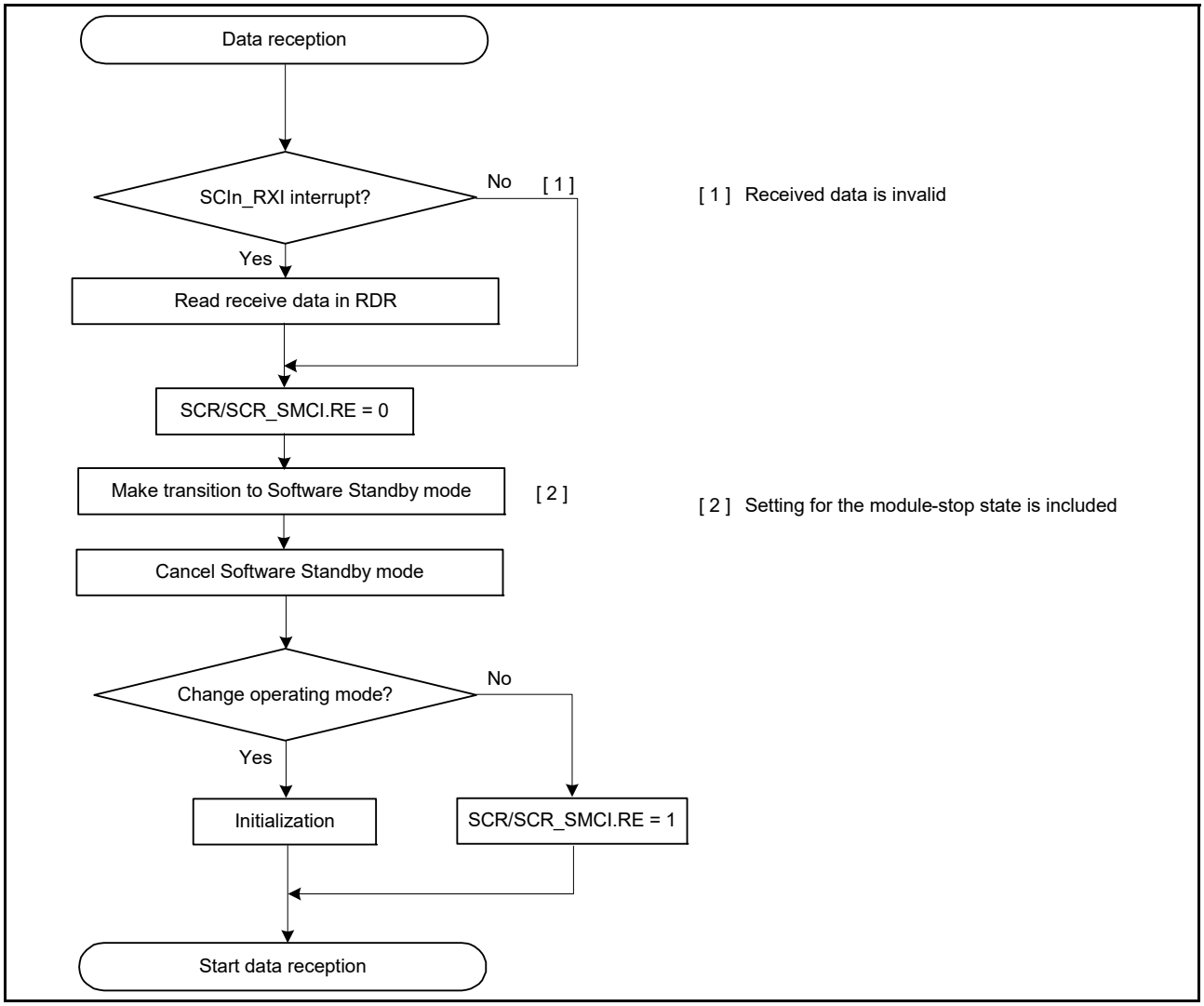


Figure 27.76 Example flow of transition to Software Standby mode during reception

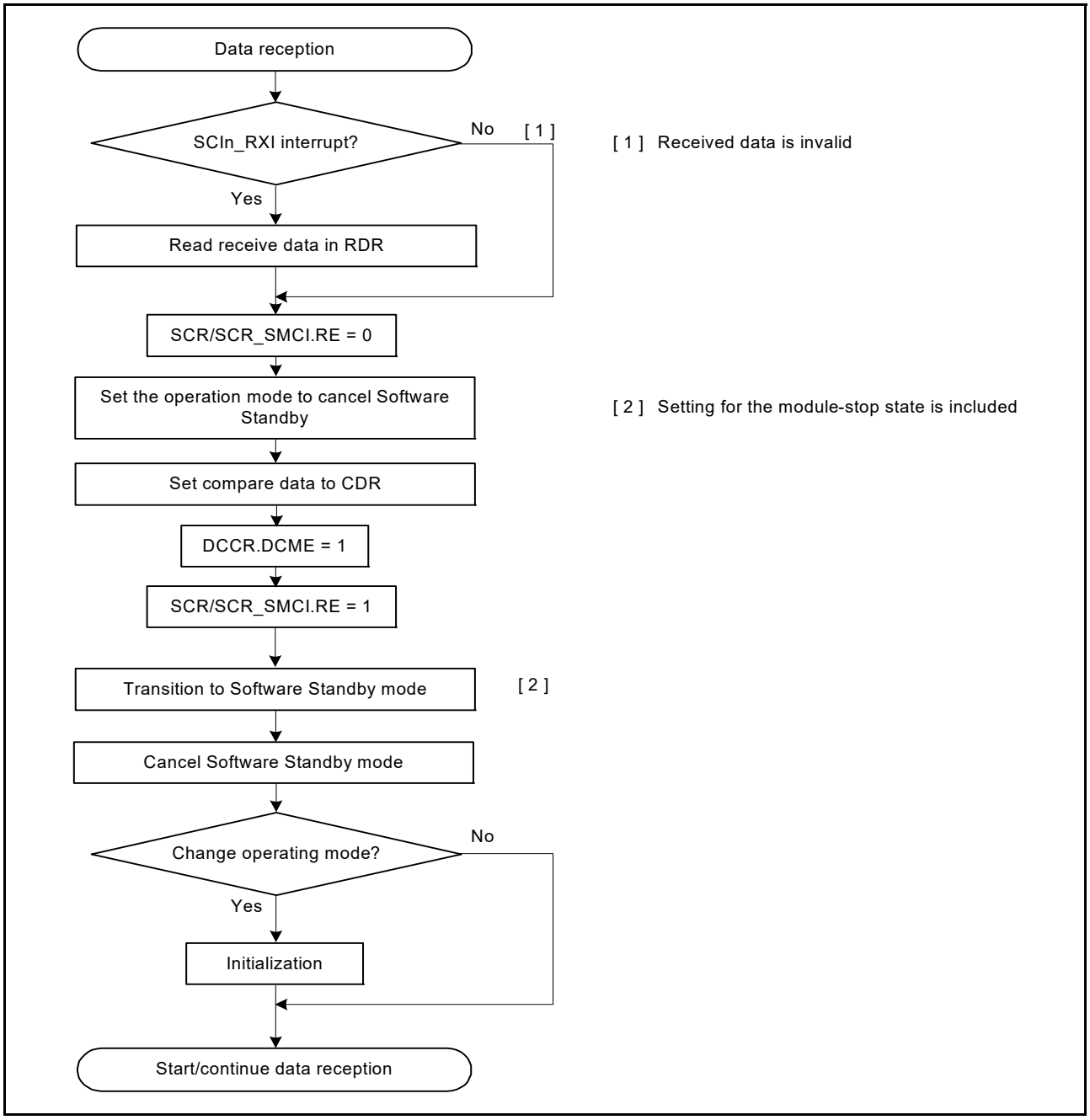


Figure 27.77 Example flow of transition to Software Standby mode during reception with address match

27.14.3 Break Detection and Processing

(1) Non-FIFO selected

When a framing error is detected, a break can be detected by reading the RXDn pin value directly. In a break, the input from the RXDn pin becomes all 0s, and the SSR.FER flag is set to 1 to indicate a framing error. The SSR.PER flag might also be set to 1 to indicate a parity error. The SCI continues the receive operation even after a break is received. Therefore, if the FER flag is 0, indicating that no framing error occurred, it is set to 1 again. When the SEMR.RXDESEL bit is 1, the SCI sets the SSR.FER flag to 1 and stops receiving operations until a start bit of the next data frame is detected. If the SSR.FER flag is 0, the SSR.FER flag retains 0 during the break.

When the RXDn pin is set to 1 and the break ends, detecting the beginning of the start bit on the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

(2) FIFO selected

After a framing error is detected and when the SCI detects that continuous receive data is 0 for one frame, reception stops. When a framing error is detected, a break can be detected by reading the SPTR.RXDMON bit value. After the RXDn signal is in the mark state and the break ends, reception of data to the FRDRHL register resumes.

27.14.4 Mark State and Production of Breaks

When the SCR/SCR_SMCI.TE bit is 0, disabling serial transmission, the state of the TXDn pin can be set using the SPTR.SPB2IO and SPTR.SPB2DT bits. With this approach, a TXDn pin can be placed in the mark state to transmit a break.

Before setting the SCR/SCR_SMCI.TE bit to 1, enabling serial transmission, set the SPB2IO and SPB2DT bits to put the communication line in the mark state (the state of 1), and change the TxDn pin using I/O port function. To output a break on data transmission, after setting the TXDn pin to output 0 by setting the SPB2IO and SPB2DT bits, change the TXDn pin using the I/O port function and set the SCR/SCR_SMCI.TE bit to 0. When the SCR/SCR_SMCI.TE bit is set to 0, the transmitter is initialized regardless of the current state of transmission.

27.14.5 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 TDR or FTDRL*1. 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.

27.14.6 Restrictions on Clock Synchronous Transmission in Clock Synchronous Mode and Simple SPI Mode

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of transmission

Wait at least the following time from writing transmit data to TDR to the start of the external clock input:

1 PCLKB cycle + data output delay time for the slave (t_{DO}) + setup time for the master (t_{SU}).

See Figure 27.78.

(2) Continuous transmission

Write the next transmit data to TDR or TDRHL before the falling edge of the transmit clock for bit [7]. See Figure 27.78.

When updating TDR after bit [7] starts to transmit, update TDR while the synchronization clock is in the low-level period, and set the high-level width of the transmit clock, bit [7] (D7) to 4 PCLKB cycles or longer. See Figure 27.78.

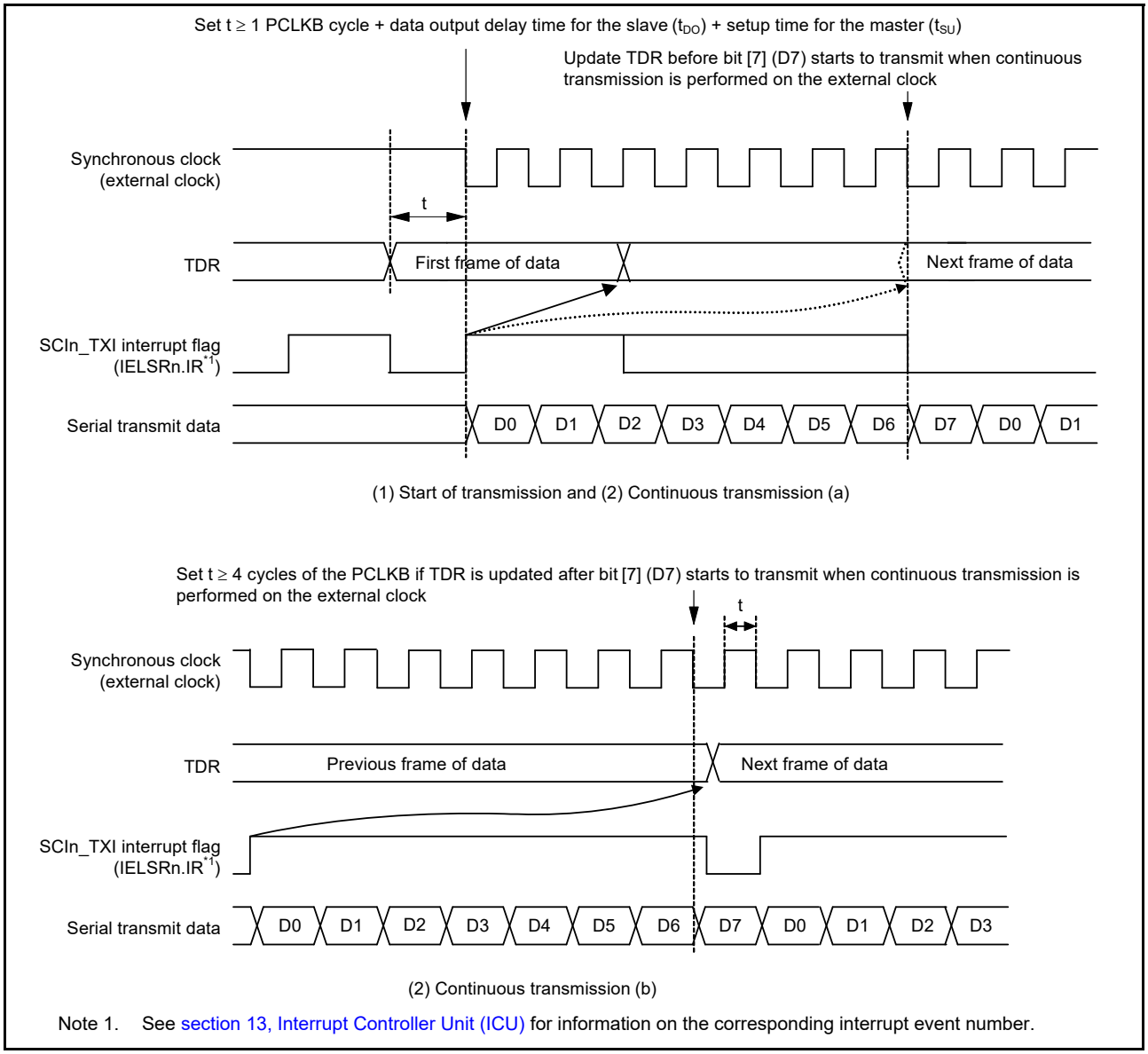


Figure 27.78 Restrictions on the use of external clock in clock synchronous transmission

27.14.7 Restrictions on Using DTC

During transmission or reception operations using the DTC, do not set transfer data for the DTC.

(1) Writing data to TDR (FTDRHL)

(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 DTC, be sure to write transmit data to TDR or TDRHL in the SCIn_TXI interrupt request handling routine.

(b) FIFO selected

It is possible to write data to the FTDRH and FTDRL registers when SCR.TE is 1. Confirm the amount of writable data using the FDR.T[4:0] bits.

(2) Reading data from RDR (FRDRHL)

When using the 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.

27.14.8 Notes on Starting Transfer

When transfer starts after 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 13, 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.

27.14.9 External Clock Input in Clock Synchronous Mode and Simple SPI Mode

In clock synchronous mode and simple SPI mode, the external clock (SCKn) must be input as follows:

High-pulse period, low-pulse period = 2 PCLKB cycles or more, period = 6 PCLKB cycles or more.

27.14.10 Limitations on Simple SPI Mode

(1) Master mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set in the SPMR.CKPH and CKPOL bits when the SPMR.SSE bit is 1. This prevents the clock line from being placed in the high-impedance state when the SCR.TE bit is set to 0 or unexpected edges from being generated on the clock line when the SCR.TE bit changes from 0 to 1. When the SPMR.SSE bit is 0 in single master mode, pulling up or pulling down the clock line is not required because the clock line is not placed in the high-impedance state even when the SCR.TE bit is set to 0.
- For the clock delay setting (SPMR.CKPH bit is 1), the receive data full interrupt (SCIn_RXI) is generated before the final clock edge on the SCKn pin, as indicated in Figure 27.79. 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 a character is being transferred, stopping supply of the clock signal to the connected slave. Reset the connected slave to avoid misaligned bits when transfer is restarted.

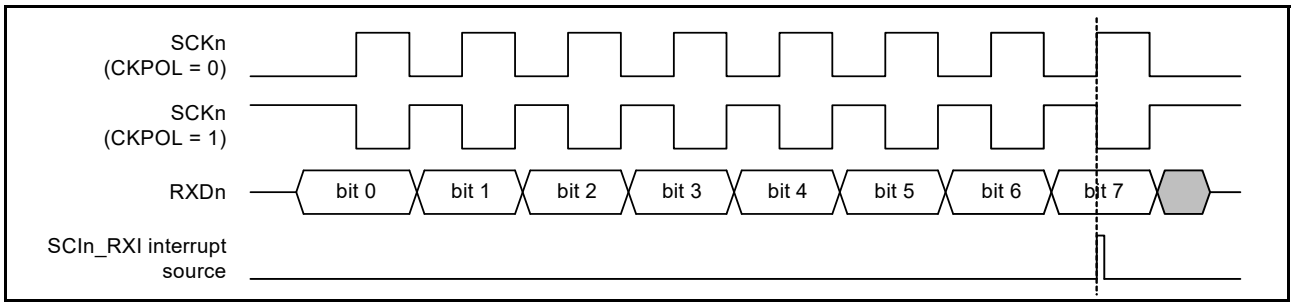


Figure 27.79 Timing of SCIn_RXI interrupt in simple SPI mode with clock delay

(2) Slave mode

- Wait at least the following time from writing transmit data to the TDR register to the start of the external clock input:
  1 PCLKB cycle + data output delay time for the slave (t_{DO}) + setup time for the master (t_{SU})

  Also wait at least 5 PCLKB 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 the SCR to 0 and, after restoring the settings, restart transfer of the first byte.

27.14.11 Notes on Transmit Enable Bit (SCR.TE)

In initial resister value, when SCR.TE = 0, the state of the TXDn pin is high impedance. The TXDn line should not be high impedance by the following one of ways.

1. The pull-up resistance is connected to the TXDn line.
2. Before setting the SCR.TE bit to 0, the function of the pin should be changed to a general-purpose output port. After that, set the SCR.TE bit to 1, and then change the function of the pin to TXDn.
3. In asynchronous mode, set SPTR register and decided level of TXDn pin during SCR.TE = 0.

In the Simple SPI mode slave operation, the MISOn pin operates in the same way as the above TXDn pin. The MISOn pin, the same as TXDn pin, should not be high impedance by the above list number 1 or list number 2.

27.14.12 Note on Stopping Reception When Using the RTS Function in Asynchronous Mode

One clock cycle of PCLK is required for the time from setting the SCR.RE bit to 0 to stopping the RTS signal generator in asynchronous mode.

When reading the RDR (or RDRHL) register after setting the SCR.RE bit to 0, confirm that the SCR.RE bit has been set to 0 before reading the RDR (or RDRHL) register to prevent these two processes form being performed consecutively.

28. I2C Bus Interface (IIC)

28.1 Overview

The MCU has a 2-channel I2C Bus Interface (IIC). The IIC module conforms with and provides a subset of the NXP I2C (Inter-Integrated Circuit) bus interface functions.

Table 28.1 lists the IIC specifications, Figure 28.1 shows a block diagram, and Figure 28.2 shows an example of I/O pin connections to external circuits. Table 28.2 lists the I/O pins.

Table 28.1 IIC specifications (1 of 2)

Parameter	Specifications
Communications format	• I2C bus format or SMBus format • Master mode or slave mode selectable • Automatic securing of the setup times, hold times, and bus-free times for the transfer rate.
Transfer rate	Fast mode supported, up to 400 kbps
SCL clock	For master operation, the duty cycle of the SCL clock is selectable in the range from 4% to 96%
Issuing and detecting conditions	• Start, restart, and stop conditions are automatically generated • Start conditions (including restart conditions) and stop conditions are detectable.
Slave address	• Configurable for up to three different slave addresses • 7-bit and 10-bit address formats supported, including simultaneous use • General call addresses, device ID addresses, and SMBus host addresses detectable.
Acknowledgment	• For transmission, automatic loading of the acknowledge bit. Transfer of the next transmit data can be automatically suspended on detection of a not-acknowledge bit. • For reception, automatic transmission of the acknowledge bit. If a wait between the 8th and 9th clock cycles is selected, software can control the value in the acknowledge field in response to the received value.
Wait function	During reception, the following wait periods are available by holding the SCL clock low: • Waiting between the 8th and 9th clock cycles • Waiting between the 9th clock cycle and the first clock cycle of the next transfer.
SDA output delay function	Output timing of transmitted data, including the acknowledge bit, can be delayed
Arbitration	• For multi-master operation: - SCL clock synchronization is possible when conflict occurs with the SCL signal from another master - When issuing the start condition creates conflict on the bus, loss of arbitration is detected by testing for mismatching between the internal signal for the SDAn line and the level on the SDAn line - In master operation, loss of arbitration is detected by testing for mismatching between the signal on the SDAn line and the internal signal for the SDAn 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 SDAn line and the level on the SDAn 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 event occurrence (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 to reduce power consumption
IIC operating modes	• Master transmit • Master receive • Slave transmit • Slave receive.

Table 28.1 IIC specifications (2 of 2)

Parameter	Specifications
Event link function (output)	• Transfer error or event occurrence (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 Software Standby or Snooze mode using a wakeup event

Note 1. This function is only available for IIC channel IIC0. IIC1 is not supported.

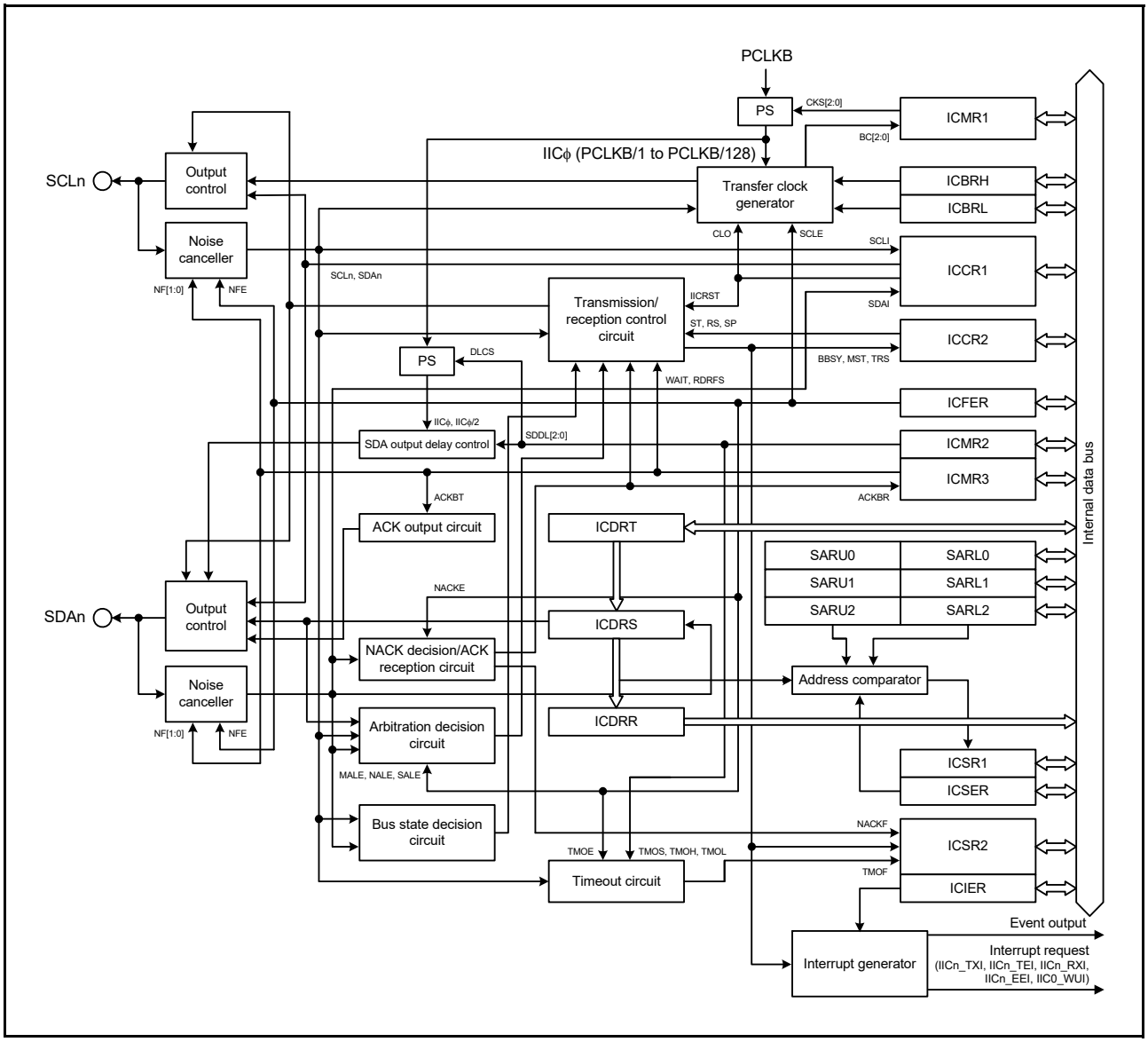


Figure 28.1 IIC block diagram

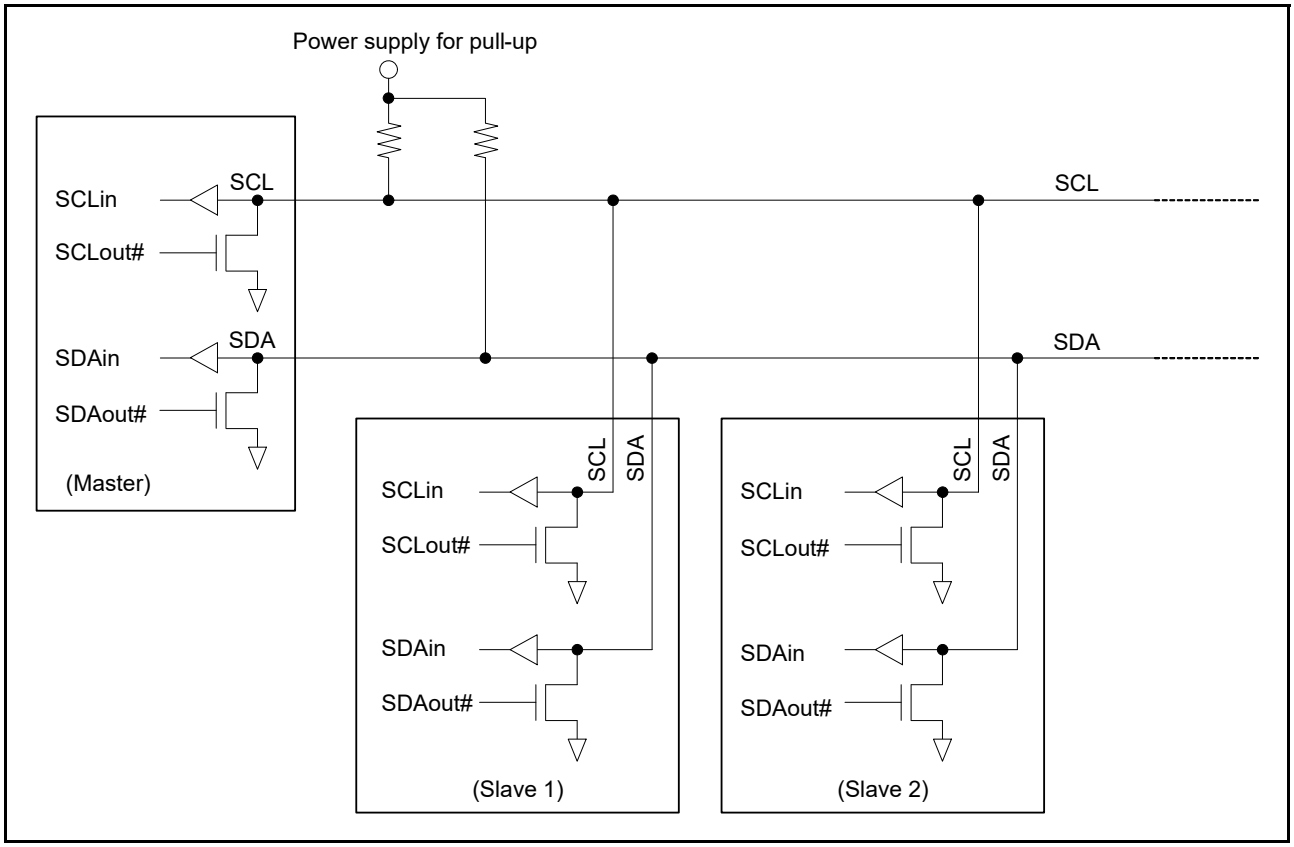


Figure 28.2 I/O pin connection to the external circuit (I2C bus configuration example)

The input level of the signals for IIC is CMOS when I2C bus is selected (ICMR3.SMBS = 0), or TTL when SMBus is selected (ICMR3.SMBS = 1).

Table 28.2 IIC I/O pins

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

28.2 Register Descriptions

28.2.1 I2C 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	SDAn Line Monitor	0: SDAn line is low 1: SDAn line is high.	R
b1	SCLI	SCLn Line Monitor	0: SCLn line is low 1: SCLn line is high.	R
b2	SDAO	SDA Output Control/Monitor	• Read: 0: IIC drives SDAn pin low 1: IIC releases 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 SCLn pin low 1: IIC releases 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	I2C bus Interface Internal Reset	0: Release IIC reset or internal reset 1: Initiate IIC reset or internal reset. This setting clears the bit counter and the SCLn/SDAn output latch.	R/W
b7	ICE	I2C bus Interface Enable	0: Disable (SCLn and SDAn pins in inactive state) 1: Enable (SCLn and SDAn pins in active state). Use 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 the bit to 1 in a normal communication state causes a communication error. For details on this function, see section 28.12.2, Extra SCL Clock Cycle Output Function.

IICRST bit (I2C bus Interface Internal Reset)

The IICRST bit initiates an internal state reset of the IIC. Setting this bit to 1 initiates an IIC reset or internal reset. Whether an IIC reset or internal reset is initiated is determined by setting this bit in combination with the ICE bit. Table 28.3 lists the IIC resets.

The IIC reset initializes all registers except ICCR1.ICE and ICCR1.IICRST bits, and internal states of the IIC. In addition to the internal states of the IIC, the internal reset initializes the following:

- Bit counter (ICMR1.BC[2:0] bits)
- I2C Bus Shift Register (ICDRS)
- I2C Bus Status Registers (ICSR1 and ICSR2)
- SDAO and SCLO Output Control/Monitor (ICCR1.SCLO and ICCR1.SDAO bits)
- I2C Bus Control Register 2 (except ICCR2.BBSY bit).

For the reset conditions of each register, see section 28.15, 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 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, then issue a restart condition from the master device, or issue a stop condition and resume communication from the start condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start or restart condition from the master device, synchronization is lost because the master and slave devices operate asynchronously.

Table 28.3 IIC resets

IICRST	ICE	State	Specifications
1	0	IIC reset	Resets all registers except ICCR1.ICE and ICCR1.IICRST bits, and the internal states of the IIC
	1	Internal reset	Resets the following: • ICMR1.BC[2:0] bits • ICSR1, ICSR2, ICDRS registers • SDAO and SCLO Output Control/Monitor (ICCR1.SCLO and ICCR1.SDAO bits) • I2C-Bus Control Register 2 (except ICCR2.BBSY bit) • Internal states of the IIC.

ICE bit (I2C bus Interface Enable)

The ICE bit selects the active or inactive state of the SCLn and SDAn pins. It can also be combined with the IICRST bit to initiate one of two types of resets. See Table 28.3 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.

28.2.2 I2C 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: I2C bus released (bus free state) 1: I2C bus occupied (bus busy state).	R

Note 1. The MST and TRS bits can be written to when the ICMR1.MTWP bit is set to 1.

ST bit (Start Condition Issuance Request)

The ST bit requests transition to master mode and issues a start condition.

When this bit is set to 1, a start condition is issued when the BBSY flag is set to 0 (bus free state). For details on issuing start conditions, see section 28.11, 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 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 restart conditions, see section 28.11, Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the RS bit with the BBSY flag in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition is issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Do not set the RS bit to 1 while issuing a stop condition.
Note: If 1 (restart condition request) is written to the RS bit in slave mode, the restart condition is not issued, but the RS bit remains set to 1. If the operating mode changes to master mode without the bit being cleared, a restart condition might be issued.

SP bit (Stop Condition Issuance Request)

The SP bit requests that a stop condition be issued in master mode.

When this bit is set to 1, a stop condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on issuing stop conditions, see section 28.11, Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the MST bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition is issued (a stop condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Writing to the SP bit is not possible when the BBSY flag is 0 (bus free state).
Note: Do not set the SP bit to 1 when a restart condition is issued.

TRS bit (Transmit/Receive Mode)

The TRS bit indicates transmit or receive mode.

The IIC is in receive mode when the TRS bit is 0 and in transmit mode when the bit is 1. The combination of 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 this bit is not required during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1)
- When a restart condition is issued normally because of a restart condition request (when a restart condition is detected with the RS bit set to 1)
- When the R/W# bit appended to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in the ICSER register, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected

- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When the R/W# bit appended to the slave address is set to 1 in master mode
- In slave mode, on a match between the received address and the address enabled in ICSER when the value of the received R/W# bit is 0, including when the received address is the general call address
- In slave mode, when a restart condition is detected (a restart condition is detected with ICCR2.BBSY = 1 and ICCR2.MST = 0)
- When 0 is written to the TRS bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

MST bit (Master/Slave Mode)

The MST bit indicates master or slave mode. The IIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. The combination of the MST bit and the TRS bit indicates the IIC operating mode.

The value of the MST bit automatically changes to 1 for master mode or 0 for slave mode when a start condition is issued or a stop condition is issued or detected. Although writing to the MST bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not required during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 0 is written to the MST bit with the MTWP bit in ICMR1set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

BBSY flag (Bus Busy Detection Flag)

The BBSY flag indicates whether the 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 register 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 register 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).

28.2.3 I2C 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] are read/write bits, it is not required to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one, for an additional acknowledge bit, between transferred frames when the SCLn line is at a low level.

The value in the BC[2:0] bits returns to 000b at the end of a data transfer, including the acknowledge bit, or when a start or restart condition is detected.

28.2.4 I2C 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 when SCLn line is low 1: Enable count when SCLn line is low.	R/W
b2	TMOH	Timeout H Count Control	0: Disable count when SCLn line is high 1: Enable count when 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 SDA output delay counter.*1	R/W

Note 1. The setting DLCS = 1 (IICϕ/2) is only valid when SCL is low. When SCL is high, the DLCS = 1 setting 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 = 1). When this bit is set to 0, long mode is selected. When it is set to 1, short mode is selected.

In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14 bit-counter. While the SCLn line is in the state that enables this counter as specified in the TMOH and TMOL bits, the counter counts up in sync with the internal reference clock (IICϕ) as a count source.

For details on the timeout function, see section 28.12.1, Timeout Function.

TMOL bit (Timeout L Count Control)

The TMOL bit enables or disables up-counting on the internal counter of the timeout function when the SCLn line is held low and the timeout function is enabled (ICFER.TMOE = 1).

TMOH bit (Timeout H Count Control)

The TMOH bit enables or disables up-counting on the internal counter of the timeout function while the SCLn line is held high and the timeout function is enabled (ICFER.TMOE = 1).

SDDL[2:0] bits (SDA Output Delay Counter)

The SDDL[2:0] bits can be used to delay the SDA output. This counter works with the clock source selected in the DLCS bit. 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 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 the devices might malfunction or falsely indicate a start or stop condition, depending on the bus state.

For details on this function, see section 28.5, 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)

28.2.5 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: Filter out noise of up to 1 IICϕ cycle (single-stage filter) 0 1: Filter out noise of up to 2 IICϕ cycles (2-stage filter) 1 0: Filter out noise of up to 3 IICϕ cycles (3-stage filter) 1 1: Filter out noise of up to 4 IICϕ cycles (4-stage filter).	R/W
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: Send 0 as the acknowledge bit (ACK transmission) 1: Send 1 as the acknowledge bit (NACK transmission).	R/W*1
b4	ACKWP	ACKBT Write Protect	0: Write protect ACKBT bit 1: Write enable 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 The period between 9th clock cycle and first clock cycle is not held low 1: Wait The period between 9th clock cycle and first clock cycle is held low. Low-hold is released by reading ICDRR.	R/W*2
b7	SMBS	SMBus/I2C bus Select	0: I2C bus selected 1: SMBus selected.	R/W

Note 1. Write to the ACKBT bit only when the ACKWP bit is 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 28.6, Digital Noise Filter Circuits.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCLn line high-level period or low-level period. If the noise range is set to a value of [SCL clock width: high-level period or low-level period, whichever is shorter] - [1.5 internal reference clock (IICϕ) cycles + analog noise filter: 120 ns (reference values)] or more, the SCL clock is regarded as noise 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 sets the acknowledge bit to be sent in receive mode.

[Setting condition]

- When 1 is written to this bit with the ACKWP bit set to 1.

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1
- When a stop condition request is detected with the SP bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 when the ICE bit in ICCR1 is 0 (IIC reset).

ACKWP bit (ACKBT Write Protect)

The ACKWP bit controls write enabling of the ACKBT bit.

RDRFS bit (RDRF Flag Set Timing Select)

The RDRFS bit selects the RDRF flag set timing in receive mode and also selects whether to hold the SCLn line low on the falling edge of the 8th SCL clock cycle.

When the RDRFS bit is 0, the SCLn line is not held low on the falling edge of the 8th SCL clock cycle, and the RDRF flag is set to 1 on the rising edge of the 9th SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 on the rising edge of the 8th SCL clock cycle, and the SCLn line is held low on the falling edge of the 8th SCL clock cycle. The low-hold of the SCLn line is released by a write to the ACKBT bit.

After data is received with this setting, the SCLn line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1), based on the receive data.

WAIT bit (WAIT)

The WAIT bit controls whether to hold the period between the 9th SCL clock cycle and the 1st SCL clock cycle low until the receive data buffer (ICDRR) is completely read each time a single-byte data is received in receive mode.

When the WAIT bit 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 bit is 1, the SCLn line is held low from the falling edge of the 9th clock cycle until the ICDRR value is

read each time a single-byte data is received. This enables receive operation in byte units.

Note: When the value of the WAIT bit is to be read, be sure to read the ICDRR register first.

SMBS bit (SMBus/I2C bus Select)

Setting the SMBS bit to 1 selects the SMBus and enables the HOAE bit in ICSER.

28.2.6 I2C 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 28.12.1, Timeout Function.

MALE bit (Master Arbitration-Lost Detection Enable)

The MALE bit specifies whether to use the arbitration-lost detection function in master mode. For normal operation, set this bit to 1.

NALE bit (NACK Transmission Arbitration-Lost Detection Enable)

The NALE bit specifies whether to cause lost of arbitration when ACK is detected during the transmission of NACK in receive mode, for example, when slaves with the same address exist on the bus, or when two or more masters select the same slave device simultaneously with a different number of receive bytes.

SALE bit (Slave Arbitration-Lost Detection Enable)

The SALE bit specifies whether to cause loss of arbitration when a value different from the value being transmitted is detected on the bus in slave transmit mode, for example, when slaves with the same address exist on the bus, or when a

mismatch with the transmit data occurs because of noise.

NACKE bit (NACK Reception Transfer Suspension Enable)

The NACKE bit specifies whether to continue or discontinue the transfer operation when NACK is received in transmit mode. For normal operation, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended. When the NACKE bit is 0, the next transfer operation continues regardless of the received acknowledge content.

For details, see section 28.9.2, NACK Reception Transfer Suspension Function.

SCLE bit (SCL Synchronous Circuit Enable)

The SCLE bit specifies whether to synchronize the SCL clock with the SCLn input clock. For normal operation, set this bit to 1.

When the SCLE bit is set to 0 (no SCL synchronous circuit used), the IIC does not synchronize the SCL clock with the SCLn input clock. In this setting, the IIC outputs the SCL clock with the transfer rate set in ICBRH and ICBRL, regardless of the SCLn line state. For this reason, if the bus load of the I2C bus line is much larger than the specification value, or if the SCLn clock output overlaps in multiple masters, a short-cycle SCL clock that does not meet the specification might be output. When no SCL synchronous circuit is used, it also affects the issuing of the start, restart, and stop conditions, and the continuous output of extra SCL clock cycles.

Do not set the SCLE bit to 0 except when checking the output of the set transfer rate.

28.2.7 I2C 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 the SARLy and SARUy registers.

When this bit is set to 1, the slave address set in the SARLy and SARUy registers is enabled and is compared with the received slave address. When this bit is set to 0, the slave address set in SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

GCAE bit (General Call Address Enable)

The GCAE bit specifies whether to ignore the general call address (0000 000b + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the IIC recognizes the received slave address as the general call address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2), and performs the data receive operation. When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

DIDE bit (Device ID Address Detection Enable)

The DIDE bit specifies whether to recognize and execute the device ID address when a device ID (1111 100b) is received in the first frame after a start or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the IIC recognizes that the device ID address was received. When the subsequent R/W# bit is 0 [W], the IIC recognizes the second and subsequent frames as slave addresses and continues the receive operation. When this bit is set to 0, the IIC ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.

For details, see section 28.7.3, Device ID Address Detection.

HOAE bit (Host Address Enable)

The HOAE bit specifies whether to ignore the received host address (0001 000b) when the SMBS bit in ICMR3 is 1.

When this bit is set to 1 while the SMBS bit in ICMR3 is 1, if the received slave address matches the host address, the IIC recognizes the received slave address as the host address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs the receive operation.

When the SMBS bit in ICMR3 or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

28.2.8 I2C 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 the ICSR2 register 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 the ICSR2 register is 1. To cancel an ALI interrupt request, set the AL flag or the ALIE bit to 0.

STIE bit (Start Condition Detection Interrupt Request Enable)

The STIE bit enables or disables start condition detection interrupt (STIn) requests when the START flag in the ICSR2 register is 1. To cancel an STI interrupt request, set the START flag or the STIE bit to 0.

SPIE bit (Stop Condition Detection Interrupt Request Enable)

The SPIE bit enables or disables stop condition detection interrupt (SPIn) requests when the STOP flag in the ICSR2 register is 1. To cancel an SPI interrupt request, set the STOP flag or the SPIE bit to 0.

NAKIE bit (NACK Reception Interrupt Request Enable)

The NAKIE bit enables or disables NACK reception interrupt (NAKIn) requests when the NACKF flag in the ICSR2 register is 1. To cancel an NAKI interrupt request, set the NACKF flag or the NAKIE bit to 0.

RIE bit (Receive Data Full Interrupt Request Enable)

The RIE bit enables or disables receive data full interrupt (IICn_RXI) requests when the RDRF flag in the ICSR2 register is 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 the ICSR2 register is 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 the ICSR2 register is 1.

28.2.9 I2C 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

Bit	Symbol	Bit name	Description	R/W
b5	DID	Device ID Address Detection Flag	0: Device ID command not detected 1: Device ID command detected. This bit is set to 1 when the first frame received immediately after a start condition is detected, matches a value of (device ID (1111 100b) + 0[W]).	R/(W) *1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOA	Host Address Detection Flag	0: Host address not detected 1: Host address detected. This bit is set to 1 when the received slave address matches the host address (0001 000b).	R/(W) *1

Note 1. Only 0 can be written to clear the flag.

AASy flag (Slave Address y Detection Flag) (y = 0 to 2)

The AASy flag indicates whether slave address y was detected.

[Setting conditions]

For 7-bit address format (SARUy.FS = 0):

- When the received slave address matches the SVA[6:0] value in the SARLy register, with the SARyE bit in the ICSER register set to 1 (slave address y detection enabled). The AASy 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 the ICSER register set to 1 (slave address y detection enabled). The AASy 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 the ICCR1 register to initiate an IIC reset or an internal reset.

For 7-bit address format (SARUy.FS = 0):

- When the received slave address does not match the SVA[6:0] value in SARLy, with the SARyE bit in the ICSER register set to 1 (slave address y detection enabled). The AASy 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 the ICSER register set to 1 (slave address y detection enabled). The AASy 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). The AASy 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)

The GCA flag indicates whether the general call address was detected.

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 [W]), with the GCAE bit in the ICSER register set to 1 (general call address detection enabled). The GCA 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 enabled). The GCA 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)

The DID flag indicates whether the device ID address was detected.

[Setting condition]

- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), with the DIDE bit in ICSER set to 1 (device ID address detection enabled). The DID flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID bit after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start or restart condition is detected does not match a value of (device ID (1111 100b)), with the DIDE bit in ICSER set to 1 (device ID address detection enabled). The DID flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame
- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), and the second frame does not match any slave address from 0 to 2 with the DIDE bit in ICSER set to 1 (device ID address detection enabled). The DID 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)

The HOA flag indicates whether the host address was detected.

[Setting condition]

- When the received slave address matches the host address (0001 000b), with the HOAE bit in ICSER set to 1 (host address detection enabled). The HOA flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA flag after reading HOA = 1
- When a stop condition is detected
- When the received slave address does not match the host address (0001 000b), with the HOAE bit in ICSER set to 1 (host address detection enabled). The HOA flag is set to 0 on the rising edge of the 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.

28.2.10 I2C 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
b6	TEND	Transmit End Flag	0: Data being transmitted 1: Data transmit complete.	R/(W) *1
b7	TDRE	Transmit Data Empty Flag	0: ICDRT contains transmit data 1: ICDRT contains no transmit data.	R

Note 1. Only 0 can be written to clear the flag.

TMOF flag (Timeout Detection Flag)

The TMOF flag is set to 1 when the IIC detects a timeout because the SCLn line state remained unchanged for the set period.

[Setting condition]

- When the SCLn line state remains unchanged for the period specified in the ICMR2.TMOH, TMOL, and TMOS bits while the ICFER.TMOE bit is 1 (the timeout function enabled) in master or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF flag after reading TMOF = 1
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

AL flag (Arbitration-Lost Flag)

The AL flag indicates that the 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. If the level on the line does not match the value of the bit being output, the IIC 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 or during data transmission.

[Setting conditions]

When master arbitration-lost detection is enabled (ICFER.MALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock, except for the ACK period during data transmission in master transmit mode

- When a start condition is detected while the ST bit in ICCR2 is 1 (start condition requested) or the internal SDA output state does not match the SDAn line level
- When the ST bit in ICCR2 is 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 the SCL clock in the ACK period during NACK transmission in receive mode.

When slave arbitration-lost detection is enabled (ICFER.SALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock, except for the ACK period during data transmission in slave transmit mode.

[Clearing conditions]

- When 0 is written to the AL flag after reading AL = 1
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

Table 28.4 Relationship between arbitration-lost generation sources and arbitration-lost enable functions

ICFER			ICSR2		
MALE	NALE	SALE	AL	Error	Arbitration-lost generation source
1	x	x	1	Start condition issuance error	When internal SDA output state does not match 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 is 1
			1	Transmit data mismatch	When transmit data including slave address does not match the bus state in master transmit mode
x	1	x	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master or slave receive mode
x	x	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

x: Don't care

START flag (Start Condition Detection Flag)

The START flag indicates whether a start or restart condition is detected.

[Setting condition]

- When a start or a restart condition is detected.

[Clearing conditions]

- When 0 is written to the START flag after reading START = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

STOP flag (Stop Condition Detection Flag)

The STOP flag indicates whether a stop condition is detected.

[Setting condition]

- When a stop condition is detected.

[Clearing conditions]

- When 0 is written to the STOP flag after reading STOP = 1
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

NACKF flag (NACK Detection Flag)

The NACKF flag indicates whether a NACK was detected.

[Setting condition]

- When acknowledge is not received (NACK received) from the receive device in transmit mode, with the NACKE bit in ICFER set to 1 (transfer suspension enabled).

[Clearing conditions]

- When 0 is written to the NACKF flag after reading NACKF = 1
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1, the IIC suspends data transmission and reception. Writing to ICDRT in transmit mode or reading from ICDRR in receive mode with the NACKF flag set to 1 does not enable data transmit or receive operation. To restart data transmission or reception, set the NACKF flag to 0.

RDRF flag (Receive Data Full Flag)

The RDRF flag indicates whether the IDCRR contains receive data.

[Setting conditions]

- When receive data is transferred from ICDRS to ICDRR. The RDRF flag is set to 1 on the rising edge of the 8th or 9th SCL clock cycle (selected in the RDRFS bit in ICMR3)
- When the received slave address matches, after a start or a restart condition is detected, with the TRS bit in ICCR2 set to 0.

[Clearing conditions]

- When 0 is written to the RDRF flag after reading RDRF = 1
- When data is read from ICDRR
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

TEND flag (Transmit End Flag)

The TEND flag indicates whether data transmission is still being transmitted or is complete.

[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 flag after reading TEND = 1
- When data is written to ICDRT
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

TDRE flag (Transmit Data Empty Flag)

The TDRE flag indicates whether the ICDRT contains transmit data.

[Setting conditions]

- When data is transferred from ICDRT to ICDRS and ICDRT becomes empty
- When the TRS bit in ICCR2 is set to 1
- When the received slave address matches while the TRS bit is 1.

[Clearing conditions]

- When data is written to ICDRT
- When the TRS bit in ICCR2 is set to 0

- When 1 is written to the IICRST bit in ICCR1 to 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.

28.2.11 I2C-Bus Wakeup Unit Register (ICWUR)

Address(es): IIC0.ICWUR 4005 3016h

	b7	b6	b5	b4	b3	b2	b1	b0
	WUE	WUIE	WUF	WUACK	—	—	—	WUAFA
Value after reset:	0	0	0	1	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	WUAFA	Wakeup Analog Filter Additional Selection	0: Do not add the wakeup analog filter 1: Add the wakeup analog filter.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	WUACK	ACK bit for Wakeup Mode	Choice of four response modes in combination with ICCR1.IICRST and WUACK. See Table 28.5.	R/W
b5	WUF	Wakeup Event Occurrence Flag	0: Slave address not matching during wakeup 1: Slave address matching during wakeup.	R/W
b6	WUIE	Wakeup Interrupt Request Enable	0: Wakeup Interrupt Request (IIC0_WUI) disabled 1: Wakeup Interrupt Request (IIC0_WUI) enabled.	R/W
b7	WUE	Wakeup Function Enable	0: Wakeup function disabled 1: Wakeup function enabled.	R/W

Table 28.5 Wakeup mode

IICRST	WUACK	Operation mode	Description
0	0	Normal wakeup mode 1	ACK response on 9th SCL and SCL low hold after 9th SCL
0	1	Normal wakeup mode 2	No ACK response immediately and SCL low hold between 8th and 9th SCL. SCL low-hold release and ACK response at the 8th Low from the beginning of the SCL clock.
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)

The WUF flag indicates whether the slave address is matching during wakeup.

[Setting condition]

- When PCLKB is supplied after a slave-address match in the first 8th SCL low during wakeup mode.

[Clearing conditions]

- When 0 is written to the WUF flag after reading WUF = 1
- When ICCR1.ICE = 0 and IICRST = 1.

28.2.12 I2C 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 as 1. The write value should be 1.	R/W

WUSEN bit (Wakeup Function Synchronous Enable)

The WUSEN bit is used in combination with the WUASYF flag (or WUSYF flag) to switch between 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 from asynchronous to synchronous operation:

- When 1 is written to the WUSEN bit with the WUASYF flag at 1, when a wakeup event is detected. After writing 1, the WUASYF flag immediately becomes 0.
- When a stop condition is detected with a wakeup event undetected.

WUASYF flag (Wakeup Function Asynchronous Operation Status Flag)

The WUASYF flag can place the IIC in PCLKB asynchronous operation when the wakeup 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 WUASYF 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 is 0 and IICRST is 1 (IIC reset)
- When CWUR.WUE is 0.

WUSYF flag (Wakeup Function Synchronous Operation Status Flag)

The WUSYF 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 the 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 is 0 and IICRST is 1 (IIC reset)
- When ICWUR.WUE is 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.

28.2.13 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

	b7	b6	b5	b4	b3	b2	b1	b0
	SVA[6:0]							SVA0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SVA0	10-Bit Address LSB	Slave address setting	R/W
b7 to b1	SVA[6:0]	7-Bit Address/10-Bit Address Lower Bits	Slave address setting	R/W

SVA0 bit (10-Bit Address LSB)

When the 10-bit address format is selected (SARUy.FS = 1), 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.

28.2.14 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 or 10-bit format 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 SVA[1:0] and SVA0 settings in the SARLy register are ignored.

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y, and the SVA[1:0] and SARLy settings are valid.

When the SARyE bit in ICSER is 0 (SARLy and SARUy disabled), the SARUy.FS 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.

These bits are valid when the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting of these bits is ignored.

28.2.15 I2C 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

ICBRL is a 5-bit register that sets the low-level period of the SCL clock. ICBRL also generates the data setup time for automatic SCL low-hold operation, see section 28.9, Automatic Low-Hold Function for SCL.

BRL[4:0] bits (Bit Rate Low-Level Period)

The BRL[4:0] bits set the low-level period of the SCL clock. ICBRL counts the low-level period with the internal reference clock source (IICφ) specified by the CKS[2:0] bits in ICMR1. When the IIC is used in slave mode, the BRL[4:0] bits must be set to a value longer than the data setup time*1.

If the digital noise filter is enabled (NFE bit in ICFER is 1), set the BRL[4:0] bits to a value at least one greater than the number of stages in the noise filter. For this number, see the description of the ICMR3.NF[1:0] bits.

Note 1. Data setup time ($t_{SU:DAT}$)
250 ns for up to 100 kbps: Standard mode (Sm)
100 ns for up to 400 kbps: Fast mode (Fm)

28.2.16 I2C 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

ICBRH is a 5-bit register that sets the high-level period of the SCL clock. ICBRH is valid in master mode.

BRH[4:0] bits (Bit Rate High-Level Period)

The BRH[4:0] bits set the high-level period of the SCL clock. 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 (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 this number, see the description of the ICMR3.NF[1:0] bits.

The IIC transfer rate and the SCL clock duty are calculated using the following expressions:

1) ICFER.SCLE = 0
   Transfer rate = 1 / {[(BRH + 1) + (BRL + 1)]/IICφ*1 + tr*2 + tf*2}
   Duty cycle = {tr + [(BRH + 1)/IICφ]}/{tr + tf + [(BRH + 1) + (BRL + 1)]/IICφ}

2) ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] = 000b (IICφ = PCLKB)
   Transfer rate = 1 / {[(BRH + 3) + (BRL + 3)]/IICφ + tr + tf}
   Duty cycle = {tr + [(BRH + 3)/IICφ]}/{tr + tf + [(BRH + 3) + (BRL + 3)]/IICφ}

3) ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] = 000b (IICφ = PCLKB)
   Transfer rate = 1 / {[(BRH + 3 + nf*3) + (BRL + 3 + nf)]/IICφ + tr + tf}
   Duty cycle = {tr + [(BRH + 3 + nf)/IICφ]}/{tr + tf + [(BRH + 3 + nf) + (BRL + 3 + nf)]/IICφ}

4) ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] ≠ 000b
   Transfer rate = 1 / {[(BRH + 2) + (BRL + 2)]/IICφ + tr + tf}
   Duty cycle = {tr + [(BRH + 2)/IICφ]}/{tr + tf + [(BRH + 2) + (BRL + 2)]/IICφ}

5) ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] ≠ 000b
   Transfer rate = 1 / {[(BRH + 2 + nf) + (BRL + 2 + nf)]/IICφ + tr + tf}
   Duty cycle = {tr + [(BRH + 2 + nf)/IICφ]}/{tr + tf + [(BRH + 2 + nf) + (BRL + 2 + nf)]/IICφ}

Note 1. IICφ = PCLKB × Division ratio.
Note 2. The SCLn line rise time [tr] and SCLn line fall time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, see the I2C bus standard from NXP Semiconductors.
Note 3. nf = Number of digital noise filter stages selected in the ICMR3.NF bit.

Table 28.6 Example of ICBRH/ICBRL settings for transfer rate when SCLE = 0

Transfer rate (kbps)	CKS[2:0]	BRH[4:0]	BRL[4:0]	PCLKB[MHz]	NF[1:0]	Computation expression
100	011	15 (EFh)	18 (F2h)	32	—	1)
400	001	9 (E9h)	20 (F4h)	32	—	1)

Note: SCLn line rising time (tr): 100 kbps or less, [Sm]: 1000 ns, 400 kbps or less, [Fm]: 300 ns.
SCLn line falling time (tf): 400 kbps or less, [Sm/Fm]: 300 ns.

Table 28.7 Example of ICBRH/ICBRL settings for transfer rate when SCLE = 1 and NFE = 0

Transfer rate (kbps)	CKS[2:0]	BRH[4:0]	BRL[4:0]	PCLKB[MHz]	NF[1:0]	Computation expression
100	011	14 (EEh)	17 (F1h)	32	—	4)
400	001	8 (E8h)	19 (F3h)	32	—	4)

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

Table 28.8 Example of ICBRH/ICBRL settings for transfer rate when SCLE = 1 and NFE = 1

Transfer rate (kbps)	CKS[2:0]	BRH[4:0]	BRL[4:0]	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.

28.2.17 I2C 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 I2C-Bus Shift Register (ICDRS), it transfers the transmit data that was written to ICDRT to ICDRS and starts transmitting data in transmit mode.

The double-buffer structure of ICDRT and ICDRS allows continuous transmit operation if the next transmit data is written to ICDRT while the ICDRS data is being transmitted.

ICDRT can always be read and written to. Write transmit data to ICDRT once when a transmit data empty interrupt (IICn_TXI) request is generated.

28.2.18 I2C 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 I2C Bus Shift Register (ICDRS) to ICDRR to enable the next data to be received.

The double-buffer structure of ICDRS and ICDRR allows continuous receive operation if the received data is read from ICDRR while ICDRS is receiving data. ICDRR cannot be written to. Read data from ICDRR 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 low for 1 clock cycle before the RDRF flag is set to 1 again.

28.2.19 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.

28.3 Operation

28.3.1 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 28.3 shows the I2C bus format and Figure 28.4 shows the I2C bus timing.

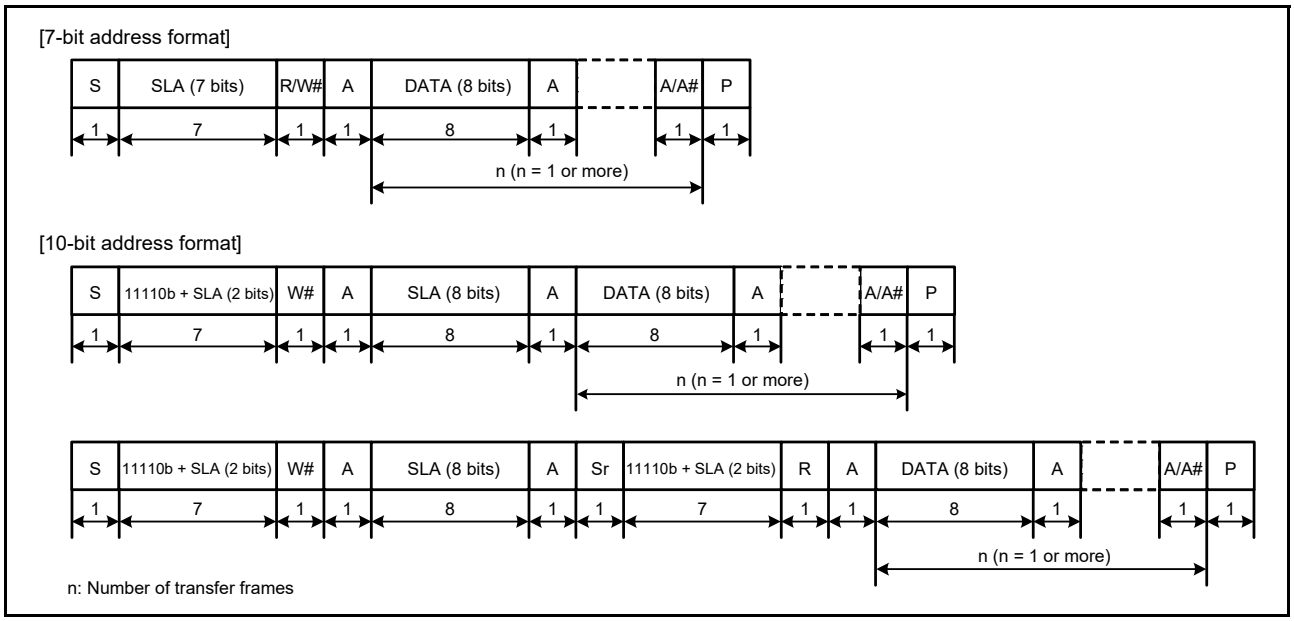


Figure 28.3 I2C bus format

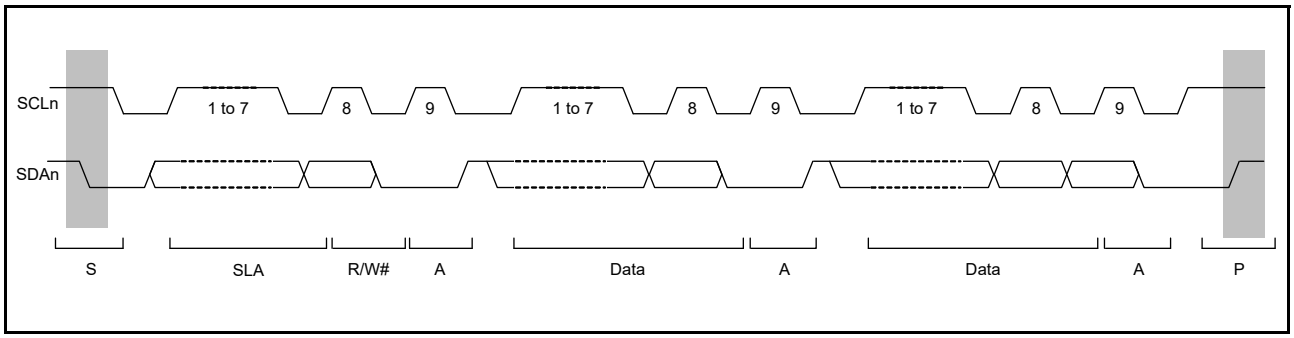


Figure 28.4 I2C bus timing (SLA = 7 bits)

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 when the SCLn line is high.

28.3.2 Initial Settings

Before starting data transmission or reception, initialize the IIC using the procedure shown in Figure 28.5.

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 28.5.
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.

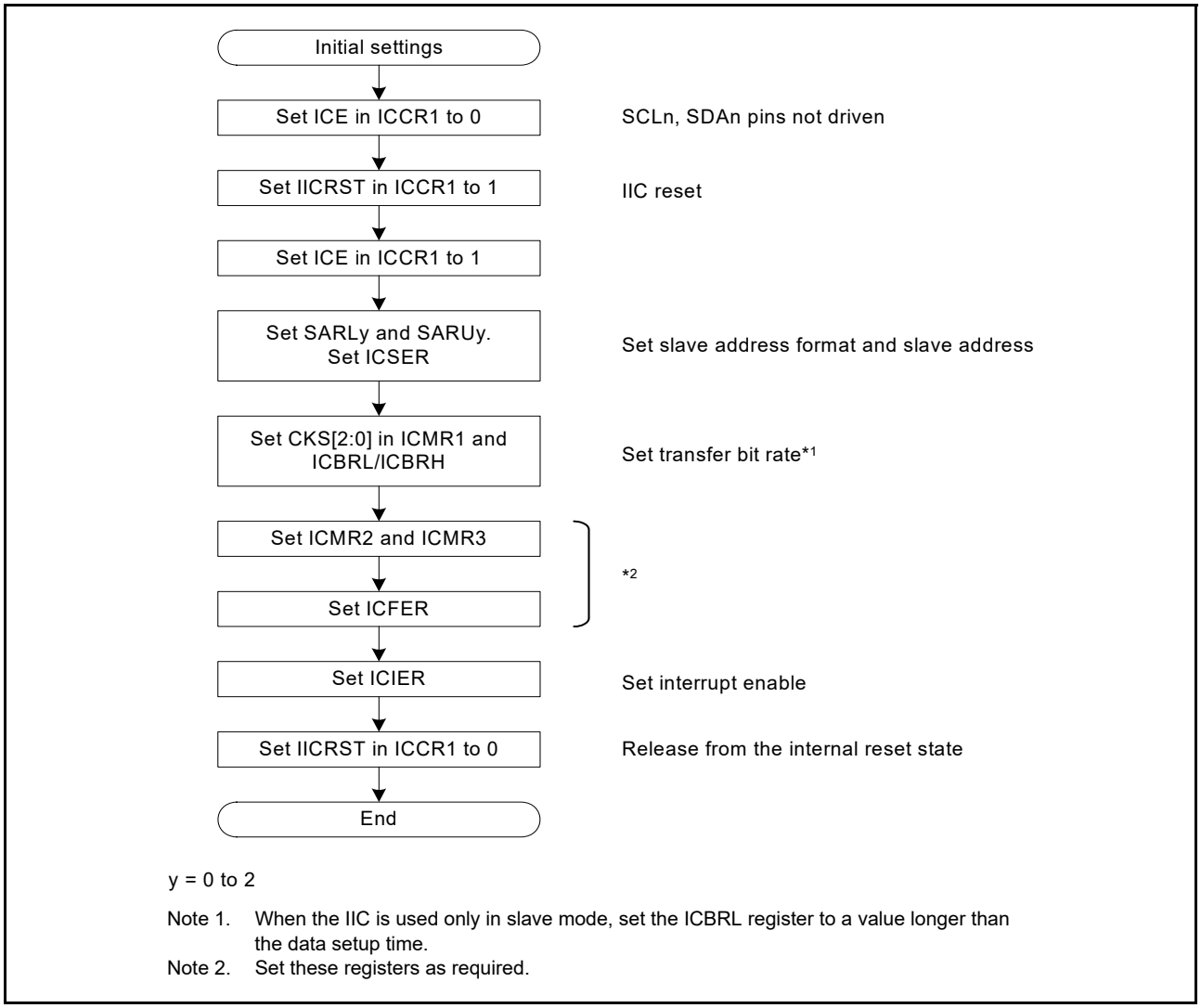


Figure 28.5 Example of IIC initialization flow

28.3.3 Master Transmit Operation

In a master transmit operation, the IIC outputs the SCL clock and transmitted data signals as the master device, and the slave device returns acknowledgments. Figure 28.6 shows an example of master transmission, and Figure 28.7 to Figure 28.9 show the operation timing in master transmission.

To set up and perform master transmission:

1. Initialize the IIC using the procedure in section 28.3.2, Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is open, then set the ST bit in ICCR2 to 1 (start condition 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 also automatically set to 1 in response to TRS = 1.
3. Check that the TDRE flag in ICSR2 is 1, 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 is again set to 1. After the byte containing the slave address and R/W# bit is transmitted, the value of the TRS bit automatically updates to select master transmit or master receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit is 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 two upper bits of the slave address, and W (= 0) to ICDRT as the first address transmission. For the second address transmission, write the 8 lower bits of the slave address to ICDRT.

4. Check that the TDRE flag in ICSR2 is 1, then write the transmit data to the ICDRT register. The IIC automatically holds the SCLn line low until the transmit data is ready or a stop condition is issued.
5. After all bytes of transmit data are written to the ICDRT register, wait until the value in the TEND flag in ICSR2 returns to 1, 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 28.11.3, Issuing a Stop Condition.
6. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, the IIC automatically sets the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 to 1.
7. Check that the ICSR2.STOP flag is 1, then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

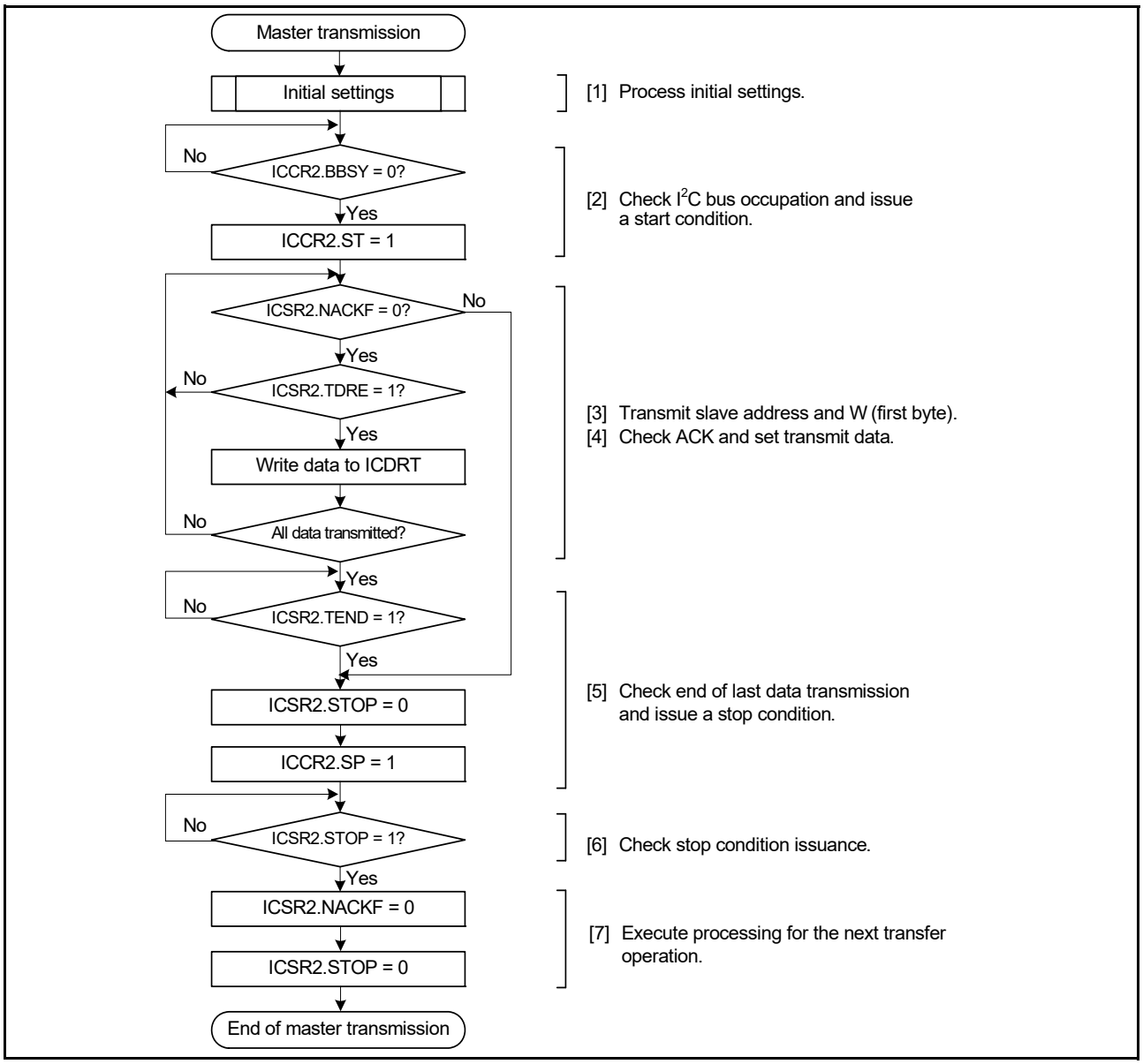


Figure 28.6 Example of master transmission flow

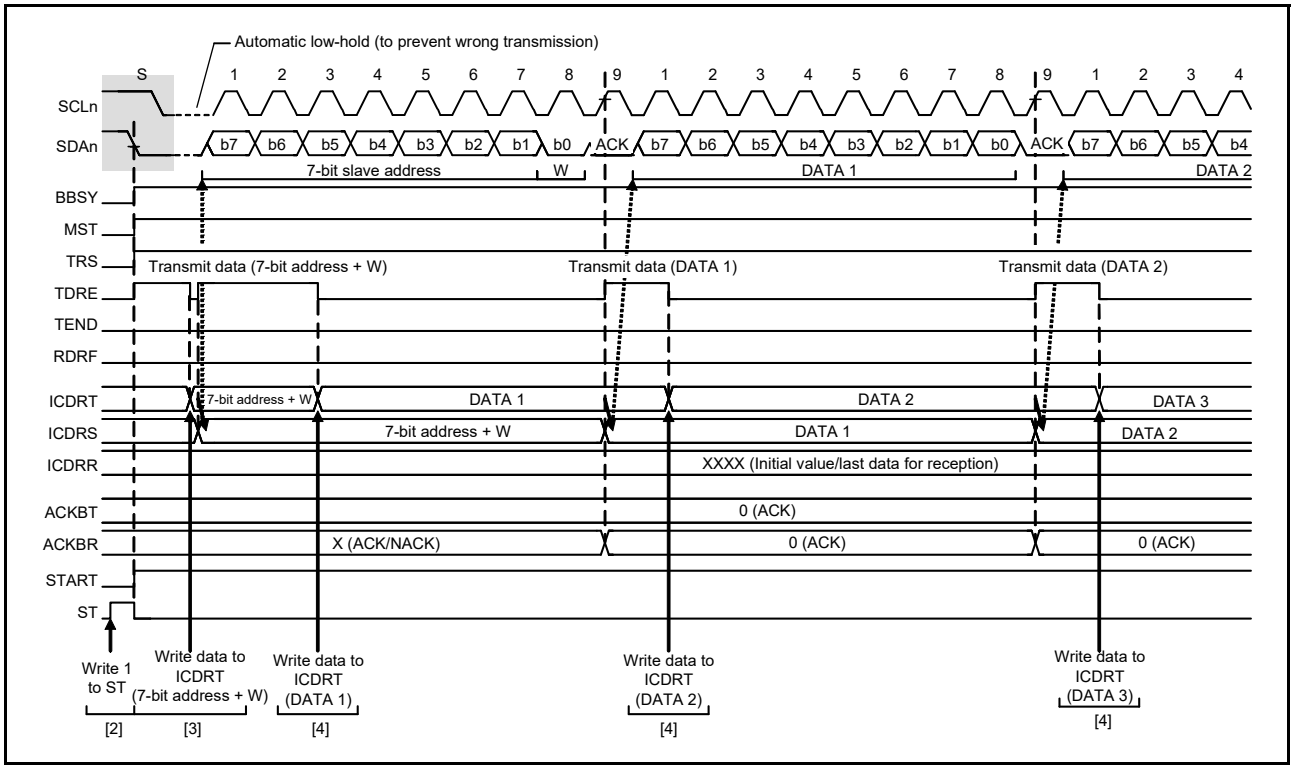


Figure 28.7 Master transmit operation timing (1) with 7-bit address format

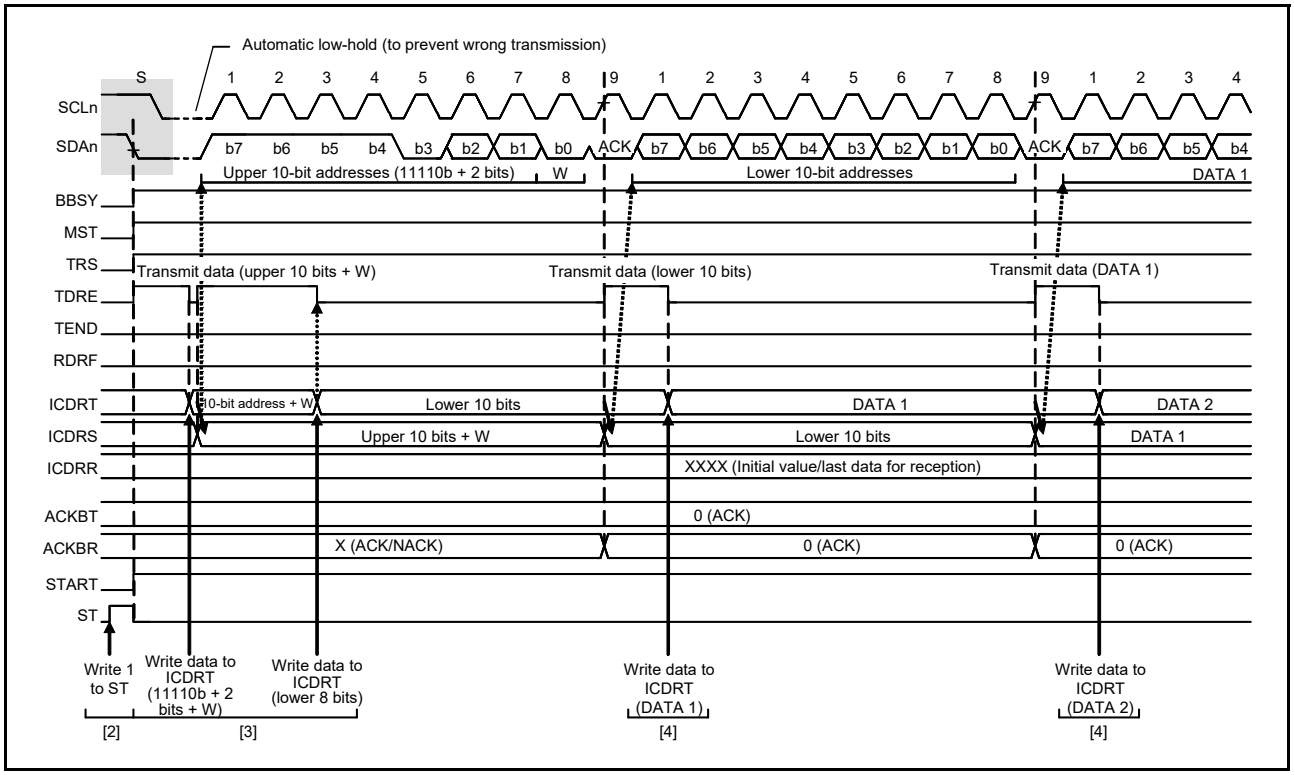


Figure 28.8 Master transmit operation timing (2) with 10-bit address format

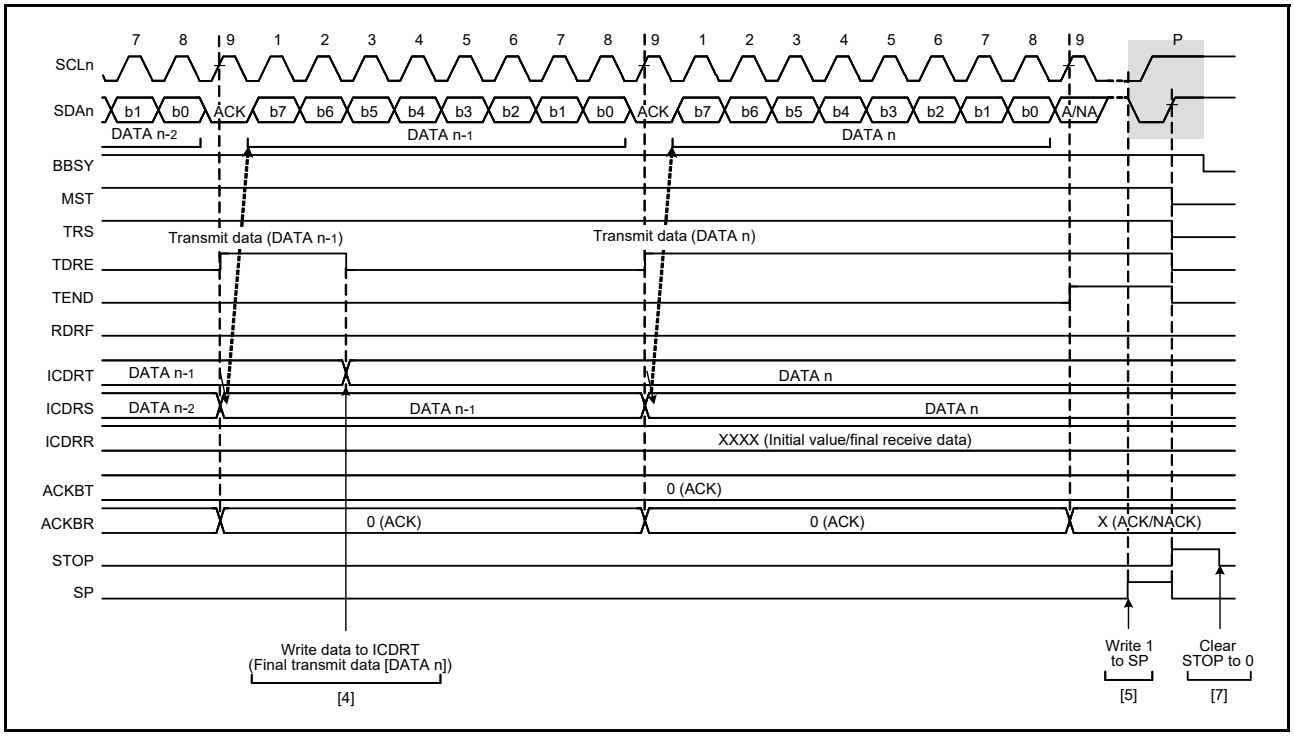


Figure 28.9 Master transmit operation timing (3)

28.3.4 Master Receive Operation

In a 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 part of the procedure is performed in master transmit mode, but the subsequent steps are performed in master receive mode.

Figure 28.10 and Figure 28.11 show examples of master reception (7-bit address format). Figure 28.12 to Figure 28.14 show the timing of operations in master reception.

To set up and perform master reception:

1. Initialize the IIC using the procedure in section 28.3.2, Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is open, then set the ST bit in ICCR2 to 1 (start condition request). On receiving the request, the IIC issues a start condition. When the IIC detects the start condition, the BBSY and START flags in ICSR2 automatically set to 1, and the ST bit automatically sets 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 requested by the ST bit has successfully completed, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 in response to TRS = 1.
3. Check that the TDRE flag in ICSR2 is 1, then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag is automatically set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag is again set to 1. When the byte containing the slave address and R/W# bit is transmitted, the value of the ICCR2.TRS bit automatically updates to select transmit or receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit is 1, the TRS bit is set to 0 on the rising edge of the 9th cycle of the SCL clock, placing the IIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag is automatically set to 1.
   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.
   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 (= 1) bit to place the IIC in master receive mode.

4. Dummy read the 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 the SCL clock, as selected in the RDRFS bit in ICMR3. Reading ICDRR produces the received data, and automatically sets the RDRF flag is 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 ICDRR, 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), 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 MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, detection of the stop condition sets the ICSR2.STOP flag to 1.
9. Check that the ICSR2.STOP flag is 1, then set the NACKF and STOP flags in ICSR2 to 0 for the next transfer operation.

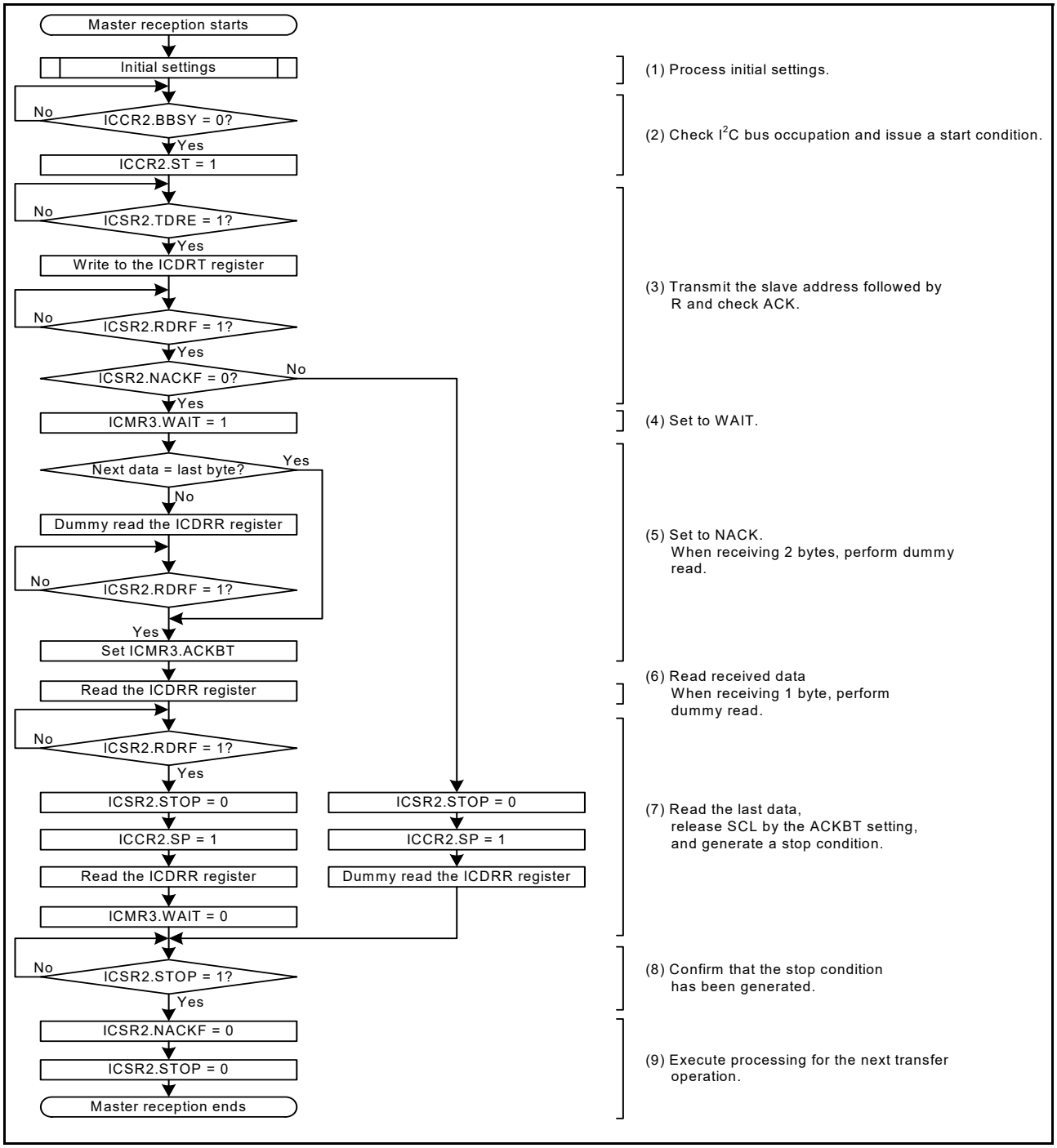


Figure 28.10 Example of master reception flow with 7-bit address format, and 1 or 2 bytes

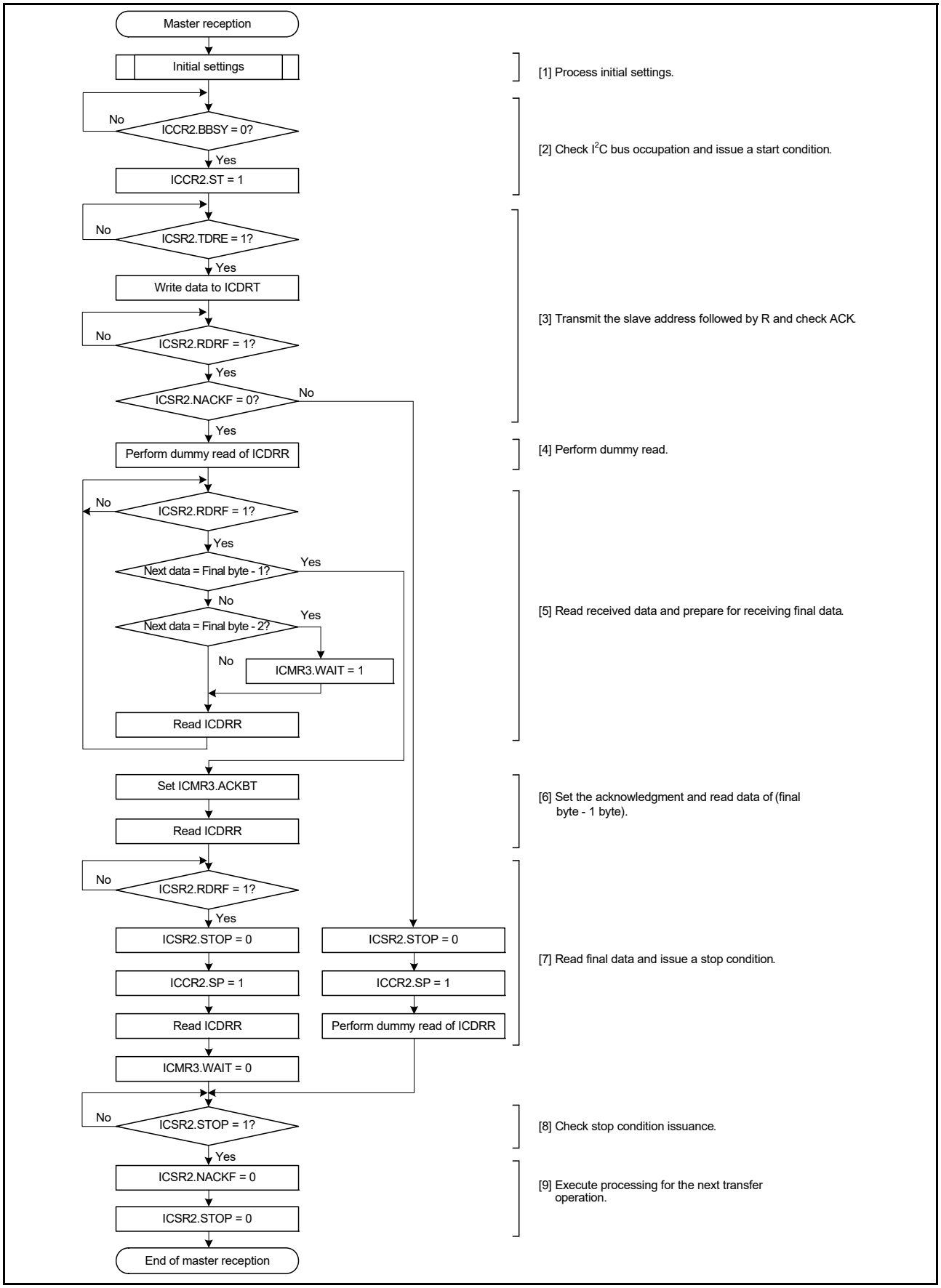


Figure 28.11 Example master reception flow with 7-bit address format, and 3 or more bytes

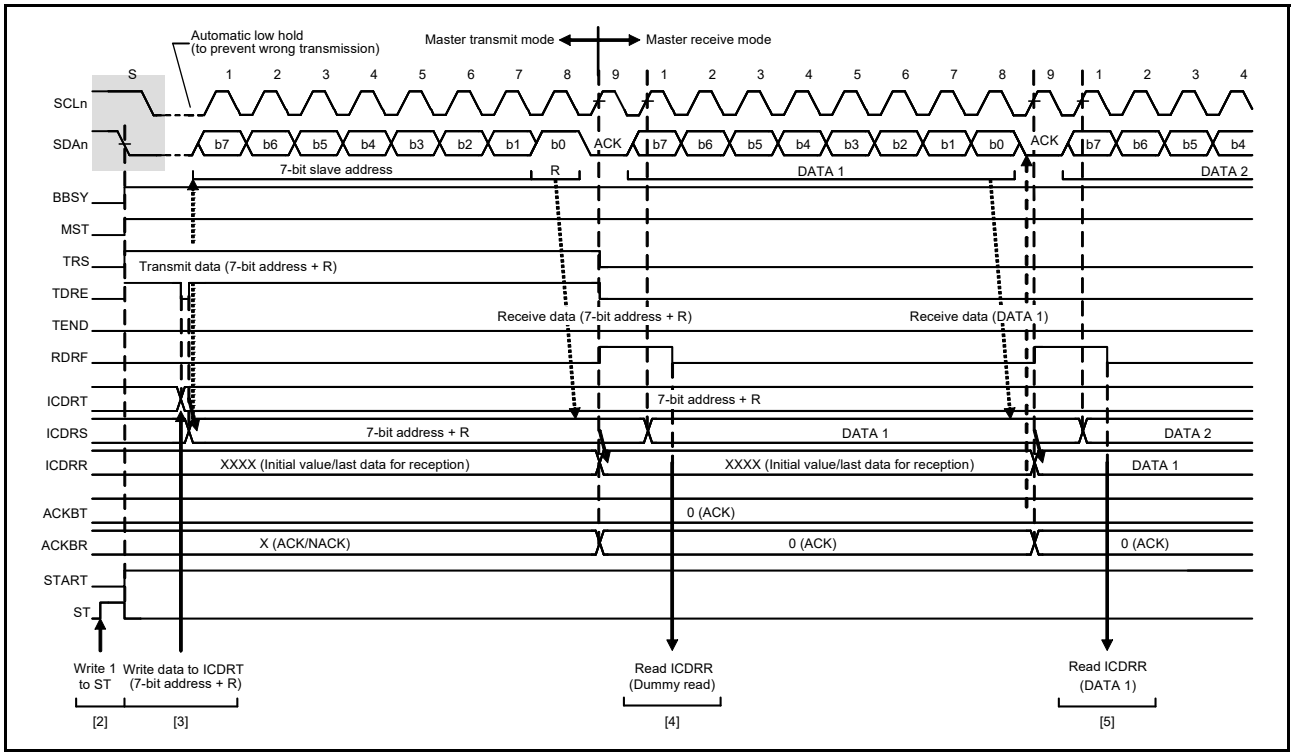


Figure 28.12 Master receive operation timing (1) with 7-bit address format, when RDRFS = 0

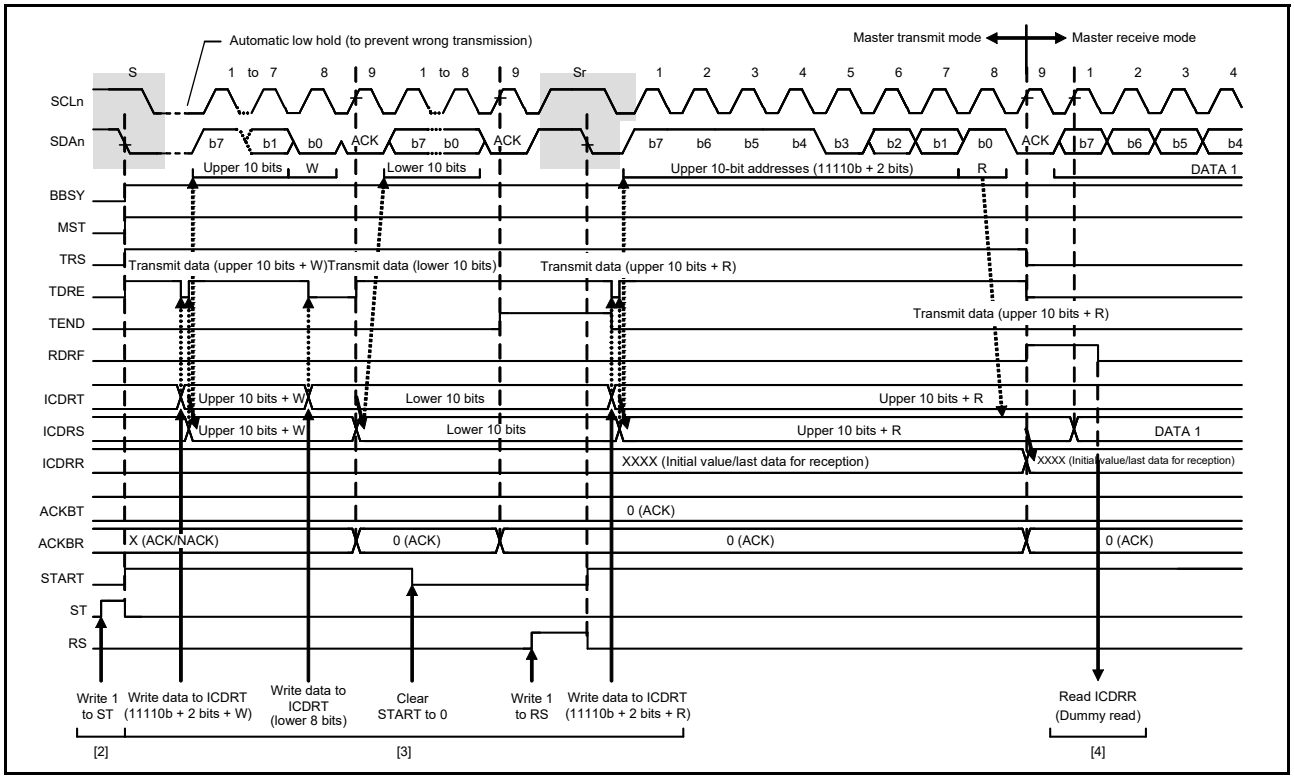


Figure 28.13 Master receive operation timing (2) with 10-bit address format, when RDRFS = 0

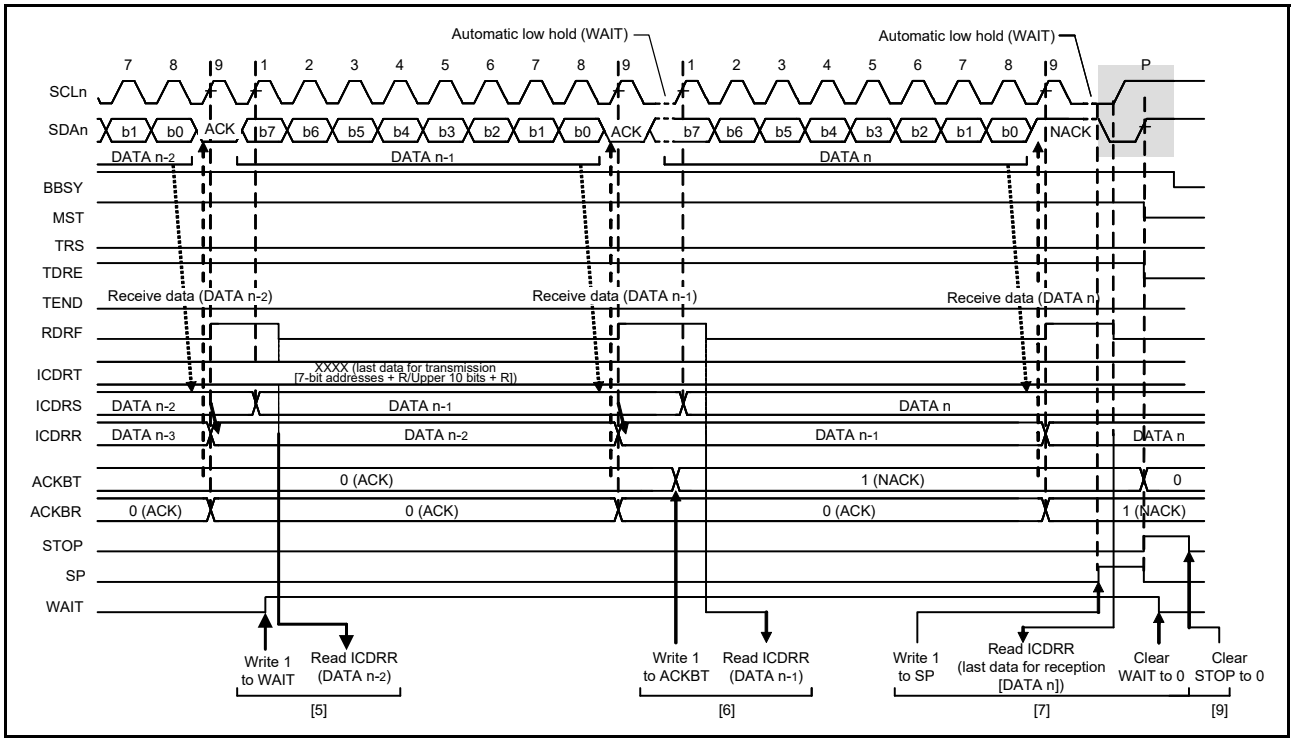


Figure 28.14 Master receive operation timing (3) when RDRFS = 0

28.3.5 Slave Transmit Operation

In a slave transmit operation, the master device outputs the SCL clock, the IIC transmits data as a slave device, and the master device returns the acknowledgments.

Figure 28.15 shows an example of slave transmission, and Figure 28.16 and Figure 28.17 show the operation timing in slave transmission.

To set up and perform slave transmission:

1. Initialize the IIC using the procedure in section 28.3.2, Initial Settings.
   After the initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated HOA, GCA, and AASy (y = 0 to 2) flags in ICSR1 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 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, then write the transmit data to the ICDRT register. If the IIC receives no acknowledge from the master device (receives an NACK signal) when 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 NACKF or TEND flag in ICSR2 is 1, the IIC drives the SCLn line low on the 9th falling edge of the SCL clock.
5. When the NACKF or TEND flag in ICSR2 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 bits (y = 0 to 2), the TDRE and TEND flags in ICSR2, and the ICCR2.TRS bit to 0, and enters slave receive mode.
7. Check that the ICSR2.STOP flag is 1, then set the NACKF and STOP flags in ICSR2 to 0 for the next transfer operation.

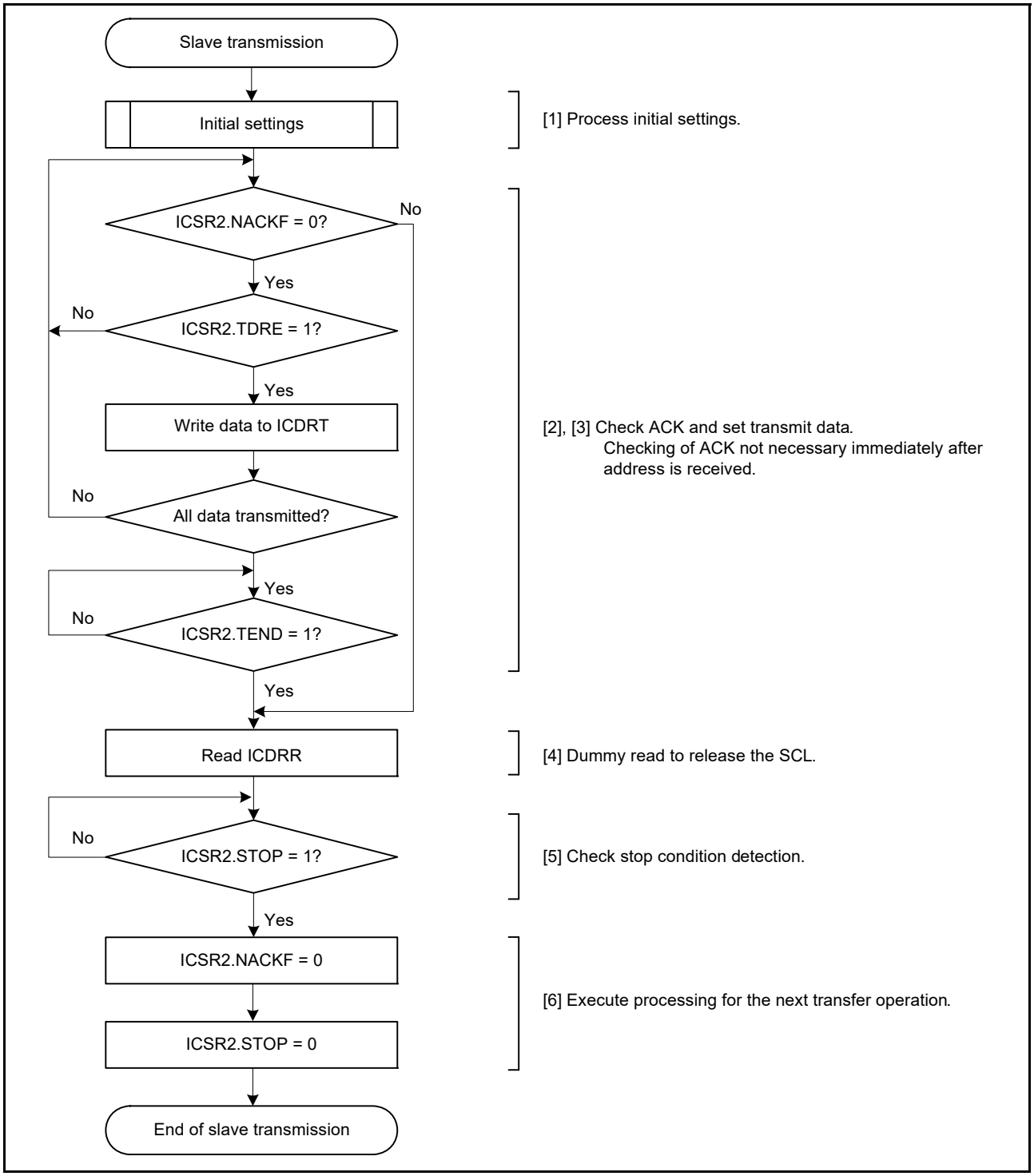


Figure 28.15 Example slave transmission flow

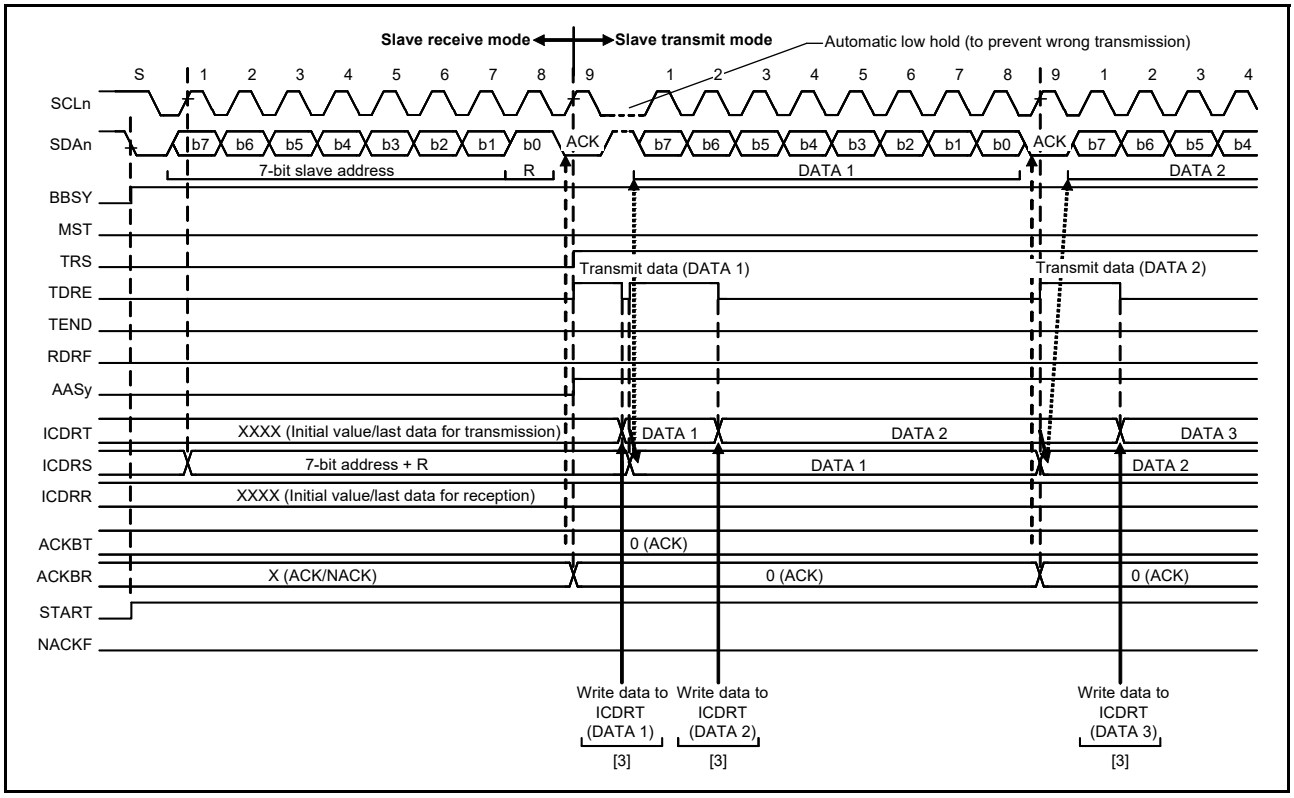


Figure 28.16 Slave transmit operation timing (1) with 7-bit address format

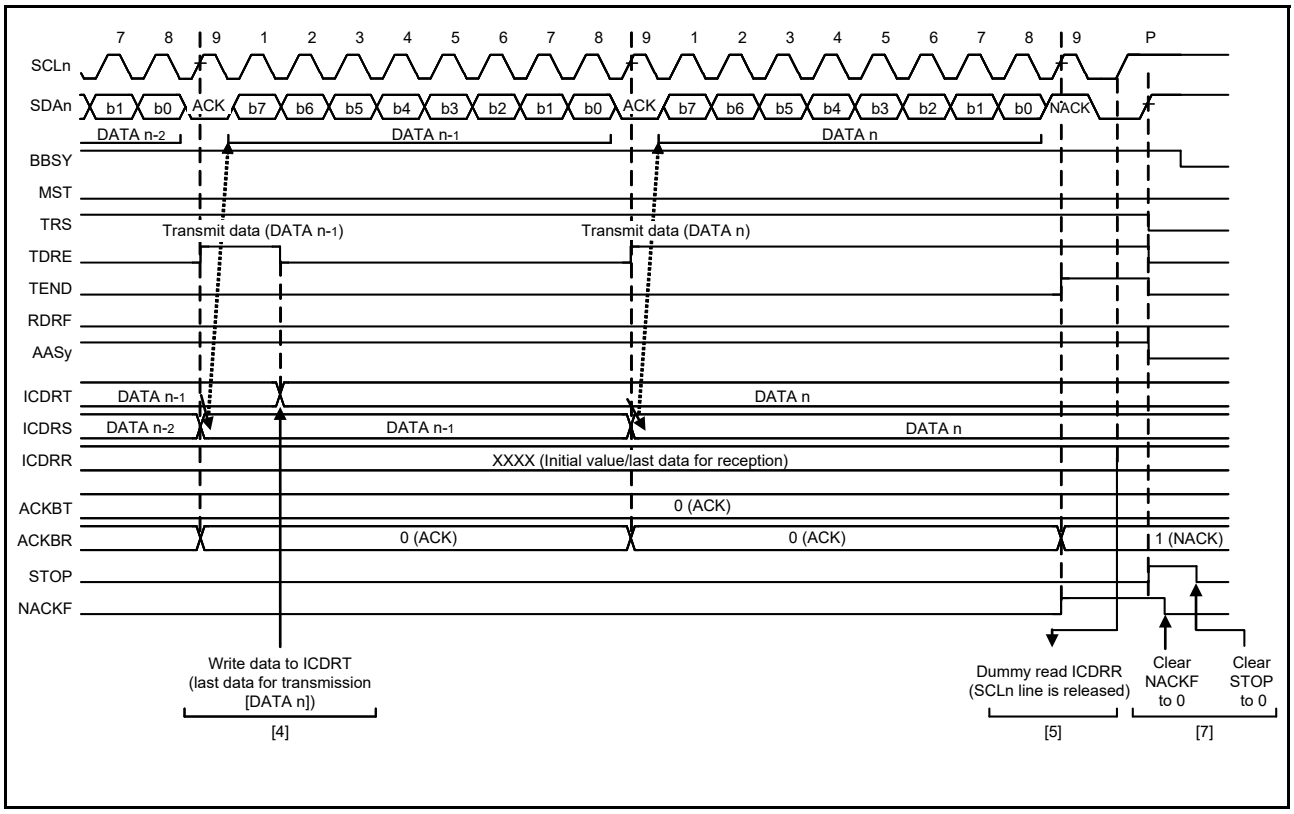


Figure 28.17 Slave transmit operation timing (2)

28.3.6 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 28.18 shows an example of slave reception. Figure 28.19 and Figure 28.20 show the operation timing in slave reception.

To set up and perform slave reception:

1. Initialize the IIC using the procedure in section 28.3.2, Initial Settings.
   After initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated HOA, GCA, and AASy (y = 0 to 2) flags in ICSR1 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, 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 a next byte is received while the RDRF flag is still set to 1, the IIC holds the SCLn line low from 1 SCL cycle before the point where RDRF must be set. In this case, reading ICDRR releases the SCLn line from being held low.
   When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read ICDRR until all the data is completely received.
5. On detecting the stop condition, the IIC automatically sets the HOA, GCA, and AASy (y = 0 to 2) flags in ICSR1 to 0.
6. Check that the ICSR2.STOP flag is 1, then set the ICSR2.STOP flag to 0 for the next transfer operation.

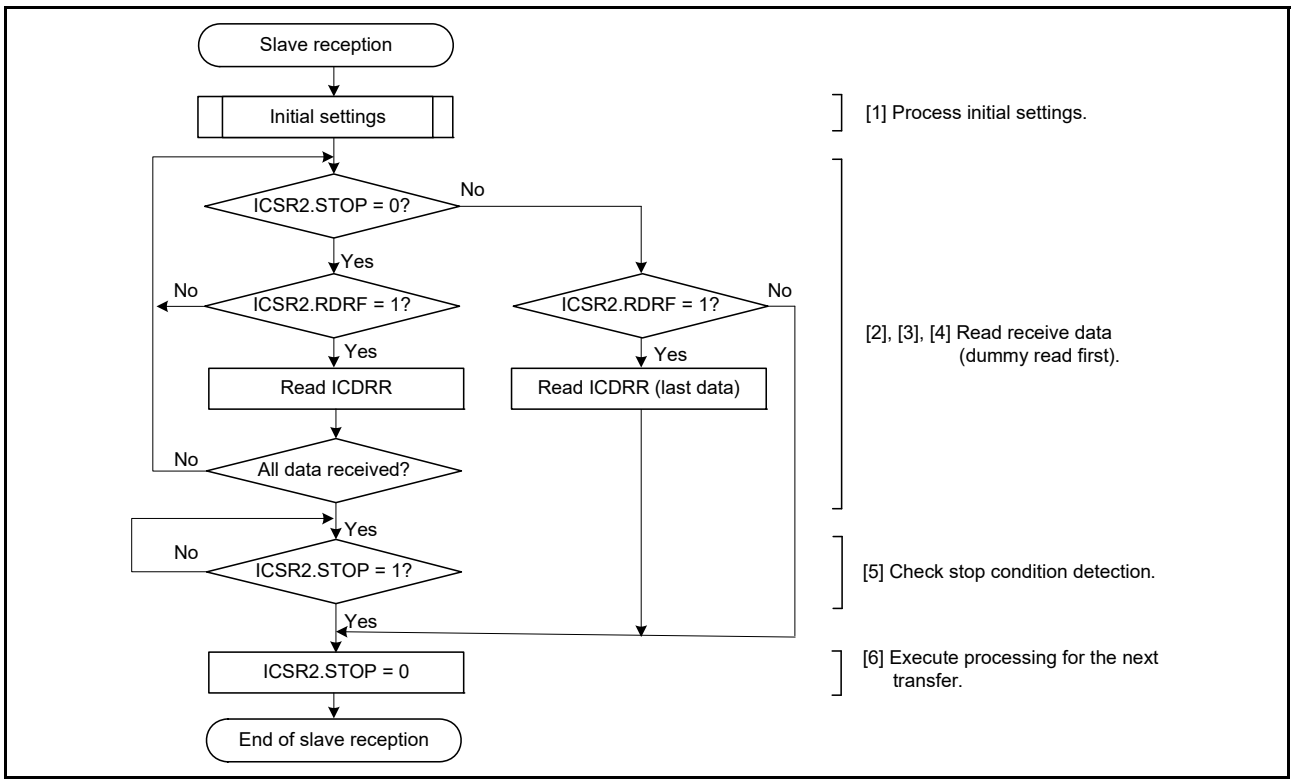


Figure 28.18 Example slave reception flow

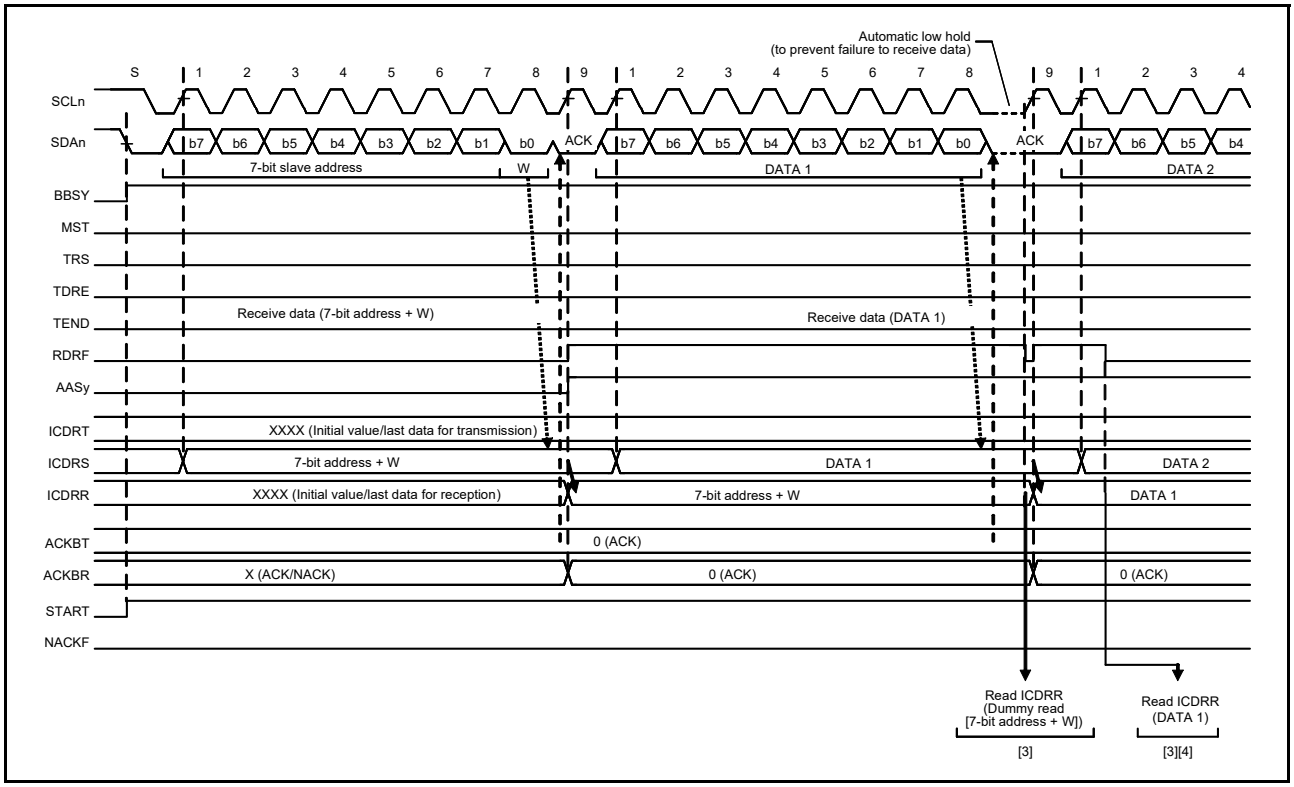


Figure 28.19 Slave receive operation timing (1) with 7-bit address format, when RDRFS = 0

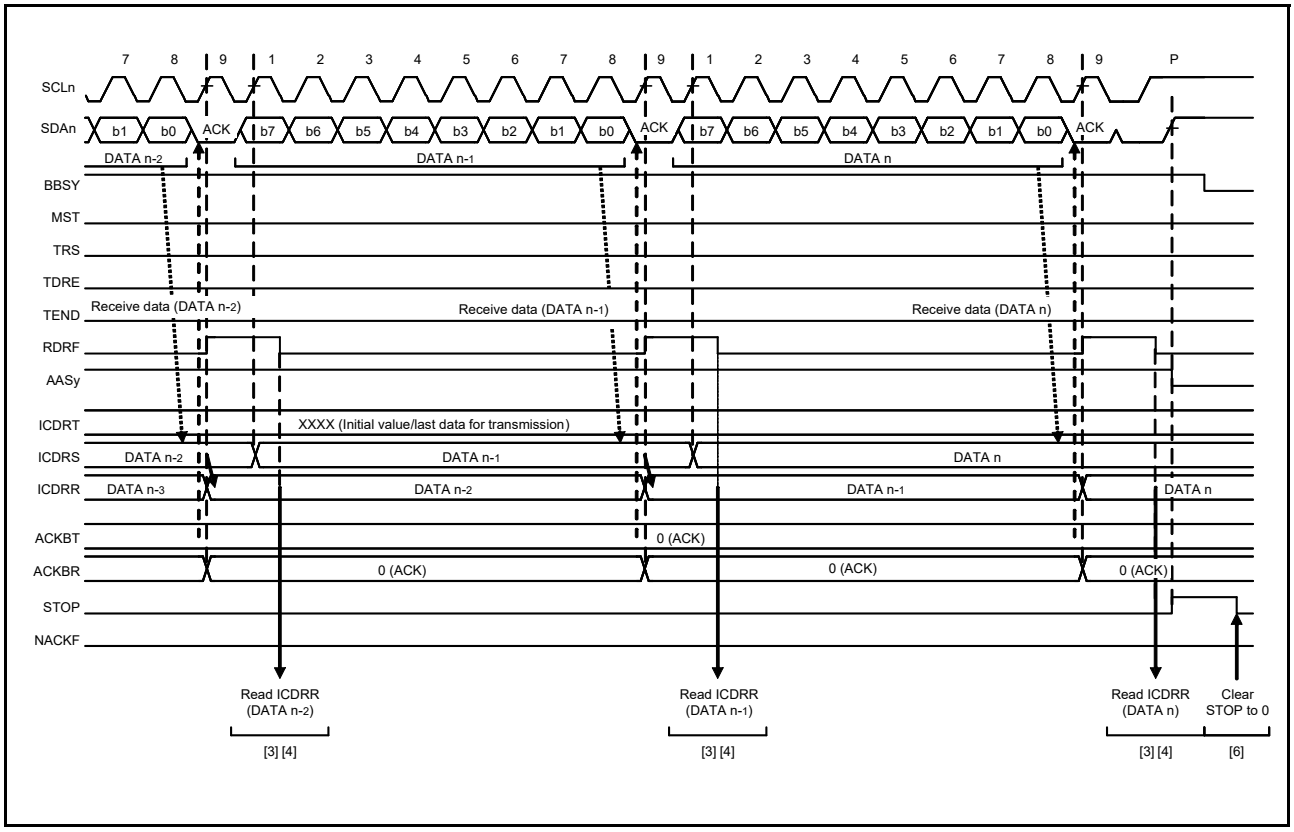


Figure 28.20 Slave receive operation timing (2) when RDRFS = 0

28.4 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 to release the line. If the low-level period of the SCL clock signal from the other master device is longer than the low-level period set in the IIC, the low-level period of the SCL signal is extended. When the low-level period for the other master device ends, the SCL signal rises because the SCLn line is released.

When the IIC finishes outputting the low-level period of the SCL clock, the SCLn line is released and the SCL clock rises. That is, when SCL signals from more than one master are contending, the high-level period of the SCL signal is synchronized with that of the clock with the narrower period, and the low-level period of the SCL signal is synchronized with that of the clock with the broader period. However, such synchronization of the SCL signal is only enabled when the SCLE bit in ICFER is set to 1.

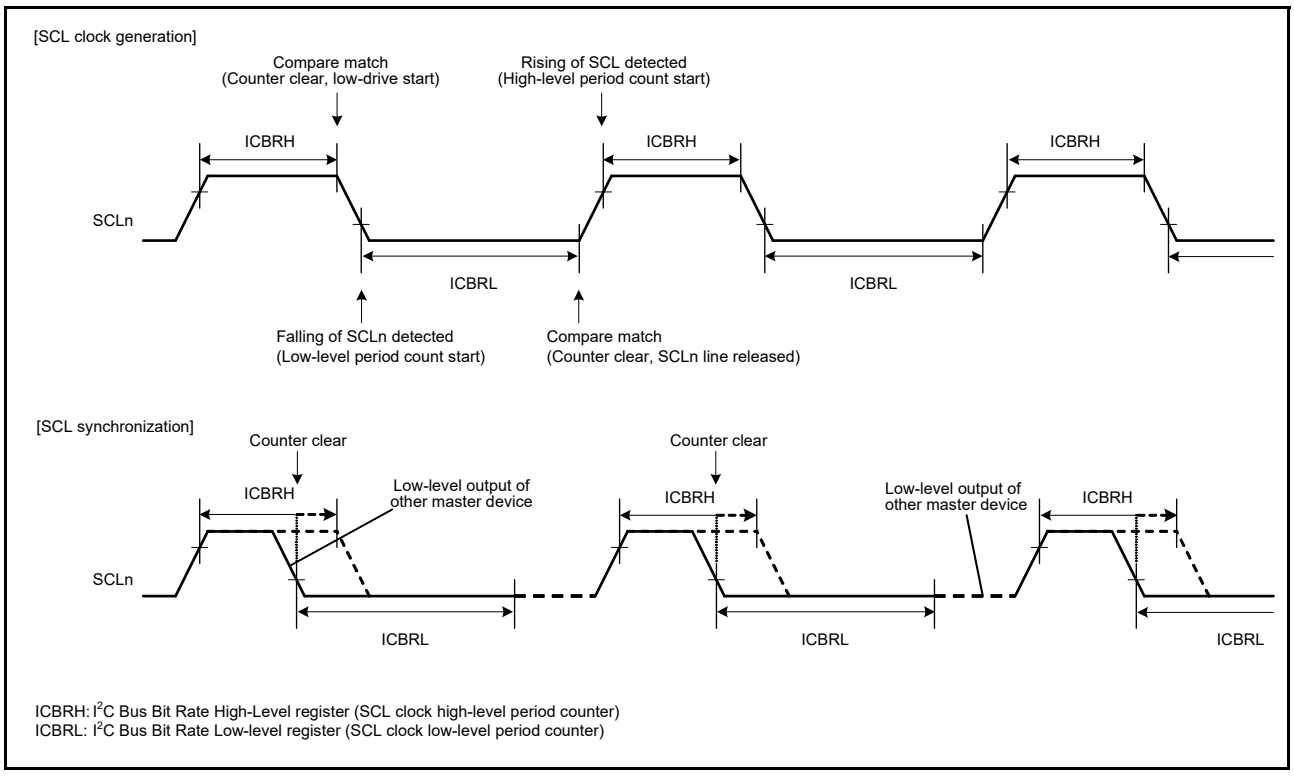


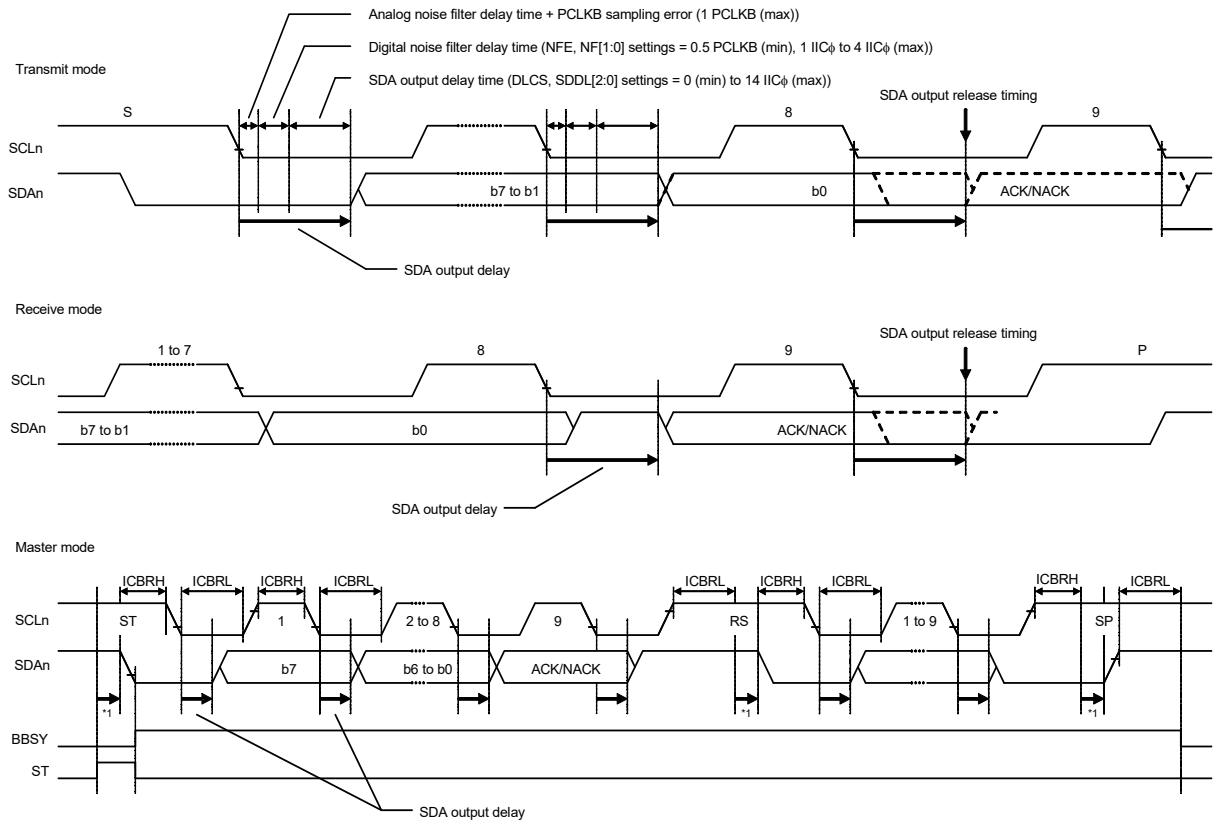
Figure 28.21 Generation and synchronization of SCL signal from the IIC

28.5 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 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.

While 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 SDAn line.



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 28.22 SDA output delay function

28.6 Digital Noise Filter Circuits

The internal circuitry sees the states of the SCLn and SDAn pins through analog and digital noise-filter circuits. Figure 28.23 shows a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the IIC consists of four flip-flop circuit stages connected in series, and a match-detection circuit.

The number of valid stages in the digital noise filter is selected in the NF[1:0] bits in ICMR3. The selected number of valid stages determines the noise-filtering capability as a period from 1 to 4 IICϕ cycles.

The input signal to the SCLn pin (or SDAn pin) is sampled on falling edges of the IICϕ signal. When the input signal level matches the output level of the number of valid flip-flop circuit stages as selected in the NF[1:0] bits in ICMR3, the signal level is seen in the subsequent stage. If the signal levels do not match, the previous value is saved.

If the ratio between the frequency of the internal operating clock (PCLKB) and the transfer rate is small, for instance, if data transfer at 400 kbps with PCLKB at 4 MHz, the characteristics of 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.

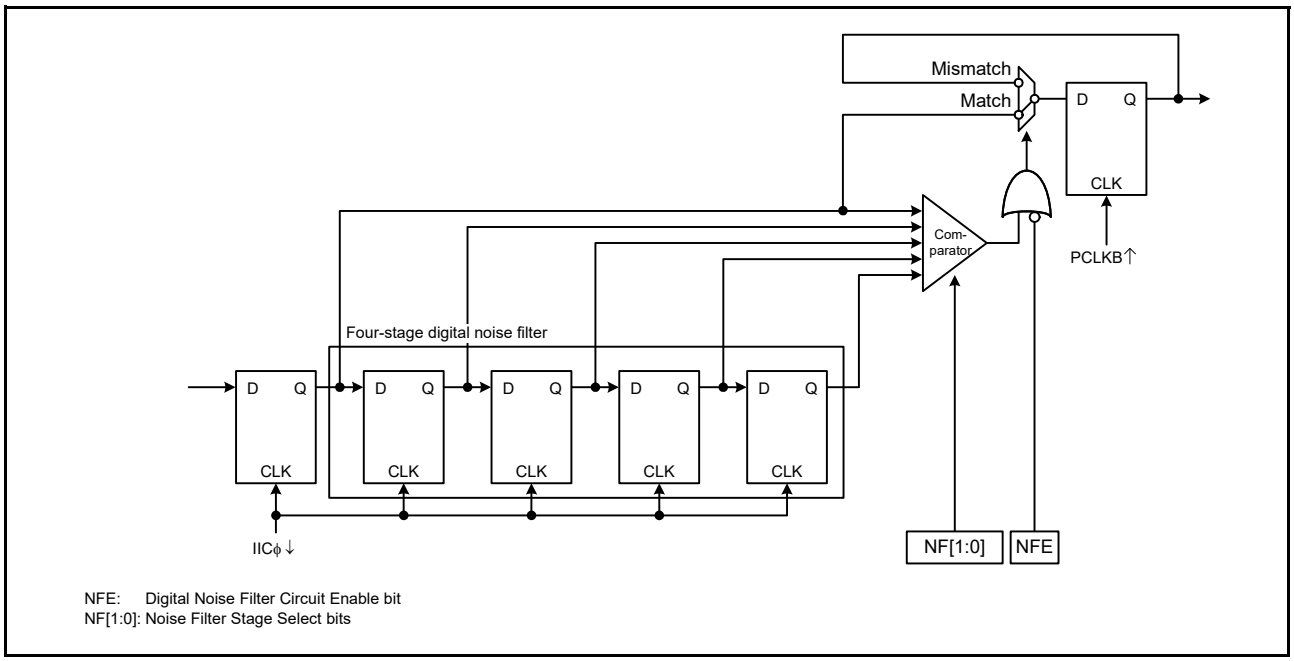


Figure 28.23 Digital noise-filter circuit block diagram

28.7 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.

28.7.1 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 (y = 0 to 2) flag in ICSR1 is set to 1 on the rising edge of the 9th SCL clock cycle, and the RDRF flag or 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 28.24 to Figure 28.26 show the AASy flag set timing in three cases.

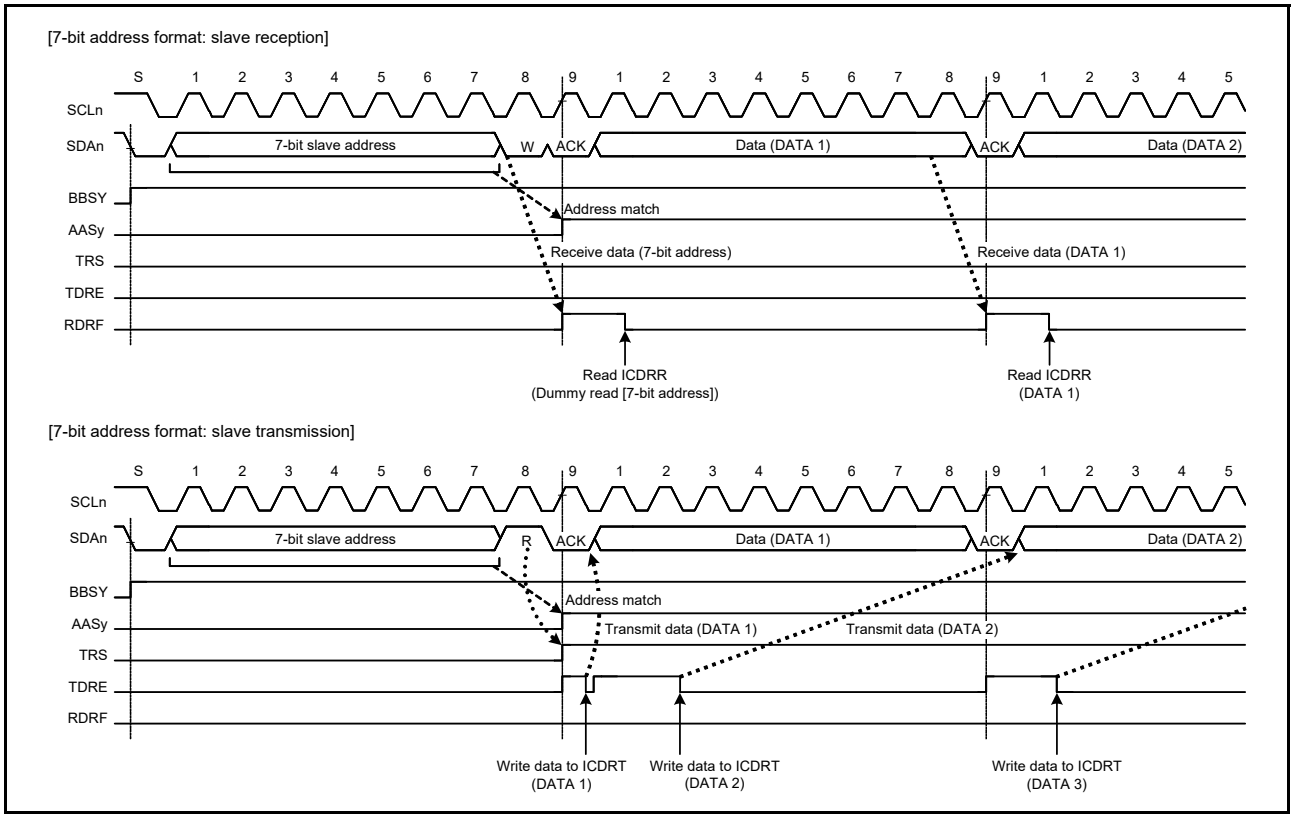


Figure 28.24 AASy flag set timing with 7-bit address format

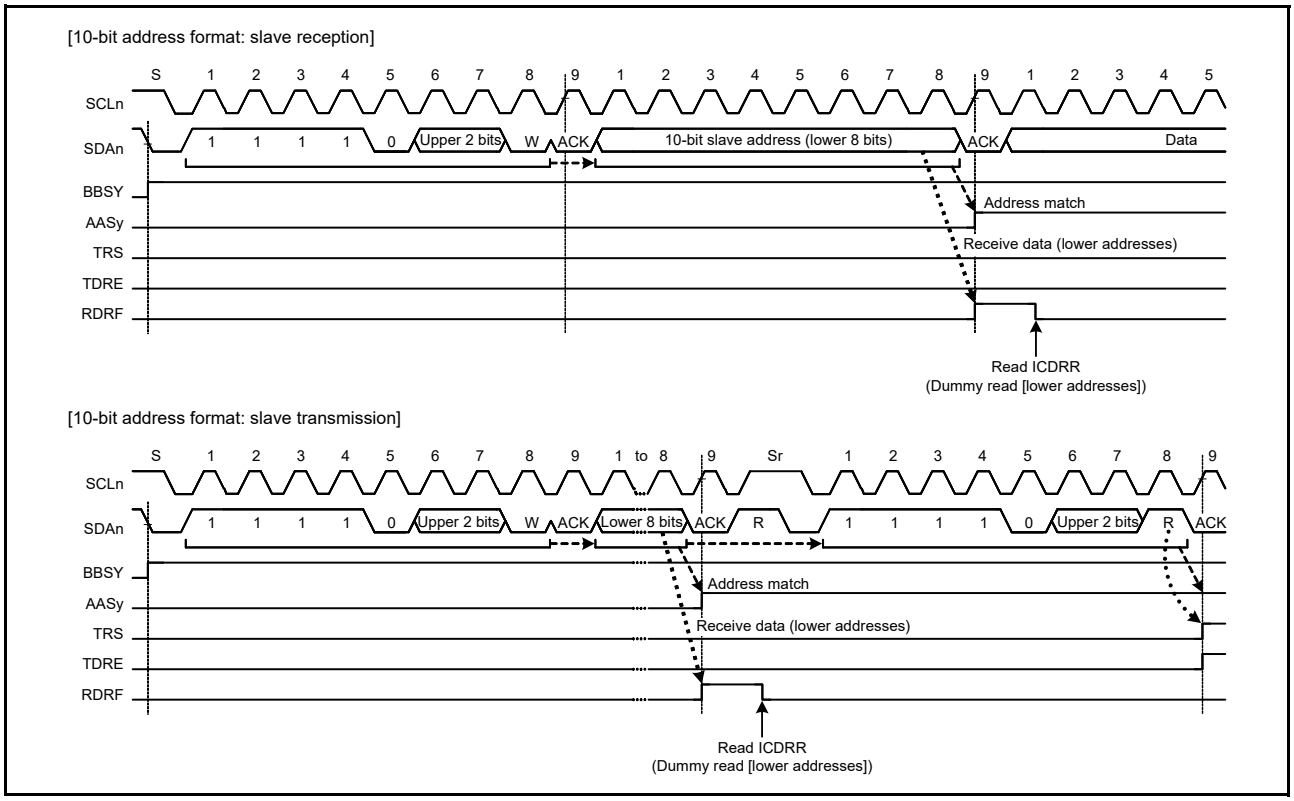


Figure 28.25 AASy flag set timing with 10-bit address format

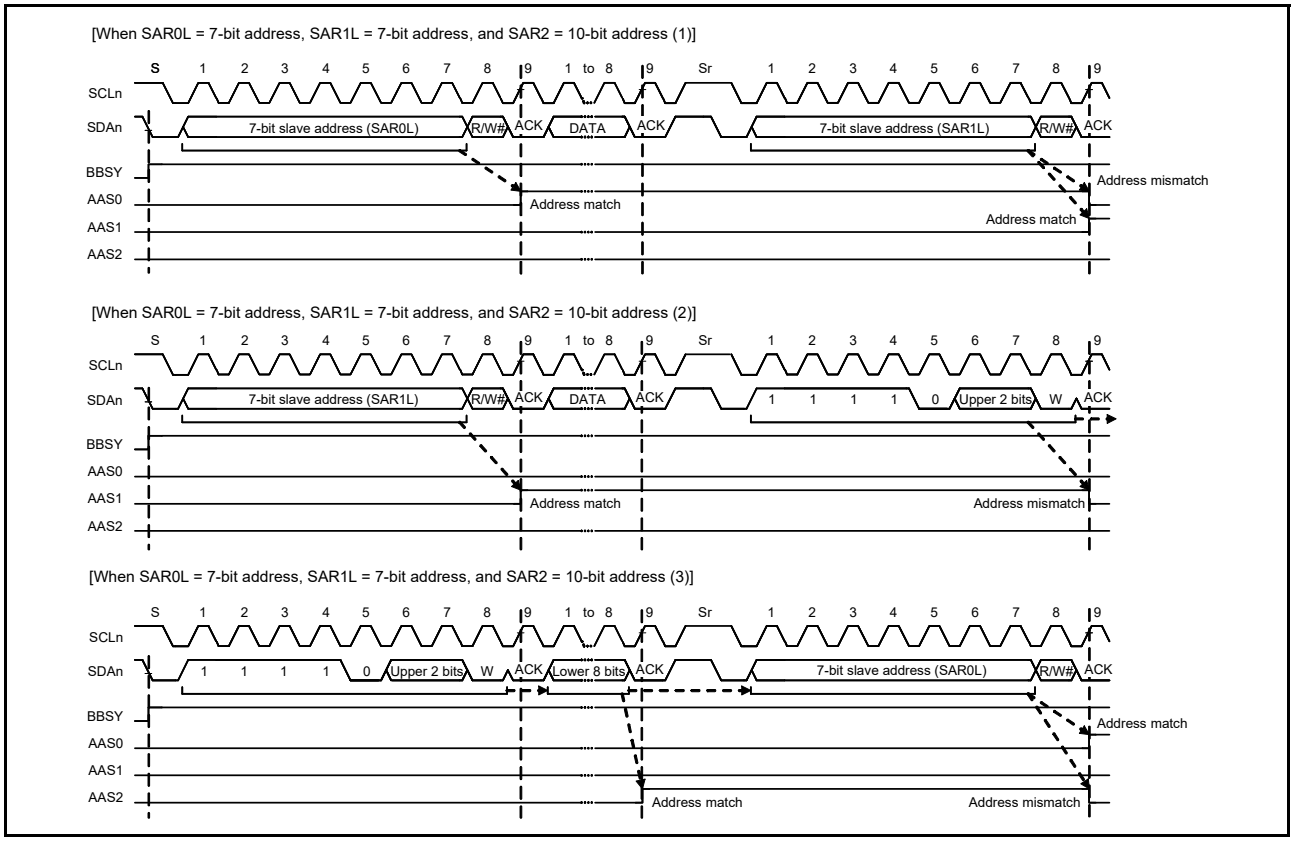


Figure 28.26 AASy flag set and clear timing with 7-bit and 10-bit address formats mixed

28.7.2 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 the SCL clock. This leads to the generation of a receive data full interrupt (IICn_RXI). The value of the GCA flag can be checked to confirm that the general call address was transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

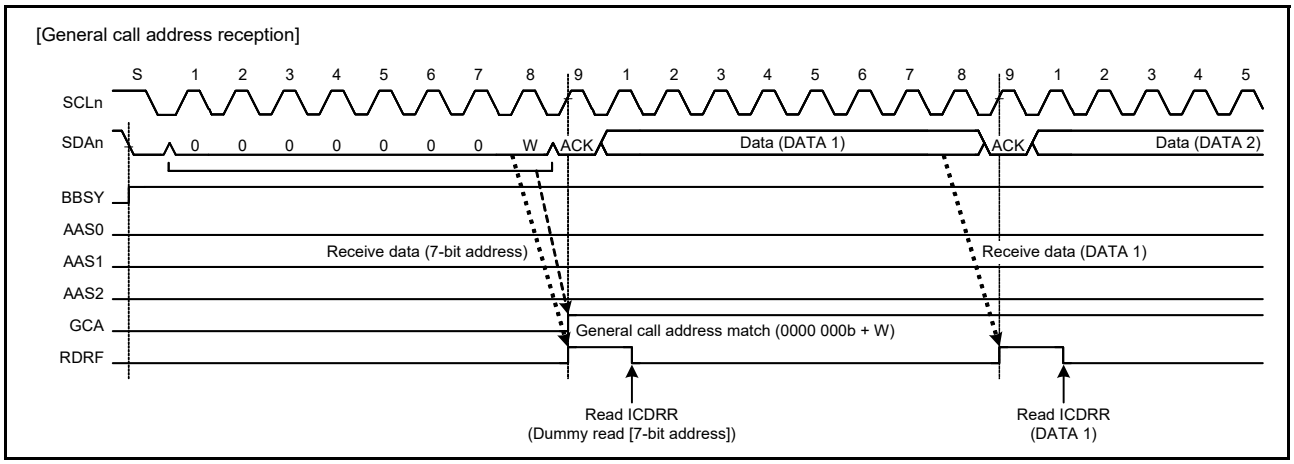


Figure 28.27 Timing of GCA flag setting during reception of general call address

28.7.3 Device ID Address Detection

The IIC module provides detection of device ID address in compliance with the I2C bus specification, revision 03. When the IIC receives 1111 100b as the first byte after a start or restart condition was issued with the DIDE bit in ICSER set to 1, it recognizes the address as a device ID, sets the DID flag in ICSR1 to 1 on the rising edge of the 9th SCL clock cycle when the subsequent R/W# bit is 0, then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the IIC sets the associated AASy (y = 0 to 2) flag 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 the detection of a start or restart condition matches the device ID address (1111 100b) and the R/W# bit is 0, the IIC sets the DID flag to 1 and compares the second and subsequent bytes with the slave address of the IIC. If the R/W# bit is 1, the DID flag holds the previous value and the IIC does not compare the second and subsequent bytes. Therefore, the reception of a device ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Additionally, prepare the device ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device ID field as normal transmit data. For details on the information that must be included in device ID fields, contact NXP Semiconductors.

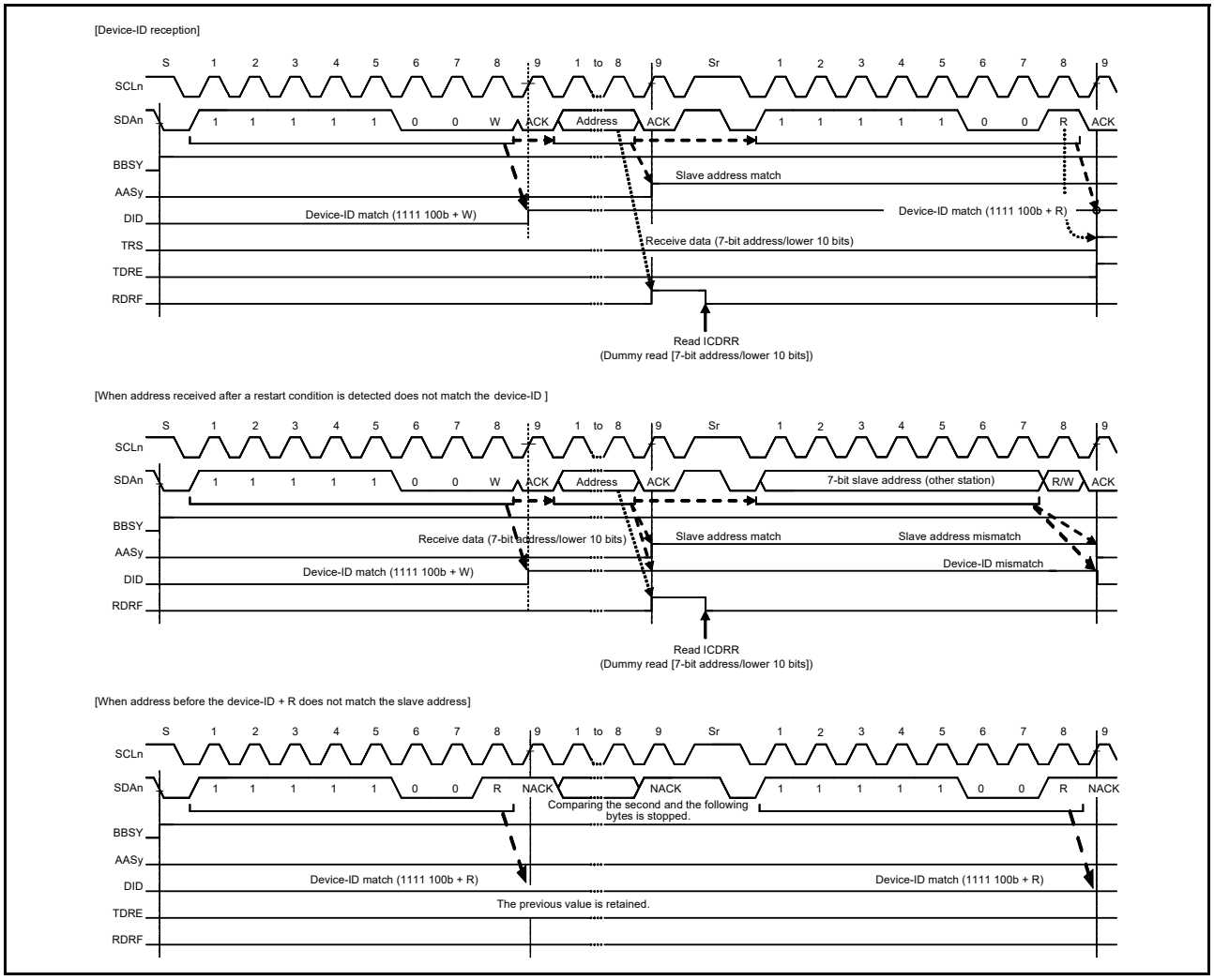


Figure 28.28 AASy/DID flag set and clear timing during reception of device ID

28.7.4 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 (MST = 0 and TRS = 0 in ICCR2).

When the IIC detects the host address, the HOA flag in ICSR1 is set to 1 on the rising edge of the 9th SCL clock cycle. At the same time, the RDRF flag in ICSR2 is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (IICn_RXI) to be generated. The HOA flag indicates that the host address was sent from another device.

If the bit following the host address (0001 000b) is a read bit (R/W# = 1), the IIC can also detect the host address. After the host address is detected, the IIC operates in the same manner as in normal slave operation.

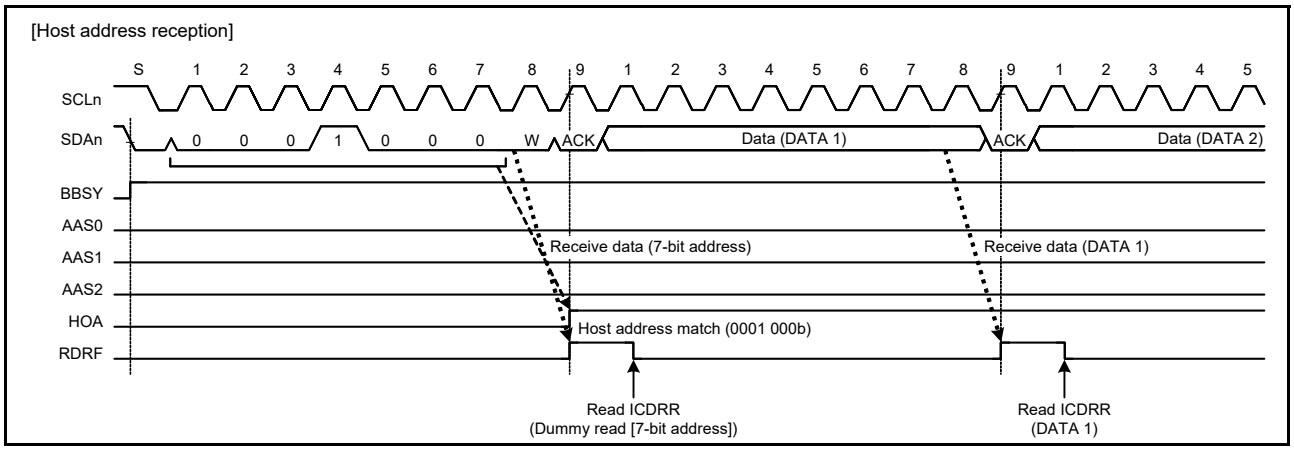


Figure 28.29 HOA flag set timing during reception of host address

28.8 Wakeup Function

The IIC provides a wakeup function that causes the MCU to transition from Software Standby mode or Snooze mode to normal operation. The wakeup function enables the reception of data when the system clock (PCLKB) is stopped, and generates a wakeup interrupt signal on a match of the slave address of the received data. This wakeup interrupt signal triggers the return to normal operation. After the wakeup interrupt occurs, switch the IIC to PCLKB synchronous operation so that communication can continue.

The wakeup function has four operation modes:

- Normal wakeup mode 1
- Normal wakeup mode 2
- Command recovery mode
- EEP response mode.

Table 28.9 describes the behavior in these modes.

Table 28.9 Wakeup operation modes

Operation mode	ACK response timing	ACK response before wakeup to PCLKB synchronous operation	SCL state during wakeup to PCLKB synchronous operation
Normal wakeup mode 1	Before wakeup to PCLKB synchronous operation*1	ACK	Fixed low
Normal wakeup mode 2	After wakeup to PCLKB synchronous operation*2	Before wakeup: no response (NACK level retained) After wakeup: ACK response	Fixed low
Command recovery mode	Before wakeup to PCLKB synchronous operation*1	ACK	Open
EEP response mode	Before recovery to PCLKB synchronous operation*1	NACK	Open

Note 1. Switching timing from PCLKB asynchronous operation to PCLKB synchronous operation is the falling edge of the 9th clock of the SCL.
Note 2. Switching timing from PCLKB asynchronous operation to PCLKB synchronous operation is the falling edge of the 8th clock of the SCL.

The following can be selected as wakeup interrupt sources:

- Host address detection (valid when ICSER.HOAE = 1)
- General call address detection (valid when ICSER.GCAE = 1)
- Slave address 0 *1 detection (valid when ICSER.SAR0E = 1)
- Slave address 1 *1 detection (valid when ICSER.SAR1E = 1)
- Slave address 2 *1 detection (valid when ICSER.SAR2E = 1).

Note 1. Only 7-bit address can be set. Set the FS bit in SARUy (y = 0 to 2) to 0.

Precautions on the use of the wakeup function

- Do not change the content of the IIC registers except the WUSEN bit in ICWUR2 while the WUASYF flag in ICWUR2 is 1 (during PCLKB asynchronous operation)
- Set ICWUR.WUE and ICWUR.WUIE to 1, and ICCR2.MST and ICCR2.TRS to 0 (slave reception mode) before switching to PCLKB asynchronous mode
- The device ID and the 10-bit slave address cannot be selected for the wakeup interrupt source. Set the DIDE bit in ICSER and FS bit in SARUy (y = 0 to 2) to 0.
- Set bits TIE, TEIE, RIE, NAKIE, SPIE, STIE, ALIE, and TMOIE in the ICIER register to 0 (interrupt disabled) before switching to the asynchronous operation
- When the wakeup function is enabled, do not use the timeout function (ICWUR.WUE = 1)
- Even when a wakeup interrupt is generated during PCLKB asynchronous operation (when ICWUR2.WUASYF = 1), if the slave addresses match in PCLKB synchronous mode (ICWUR2.WUASYF = 0), the wakeup interrupt does not occur and the WUF flag is not set
- If the timing of writing 0 to the ICWUR2.WUSEN bit and the timing of detecting a start condition conflict, the IIC might start the next reception in PCLKB synchronous operation mode. In this case, ICWUR2.WUASYF flag becomes 1 (switch to PCLKB asynchronous mode) when data communication is complete, a stop condition is detected, and detection of a wakeup event starts.
- After writing 0 to the WUSEN bit in ICWUR2, do not change registers relate to the IIC operation mode setting (ICMR3, ICSER, and SARLy) until the mode is switched to PCLKB asynchronous operation from PCLKB synchronous operation (while the ICWUR2.WUASYF flag is 1). If the register value changes during this period by an interrupt handling or another factor, the IIC might malfunction before switching to the asynchronous operation.

28.8.1 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 28.32 shows an operation example, and Figure 28.30 shows the detailed timing.

Note 1. Between the 9th clock cycle and 1st clock cycle during wakeup, WAIT = 1 is invalid.

If the transition from Software Standby mode or Snooze mode is triggered by an interrupt other than a wakeup interrupt, for example the IRQn, the WUF flag is not set to 1. Figure 28.31 shows an operation example.

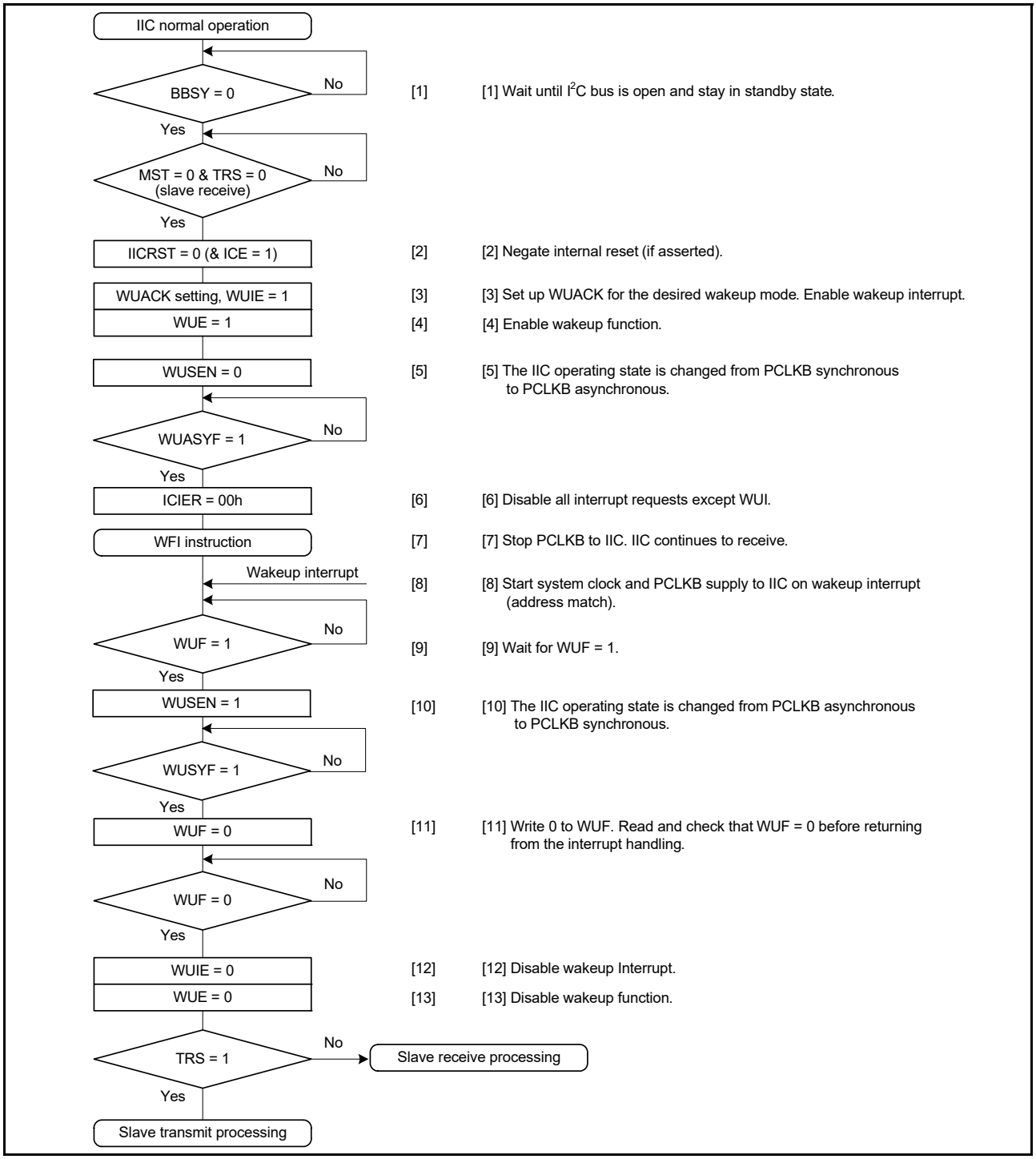


Figure 28.30 Example operation of normal wakeup mode 1 when wakeup is triggered by a wakeup interrupt on match of the slave address

Note: See Precautions on the use of the wakeup function.

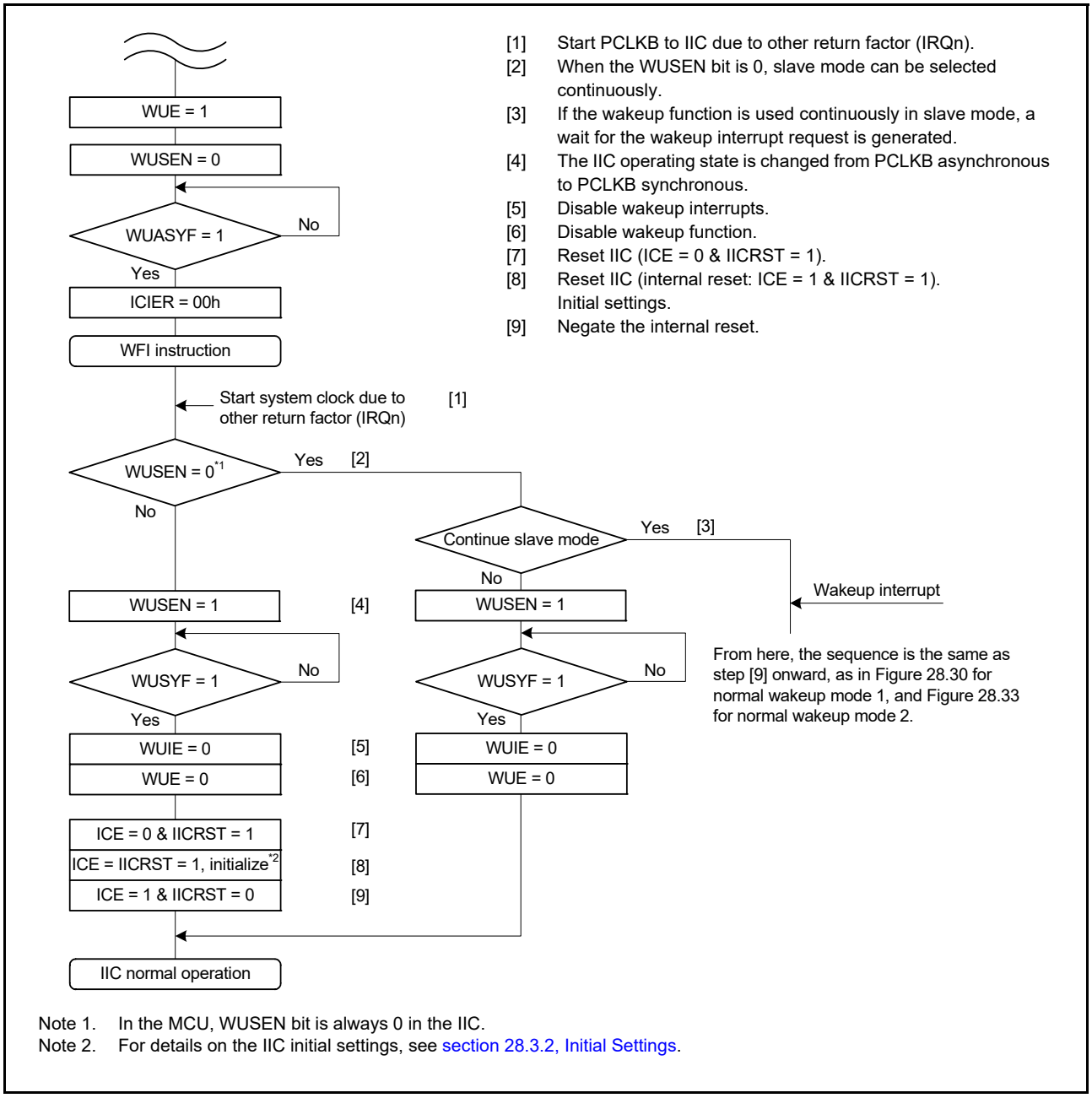


Figure 28.31 Example operation of normal wakeup modes 1 and 2 when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example, the IRQn

Note: For details on the IIC initial settings, see section 28.3.2, Initial Settings.

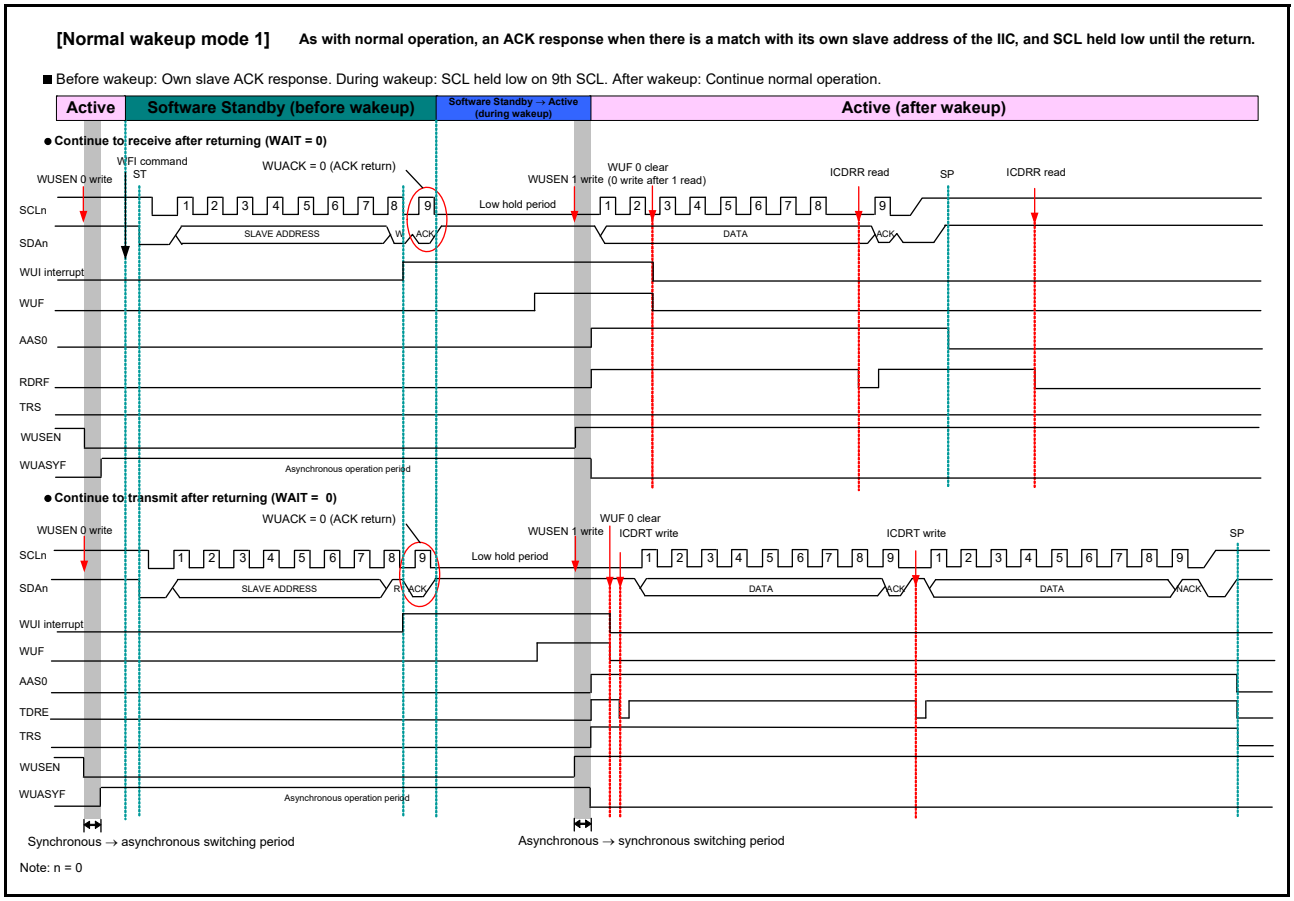


Figure 28.32 Timing of normal wakeup mode 1

28.8.2 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 a match of the slave address initiates the transition to normal operation as follows:

Before wakeup: No response to data received with its own slave address until the end of the 8th SCL cycle.
During wakeup: SCL line held low during the 8th and 9th clock cycles.
After wakeup: ACK returns on the 9th clock cycle of SCL, and normal operation continues.

If the slave address does not match, the SCL line is not held low after the 8th SCL clock cycle, and the slave operation continues.

Figure 28.33 shows an example operation, and Figure 28.34 shows the detailed timing.

If the transaction from Software Standby mode or Snooze mode is triggered by an interrupt other than a wakeup interrupt, for example IRQn. WUF is not set to 1. Figure 28.31shows an operation example.

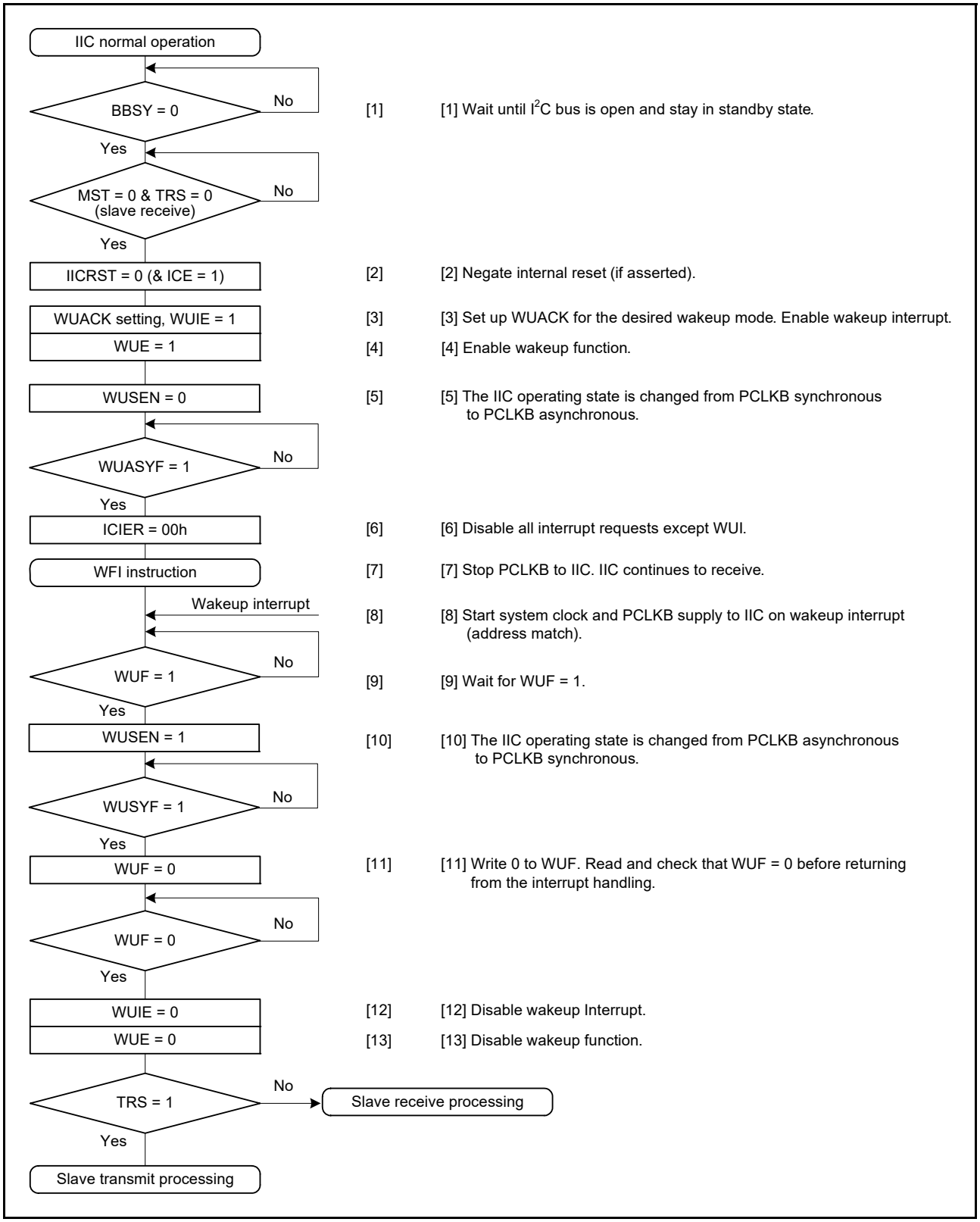


Figure 28.33 Example operation of normal wakeup mode 2 when wakeup is triggered by a wakeup interrupt on match of the slave address

Note: See Precautions on the use of the wakeup function.

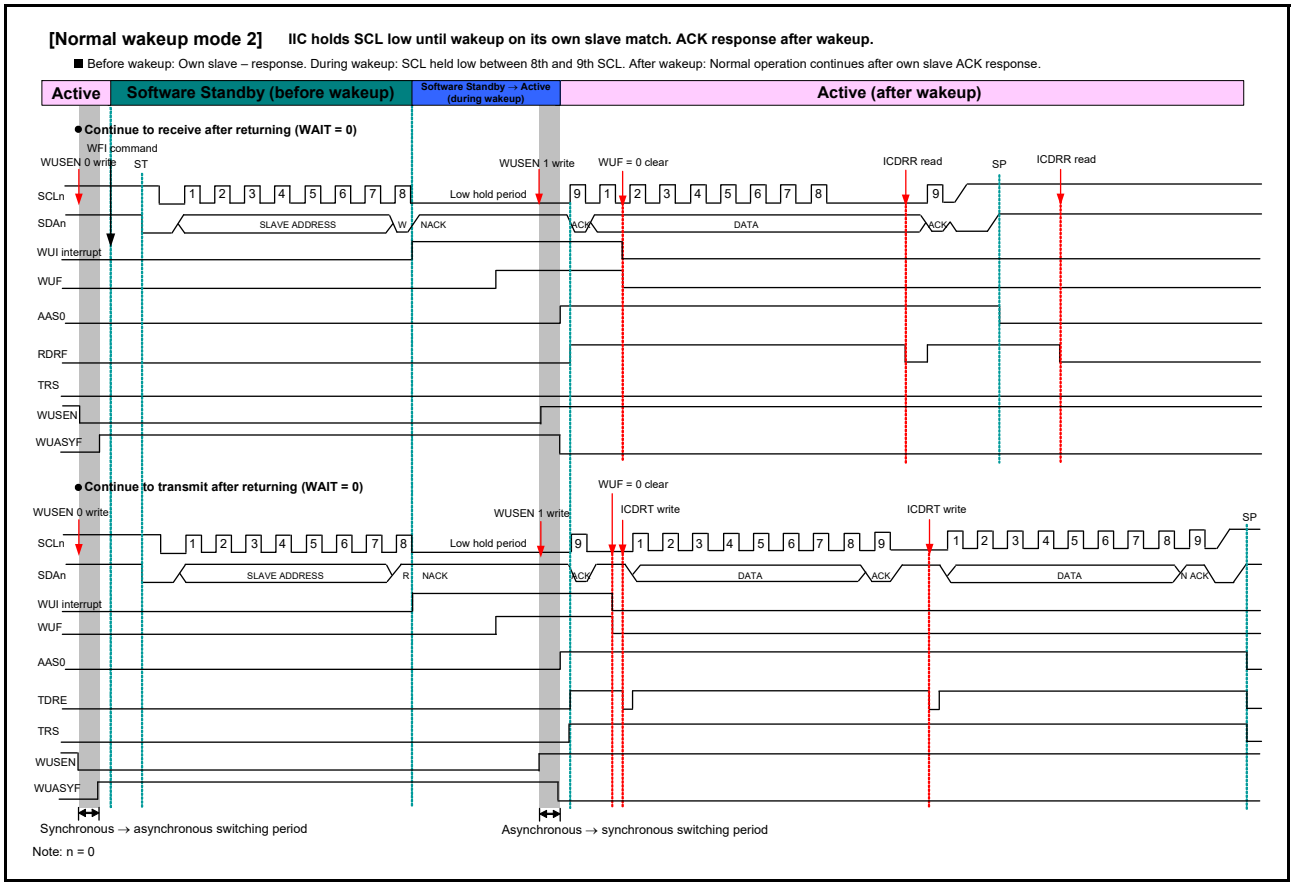


Figure 28.34 Timing of normal wakeup mode 2

28.8.3 Command Recovery Mode and EEP Response Mode (Special Wakeup Modes)

This section describes the behavior, timing, and an example operation in the command recovery and EEP response modes.

In the command recovery and EEP response modes, the SCLn line is not held low during the wakeup period (after the rise of the 9th clock cycle of SCL). Therefore, other IIC devices can use the I2C bus during this period.

A wakeup interrupt triggered by the match of the slave address initiates the transition to normal operation as follows:

Before wakeup: In response to data received with its own slave address, the IIC returns ACK (command recovery mode) or NACK (EEP response mode).
During wakeup: The SCLn line is not held low.
After wakeup: Normal operation continues after IIC initialization.

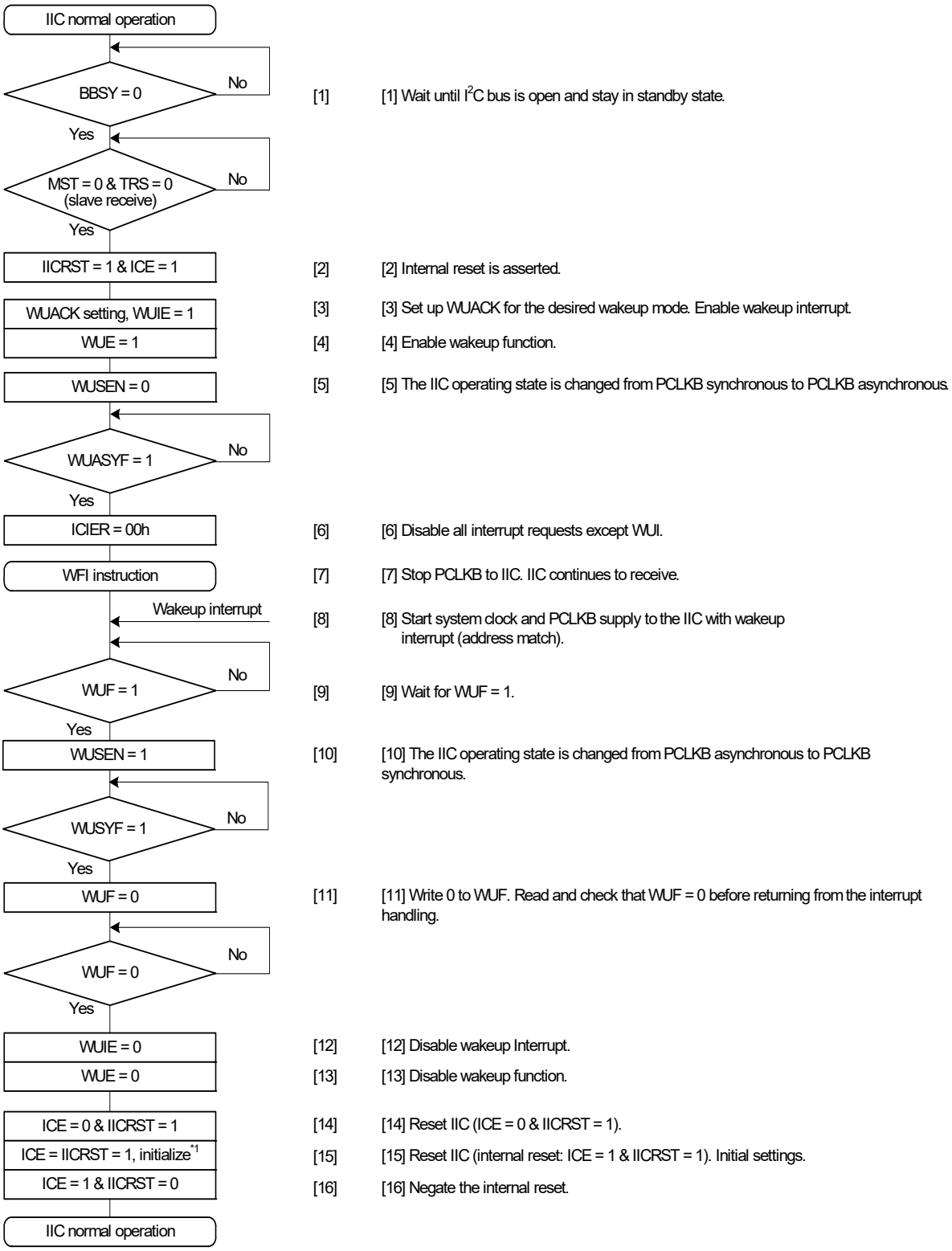
If the slave address does not match, the slave operation continues.

Note: Because the SCLn 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.

Figure 28.35 shows an example operation in command recovery and EEP response modes. Figure 28.37 shows the detailed timing.

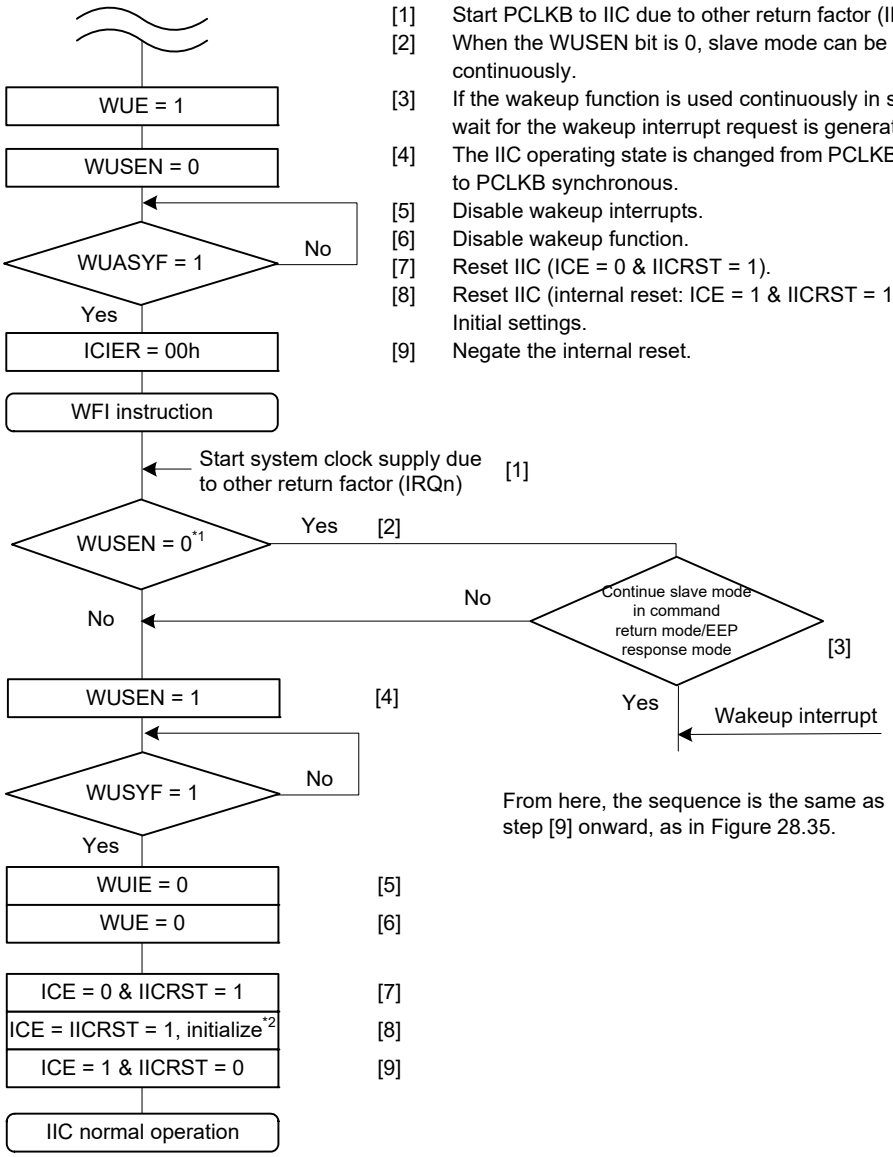
If the transaction from Software Standby mode or Snooze mode is triggered by an interrupt other than a wakeup interrupt, for example the IRQn, the WUF flag is not set to 1. Figure 28.36 shows an operation example.



Note 1. For details on the IIC initial settings, see section 28.3.2, Initial Settings.

Figure 28.35 Example operation of command recovery and EEP response modes when wakeup is triggered by a wakeup interrupt on match of the slave address

Note: See Precautions on the use of the wakeup function.



[1] Start PCLKB to IIC due to other return factor (IRQn).
[2] When the WUSEN bit is 0, slave mode can be selected continuously.
[3] If the wakeup function is used continuously in slave mode, a wait for the wakeup interrupt request is generated.
[4] The IIC operating state is changed from PCLKB asynchronous to PCLKB synchronous.
[5] Disable wakeup interrupts.
[6] Disable wakeup function.
[7] Reset IIC (ICE = 0 & IICRST = 1).
[8] Reset IIC (internal reset: ICE = 1 & IICRST = 1). Initial settings.
[9] Negate the internal reset.

Note 1. In the MCU, WUSEN bit is always 0 in the IIC.
Note 2. For details on the IIC initial settings, see section 28.3.2, Initial Settings.

Figure 28.36 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

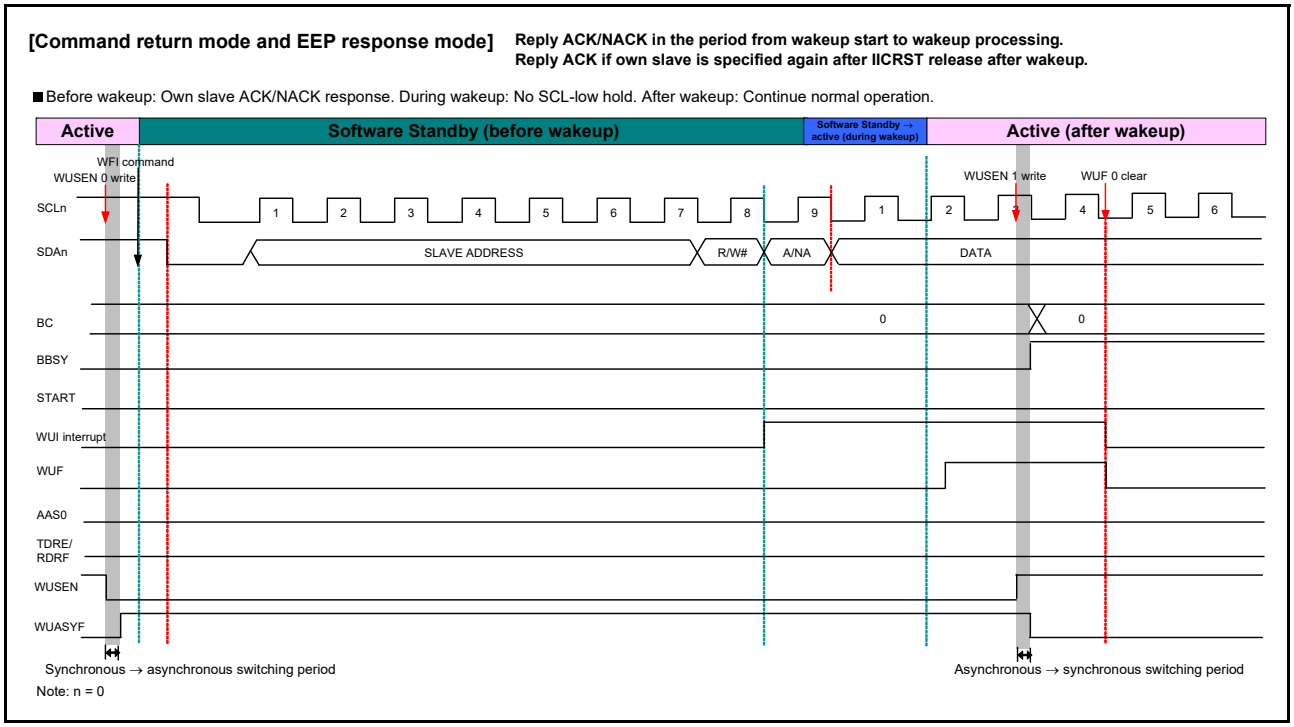


Figure 28.37 Timing of command recovery and EEP response modes

28.9 Automatic Low-Hold Function for SCL

28.9.1 Function to Prevent Wrong Transmission of Transmit Data

If the I2C Bus Shift Register (ICDRS) is empty when data has not been written to the I2C Bus Transmit Data Register (ICDRT) with the IIC in transmission mode (ICCR2.TRS = 1), the SCLn line is automatically held low over the subsequent intervals. This low-hold period is extended until the transmit data is written, which prevents the unintended transmission of erroneous data.

Master transmit mode:

- Low-level interval after a start or restart condition is issued
- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

Slave transmit mode:

- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

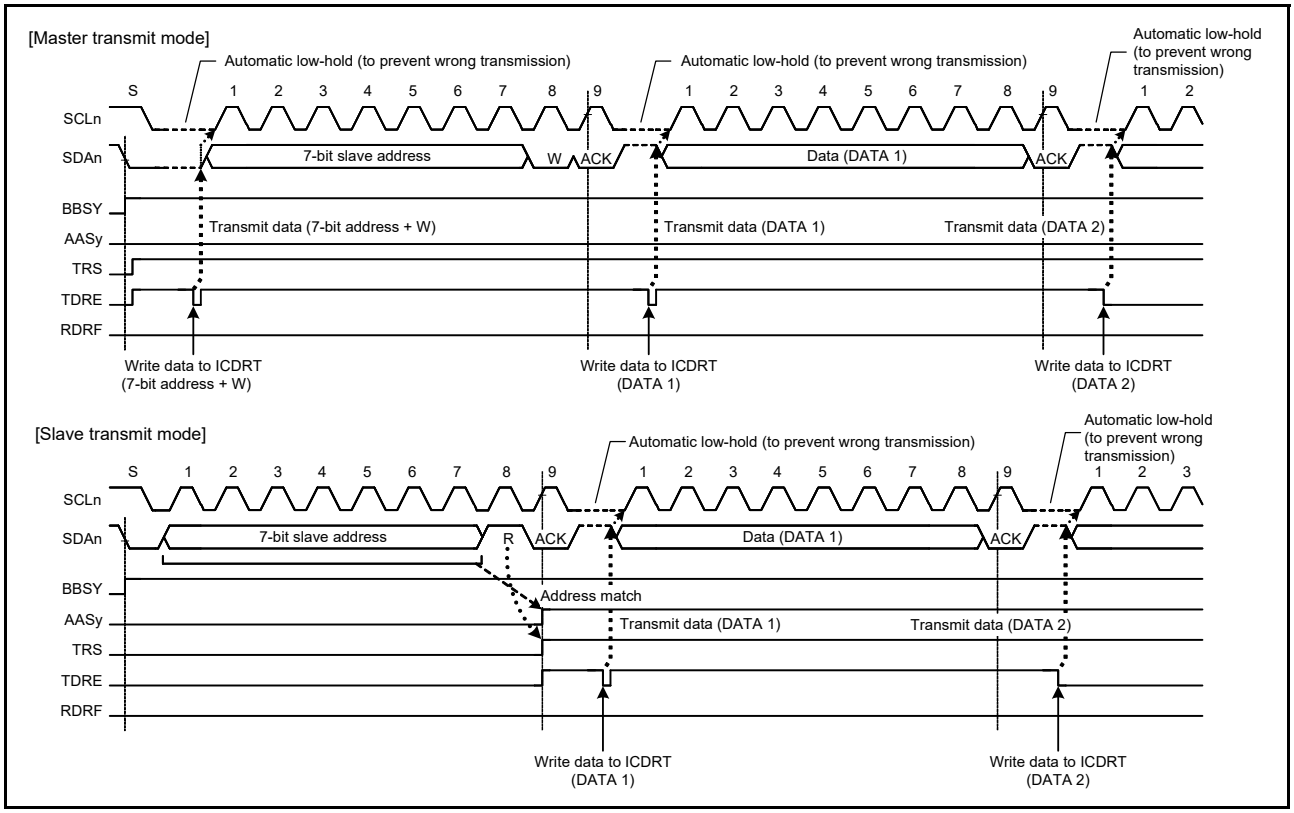


Figure 28.38 Automatic low-hold operation in transmit mode

28.9.2 NACK Reception Transfer Suspension Function

This function suspends transfer operation when NACK is received in transmit mode (ICCR2.TRS = 1). This function is enabled when the NACKE bit in ICFER is set to 1 (transfer suspension enabled). If the next transmit data is already written (TDRE = 0 in ICSR2) when NACK is received, the next data transmission on the falling edge of the 9th SCL clock cycle is automatically suspended. This prevents the SDAn line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (NACKF = 1 in ICSR2), 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, then issue a start condition again.

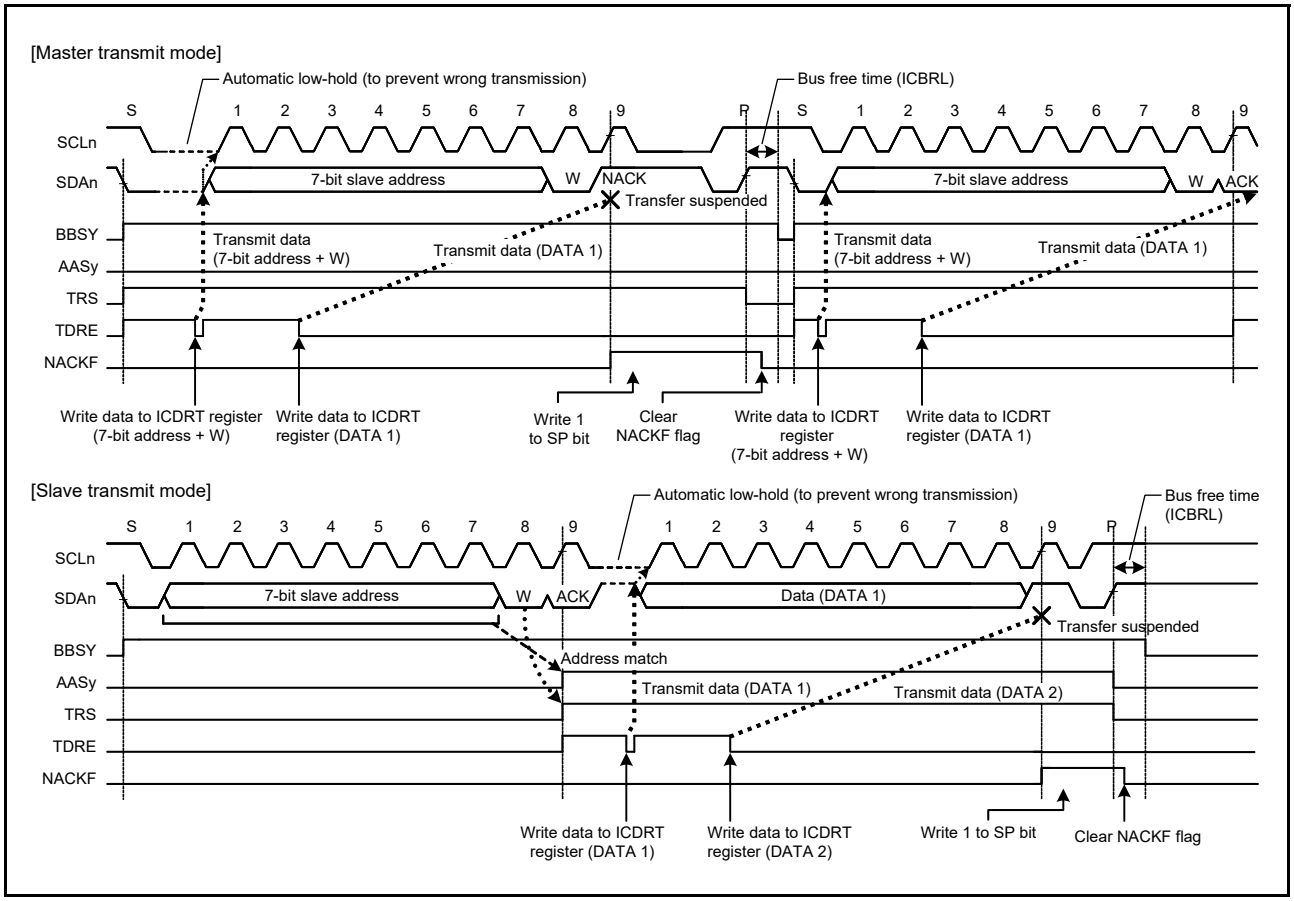


Figure 28.39 Suspension of data transfer when NACK is received (NACKE = 1)

28.9.3 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 (RDRF = 1 in ICSR2) in receive mode (TRS = 0 in ICCR2), the IIC holds the SCLn line low automatically immediately before the next data is received to prevent 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 with other communication because the IIC does not hold the SCLn line low when a mismatch with its own slave address occurs after a stop condition is issued.

Periods in which the SCLn line is held low can be selected with a combination of the WAIT and RDRFS bits in ICMR3.

(1) 1-byte receive operation and automatic low-hold function using the WAIT bit

When the WAIT bit in ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the WAIT bit function. Additionally, when the ICMR3.RDRFS bit is 0, the IIC automatically sends the ACKBT bit value in ICMR3 for the acknowledge bit in the period from the falling edge of the 8th SCL clock cycle to the falling edge of the 9th SCL clock cycle, and automatically holds the SCLn line low on the falling edge of the 9th SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from ICDRR, which enables byte-wise receive operation.

The WAIT bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master 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 ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the RDRF flag in ICSR2 is set to 1 (receive data full) on the rising edge of the 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 a value to the ACKBT bit in ICMR3, but cannot be released by reading data from ICDRR, which enables receive operation through the ACK or NACK transmission control based on the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master receive mode or slave receive mode.

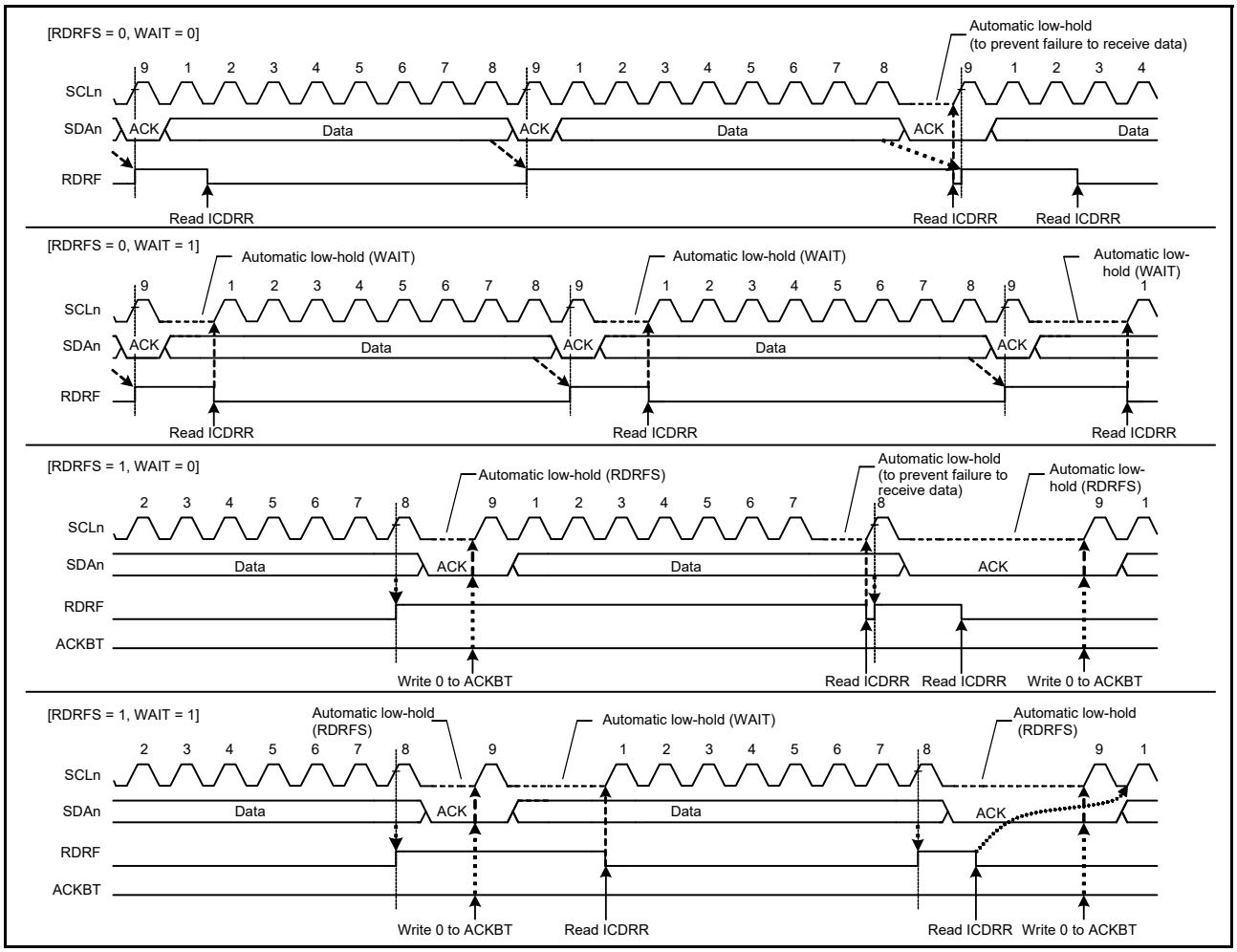


Figure 28.40 Automatic low-hold operation in receive mode using RDRFS and WAIT bits

28.10 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I2C bus standard, the IIC provides functions to prevent double-issue of a start condition, detect arbitration-lost during transmission of NACK, and detect arbitration-lost in slave transmit mode.

28.10.1 Master Arbitration-Lost Detection (MALE Bit)

The IIC drives the SDAn line low to issue a start condition. However, if the SDAn line was already driven low by another master device issuing a start condition, the IIC regards its own start condition as an error and considers this a loss in arbitration. Priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the ST bit in ICCR2 to 1 while the bus is busy (BBSY = 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 being issued while transfer is in progress.

When a start condition is issued successfully, if the transmit data including the address bits (internal SDA output level) and the level on the SDAn line does not match (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state) and a low level is detected on the SDAn line, the IIC loses 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 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 ICCR2.ST bit to 1 while the BBSY flag in ICCR2 is set to 0 (erroneous issuing of a start condition)
- Setting 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 = 1 and TRS = 1 in ICCR2).

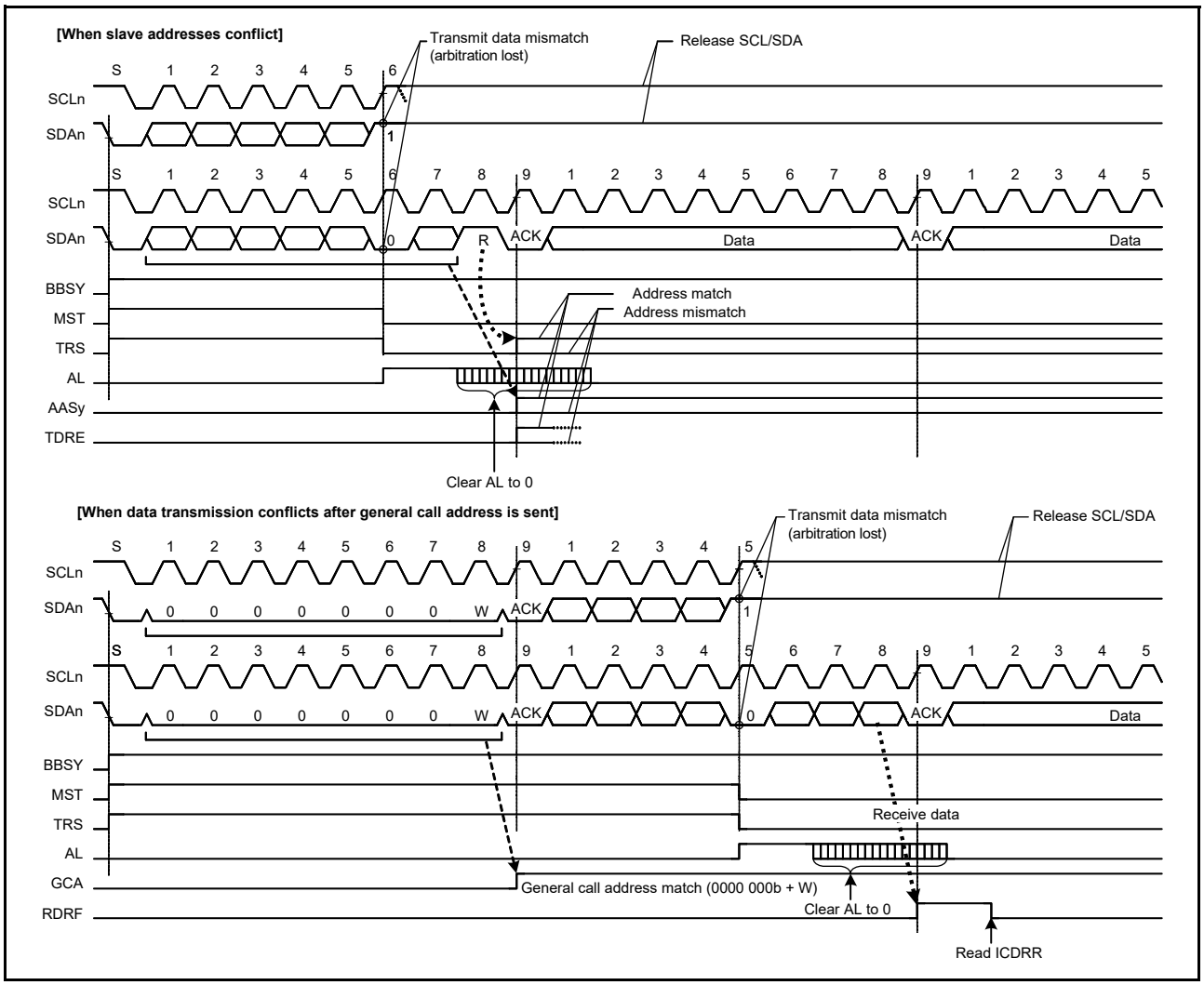


Figure 28.41 Examples of master arbitration-lost detection (MALE = 1)

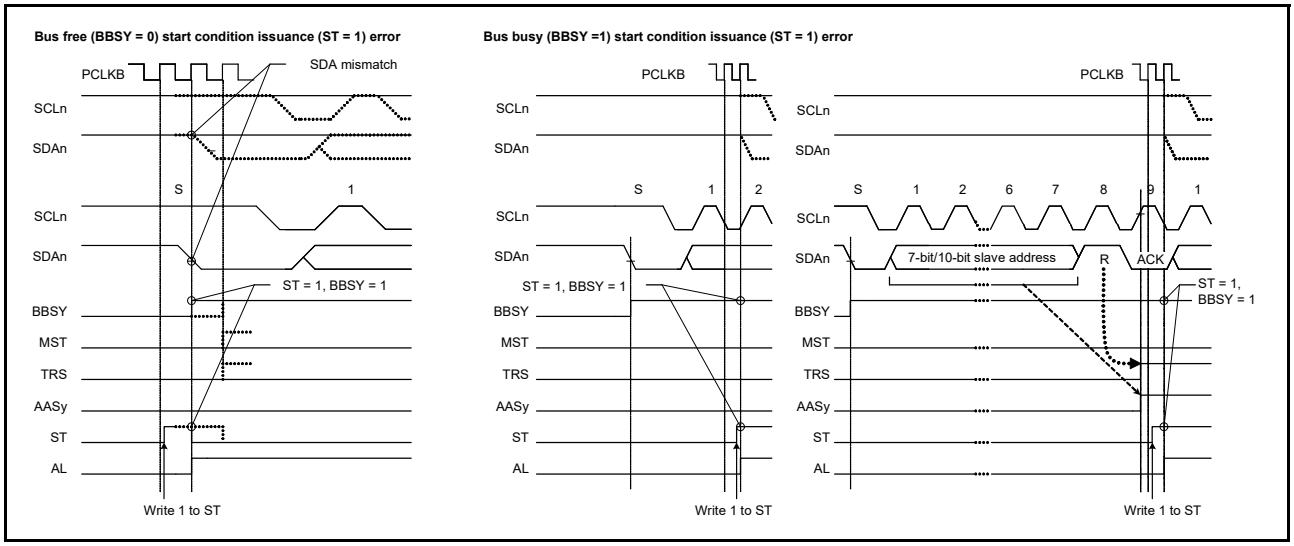


Figure 28.42 Arbitration-lost when start condition is issued (MALE = 1)

28.10.2 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 conflict occurs when multiple master devices send or receive the same information through a single slave device. Figure 28.43 shows an example of arbitration-lost detection during transmission of NACK.

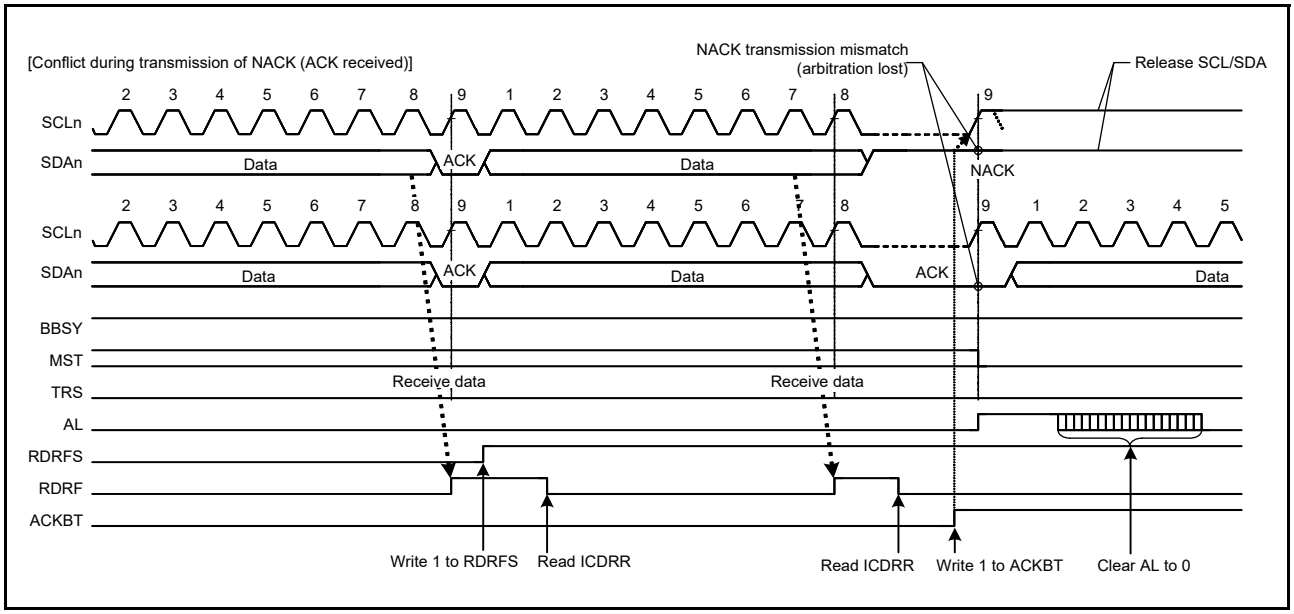


Figure 28.43 Example of arbitration-lost detection during transmission of NACK (NALE = 1)

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 or 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 to eliminate 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 (ACKBT = 1 in ICMR3).

28.10.3 Slave Arbitration-Lost Detection (SALE Bit)

This function causes arbitration to be lost if the transmit data (internal SDA output level) and the level on the SDAn line does 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 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 = 0 and TRS = 1 in ICCR2).

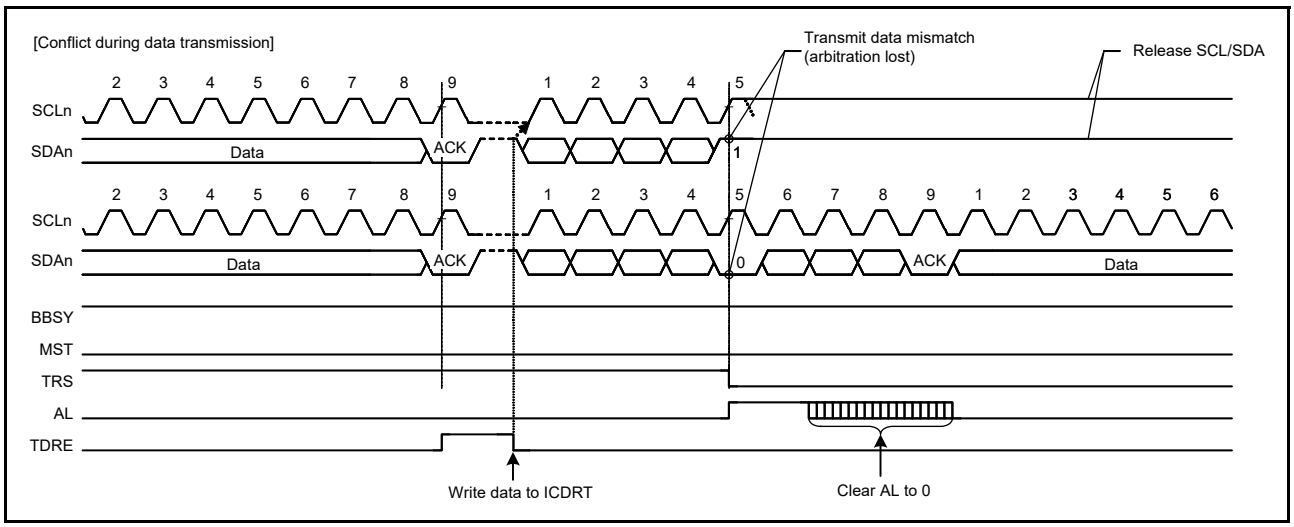


Figure 28.44 Example of slave arbitration-lost detection (SALE = 1)

28.11 Start, Restart, and Stop Condition Issuing Function

28.11.1 Issuing a Start Condition

The IIC issues a start condition when the ST bit in the ICCR2 register is set to 1.

When the ST bit is set to 1, a start condition request is made, and the IIC issues a start condition when the BBSY flag in ICCR2 is 0 (bus free state). When a start condition is issued normally, the IIC automatically shifts to the master transmit mode.

To issue a start condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure that the time set in ICBRH.BRH[4:0] 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 the low-level period of the SCLn line set in ICBRL.BRL[4:0] elapses.

28.11.2 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.BRL[4:0] elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level on the SCLn line and ensure the time set in ICBRL.BRL[4:0] 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.BRH[4:0] and the restart condition hold time elapse.
7. Drive the SCLn line low (high level to low level).
8. Detect a low level on the SCLn line and ensure the low-level period of SCLn line set in ICBRL.BRL[4:0] elapses.

Note: When issuing restart condition requests, write the slave address to ICDRT after confirming that ICCR2.RS = 0. Data written while ICCR2.RS = 1 is not forwarded because of the retransmission condition before the occurrence.

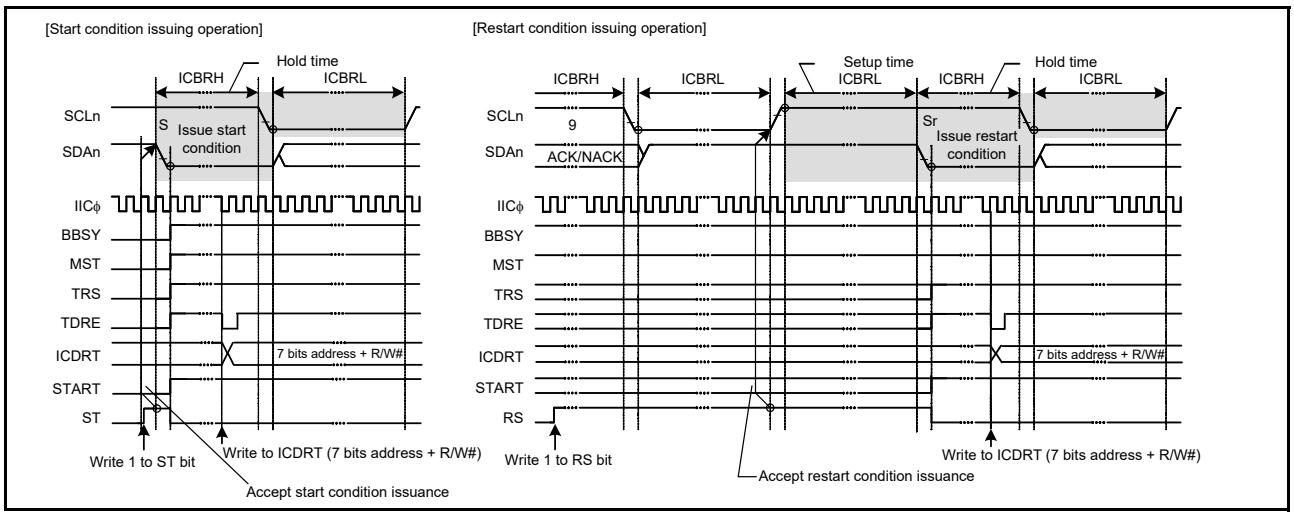


Figure 28.45 Start and restart condition issue timing using the ST and RS bits

Figure 28.46 shows the operation timing when a restart condition is issued after the master transmission.

To issue a restart condition after the master transmission:

1. Initialize the IIC using the procedure in section 28.3.2, Initial Settings.
2. Read the IICR2.BBSY flag to check that the bus is open, 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 BBSY and the START flags in ICSR2 are automatically set to 1 and the ST bit is automatically set to 0. If the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that a start condition is successfully issued as requested by the ST bit. The MST and TRS bits in ICCR2 are automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 when the ICCR2.TRS bit is set to 1.
3. Check that the ICSR2.TDRE flag is 1, then write the value for transmission (the slave address and the R/W# bit) to the ICDRT register. After the transmit data are written to the ICDRT register, the TDRE flag is automatically set to 0, the data is transferred from the ICDRT register to the ICDRS register, and the TDRE flag again sets to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode according to the value of the transmitted R/W# bit. If the value of the R/W# bit is 0, the IIC continues in master transmit mode. If the NACKF.ICSR2 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 (= 0) to the ICDRT register as the first address transmission. Then, as the second address transmission, write the 8 lower bits of the slave address to the ICDRT register.
4. After confirming that the ICSR2.TDRE flag is 1, write the data for transmission to the ICDRT register. The IIC automatically holds the SCLn line low until data for transmission is ready, and a restart condition or a stop condition is issued.
5. After all bytes of data for transmission are written to the ICDRT register, wait until the value of the ICSR2.TEND flag returns to 1. Then, after checking that the ICSR2.START flag is 1, set the ICSR2.START flag to 0.
6. Set the ICCR2.RS bit to 1 (restart condition request). On receiving the request, the IIC issues a restart condition.
7. After checking that the ICSR2.START flag is 1, write the value for transmission (the slave address and the R/W# bit) to the ICDRT register.

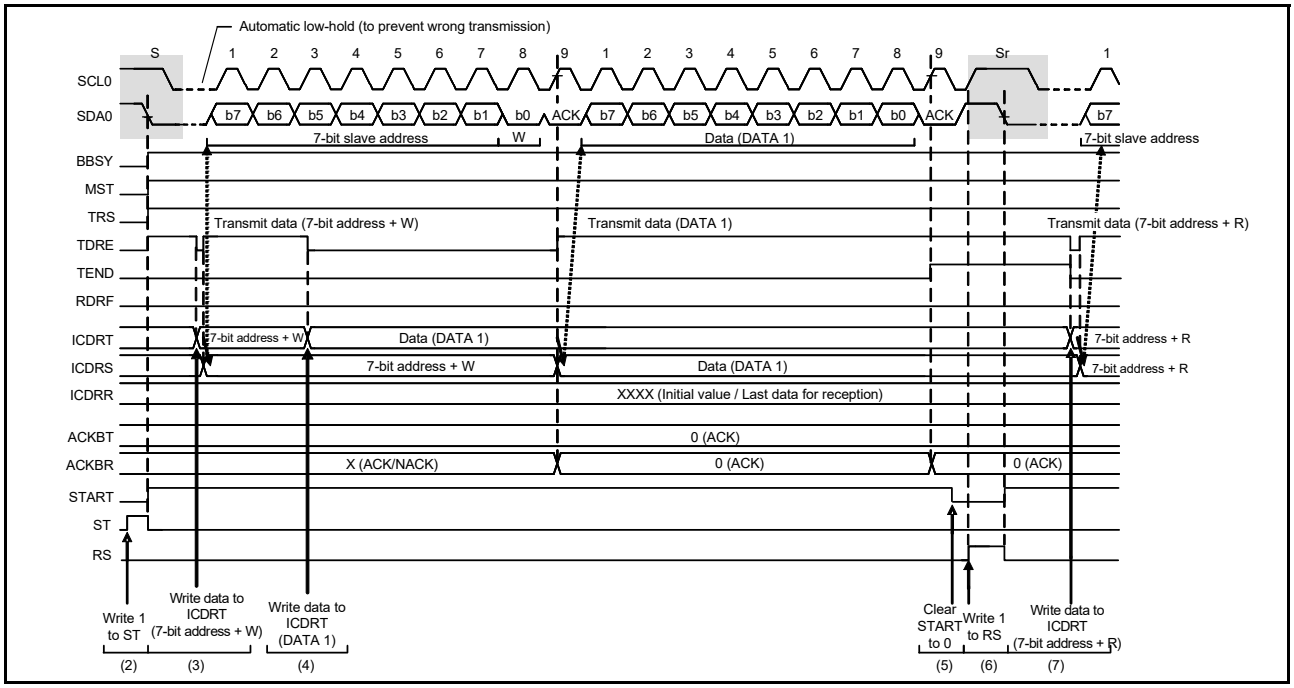


Figure 28.46 Restart condition issue timing after master transmission

28.11.3 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 BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a stop condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure the low-level period of SCLn line set in ICBRL.BRL[4:0] elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level of the SCLn line and ensure the time set in ICBRH.BRH[4:0] and the stop condition setup time elapse.
5. Release the SDAn line (low level to high level).
6. Ensure the time set in ICBRL.BRL[4:0] and the bus free time elapse.
7. Clear the BBSY flag to 0 to release the bus mastership.

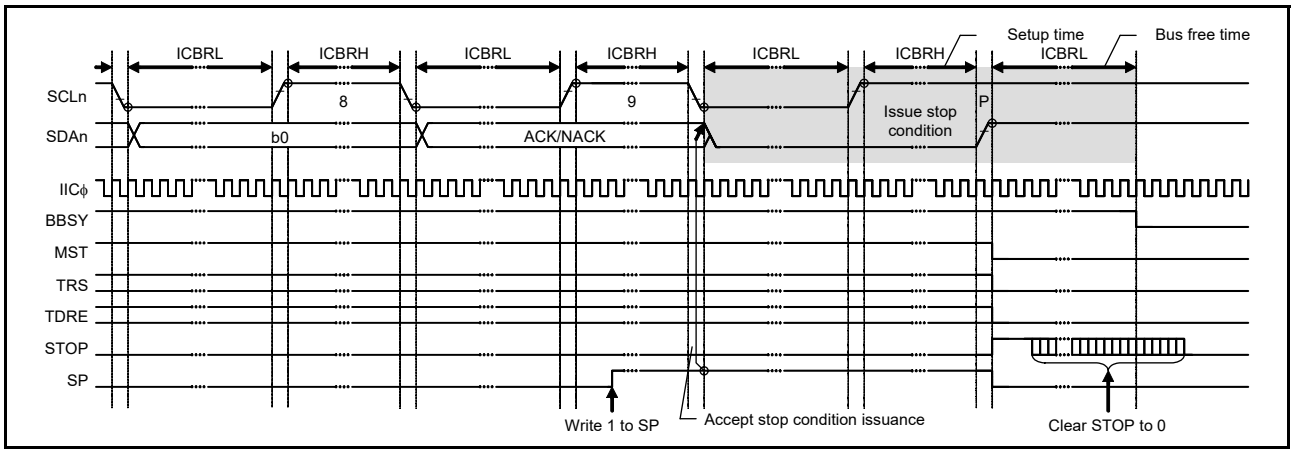


Figure 28.47 Stop condition issue timing using the SP bit

28.12 Bus Hanging

If the clock signals from the master and slave devices are out of synchronization because of noise or other factors, the I2C bus might hang with a fixed level on the SCLn line or SDAn line.

To manage bus hanging, the IIC has:

- A timeout function to detect hanging by monitoring the SCLn line
- A function for the output of an extra SCL clock cycle to release the bus from a hung state because of clock signals being out of synchronization
- The IIC reset function
- An internal reset function.

By checking the SCLO, SDAO, SCLI, and SDAI bits in ICCR1, it is possible to determine whether the IIC or its communicating partner is placing the low level on the SCLn or SDAn lines.

28.12.1 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 open (ICCR2.BBSY flag is 0) while a start condition is requested (ICCR2.ST bit is 1).

The internal counter of the timeout function uses the internal reference clock (IICϕ) set in the CKS[2:0] bits in ICMR1 as a count source. It functions as a 16-bit counter when long mode is selected (TMOS = 0 in ICMR2) or a 14-bit counter when short mode is selected (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.

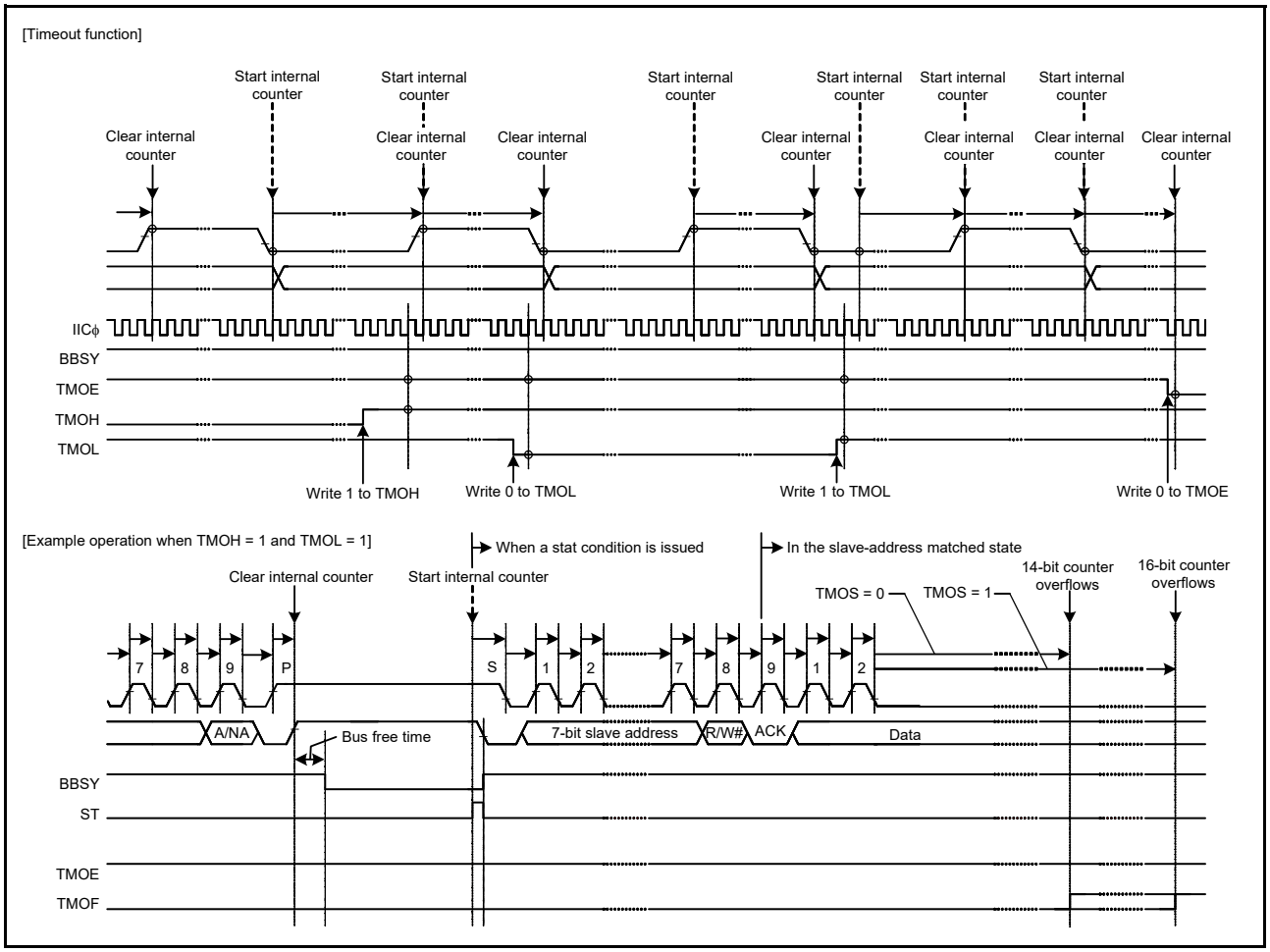


Figure 28.48 Timeout function using the TMOE, TMOS, TMOH, and TMOL bits

28.12.2 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 where 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 malfunctioning.

When the CLO bit in ICCR1 is set to 1 in master mode, a single cycle of the SCL clock at the transfer rate specified in the CKS[2:0] bits in ICMR1, the BRH[4:0] bits in ICBRH, and the BRL[4:0] bits in ICBRL, 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. If the BBSY flag is 1, SCL terminal keeps low output, if BBSY flag is 0, SCL terminal keeps high output.

Additional clock cycles can be output consecutively by writing 1 to the CLO bit with software after reading the bit as 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 due to 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.

[Output conditions for using the CLO bit in ICCR1]:

- When the bus is free (BBSY flag in ICCR2 = 0) or in master mode (MST = 1 and BBSY = 1 in ICCR2)
- When the communication device does not hold the SCLn line low.

Figure 28.49 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

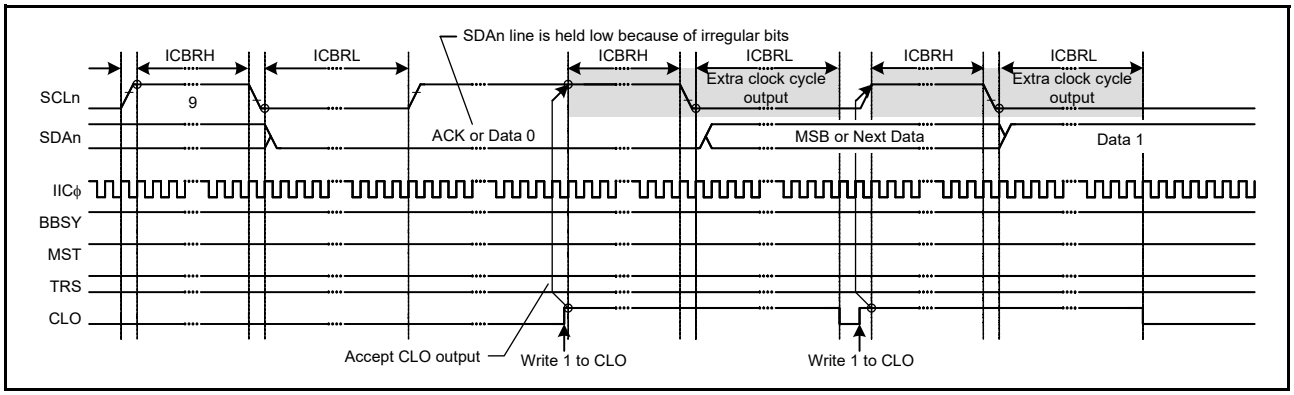


Figure 28.49 Extra SCL clock cycle output function using the CLO bit

28.12.3 IIC Reset and Internal Reset

The IIC module 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 IICRST bit in ICCR1 to 0. Both types of resets are valid for release from bus-hung states, because both restore the output state of the SCLn and SDAn pins to the high-impedance state.

Issuing a reset during slave operation might lead to a loss of synchronization between the master device clock and the slave device clock, so avoid this when possible. In addition, monitoring of the bus state, such as for the presence of a start condition, is not possible during an IIC reset (ICE = 0 and IICRST = 1 in ICCR1).

For a detailed description of the IIC and internal resets, see section 28.15, State of Registers When Issuing Each Condition.

28.13 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 CKS[2:0] bits in ICMR1, BRH[4:0] bits in ICBRH, and BRL[4:0] bits in ICBRL. In addition, specify the values of the DLCS bit in ICMR2 and the SDDL[2:0] bits in ICMR2 to meet the data hold time specification of 300 ns or more. When the IIC is used only as a slave device, the transfer rate setting is not required, but ICBRL.BRL[4:0] 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 SALE bit in ICFER to 1 to enable the slave arbitration-lost detection function.

28.13.1 SMBus Timeout Measurement

(1) Measuring slave device timeout

The following period (timeout interval: $T_{LOW: SEXT}$) must be measured for slave devices in SMBus communication:

- From start condition to stop condition.

To measure timeout for slave devices, measure the period from start condition detection to stop condition detection with the GPT using the IIC start condition detection interrupt (STIn) and stop condition detection interrupt (SPIn). The measured timeout period must be within the total clock low-level period [slave device] $T_{LOW: SEXT}$: 25 ms (maximum) of the SMBus standard.

If the time measured with the GPT exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (minimum) of the SMBus standard, the slave device must release the bus by writing 1 to the IICRST bit in ICCR1 to issue an internal reset of the IIC. When an internal reset is issued, the IIC stops driving the bus for the SCLn and SDAn pins and makes the SCLn and 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), 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 the start condition to the stop condition must be within $T_{LOW: SEXT}$: 25 ms (maximum).

For the ACK receive timing (rising edge of the 9th SCL clock cycle), monitor the TEND flag in ICSR2 in master transmit mode (master transmitter) and the RDRF flag in ICSR2 in master receive mode (master receiver). Perform byte-wise transmit operations in master transmit mode, and hold the RDRFS bit in ICMR3 at 0 until the byte immediately before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 on the rising edge of the 9th SCL clock cycle.

If the period measured with the GPT exceeds the total clock low-level extended period (master device) $T_{LOW: MEXT}$: 10 ms (maximum) of the SMBus standard or the total of measured periods exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (minimum) of the SMBus standard, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (stop writing data to ICDRT).

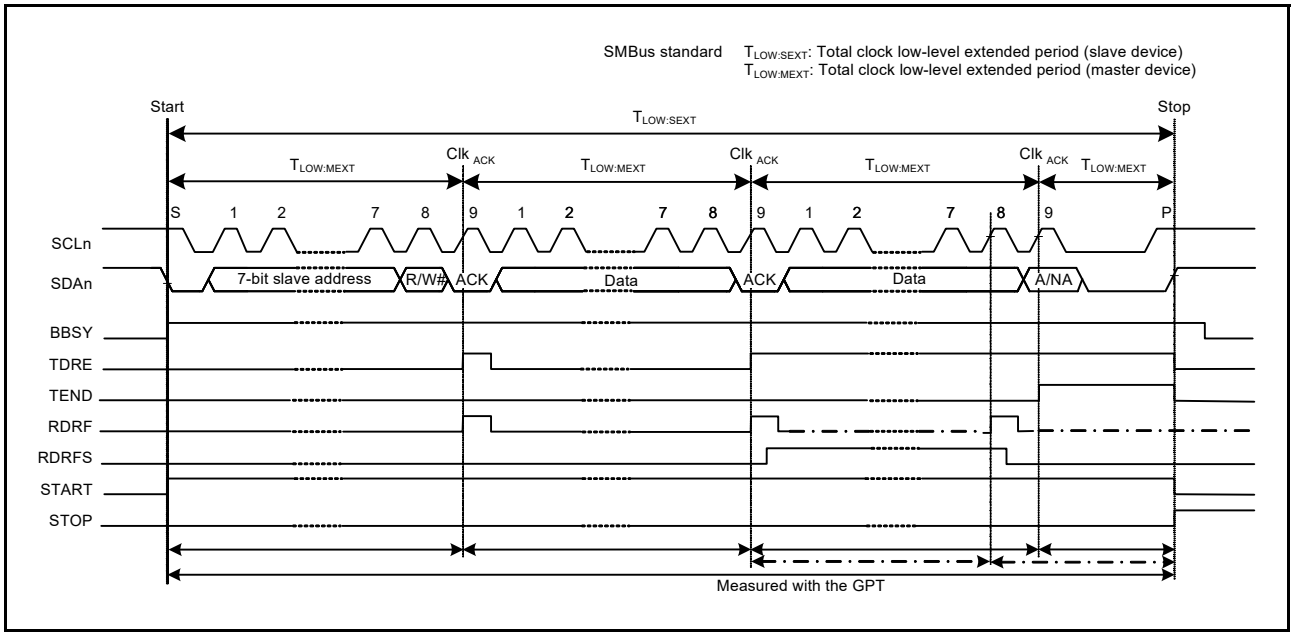


Figure 28.50 SMBus timeout measurement

28.13.2 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 CRCgenerating polynomials of the CRC calculator, see section 31, Cyclic Redundancy Check (CRC) Calculator.

In master transmit mode, the PEC data can be generated by writing all transmit data to the CRC Data Input Register (CRCDIR) in the CRC calculator.

In master receive mode, the PEC data can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC Data Output Register (CRCDOR) with the received PEC data.

To send ACK or NACK based on the match or mismatch result when the final byte is received as a result of the PEC code check, set the RDRFS bit in ICMR3 to 1 before the rising edge of the 8th SCL clock cycle during reception of the final byte, and hold the SCLn line low on the falling edge of the 8th clock cycle.

28.13.3 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, so the IIC has a function for detecting the host address. To detect the host address as a slave address, set the SMBS bit in ICMR3 and the HOAE bit in ICSER to 1. Operation after the host address is detected is the same as normal slave operation.

28.14 Interrupt Sources

The IIC issues five types of interrupt requests:

- Transfer error or event generation (arbitration-lost, NACK detection, timeout detection, start or restart condition detection, and stop condition detection)
- Receive data full
- Transmit data empty
- Transmit end
- Address match during wakeup function.

Table 28.10 lists details of the interrupt requests. The receive data full and transmit data empty interrupts can activate data transfer by the DTC.

Table 28.10 Interrupt sources

Symbol	Interrupt source	Interrupt flag	DTC activation	Interrupt condition
IICn_EEI*5	Transfer error/event generation	AL	Not possible	AL = 1, ALIE = 1
		NACKF		NACKF = 1, NAKIE = 1
		TMOF		TMOF = 1, TMOIE = 1
		START		START = 1, STIE = 1
		STOP		STOP = 1, SPIE = 1
IICn_RXI*2, *5	Receive data full	RDRF	Possible	RDRF = 1, RIE = 1
IICn_TXI*1, *5	Transmit data empty	TDRE	Possible	TDRE = 1, TIE = 1
IICn_TEI*3, *5	Transmit end	TEND	Not possible	TEND = 1, TEIE = 1
IIC0_WUI*4	Address match during wakeup function	WUF, WUSYF or WUASYF	Not possible	• Address match • WUASYF = 1 • WUIE = 1 or • WUF = 1 • WUSYF = 1 • WUIE = 1

Note: There is a delay between the execution of a write instruction for a peripheral module by the CPU and the actual writing to the module. When an interrupt flag is cleared or masked, read the relevant flag again to check whether clearing or masking is complete, 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 TDRE flag in ICSR2 (condition for IICn_TXI) is automatically set to 0 when transmit data is written to ICDRT or a stop condition is detected (STOP = 1 in ICSR2).

Note 2. Because IICn_RXI is an edge-detected interrupt, it does not require clearing. Additionally, the RDRF flag in ICSR2 (condition for IICn_RXI) is automatically set to 0 when data is read from ICDRR.

Note 3. When using the IICn_TEI interrupt, clear the TEND flag in ICSR2 in the IICn_TEI interrupt handling.
The TEND flag in ICSR2 is automatically set to 0 when transmit data is written to ICDRT or a stop condition is detected (STOP = 1 in ICSR2).

Note 4. Only channel 0 has a wakeup function, so IIC0_WUI is for channel 0 only.

Note 5. Channel number (n = 0, 1).

Clear or mask each flag during interrupt handling.

28.14.1 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 ICU.IELSRn.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 ICU.IELSRn.IR flag becomes 0. Internally saved interrupt requests are automatically cleared under normal usage conditions. They can also be cleared by writing 0 to the interrupt enable bit within the associated peripheral module.

28.15 State of Registers When Issuing Each Condition

The IIC has two dedicated resets, IIC reset and internal reset. Table 28.11 lists the register states when issuing each condition.

Table 28.11 Register states when issuing each condition

Registers		Reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
ICCR1	ICE, IICRST	Reset	Saved	Saved	Saved	Saved
	SCLO, SDAO		Reset	Reset		
	Others			Saved		
ICCR2	BBSY	Reset	Reset	Saved	Set	Reset
	ST, RS			Reset	Reset	Saved
	SP					Reset
	TRS				Set or saved	
	MST					
ICMR1	BC[2:0]	Reset	Reset	Reset	Reset	Saved
	Others			Saved	Saved	
ICMR2		Reset	Reset	Saved	Saved	Saved
ICMR3	ACKBIT	Reset	Reset	Saved	Saved	Reset
	Others					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	TEND	Reset	Reset	Reset	Saved	Reset
	TDRE				Set or Saved	
	START				Set	
	STOP				Saved	Set
	Others					Saved
ICWUR		Reset	Reset	Saved	Saved	Saved
SARL0, SARL1, SARL2 SARU0, SARU1, SARU2		Reset	Reset	Saved	Saved	Saved
ICBRH, ICBRL		Reset	Reset	Saved	Saved	Saved
ICDRT		Reset	Reset	Saved	Saved	Saved
ICDRR		Reset	Reset	Saved	Saved	Saved
ICDRS		Reset	Reset	Reset	Saved	Saved
ICWUR2	WUSEN	Reset	Reset	Saved	Saved	Saved
	Others					Set, reset, or saved
Timeout function		Reset	Reset	Reset	Operation	Operation
Bus free time measurement		Reset	Reset	Operation	Operation	Operation

28.16 Event Link Output

IIC0 and IIC1 modules 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.

28.16.1 Interrupt Handling and Event Linking

Each of the IIC interrupt types (see Table 28.10) 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 28.10.

28.17 Usage Notes

28.17.1 Settings for the Module-Stop State

The Module Stop Control Register B (MSTPCRB) can enable or disable IIC operation. The IIC 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.

28.17.2 Notes on Starting Transfer

If the ICU.IELSRn.IR flag associated with the IIC interrupt is 1 when transfer is started (ICCR1.ICE = 1), follow the procedure in this section to clear the interrupts before enabling operations. Starting transfer with the IR flag set to 1 while the ICCR1.ICE bit is 1 leads to an interrupt request being internally saved after transfer starts, and this can lead to unexpected behavior of the IR flag.

To clear interrupts before starting transfer operation:

1. Confirm that the ICCR1.ICE bit is 0.
2. Set the relevant interrupt enable bits, such as ICIER.TIE to 0.
3. Read the relevant interrupt enable bits, such as ICIER.TIE, and confirm that the value is 0.
4. Set the ICU.IELSRn.IR flag to 0.

29. Controller Area Network (CAN) Module

29.1 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 29.1 lists the CAN specifications, and Figure 29.1 shows a block diagram.

Table 29.1 CAN specifications (1 of 2)

Parameter	Specifications
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 either transmission or reception, with remaining mailboxes used for receive (RX) and transmit (TX) 4-stage FIFOs.
Reception	• Support for data frame and remote frame reception • Reception ID format selectable to only standard ID, only extended ID, or mixed IDs • Programmable one-shot reception function • Selectable between overwrite mode (unread message overwritten) and overrun mode (unread message saved) • Reception complete interrupt independently enabled or disabled for each mailbox.
Acceptance filter	• Eight acceptance masks (one for every four mailboxes) • Masks independently enabled or disabled for each mailbox.
Transmission	• Support for data frame and remote frame transmission • Transmission ID format selectable to only standard ID, only extended ID, or mixed IDs) • Programmable one-shot transmission function • Broadcast messaging function • Priority mode selectable based on message ID or mailbox number • Support for transmission request abort, with abort completion confirmable in status flag • Transmission complete interrupt independently enabled or disabled for each mailbox.
Mode transition for bus-off recovery	Mode transition for the recovery from the bus-off state selectable to: • ISO11898-1 specification-compliant • Automatic invoking of CAN halt mode on bus-off entry • Automatic invoking of CAN halt mode on bus-off end • Transition to 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 to 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 29.1 CAN specifications (2 of 2)

Parameter	Specifications
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	CANMCLK
Test mode	Three test modes available for evaluation purposes: • Listen-only mode • Self-test mode 0 (external loopback) • Self-test mode 1 (internal loopback).
Module-stop function	Module-stop state can be set to reduce power consumption

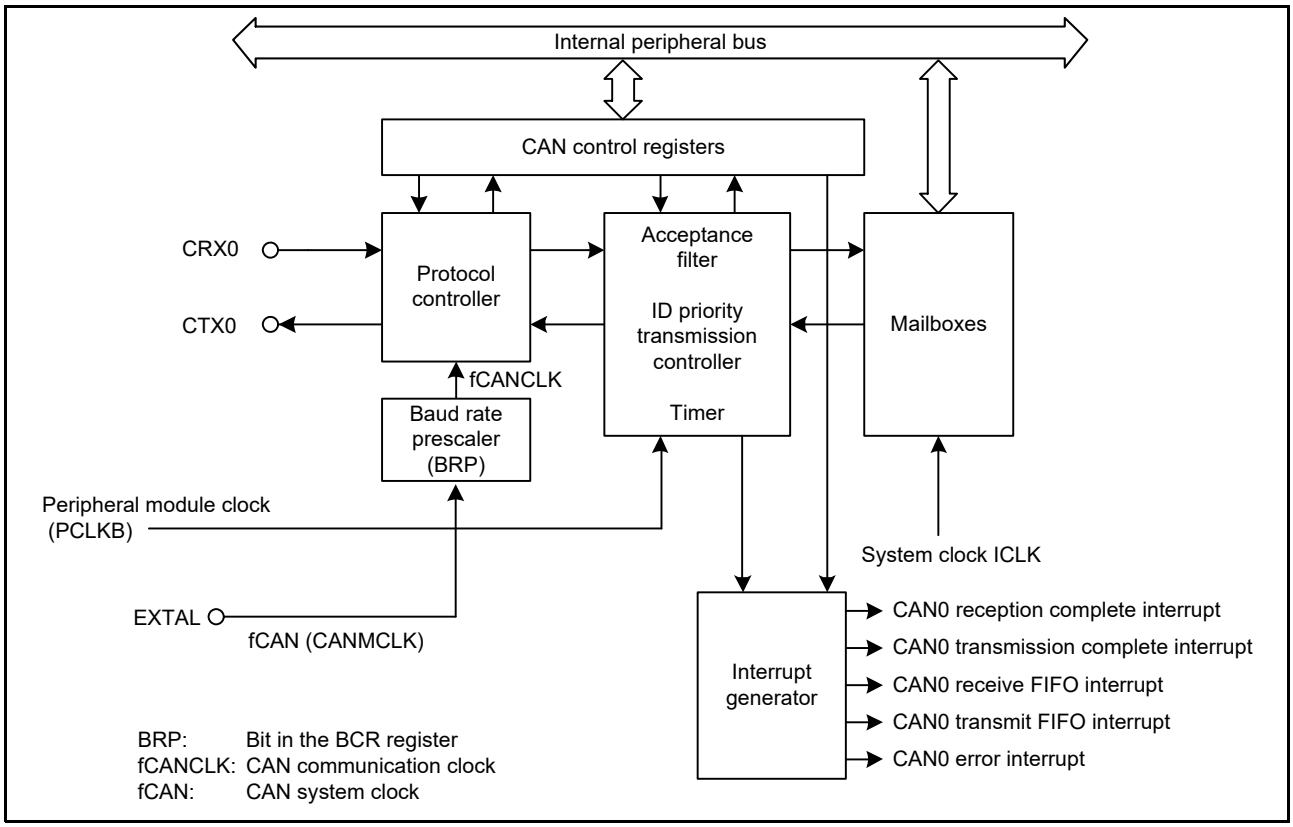


Figure 29.1 CAN module block diagram

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), data field (8 bytes), and a time stamp.
- Acceptance filter
  Performs filtering of received messages using MKRk register for the filtering process
- 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 29.2 lists the CAN module pins. These pins are multiplexed with other signals on the MCU. For details, see section 18, I/O Ports.

Table 29.2 CAN module I/O pins

Pin name	I/O	Function
CRX0	Input	Data receive pin
CTX0	Output	Data transmit pin

29.2 Register Descriptions

29.2.1 Control Register (CTLR)

Address(es): CAN0.CTLR 4005 0840h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RBOC	BOM[1:0]		SLPM	CANM[1:0]		TSPS[1:0]		TSRC	TPM	MLM	IDFM[1:0]		MBM
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MBM	CAN Mailbox Mode Select*1	0: Normal mailbox mode 1: FIFO mailbox mode.	R/W
b2, b1	IDFM[1:0]	ID Format Mode Select *1	b2 b1 0 0: Standard ID mode: All mailboxes, including FIFO mailboxes, handle only standard IDs 0 1: Extended ID mode: All mailboxes, including FIFO mailboxes, handle only extended IDs 1 0: Mixed ID mode: All mailboxes, including FIFO mailboxes, handle both standard and extended IDs. In normal mailbox mode, use the associated IDE bit to differentiate standard and extended IDs. In FIFO mailbox mode, the associated IDE bits are used for mailboxes 0 to 23, the IDE bits in FIDCR0 and FIDCR1 are used for the receive FIFO, and the IDE bit associated with mailbox 24 is used for the transmit FIFO 1 1: Setting prohibited.	R/W
b3	MLM	Message Lost Mode Select*1	0: Overwrite mode 1: Overrun mode.	R/W
b4	TPM	Transmission Priority Mode Select*1	0: ID priority transmit mode 1: Mailbox number priority transmit mode.	R/W
b5	TSRC	Time Stamp Counter Reset Command*4	0: Do not reset time stamp counter 1: Reset time stamp counter.*3	R/W
b7, b6	TSPS[1:0]	Time Stamp Prescaler Select*1	b7 b6 0 0: Every 1-bit time 0 1: Every 2-bit time 1 0: Every 4-bit time 1 1: Every 8-bit time.	R/W

Bit	Symbol	Bit name	Description	R/W
b9, b8	CANM[1:0]	CAN Mode 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: Enter CAN halt mode automatically on entering bus-off state 1 0: Enter CAN halt mode automatically at the end of bus-off state 1 1: Enter CAN halt mode during bus-off recovery period through a software request.	R/W
b13	RBOC	Forced Return from Bus-Off*2	0: No return occurred 1: Forced return from bus-off state.*3	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Write to the BOM[1:0], TSPS[1:0], TPM, MLM, IDFM[1:0], and MBM bits in CAN reset mode.
Note 2. Set the RBOC bit to 1 in the bus-off state.
Note 3. This bit automatically sets 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*1)

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 transmit FIFO
- Mailboxes 28 to 31 are configured as receive FIFO

Transmit data is written into mailbox 24, a window mailbox for the transmit FIFO. Receive data is read from mailbox 28, a window mailbox for the receive FIFO.

Table 29.3 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*1)

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 MLM is 0, all mailboxes are set to overwrite mode. Any new message received overwrites the pre-existing message.

When MLM is 1, all mailboxes are set to overrun mode. Any new message received does not overwrite the pre-existing message and the new message is discarded.

TPM bit (Transmission Priority Mode Select*1)

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 TPM is 0, ID priority transmit mode is selected and transmission priority is arbitrated, as defined in the CAN

specification (ISO11898-1). In ID priority transmit mode, mailboxes 0 to 31 (in normal mailbox mode), and mailboxes 0 to 23 (in FIFO mailbox mode), and the transmit FIFO are compared for the IDs of mailboxes configured for transmission. If two or more mailbox IDs are the same, the mailbox with the smaller number has higher priority.

Only the next message to be transmitted from the transmit FIFO is included in the transmission arbitration. If a FIFO message is currently being transmitted, the next pending message within the transmit FIFO is included in the transmission arbitration.

When TPM is 1, mailbox number transmit mode is selected and the transmit mailbox with the smallest 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*4)

The TSRC bit resets the time stamp counter. When this bit is set to 1, TSR is set to 0000h. This bit is automatically set to 0.

TSPS[1:0] bits (Time Stamp Prescaler Select*1)

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 Operation Select*5)

The CANM[1:0] bits select one of the following modes for the CAN module:

- CAN operation mode
- CAN reset mode
- CAN halt mode.

The CAN sleep mode is set in the SLPM bit. For details, see section 29.3, Operation Modes. When the CAN module enters CAN halt mode based on the BOM[1:0] setting, the CANM[1:0] bits are automatically set to 10b.

SLPM bit (CAN Sleep Mode*5,*6)

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 29.3, Operation Modes.

BOM[1:0] bits (Bus-Off Recovery Mode*1)

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 CTLR are set to 10b to enter the CAN halt mode. No bus-off recovery interrupt request is generated when recovering from bus-off, and TECR and RECR 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 the recovery from the bus-off state, after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request is generated when recovering from bus-off, and TECR and RECR 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 (Forced Return from Bus-Off*2)

When the RBOC bit is set to 1 in the bus-off state, the CAN module forcibly exits the bus-off state. The RBOC bit is automatically set to 0, and the error state changes from bus-off to error-active. When the RBOC bit is set to 1, RECR and TECR are set to 00h and the BOST bit in STR is set to 0, indicating that the CAN module is not in bus-off state. The other registers remain unchanged even when RBOC 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 29.3 Mailbox configuration

Mailbox	MBM = 0 (normal mailbox mode)	MBM = 1 (FIFO mailbox mode)*1 to *5
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 the FIFO interrupts.
Note 4. The bits in MKIVLR 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 frames and remote frames.

29.2.2 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: Setting prohibited 1: CANMCLK (generated by the main clock). This bit must be set to 1 when using the CAN module.	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
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	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 while the SCKSCR.CKSEL[2:0] bits are 011b (selecting the main clock oscillator).

For details about setting the bit timing, see section 29.4, 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 [7:0].

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*1)

The BRP[9:0] bits set the frequency of the CAN communication clock (fCANCLK). The fCANCLK cycle is 1 Tq. If the setting is P (0 to 1023), the baud rate prescaler divides fCAN by P + 1.

TSEG1[3:0] bits (Time Segment 1 Control)

The TSEG1[3:0] bits specify the total length of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1) with a time quantum (Tq) value. A value from 4 to 16 Tq can be set.

29.2.3 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	These bits are read as undefined. The write value should be 0.	R/W

For the mask function in FIFO mailbox mode, see section 29.6, 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. They are used to receive extended ID messages. When an EID[17:0] bit is set to 0, the received ID is not compared with the associated mailbox ID. When an EID[17:0] bit is set to 1, the received ID is compared with the associated mailbox ID.

SID[10:0] bits (Standard ID)

The SID[10:0] bits are the filter mask bits associated with the CAN standard ID bits. They are used to receive both standard ID and extended ID messages. When the SID[10:0] bits are set to 0, the received ID is not compared with the associated mailbox ID. When the SID[10:0] bits are set to 1, the received ID is compared with the associated mailbox ID.

29.2.4 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
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.

FIDCR0 and FIDCR1 are enabled when the MBM bit in CTLR 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 or CAN halt mode. For information on using FIDCR0 and FIDCR1, see section 29.6, 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 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 to data frames or remote frames:

- When the RTR bits in both FIDCR0 and FIDCR1 registers are set to 0, only data frames are received
- When the RTR bits in both FIDCR0 and FIDCR1 registers are set to 1, only remote frames are received
- When the RTR bits in both FIDCR0 and FIDCR1 registers are set to different values, both data and remote frames can be received.

IDE bit (ID Extension*1)

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 both IDE bits in FIDCR0 and FIDCR1 are set to 0, only standard ID frames are received
- When both IDE bits in FIDCR0 and FIDCR1 are set to 1, only extended ID frames are received
- When the IDE bits in FIDCR0 and FIDCR1 are set to different values, both standard ID and extended ID frames can be received.

29.2.5 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 MKIVLR 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).

Note: Set bits [31:24] to 0 in FIFO mailbox mode.

When a bit is set to 1, the corresponding acceptance mask register becomes invalid for the associated mailbox. When a mask invalid bit is set to 1, a message is received by the associated mailbox only if the receive message ID matches the mailbox ID exactly. Write to MKIVLR in CAN halt mode.

29.2.6 Mailbox Register j (MBj_ID, MBj_DL, MBj_Dm, MBj_TS) (j = 0 to 31, m = 0 to 7)

Table 29.4 lists the CAN0 mailbox memory mapping, and Table 29.5 lists the CAN data frame configuration. The value of the CAN0 mailbox after reset is undefined.

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 29.4 for details on register addresses.

Table 29.4 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 29.4 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 29.5 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 data and remote frames	R/W
b28 to b18	SID[10:0]	Standard ID	Standard ID of 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 CTLR 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	The TSH[7:0] and TSL[7:0] bits store the counter value of the time stamp when received messages are stored in the mailbox.	R/W
b15 to b8	TSH[7:0]	Time Stamp Higher Byte		R/W

EID[17:0] bits (Extended ID*1)

The EID[17:0] bits set the extended ID of data frames 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 frames 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 as specified in the RTR bit in the FIDCR0 and FIDCR1 registers
- The transmit FIFO mailbox transmits the data or remote frame as specified in the RTR bit in the relevant transmit message.

IDE bit (ID Extension*2)

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).

- 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 bits in FIDCR0 and FIDCR1
- The transmit FIFO mailbox transmits messages with the standard ID or extended ID specified in the IDE bit in the associated transmit message.

DLC[3:0] bits (Data Length Code*1)

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.

29.2.7 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 can enable interrupts for each mailbox independently. 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 associated 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.

29.2.8 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 a 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 because it is full.
Note 2. Buffer warning indicates a state in which the third message is stored in the receive FIFO.

The MIER_FIFO register can enable interrupts for each mailbox and FIFO independently. 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).

MB24, MB25, MB28 and MB29 specify whether the 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 associated 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 associated 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.

29.2.9 Message Control Register for Transmit (MCTL_TXj) (j = 0 to 31)

- Transmit mode (when the TRMREQ bit is 1 and the RECREQ bit is 0)

Address(es): CAN0.MCTL_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 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: One-shot transmission disabled 1: One-shot transmission enabled.	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 mailbox j to transmit or receive mode. In transmit mode, MCTL_TXj also controls and indicates the transmission status. Do not access MCTL_TXj if mailbox j is in receive mode. Only write to MCTL_TXj in CAN operation mode or CAN halt mode. Do not use the MCTL_TX24 to MCTL_TX31 registers in FIFO mailbox mode.

SENTDATA flag (Transmission Complete Flag)

The SENTDATA flag is set to 1 when data transmission from the associated mailbox is complete. 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, when a CAN bus error occurs, or when data transmission is complete.

TRMABT flag (Transmission Abort Complete Flag*1,*2)

The TRMABT flag is set to 1 in the following cases:

- Following a transmission abort request, when the transmission abort is complete 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 a 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*2,*3)

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 does not complete because of a CAN bus error or CAN bus arbitration-lost, the TRMABT flag is set to 1. Set the ONESHOT bit to 0 after the SENTDATA or TRMABT flag is set to 1.

RECREQ bit (Receive Mailbox Request)

The RECREQ bit selects the receive modes listed in Table 29.10.

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.REQREQ.

TRMREQ bit (Transmit Mailbox Request)

The TRMREQ bit selects the transmit modes listed in Table 29.10.

When the TRMREQ bit is set to 1, the associated mailbox is configured for transmission of a data or remote frame.

When the TRMREQ bit is set to 0, the associated mailbox is not configured for transmission of a data or 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.

29.2.10 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 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: One-shot reception disabled 1: One-shot reception enabled.	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 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 or receive mode. In receive mode, MCTL_RXj also controls and indicates the reception status.

Do not access MCTL_RXj if mailbox j is in transmit mode. Only write to MCTL_RXj in CAN operation mode or CAN halt mode. Do not use the MCTL_RX24 to MCTL_RX31 registers in FIFO mailbox mode.

NEWDATA flag (Reception Complete Flag*1,*2)

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*1,*2)

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*2,*3)

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 29.10.

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 the acceptance filter processing (the beginning of 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. The maximum period of hardware protection is from the beginning of the CRC field to the end of 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_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 29.10.

When the TRMREQ bit is set to 1, the associated mailbox is configured for transmission of a data frame or remote frame. When the TRMREQ bit is set to 0, the associated mailbox is not configured for transmission of a data frame or remote frame.

If the TRMREQ bit 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_RXj.TRMREQ is the mirror bit of MCTL_TXj.TRMREQ.

29.2.11 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: Receive FIFO disabled 1: Receive FIFO enabled.	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 the RFE bit to 1 in normal mailbox mode (MBM bit in CTLR = 0). Due to hardware protection, the RFE bit cannot be set to 0 through a software write during the following period:

- Hardware protection is started from acceptance filter processing (the beginning of the CRC field)
- Hardware protection is released:
  - If the receive FIFO is specified to receive the incoming message, after the received data is stored in the receive FIFO or a CAN bus error occurs. This means that the maximum period of hardware protection is from the beginning of CRC field to the end of the 7th bit of EOF
  - If the receive FIFO is not specified to receive the message, after acceptance filter processing.

RFUST[2:0] bits (Receive FIFO Unread Message Number Status)

The RFUST[2:0] bits indicate the number of unread messages in the receive FIFO. The value of the RFUST[2:0] bits is initialized to 000b when the RFE bit is set to 0.

RFMLF flag (Receive FIFO Message Lost Flag)

The RFMLF flag is set to 1 (receive FIFO message lost) when the receive FIFO receives a new message and is full. It is set to 1 at the end of the 6th bit of EOF.

The RFMLF flag is set to 0 through a software write (writing 1 has no effect). In both overwrite and overrun modes, if the receive FIFO is full and determined to 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. It 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 29.2 shows the receive FIFO mailbox operation.

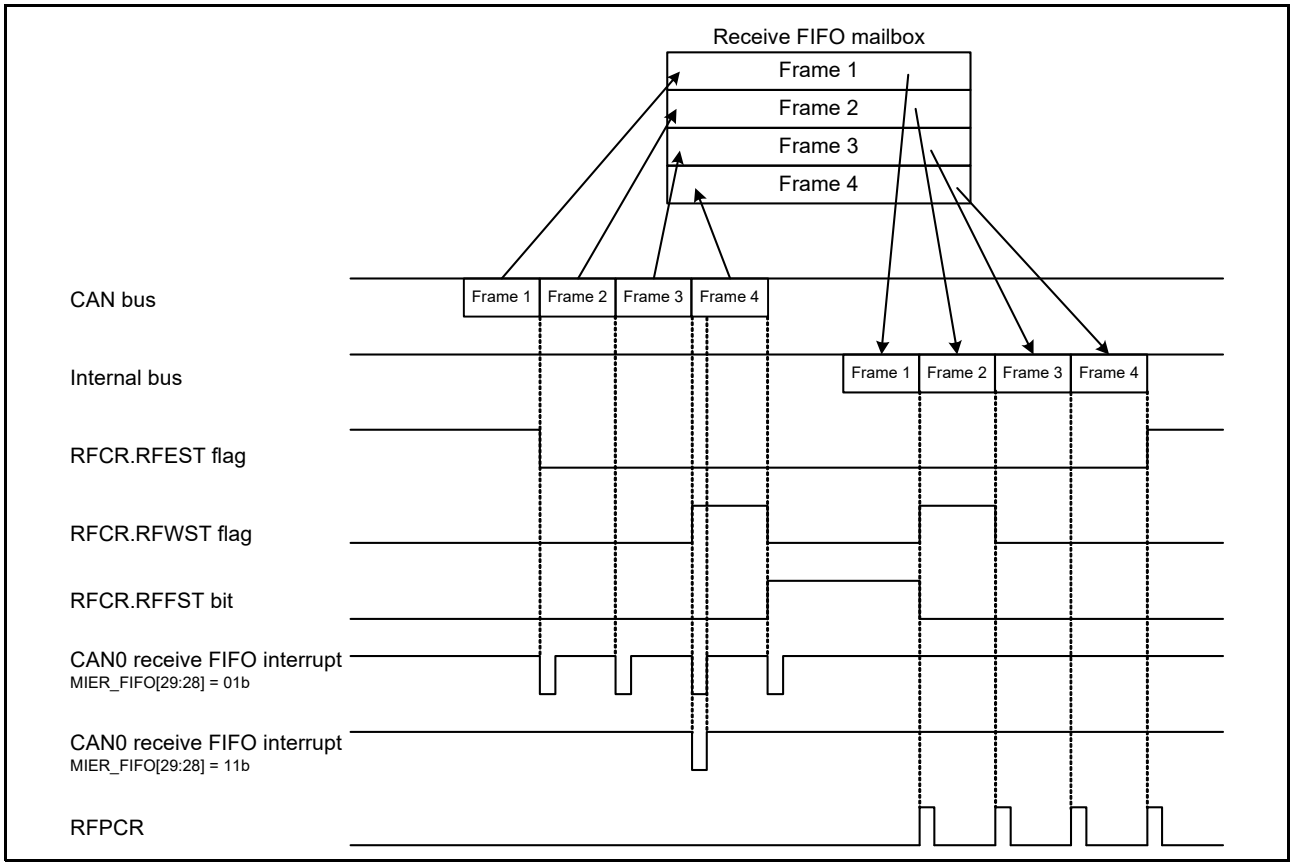


Figure 29.2 Receive FIFO mailbox operation with bits [29:28] in MIER_FIFO = 01b or 11b

29.2.12 Receive FIFO Pointer Control Register (RFPCR)

Address(es): CAN0.RFPCR 4005 0849h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Description	R/W
b7 to b0	The CPU pointer for the receive FIFO is incremented by writing FFh to RFPCR	W

When the receive FIFO is not empty, write FFh to the RFPCR register through software to increment the CPU pointer to the next mailbox location. Do not write to RFPCR when the RFE bit in RFCR is 0 (receive FIFO disabled).

Both the CAN and CPU pointers increment when a new message is received and the RFFST flag is 1 (receive FIFO is full) in overwrite mode. When RFMLF flag is 1 in this condition, the CPU pointer cannot be incremented on a software write to RFPCR.

29.2.13 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: Transmit FIFO disabled 1: Transmit FIFO enabled.	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 the TFCR register 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 = 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 is already in transmission

- On completion of transmission, on 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. These 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 is aborted. The TFEST bit is set to 0 (message in transmit FIFO) when the number of unsent messages in the transmit FIFO is not 0.

Figure 29.3 shows the transmit FIFO mailbox operation.

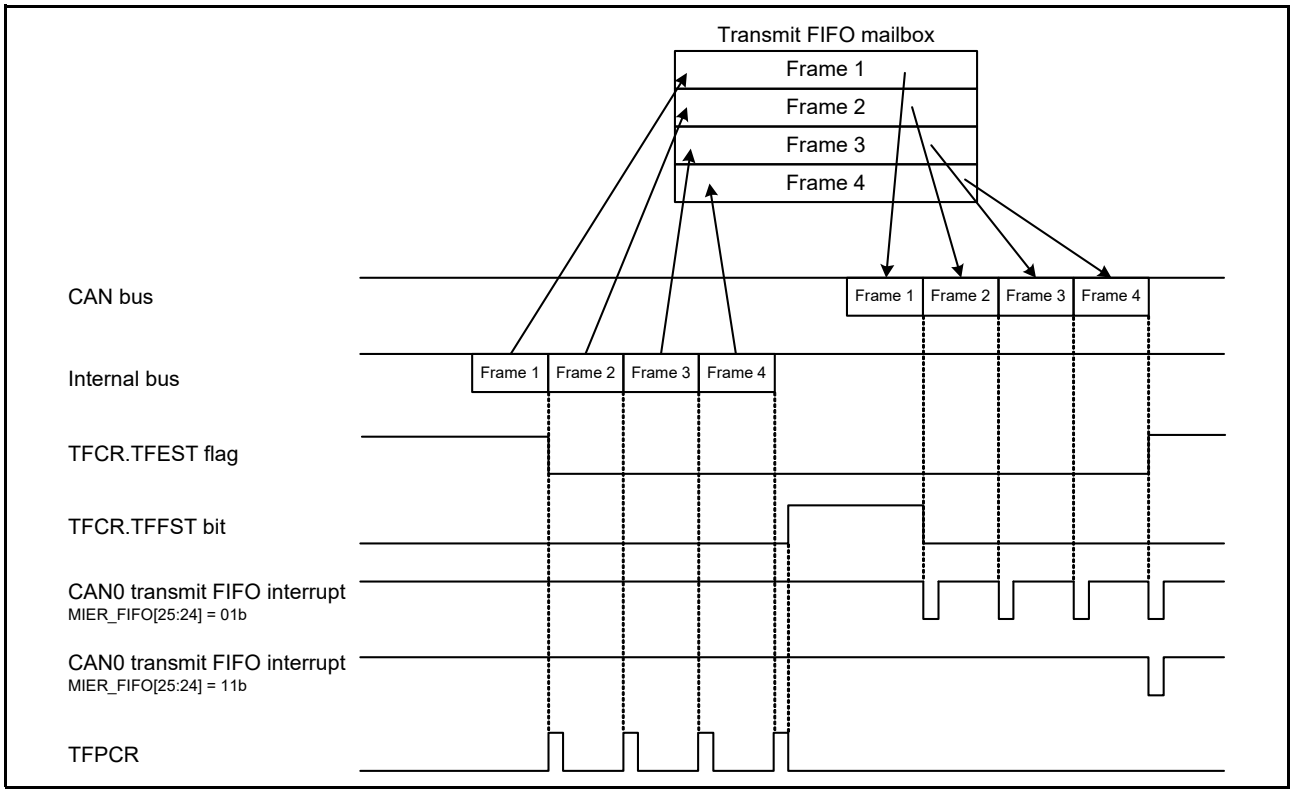


Figure 29.3 Transmit FIFO mailbox operation when bits [25:24] in MIER_FIFO = 01b or 11b

29.2.14 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 TFPCR when the TFE bit in TFCR is 0 (transmit FIFO disabled).

29.2.15 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: One or more mailboxes with NEWDATA = 1.	R
b1	SDST	SENTDATA Status Flag	0: No mailbox with SENTDATA = 1 1: One 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: One 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

Bit	Symbol	Bit name	Description	R/W
b13	TRMST	Transmit Status Flag	0: Bus idle or reception in progress 1: Transmission in progress or module in bus-off state.	R
b14	RECST	Receive Status Flag	0: Bus idle or transmission in progress 1: Reception in progress.	R
b15	—	Reserved	This bit is read as 0	R

NDST flag (NEWDATA Status Flag)

The NDST flag is set to 1 when at least one NEWDATA flag in MCTL_RXj (j = 0 to 31) registers is 1, regardless of the value of MIER or MIER_FIFO. The NDST flag is set to 0 when all NEWDATA flags are 0.

SDST flag (SENTDATA Status Flag)

The SDST flag is set to 1 when at least one SENTDATA flag in MCTL_TXj (j = 0 to 31) registers is 1, regardless of the value of MIER or MIER_FIFO. The SDST flag is set to 0 when all 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. It 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) registers 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 bit in MCTL_TXj (j = 0 to 31) registers is 1, regardless of the value of MIER or MIER_FIFO. It is set to 0 when all TRMABT bits are 0.

EST flag (Error Status Flag)

The EST flag is set to 1 when at least one error is detected by EIFR, regardless of the value of EIER. 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. It is set to 0 when the CAN module is not in CAN halt mode. It remains 1, even when the state changes from CAN halt mode to CAN sleep mode.

SLPST flag (CAN Sleep Status Flag)

The SLPST flag is set to 1 when the CAN module is in CAN sleep mode. The SLPST flag is set to 0 when the CAN module is not in CAN sleep mode.

EPST flag (Error-Passive Status Flag)

The EPST flag is set to 1 when the value of TECR or RECR exceeds 127 and the CAN module is in an 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 an error-passive state.

BOST flag (Bus-Off Status Flag)

The BOST flag is set to 1 when the value in TECR exceeds 255 and the CAN module is in the bus-off state (TEC ≥ 256). The BOST flag is set to 0 when the CAN module is not in the bus-off state.

TRMST flag (Transmit Status Flag)

The TRMST flag is set to 1 when the CAN module performs as a transmitter node or is in the bus-off state. It 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.

29.2.16 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 MCTL_RXj (j = 0 to 31) registers for the normal mailbox and the RFEST flag in RFCR.

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 MCTL_TXj (j = 0 to 31).

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 MCTL_RXj for the normal mailbox, and the RFMLF flag in RFCR.

When the MBSM[1:0] bits are 11b, channel search mode is selected. In this mode, the search target is CSSR. See section 29.2.18, Channel Search Support Register (CSSR).

29.2.17 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 associated NEWDATA, SENTDATA, or MSGLOST flag is set to 0 for a mailbox output by MBNST[4:0]
- When the associated 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 and 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 29.6 lists the behavior of the MBNST[4:0] bits in FIFO mailbox mode.

In channel search mode, the MBNST[4:0] bits output the associated channel number. After the MSSR register is read by software, the next target channel number is output.

SEST bit (Search Result Status)

The SEST bit is set to 1 (no search result) when no associated mailbox is found after searching all mailboxes. For example, in transmit mailbox search mode, the SEST bit is set to 1 when no SENTDATA 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 29.6 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 (new message is being stored or was stored to the mailbox) and the receive FIFO is not empty
01b		Mailbox 28 is not output
10b		Mailbox 28 is output when no MCTL_RXj.MSGLOST flag for the normal mailboxes is set to 1 (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

29.2.18 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 CSSR only when the MSMR.MBSM[1:0] bits are 11b (channel search mode). Write to CSSR in CAN operation mode or CAN halt mode.

Figure 29.4 shows the write and read operations of the CSSR and MSSR registers.

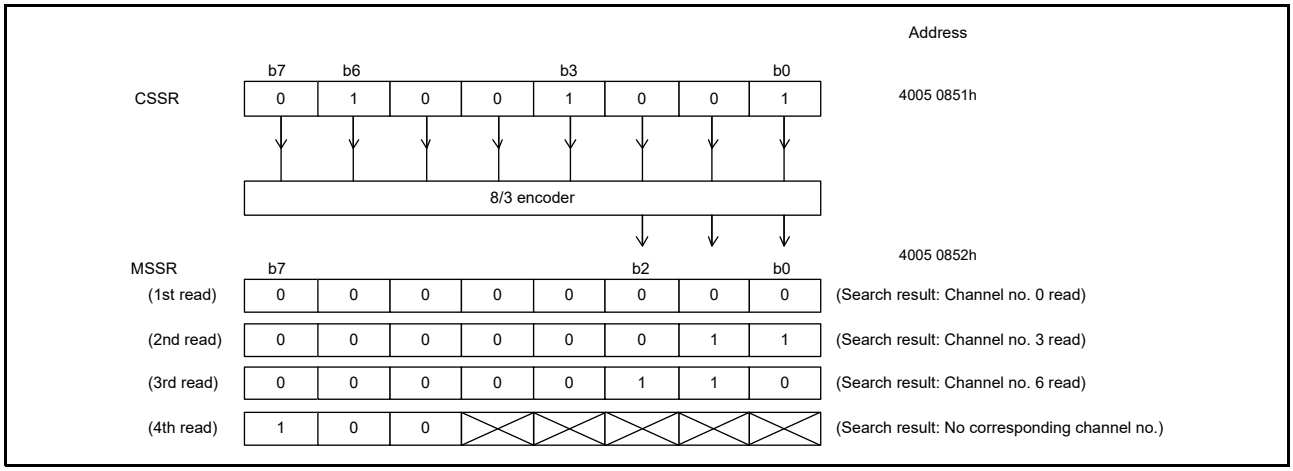


Figure 29.4 Write and read operations of the CSSR and MSSR registers

The value of CSSR is also updated whenever MSSR is read. On this read, the value prior to conversion by the 8/3 encoder can be read.

29.2.19 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.

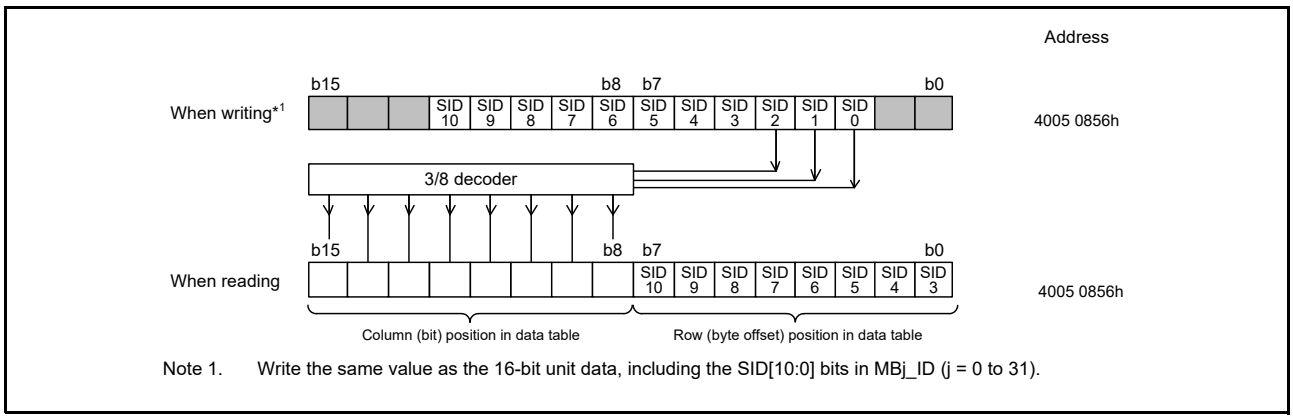
The acceptance filter support unit (ASU) can be used for data table (8 bits × 256) searches. In the data table, all standard IDs that are created are set to be valid or invalid in bit units. When AFSR is written with data in 16-bit units including the SID[10:0] bits in MBj_ID (j = 0 to 31), in which a received standard ID is stored, a decoded row (byte offset) position and column (bit) position for data table search can be read. The ASU can be used for standard (11-bit) IDs only.

The ASU is enabled in the following cases:

- When the 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 29.5 shows the write and read operations in the AFSR register.



Note 1. Write the same value as the 16-bit unit data, including the SID[10:0] bits in MBj_ID (j = 0 to 31).

Figure 29.5 Write and read operations in the AFSR register

29.2.20 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: Interrupt disabled 1: Interrupt enabled.	R/W
b1	EWIE	Error-Warning Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled.	R/W
b2	EPIE	Error-Passive Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled.	R/W
b3	BOEIE	Bus-Off Entry Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled.	R/W
b4	BORIE	Bus-Off Recovery Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled.	R/W
b5	ORIE	Overrun Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled.	R/W
b6	OLIE	Overload Frame Transmit Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled.	R/W
b7	BLIE	Bus Lock Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled.	R/W

The EIER register independently enables or disables the error interrupt for each error interrupt source. Write to EIER in CAN reset mode.

BEIE bit (Bus Error Interrupt Enable)

When the BEIE bit is 0, no error interrupt request occurs even if the BEIF flag in EIFR is 1. When the BEIE bit is 1, an error interrupt request occurs if the BEIF flag is set to 1.

EWIE bit (Error-Warning Interrupt Enable)

When the EWIE bit is 0, no error interrupt request occurs even if the EWIF flag in EIFR is 1. When the EWIE bit is 1, an error interrupt request occurs if the EWIF flag is set to 1.

EPIE bit (Error-Passive Interrupt Enable)

When the EPIE bit is 0, no error interrupt request occurs even if the EPIF flag in EIFR is 1. When the EPIE bit is 1, an error interrupt request occurs 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 occurs even if the BOEIF flag in EIFR is 1. When the BOEIE bit is 1, an error interrupt request occurs 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 occurs even if the BORIF flag in EIFR is 1. When the BORIE bit is set to 1, an error interrupt request occurs if the BORIF flag is set to 1.

ORIE bit (Overrun Interrupt Enable)

When the ORIE bit is 0, no error interrupt request occurs even if the ORIF flag in EIFR is 1. When the ORIE bit is 1, an error interrupt request occurs 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 occurs even if the OLIF flag in EIFR is 1. When the OLIE bit is 1, an error interrupt request occurs if the OLIF flag is set to 1.

BLIE bit (Bus Lock Interrupt Enable)

When the BLIE bit is 0, no error interrupt request occurs even if the BLIF bit in EIFR is 1. When the BLIE bit is 1, an error interrupt request occurs if the BLIF flag is set to 1.

29.2.21 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

Bit	Symbol	Bit name	Description	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 EIER setting.

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, while 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 REC or TEC remains greater than 127, the EPIF flag is not set to 1 until REC or TEC 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. This 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 29.7 lists the behavior of the BOEIF and BORIF flags for each CTLR.BOM[1:0] setting.

Table 29.7 Behavior of BOEIF and BORIF flags for each CTLR.BOM[1:0] setting

BOM[1:0] bits	BOEIF flag	BORIF flag
00b	Set to 1 on entry to bus-off state	Set to 1 on exit from 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. It is not set to 1 in overwrite mode. In this mode, a reception complete interrupt request occurs 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 to 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 to 23, or the receive FIFO, this 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, 32 consecutive dominant bits are detected again in either of the following conditions:

- Recessive bits are detected after the BLIF flag changes to 0 from 1
- The CAN module enters CAN reset or halt mode and then enters CAN operation mode again, after the BLIF flag changes to 0 from 1.

29.2.22 Receive Error Count Register (RECR)

Address(es): CAN0.RECR 4005 084Eh

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

Bit	Description	R/W
b7 to b0	Receive error count function. RECR increments or decrements the counter value based on the error status of the CAN module during reception.	R

The RECR register indicates the value of the receive error counter. See the CAN specification (ISO11898-1) for the increment and decrement conditions of the receive error counter. The value of RECR in the bus-off state is undefined.

29.2.23 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 TECR in the bus-off state is undefined.

29.2.24 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 first detected error code 1: Output 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 halt mode.
Note 4. If more than one error condition is detected simultaneously, all the related bits are set to 1.

The ECSR register indicates whether an error occurred on the CAN bus. See the CAN specification (ISO11898-1) for the conditions when each error occurs.

Clear all of the bits, except for EDPM, to 0 through a software write. If the 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*1,*2)

The SEF flag is set to 1 when a stuff error is detected.

FEF flag (Form Error Flag*1,*2)

The FEF flag is set to 1 when a form error is detected.

AEF flag (ACK Error Flag*1,*2)

The AEF flag is set to 1 when an ACK error is detected.

CEF flag (CRC Error Flag*1,*2)

The CEF flag is set to 1 when a CRC error is detected.

BE1F flag (Bit Error (recessive) Flag*1,*2)

The BE1F flag is set to 1 when a recessive bit error is detected.

BE0F flag (Bit Error (dominant) Flag*1,*2)

The BE0F flag is set to 1 when a dominant bit error is detected.

ADEF flag (ACK Delimiter Error Flag*1,*2)

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*3,*4)

The EDPM bit selects the output mode of ECSR. When the EDPM bit is set to 0, ECSR outputs the first error code. When the EDPM bit is set to 1, ECSR outputs the accumulated error code.

29.2.25 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 TSR 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.

29.2.26 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: CAN test mode disabled 1: CAN test mode enabled.	R/W
b2, b1	TSTM[1:0]	CAN Test Mode Select	b2 b1 0 0: Not CAN test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback) 1 1: Self-test mode 1 (internal loopback).	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TCR register controls the CAN test mode. Write to TCR 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 29.6 shows the connection when listen-only mode is selected.

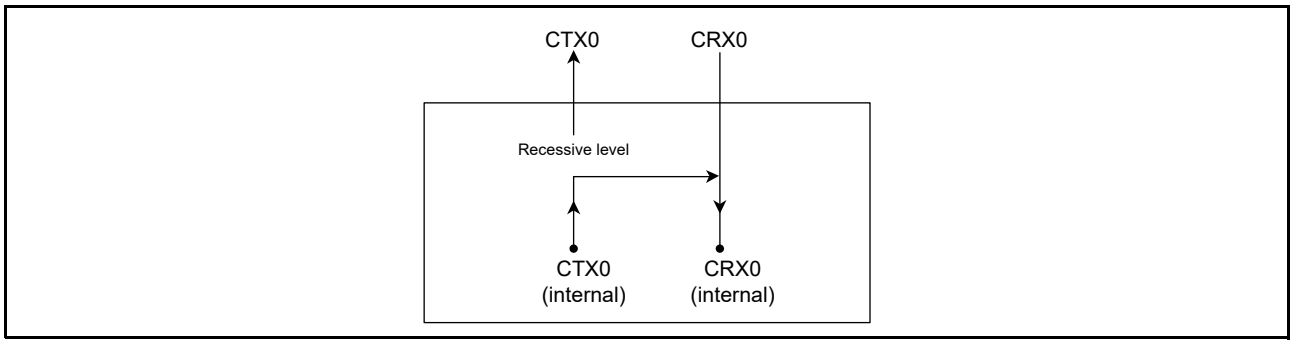


Figure 29.6 Connection when listen-only mode is selected

(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 29.7 shows the connection when self-test mode 0 is selected.

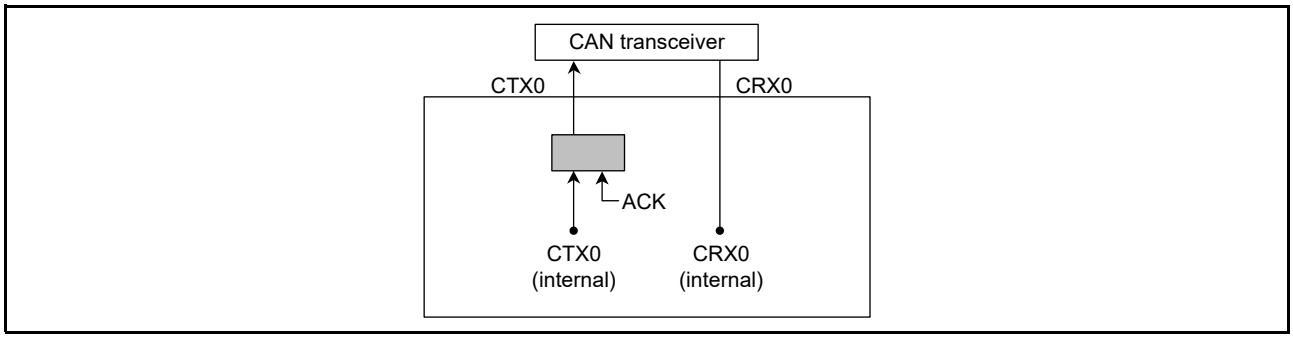


Figure 29.7 Connection when self-test mode 0 is selected

(3) Self-test mode 1 (internal loopback)

Self-test mode 1 is provided for self-test functions. In self-test mode 1, the protocol controller treats its transmitted messages as received messages and stores them into the receive mailbox. To be independent from external stimulation, the protocol controller generates the ACK bit.

In self-test mode 1, the protocol controller performs internal loopback from the internal CTX0 to the internal CRX0. 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 connect to the CAN bus or any external device.

Figure 29.8 shows the connection when self-test mode 1 is selected.

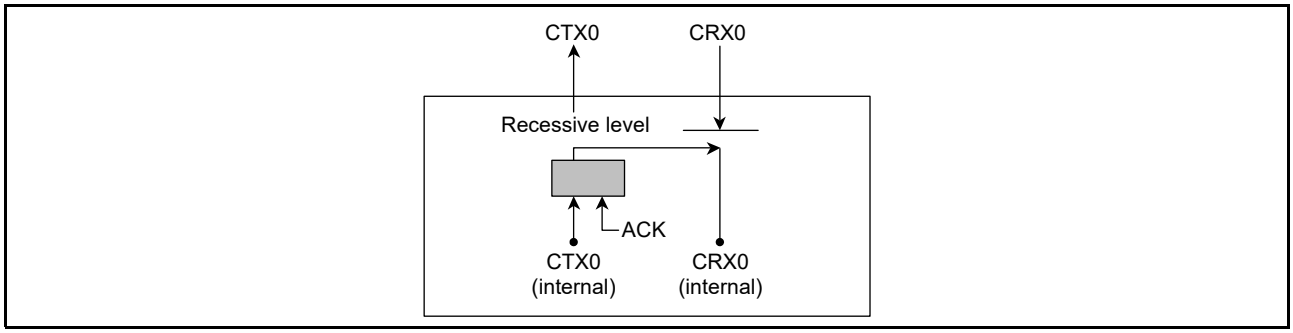


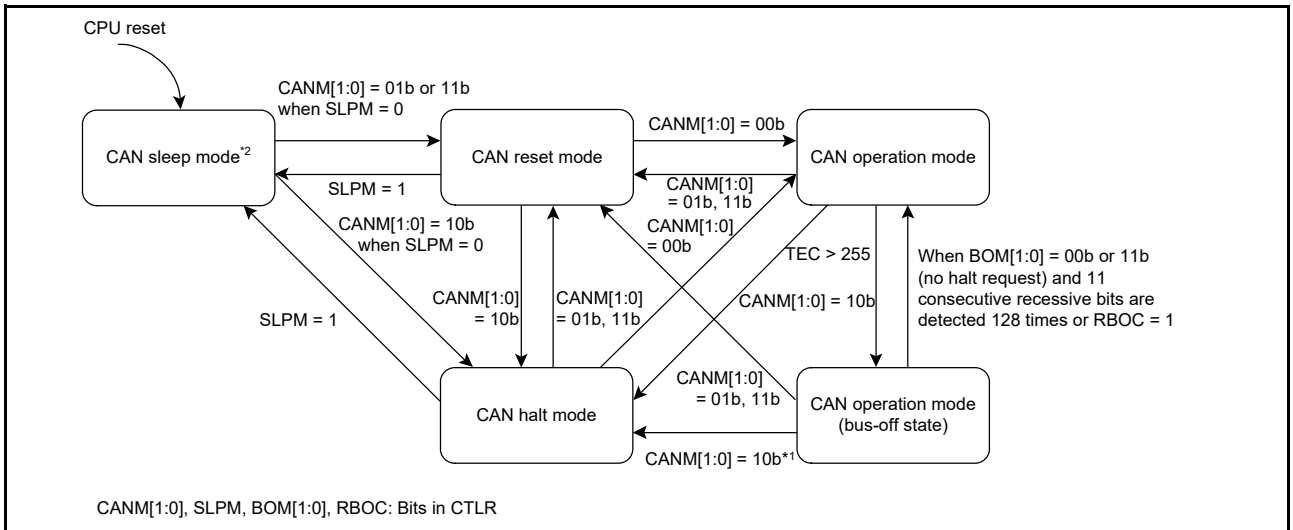
Figure 29.8 Connection when self-test mode 1 is selected

29.3 Operation Modes

The CAN module operation includes the following modes:

- CAN reset mode
- CAN halt mode
- CAN operation mode
- CAN sleep mode.

Figure 29.9 shows the transition between different operation modes.



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 29.9 Transition between different operation modes

29.3.1 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 saved 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.

29.3.2 CAN Halt Mode

CAN halt mode is used for mailbox configuration and test mode setting. When the CTLR.CANM[1:0] bits are set to 10b, CAN halt mode is selected and the STR.HLTST bit is set to 1. Do not change the CTLR.CANM[1:0] bits until the HLTST bit is 1. See Table 29.8 for the state transition conditions when transmitting or receiving.

All registers except for the RSTST, HLTST, and SLPST bits in STR remain unchanged when the CAN enters CAN halt mode. Do not change CTLR (except for the CANM[1:0] and SLPM bits) and EIER 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 29.8 Operation in CAN reset mode and CAN halt mode

Operation mode	Receiver	Transmitter	Bus-off
CAN reset mode (forced transition) CA79	CAN module enters CAN reset mode without waiting for the end of message reception	CAN module enters CAN reset mode without waiting for the end of message transmission	CAN module enters CAN reset mode without waiting for the end of bus-off recovery
CAN reset mode CANM[1:0] = 01b	CAN module enters CAN reset mode without waiting for the end of message reception	CAN module enters CAN reset mode after waiting for the end of message transmission*1,*4	CAN module enters CAN reset mode without waiting for the end of bus-off recovery
CAN halt mode	CAN module enters CAN halt mode after waiting for the end of message reception*2,*3	CAN module enters CAN halt mode after waiting for the end of message transmission*1,*4	• When the BOM[1:0] bits are 00b: A halt request from the software is accepted only after bus-off recovery • When the BOM[1:0] bits are 01b: CAN module automatically enters CAN halt mode without waiting for the end of bus-off recovery, regardless of a halt request from software • When the BOM[1:0] bits are 10b: CAN module automatically enters CAN halt mode after waiting for the end of bus-off recovery, regardless of a halt request from software • When the BOM[1:0] bits are 11b: CAN module enters CAN halt mode, without waiting for the end of bus-off recovery, if a halt is requested by software during bus-off.

Note 1. If transmission of multiple messages is requested, a mode transition occurs after completion of the first transmission. If the CAN reset mode is being requested during suspend transmission, mode transition occurs when the bus is idle, the next transmission ends, or the CAN module becomes a receiver.
Note 2. If the CAN bus is locked at the dominant level, the program can detect this state by monitoring the BLIF flag in EIFR.
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.

29.3.3 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 CTLR 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.

29.3.4 CAN Operation Mode (Excluding Bus-Off State)

CAN operation mode is used for CAN communication. When the CANM[1:0] bits in CTLR 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 0.

If 11 consecutive recessive bits are detected after entering CAN operation mode:

- The CAN module becomes an active node on the network, which enables transmission and reception of CAN messages
- Error monitoring of the CAN bus, such as receive and transmit error counters, is performed.

During CAN operation mode, the CAN module maybe in one of the following three sub-modes, depending on the status of the CAN bus:

- Idle mode: No transmission or reception occurs
- 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] = 11b) is selected.

Figure 29.10 shows the sub-modes of CAN operation mode.

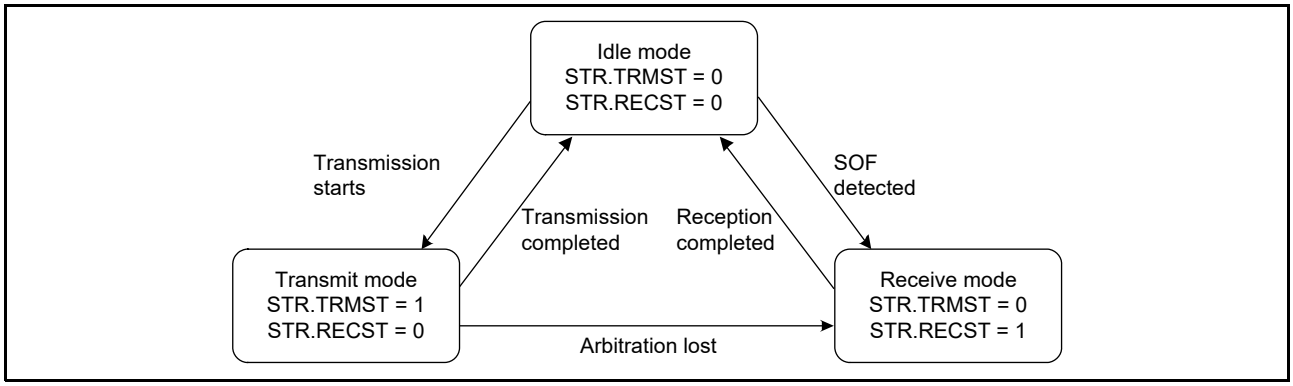


Figure 29.10 Sub-modes of CAN operation mode

29.3.5 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 in the CAN-related registers remain unchanged, except for those in STR, EIFR, RECR, TECR, and TSR.

(1) When the CTLR.BOM[1:0] bits = 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 EIFR is set to 1 (bus-off recovery detected).

(2) When the CTLR.RBOC bit = 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 1. CAN communication is enabled again after 11b consecutive recessive bits are detected. The BORIF flag is not set to 1.

(3) When the CTLR.BOM[1:0] bits = 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 CTLR.BOM[1:0] bits = 10b (automatic transition to CAN halt mode on bus-off end)

The CAN module enters CAN halt mode when it completes recovery from bus-off. The BORIF flag is set to 1.

(5) When CTLR.BOM[1:0] bits = 11b (automatic transition to CAN halt mode through software) and the CTLR.CANM[1:0] bits = 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.

29.4 Data Transfer Rate Configuration

This section describes how to configure the data transfer rate.

29.4.1 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 29.11 shows a block diagram of the CAN clock generator.

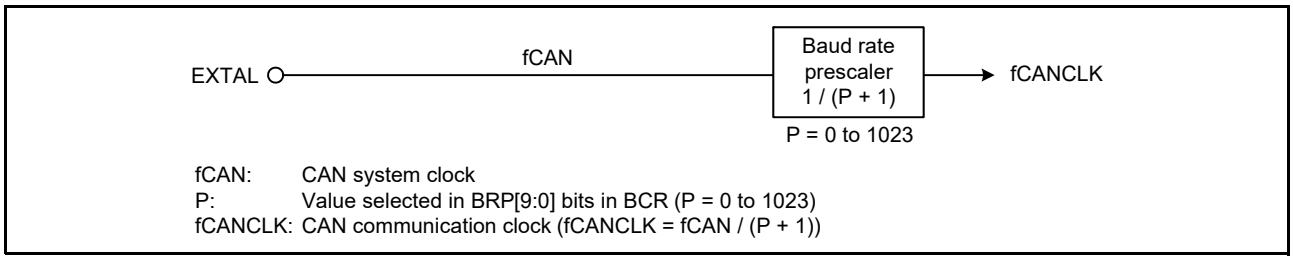
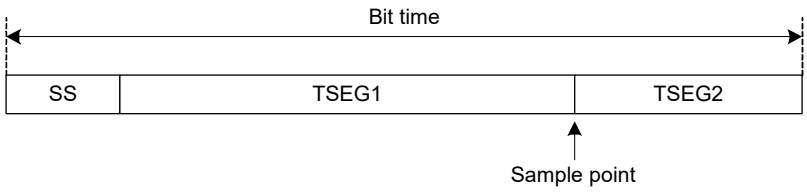


Figure 29.11 Block diagram of CAN clock generator

29.4.2 Bit Time Setting

The bit timing consists of three segments as shown in Figure 29.12.



Range of each segment: Bit time = 8 to 25 Tq
SS = 1 Tq
TSEG1 = 4 to 16 Tq
TSEG2 = 2 to 8 Tq
SJW = 1 to 4 Tq

TSEG1 and TSEG2 settings: TSEG1 > TSEG2 ≥ SJW

Figure 29.12 Bit timing

29.4.3 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.

$$\text{Data transfer rate [bps]} = \frac{\text{fCAN}}{\text{Baud rate prescaler division value*1} \times \text{Tq count for 1 bit time}} = \frac{\text{fCANCLK}}{\text{Tq count for 1 bit time}}$$

Note 1. Division value of baud rate prescaler = P + 1 (P: 0 to 1023), where P is the BRP[9:0] setting in BCR.

Table 29.9 lists data transfer rate examples.

Table 29.9 Data transfer rate examples when fCAN = 20 MHz

Data transfer rate	Tq Count	P + 1
1 Mbps	5 Tq 10 Tq	4 2
500 kbps	5 Tq 10 Tq	8 4
250 kbps	5 Tq 10 Tq	16 8
125 kbps	5 Tq 10 Tq	32 16
83.3 kbps	5 Tq 10 Tq	48 24
33.3 kbps	5 Tq 8 Tq 10 Tq	120 75 60

29.5 Mailbox and Mask Register Structure

Figure 29.13 shows the structure of the 32 mailbox registers MBj_ID, MBj_DL, MBj_Dm, and MBj_TS.

b7							b0	Address
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 29.13 Structure of the mailbox registers (j = 0 to 31)

Figure 29.14 shows the structure of the eight mask registers MKRk.

b7							b0	Address
			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 29.14 Structure of the MKRk registers (k = 0 to 7)

Figure 29.15 shows the structure of the two FIFO received ID compare registers, FIDCR0 and FIDCR1.

b7							b0	Address
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 29.15 Structure of the FIDCRn registers (n = 0, 1)

29.6 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 CTLR 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 standard ID and extended ID values are set in the IDE bits in FIDCR0 and FIDCR1, both ID formats are received. If different data and remote frame values are set in the RTR bits in FIDCR0 and FIDCR1, 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 29.16 shows the associations between mask registers and mailboxes. Figure 29.17 shows the acceptance filtering.

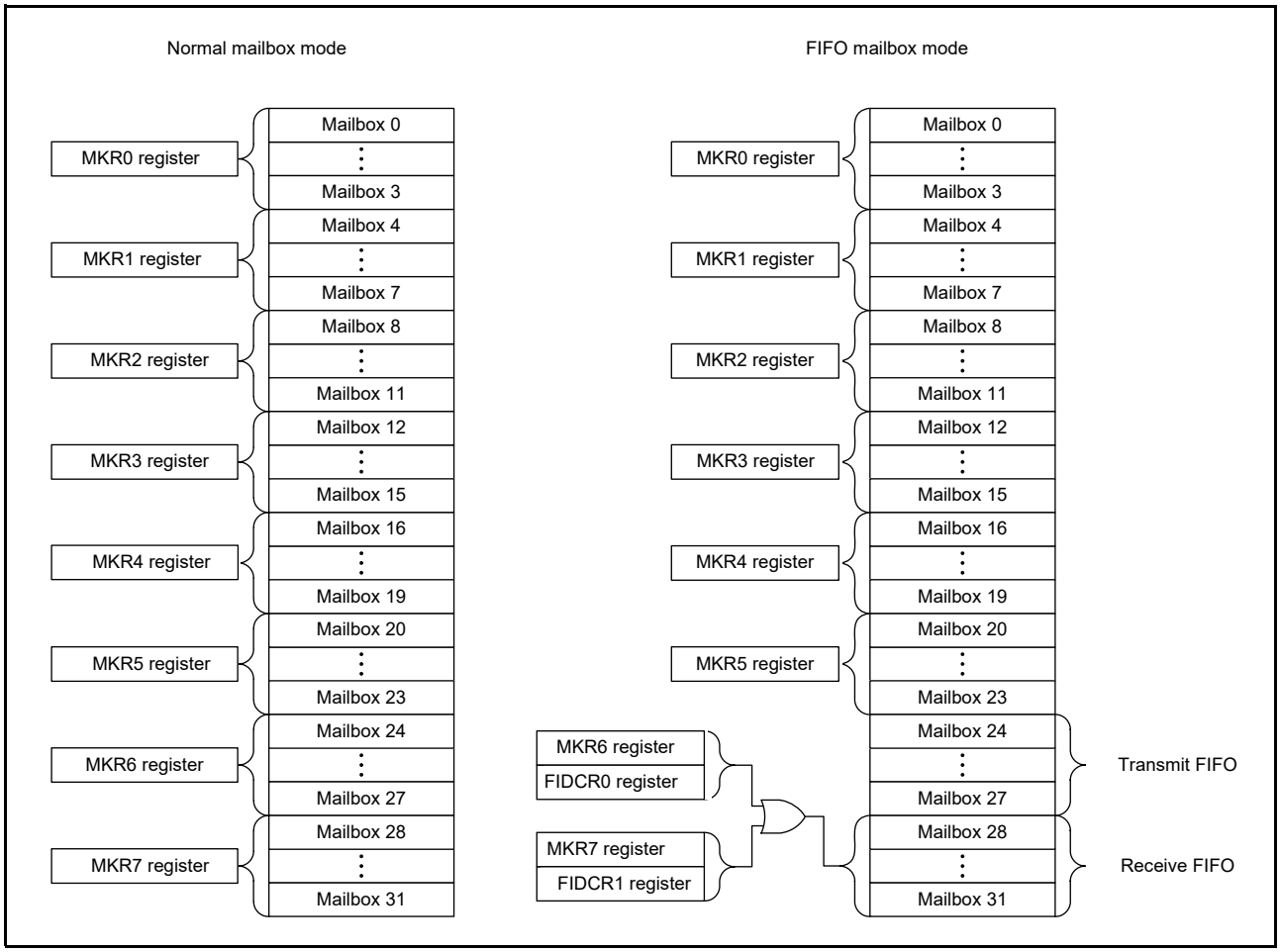


Figure 29.16 Associations between mask registers and mailboxes

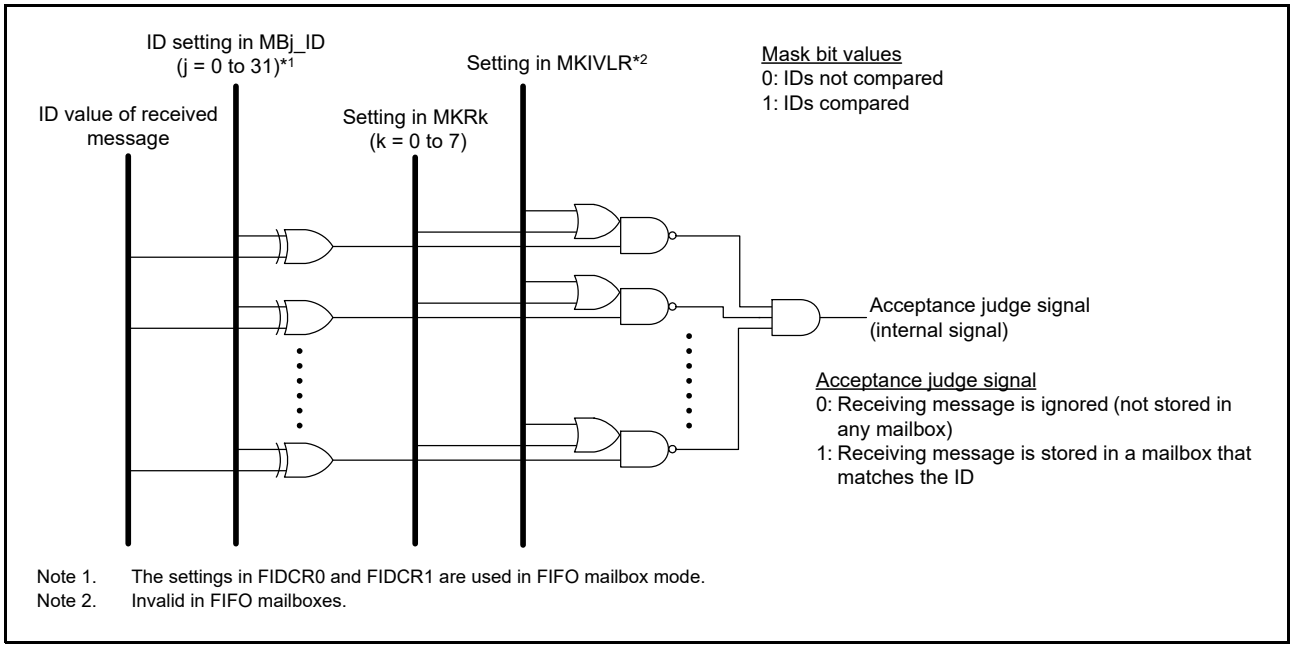


Figure 29.17 Acceptance filtering

29.7 Reception and Transmission

Table 29.10 lists the CAN communication mode settings.

Table 29.10 Settings for CAN receive and transmit modes

MCTL_TXj. and MCTL_RXj. TRMREQ	MCTL_TXj. and MCTL_RXj. RECREQ	MCTL_TXj. and MCTL_RXj. 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:

- Before configuring the mailbox, set MCTL_RXj to 00h
- A received message is stored into the first mailbox that matches the conditions resulting from the receive mode settings and acceptance filtering. The matching mailbox with the smallest number takes priority for storing the received message
- 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 MCTL_TXj is 00h and that there is no pending abort process.

29.7.1 Reception

Figure 29.18 shows an operation example of data frame reception in overwrite mode.

The example shows the overwriting of the first message when the CAN module receives two consecutive CAN messages that match the receiving conditions in MCTL_RXj (j = 0 to 31).

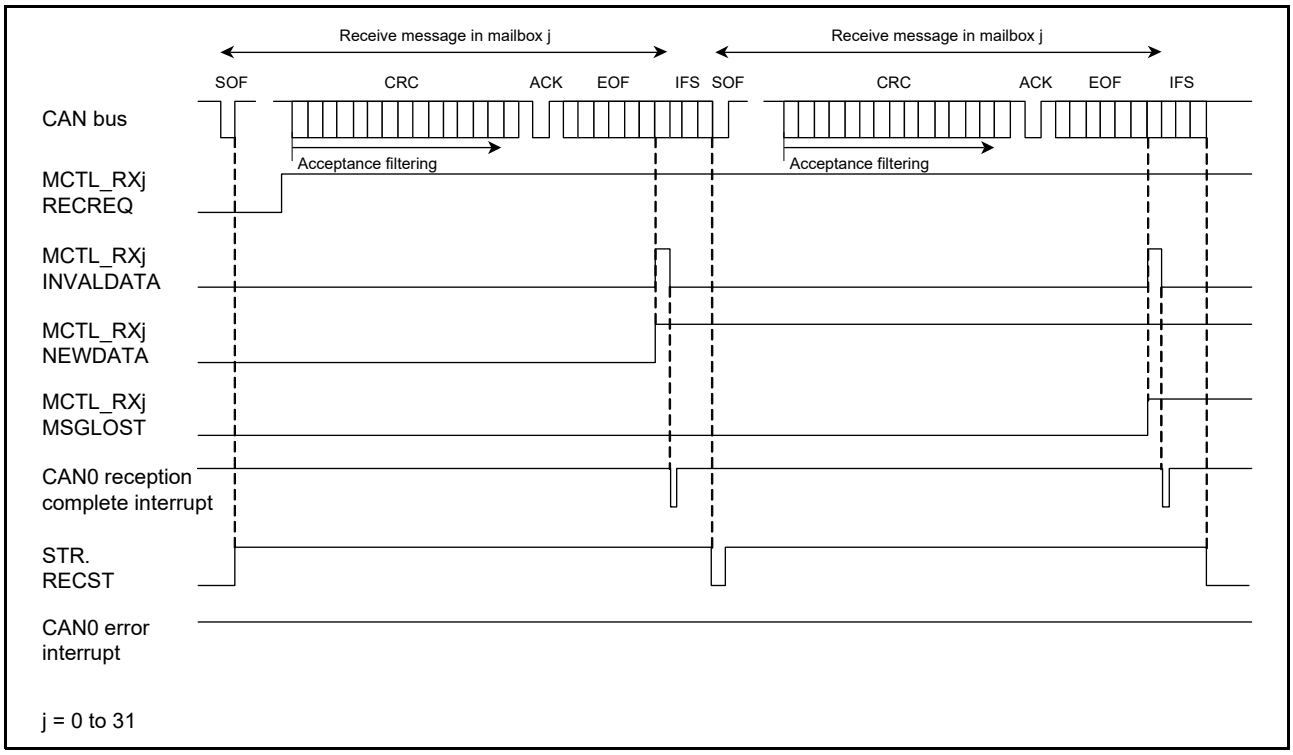


Figure 29.18 Operation example of data frame reception in overwrite mode

1. When an SOF is detected on the CAN bus, the RECST bit in STR is set to 1 (reception in progress) if the CAN module has no message ready to start transmission.
2. Acceptance filtering starts at the beginning of the CRC field to select the receive mailbox.
3. After a message is received, the NEWDATA flag in MCTL_RXj for the receive mailbox is set to 1 (new message is being stored or was stored to the mailbox). The INVALDATA flag in MCTL_RXj is set to 1 (message is 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 MIER for the receive mailbox is 1 (interrupt enabled), the INVALDATA flag is set to 0, triggering 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 flag 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 29.19 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 in MCTL_RXj (j = 0 to 31).

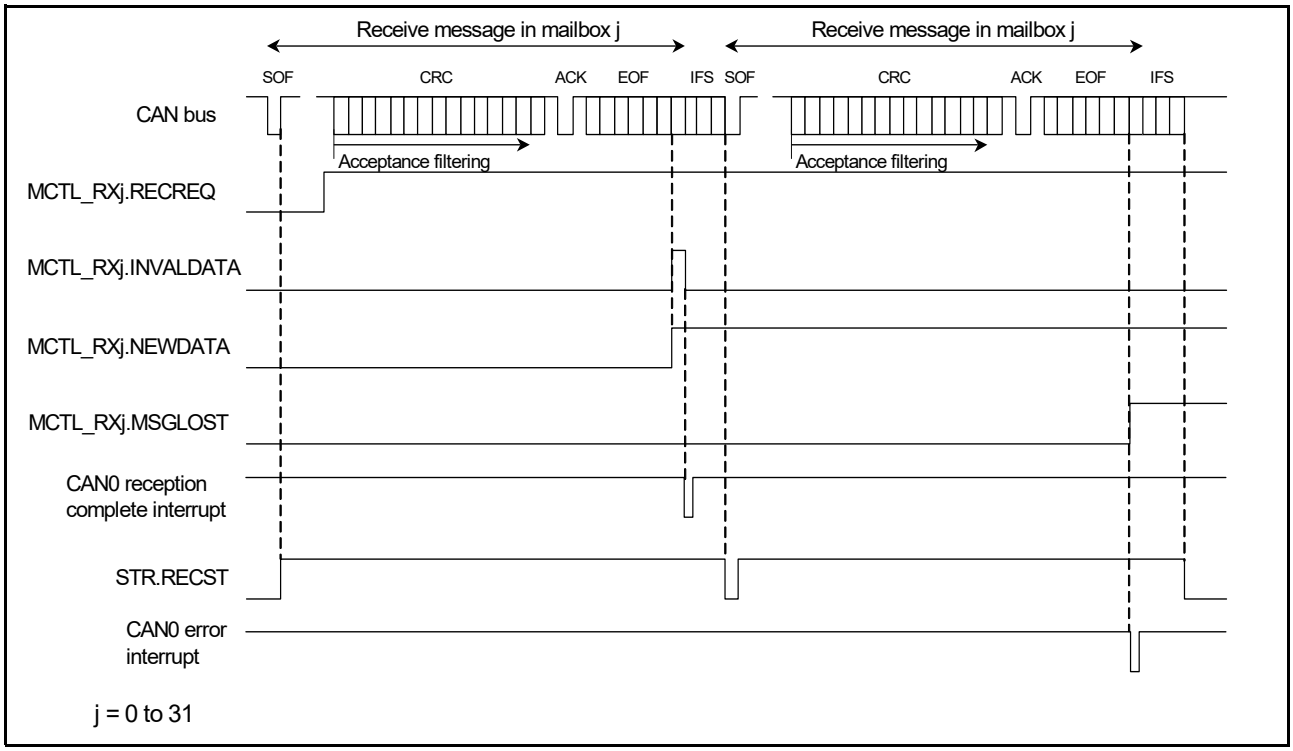


Figure 29.19 Operation example of data frame reception in overrun mode

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 occurs if the associated interrupt enable bit in EIER is 1 (interrupt enabled).

29.7.2 Transmission

Figure 29.20 shows an operation example of data frame transmission.

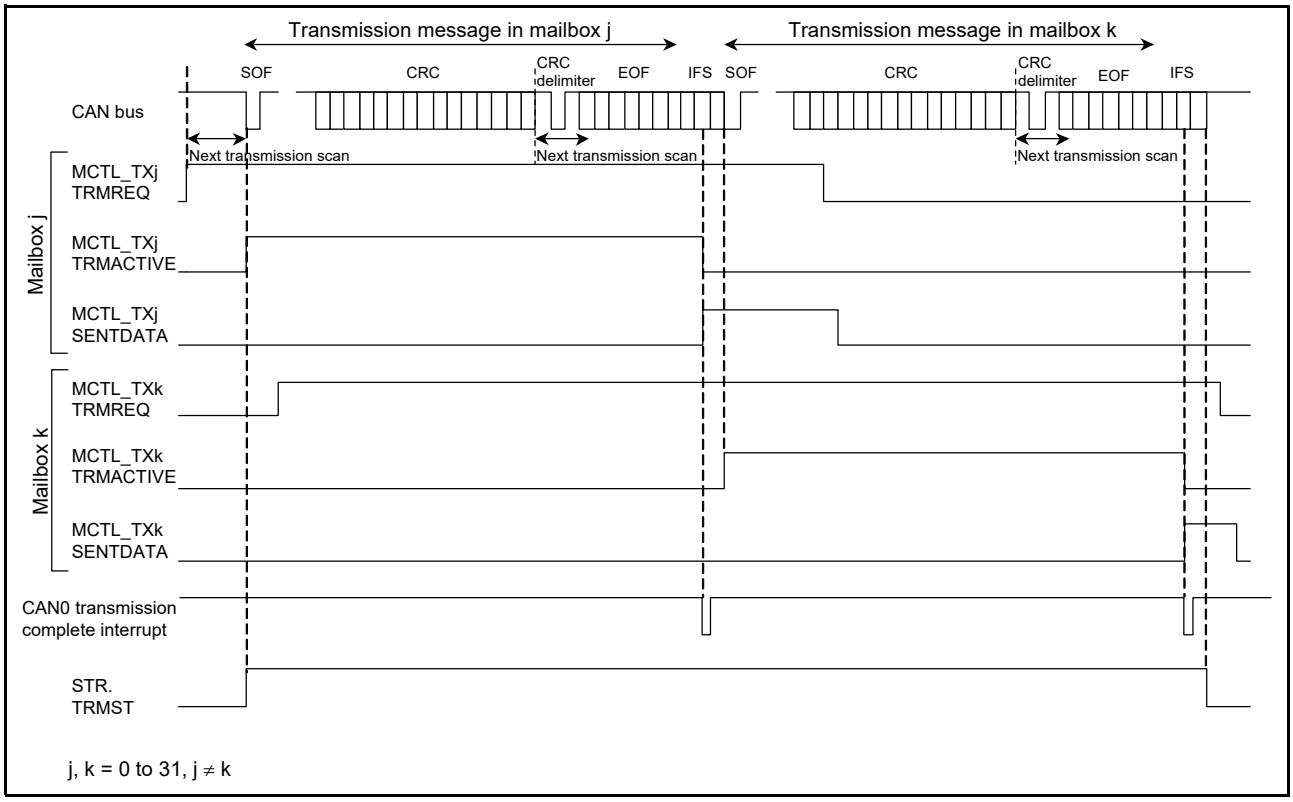


Figure 29.20 Operation example of data frame transmission

1. When a TRMREQ bit in MCTL_TXj (j = 0 to 31) is set to 1 (transmit mailbox) in the bus-idle state, mailbox scanning 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 complete without losing arbitration, the SENTDATA flag in MCTL_TXj is set to 1 (transmission completes) and the TRMACTIVE flag is set to 0 (transmission is pending or transmission is not requested). If the interrupt enable bit in MIER is 1 (interrupt enabled), the CAN0 transmission complete interrupt request is generated.
4. When requesting the next transmission from the same mailbox, set the SENTDATA flag and TRMREQ bit to 0, then set the TRMREQ bit to 1 after checking that the SENTDATA flag and TRMREQ bit 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.

29.8 Interrupts

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 29.11 lists the CAN interrupts.

Table 29.11 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 buffer warning (MIER_FIFO.MB29 = 1)	
	CAN0_TXF	Transmit FIFO message transmission complete (MIER_FIFO.MB25 = 0)	TFCR.TFUST[2:0]
		FIFO last message transmission complete (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 complete	MCTL_TX0.SENTDATA to MCTL_TX31.SENTDATA

29.9 Usage Notes

29.9.1 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.

29.9.2 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:

  PCLKB ≥ CANMCLK
- The clock frequency ratio of ICLK and PCLKB must be 2:1 when using the CAN module. Operation is not guaranteed for other settings.

30. Serial Peripheral Interface (SPI)

30.1 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 30.1 lists the SPI specifications, Figure 30.1 shows a block diagram, and Table 30.2 shows the I/O pins.

In this section, *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.

Table 30.1 SPI specifications (1 of 2)

Parameter	Specifications
Number of channels	Two channels
SPI transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (SPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method) • Transmit-only operation available • Communication mode selectable to full-duplex or transmit-only • 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 • 32-bit transmit and receive buffers.
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing PCLKB (the division ratio ranges from divided by 2 to divided by 4096) • In slave mode, the minimum PCLKB clock divided by 6 can be input as RSPCK (the maximum RSPCK frequency is that of PCLKB divided by 6) Width at high level: 3 PCLKB cycles Width at low level: 3 PCLKB cycles.
Buffer configuration	• Double buffer configuration for the transmit and receive buffers • 32 bits for the transmit and receive buffers.
Error detection	• Mode fault error detection • Underrun error detection • Overrun error detection*1 • Parity error detection.
SSL control function	• Four SSL pins (SSLn0 to SSLn3) for each channel • In single-master mode, SSLn0 to SSLn3 pins are output • In multi-master mode, SSLn0 pin for output 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.
Control in master transfer	• For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, MSB- or LSB-first, RSPCK delay, SSL negation delay, and next-access delay • Transfers can be initiated by writing to the transmit buffer • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function.
Interrupt sources	• Receive buffer full interrupt • Transmit buffer empty interrupt • SPI error interrupt (mode fault, overrun, parity error) • SPI idle interrupt (SPI idle) • Transmission-completed interrupt.

Table 30.1 SPI specifications (2 of 2)

Parameter	Specifications
Event link function (output)	The following events can be output to the Event Link Controller (ELC): • Receive buffer full signal • Transmit buffer empty signal • Mode fault, underrun, overrun, or parity error signal • SPI idle • Transmission-completed signal.
Other functions	• SPI initialization function • 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.

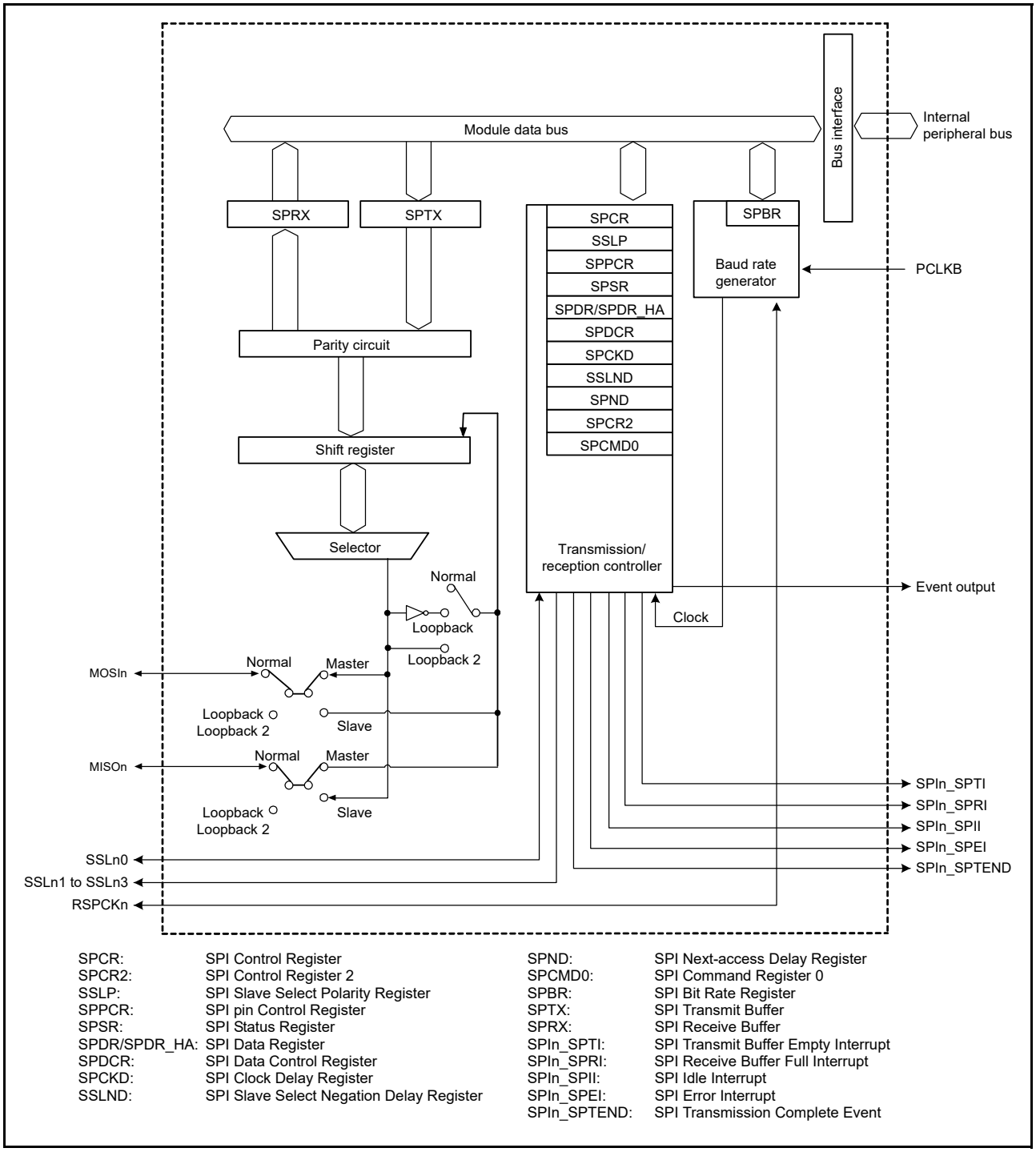


Figure 30.1 SPI block diagram

Table 30.2 lists the I/O pins used in the SPI. The SPI automatically switches the I/O direction of the SSLn0 pin. SSLn0 is an output when the SPI is a single master and an input when the SPI is a multi-master or a slave. The RSPCKn, MOSIn, and MISOn pins are automatically set as inputs or outputs based on the master or slave setting and the level input on the SSLn0 pin. For details, see section 30.3.2, Controlling the SPI Pins.

Table 30.2 SPI I/O pins

Channel	Pin name	I/O	Function
SPI0	RSPCKA	I/O	Clock input/output
	MOSIA	I/O	Master transmit data input/output
	MISOA	I/O	Slave transmit data input/output
	SSLA0	I/O	Slave selection input/output
	SSLA1	Output	Slave selection output
	SSLA2	Output	Slave selection output
	SSLA3	Output	Slave selection output
SPI1	RSPCKB	I/O	Clock input/output
	MOSIB	I/O	Master transmit data input/output
	MISOB	I/O	Slave transmit data input/output
	SSLB0	I/O	Slave selection input/output
	SSLB1	Output	Slave selection output
	SSLB2	Output	Slave selection output
	SSLB3	Output	Slave selection output

30.2 Register Descriptions

30.2.1 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 with transmit only.	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. For clock synchronous operation in master mode (SPCR.MSTR = 1), the SPCMD0.CPHA bit can be set to either 0 or 1. For clock synchronous operation in slave mode (SPCR.MSTR = 0), set the CPHA bit to 1. Do not perform the operations if the CPHA bit is set to 0 when clock synchronous operation is in slave mode.

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 30.3.6, Data Transfer Modes), and the 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 30.3.8, 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 30.3.2, 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 one of the following occurs:

- The SPI detects a mode fault error or underrun error and sets the SPSR.MODF flag to 1
- The SPI detects an overrun error and sets the SPSR.OVRF flag to 1
- The SPI detects a parity error and sets the SPSR.PERF flag to 1.

For details, see section 30.3.8, Error Detection.

SPTIE bit (Transmit Buffer Empty Interrupt Enable)

The SPTIE bit enables or disables the generation of transmit buffer empty interrupt requests when the SPI detects that the transmit buffer is empty. To generate a transmit buffer empty interrupt request when transmission starts, set the SPE and SPTIE bits to 1 at the same time or set the SPE bit to 1 after setting the SPTIE bit to 1.

When the SPTIE bit is 1, transmit buffer interrupts are generated even when the SPI function is disabled (when the SPE bit is changed to 0).

SPE bit (SPI Function Enable)

The SPE bit enables or disables the SPI function. The SPE bit cannot be set to 1 when the SPSR.MODF flag is 1. For details, see section 30.3.8, Error Detection.

Setting the SPE bit to 0 disables the SPI function and initializes a part of the module function. For details, see section 30.3.9, Initializing the SPI. In addition, a transmit buffer empty interrupt request is generated when the SPE bit is changed from 0 to 1 or 1 to 0.

SPRIE bit (SPI Receive Buffer Full Interrupt Enable)

The SPRIE bit enables or disables the generation of an SPI receive buffer full interrupt request when the SPI detects a receive buffer full write after completion of a serial transfer.

30.2.2 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: Set SSLn0 signal to active-low 1: Set SSLn0 signal to active-high.	R/W
b1	SSL1P	SSL1 Signal Polarity Setting	0: Set SSLn1 signal to active-low 1: Set SSLn1 signal to active-high.	R/W
b2	SSL2P	SSL2 Signal Polarity Setting	0: Set SSLn2 signal to active-low 1: Set SSLn2 signal to active-high.	R/W
b3	SSL3P	SSL3 Signal Polarity Setting	0: Set SSLn3 signal to active-low 1: Set SSLn3 signal to active-high.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the contents of SSLP are changed when the SPCR.SPE bit is 1, do not perform subsequent operations.

30.2.3 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 SPPCR are changed when the SPCR.SPE bit is 1, do not perform subsequent operations.

SPLP bit (SPI Loopback)

The SPLP bit selects the mode of the SPI pins. When this bit is set to 1, 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 and inverts 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 this bit is set to 1, 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 2.

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.

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. When the MOIFE bit is 0, the SPI outputs to the MOSIn pin the last data from the previous serial transfer during the SSL negation period. When the MOIFE bit is 1, the SPI outputs the fixed value set in the MOIFV bit to the MOSIn pin.

30.2.4 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 occurred 1: Overrun error occurred.	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 occurred 1: A mode fault error or underrun error occurred.	R/(W)*1
b3	PERF	Parity Error Flag	0: No parity error occurred 1: A parity error occurred.	R/(W)*1
b4	UDRF	Underrun Error Flag	0: A mode fault error occurred (MODF = 1) 1: An underrun error occurred (MODF = 1). This bit is invalid when MODF is 0.	R/W*1,*2
b5	SPTEF	SPI Transmit Buffer Empty Flag	0: Data is in the transmit buffer 1: No data is 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 is in SPDR/SPDR_HA 1: Valid data is in SPDR/SPDR_HA.	R/(W)*3

Note 1. Only 0 can be written to clear the flag after reading 1.
Note 2. Clear the UDRF flag at the same time as the MODF flag.
Note 3. The write value should be 1.

OVRF flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error. In master mode (SPCR.MSTR = 1) and when the RSPCK clock auto-stop function is enabled (SPCR2.SCKASE = 1), an overrun error does not occur, and this flag is not set to 1. For details, see section 30.3.8.1, Overrun errors.

[Setting condition]

- When the next serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer is full.

[Clearing condition]

- When SPSR is read while this flag is 1, and then 0 is written to this flag.

IDLNF flag (SPI Idle Flag)

The IDLNF flag indicates the transfer status of the SPI.

[Setting condition]

Master mode:

- When conditions 1. and 2. in the master mode, in the clearing conditions, are not satisfied.

Slave mode:

- When the SPCR.SPE bit is 1, enabling the SPI function.

[Clearing conditions]

Master mode:

- When condition 1. or conditions 2. and 3. are satisfied.

1. The SPCR.SPE bit is 0, indicating that the SPI is initialized.
2. The transmit buffer (SPTX) is empty, indicating that data for the next transfer is not set.
3. The SPI internal sequencer is in the idle state, indicating that operations up to the next-access delay are complete.

Slave mode:

- When condition 1. is satisfied.

MODF flag (Mode Fault Error Flag)

The MODF flag indicates the occurrence of a mode fault error or an underrun error. The UDRF flag indicates which error occurred.

[Setting conditions]

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 enabled), triggering a mode fault error.

Slave mode:

- When condition 1. or 2. is satisfied.

  1.The SSLni pin is negated before the RSPCK cycle required for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection 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, and then 0 is written to this flag.

PERF flag (Parity Error Flag)

The PERF flag indicates the occurrence of a parity error.

[Setting condition]

- When a serial transfer ends while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1, triggering a parity error.

[Clearing condition]

- When SPSR is read while this flag is 1, and then 0 is written to this flag.

UDRF flag (Underrun Error Flag)

The UDRF flag indicates the occurrence of an underrun error.

[Setting condition]

- When the serial transfer begins with the SPCR.MSTR bit 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 while this flag is 1, and then 0 is written to this flag.

SPTEF flag (SPI Transmit Buffer Empty Flag)

The SPTEF flag indicates the status of the transmit buffer for the SPI Data Register (SPDR/SPDR_HA).

[Setting conditions]

- When condition 1. or 2. is satisfied.
  1. The SPCR.SPE bit is 0, indicating that the SPI is initialized.
  2. Transmit data is transferred from the transmit buffer to the shift register.

[Clearing condition]

- When data is 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 condition]

- When the SPI transfers receive data 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 is not changed from 0 to 1.

[Clearing condition]

- When received data is read from SPDR/SPDR_HA.

30.2.5 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 30.2 shows the configuration of the SPDR/SPDR_HA register.

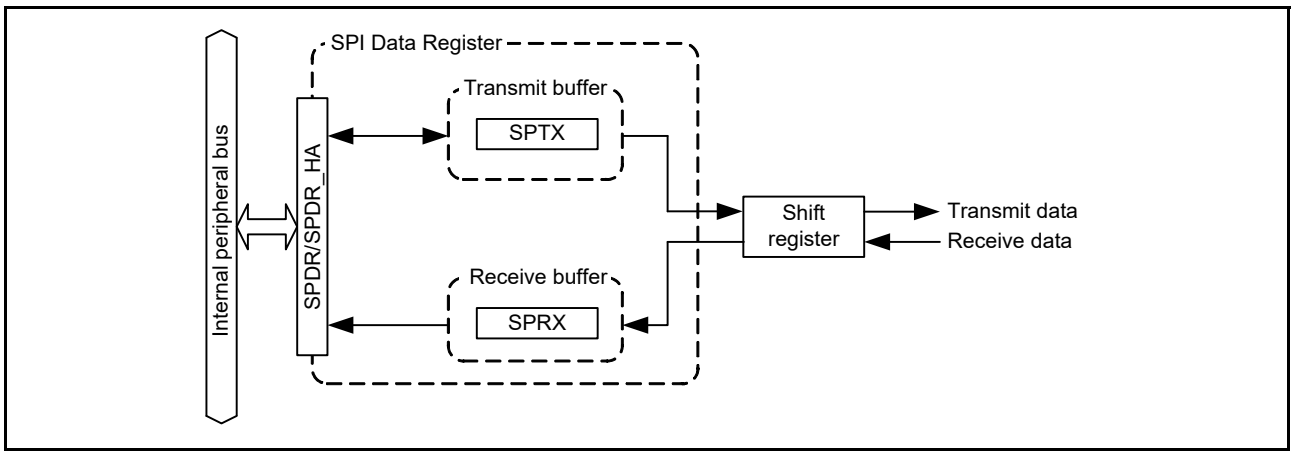


Figure 30.2 Configuration of SPDR/SPDR_HA

The transmit and receive buffers each have one stage. The two stages of the buffer are all mapped to the single address of SPDR/SPDR_HA.

Data written to SPDR/SPDR_HA is written to a transmit-buffer stage (SPTX) 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 occurs.

If the data length is not 32 bits, the bits not referred to in SPTX are stored in the associated bits in SPRX. For example, if the data length is 9 bits, received data is stored in the SPRX[8:0] bits and the SPTX[31:9] bits are stored in the SPRX[31:9] bits.

(1) Bus interface

SPDR/SPDR_HA is an interface with 32-bit wide transmit and receive buffers, each of which has one stage, for a total of 8 bytes. The 8 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 Halfword Access Specification bit in the SPI Data Control Register (SPDCR.SPLW).

Flush transmission data at the LSB end of the register and store received data at the LSB end.

The following sections describe the operations involved in writing to and reading from SPDR/SPDR_HA.

(a) Writing

Data written to SPDR/SPDR_HA is written to a transmit buffer SPTX. This is not affected by the value of the SPDCR.SPRDTD bit, unlike when reading from SPDR/SPDR_HA.

Figure 30.3 shows the configuration of the bus interface with the transmit buffer when writing to SPDR/SPDR_HA.

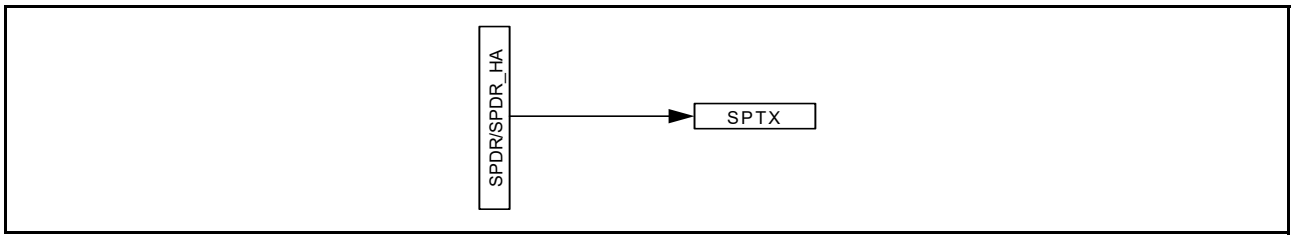


Figure 30.3 Configuration of SPDR/SPDR_HA for write access

Write the transfer data to SPTX after generating the transmit buffer empty interrupt (SPSR.SPTEF is 1). Even when data is written to SPDR/SPDR_HA, the value of the buffer is not updated after completion of the writing and before generation of the next transmit buffer empty interrupt (SPTEF is 0).

(b) Reading

SPDR/SPDR_HA can be accessed to read the value of a receive buffer (SPRX) or a transmit buffer (SPTX). The setting in the SPI Receive or Transmit Data Select bit in the SPI Data Control Register (SPDCR.SPRDTD) selects whether reading is from the receive or transmit buffer.

Figure 30.4 shows the configuration of a bus interface with the receive and transmit buffers for reading from SPDR/SPDR_HA.

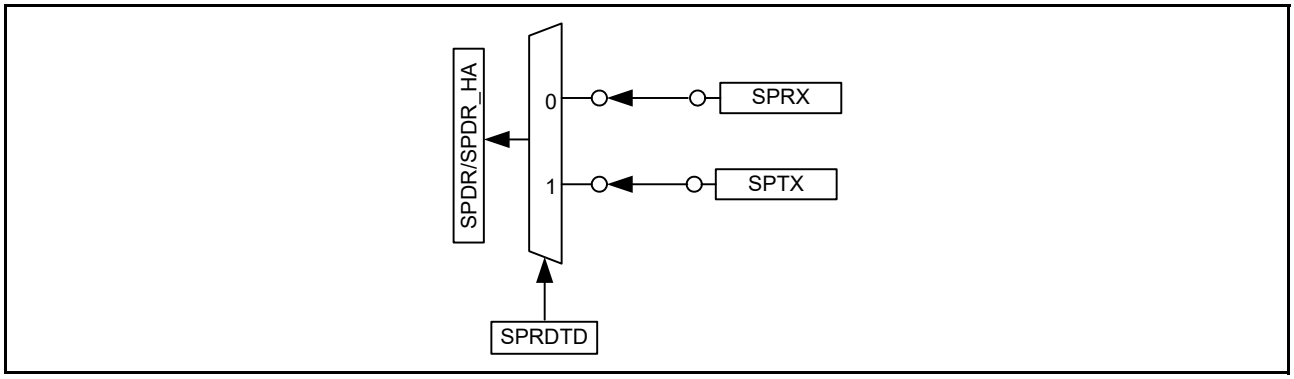


Figure 30.4 Configuration of SPDR/SPDR_HA for read access

After a transmit buffer empty interrupt is generated, the values read from the buffer are all 0s in the interval after completion of writing the data and before generation of the next buffer empty interrupt (when SPSR.SPTEF is 0).

30.2.6 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 SPCMD0.BRDV[1:0] bits (bit rate division setting). Use bit rates that satisfy the electrical characteristics of the device.

The bit rate is determined by combination of the SPBR and SPCMD0.BRDV[1:0] settings in the SPI Command Register. The equation for calculating the bit rate is as follows:

$$\text{Bit rate} = \frac{f\,(\text{PCLKB})}{2 \times (n + 1) \times 2^{N}}$$

In the equation, n denotes an SPBR setting (0, 1, 2, …, 255), and N denotes a BRDV[1:0] setting (0, 1, 2, or 3).

Table 30.3 lists examples of the relationship between the SPBR settings, BRDV[1:0] settings, and bit rates.

Table 30.3 Relationship between SPBR settings, BRDV[1:0] settings, and bit rates (1 of 2)

SPBR register (n)	BRDV[1:0] bits (N)	Division ratio	Bit rate when PCLKB = 32 MHz
0	0	2	16.0 Mbps
1	0	4	8.00 Mbps
2	0	6	5.33 Mbps

Table 30.3 Relationship between SPBR settings, BRDV[1:0] settings, and bit rates (2 of 2)

SPBR register (n)	BRDV[1:0] bits (N)	Division ratio	Bit rate when PCLKB = 32 MHz
3	0	8	4.00 Mbps
4	0	10	3.20 Mbps
5	0	12	2.67 Mbps
5	1	24	1.33 Mbps
5	2	48	667 kbps
5	3	96	333 kbps
255	3	4096	7.81 kbps

30.2.7 SPI Data Control Register (SPDCR)

Address(es): SPI0.SPDCR 4007 200Bh, SPI1.SPDCR 4007 210Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SPBYT	SPLW	SPRDTD	—	—	—	—
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	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
b6	SPBYT	SPI Byte Access Specification	0: SPDR is accessed in halfword or word (SPLW is valid). 1: SPDR is accessed in byte (SPLW is invalid).	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

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 must be done after generation of the transmit buffer empty interrupt (SPSR.SPTEF is 1).

For details, see section 30.2.5, SPI Data Register (SPDR/SPDR_HA).

SPLW bit (SPI Word Access/Halfword Access Specification)

The SPLW bit specifies the access width for the SPDR register. Access to SPDR_HA in halfwords is valid when the SPLW bit is 0 and access to the SPDR register in words is valid when the SPLW bit is 1. In addition, when the SPLW bit is 0, set the SPI data length setting bits, SPCMD0.SPB[3:0], from 8 to 16 bits. Do not perform any operations when a data length of 20, 24, or 32 bits is specified.

SPBYT bit (SPI Byte Access Specification)

This bit is used to set the data width of access to the SPI Data Register (SPDR). When SPBYT = 0, use word or half word access to SPDR. When SPBYT = 1 (in that case, SPLW is invalid), use byte access to SPDR. When SPBYT = 1, set the SPI data length bits (SPB[3:0]) in the SPI Command Register 0 (SPCMD0) to 8 bits. If SPB[3:0] are set to 9 to 16, 20, 24, or 32 bits, subsequent operation is not guaranteed.

30.2.8 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

SPCKD specifies the RSPCK delay, the period from the beginning of SSLni signal assertion to RSPCK oscillation, when the SPCMD0.SCKDEN bit is 1. If the contents of SPCKD 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 specify an RSPCK delay value when the SPCMD0.SCKDEN bit is 1. When using the SPI in slave mode, set the SCKDL[2:0] bits to 000b.

30.2.9 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 specifies the SSL negation delay, the period from the transmission of a final RSPCK edge to the negation of the SSLni signal during a serial transfer by the SPI in master mode. If the contents of SSLND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, do not perform subsequent operations.

SLNDL[2:0] bits (SSL Negation Delay Setting)

The SLNDL[2:0] bits specify an SSL negation delay value when the SPI is in master mode. When using the SPI in slave mode, set the SLNDL[2:0] bits to 000b.

30.2.10 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 PCLKB 0 0 1: 2 RSPCK + 2 PCLKB 0 1 0: 3 RSPCK + 2 PCLKB 0 1 1: 4 RSPCK + 2 PCLKB 1 0 0: 5 RSPCK + 2 PCLKB 1 0 1: 6 RSPCK + 2 PCLKB 1 1 0: 7 RSPCK + 2 PCLKB 1 1 1: 8 RSPCK + 2 PCLKB.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPND register specifies the next-access delay, the non-active period of the SSLni signal after termination of a serial transfer, when the SPCMD0.SPNDEN bit is 1. If the contents of SPND 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 specify a next-access delay when the SPCMD0.SPNDEN bit is 1. When using the SPI in slave mode, set the SPNDL[2:0] bits to 000b.

30.2.11 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: When SPCR.TXMD = 0, parity bit added to transmit data and parity bit of receive data checked When SPCR.TXMD = 1, parity bit added to transmit data but parity bit of receive data not checked.	R/W
b1	SPOE	Parity Mode	0: Even parity selected for transmission and reception 1: Odd parity selected for transmission and reception.	R/W
b2	SPIIE	SPI Idle Interrupt Enable	0: 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 SPCR.SPE bit 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 idle state is detected in the SPI and the SPSR.IDLNF flag is set to 0.

PTE bit (Parity Self-Testing)

The PTE bit enables self-diagnosis of the parity circuit to check whether the parity function is operating correctly.

SCKASE bit (RSPCK Auto-Stop Function Enable)

The SCKASE bit enables or disables the RSPCK auto-stop function. When this function is enabled, the RSPCK clock is stopped before an overrun error occurs, when data is received in master mode. For details, see section 30.3.8.1, Overrun errors.

30.2.12 SPI Command Register 0 (SPCMD0)

Address(es): SPI0.SPCMD0 4007 2010h, SPI1.SPCMD0 4007 2110h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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 RSPCKn low when idle 1: Set RSPCKn high when idle.	R/W
b3, b2	BRDV[1:0]	Bit Rate Division Setting	b3 b2 0 0: Select the base bit rate 0 1: Select the base bit rate divided by 2 1 0: Select the base bit rate divided by 4 1 1: Select the base bit rate divided by 8.	R/W
b6 to b4	SSLA[2:0]	SSL Signal Assertion Setting	b6 b4 0 0 0: SSLn0 0 0 1: SSLn1 0 1 0: SSLn2 0 1 1: SSLn3 1 x x: Setting prohibited. x: Don't care.	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
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: Select next-access delay of 1 RSPCK + 2 PCLKB 1: Select 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: Select SSL negation delay of 1 RSPCK 1: Select 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: Select RSPCK delay of 1 RSPCK 1: Select RSPCK delay equal to the setting in the SPI Clock Delay register (SPCKD).	R/W

The SPCMD0 register specifies the transfer format for the SPI in master mode.

Set this register while the transmit buffer is empty (SPSR.SPTEF is 1 and the data for the next transfer is not set), and before the setting of the data to be transmitted when this register is referenced.

If the contents of SPCMD0 are changed while the SPCR.SPE bit is 1, do not perform subsequent operations.

CPHA bit (RSPCK Phase Setting)

The CPHA bit selects the RSPCK phase of the SPI in master 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 selects 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 by combination of the settings in the BRDV[1:0] bits and the SPBR register. See section 30.2.6, 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. 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.

SPB[3:0] bits (SPI Data Length Setting)

The SPB[3:0] bits specify a transfer data length for the SPI in master or slave mode.

When the SPLW bit is 0, set these bits from 8 to 16 bits.

LSBF bit (SPI LSB First)

The LSBF bit specifies the data format of the SPI in master or slave mode to MSB-first or LSB-first.

SPNDEN bit (SPI Next-Access Delay Enable)

The SPNDEN bit specifies the 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 PCLKB. If the SPNDEN bit is 1, the SPI inserts a next-access delay according to the SPND setting.

When using the SPI in slave mode, set the SPNDEN bit to 0.

SLNDEN bit (SSL Negation Delay Setting Enable)

The SLNDEN bit specifies the SSL negation delay, the period from the time the SPI in master mode stops RSPCK oscillation until the SPI sets the SSLni signal to inactive. If the SLNDEN bit is 0, the SPI sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the SPI negates the SSLni 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 specifies the SPI clock delay, the period from the point when the SPI in master mode asserts the SSLni signal until the RSPCK starts oscillation (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.

30.3 Operation

In this section, the *serial transfer period* refers to the period from the beginning of driving valid data to the fetching of the final valid data.

30.3.1 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 30.4 lists the relationship between the SPI modes and SPCR settings, and a description of each mode.

Table 30.4 Relationship between SPCR settings and SPI modes (1 of 2)

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
RSPCKn 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

Table 30.4 Relationship between SPCR settings and SPI modes (2 of 2)

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
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 PCLKB/6	Up to PCLKB/2	Up to PCLKB/2	Up to PCLKB/6	Up to PCLKB/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				
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 transmit buffer empty interrupt request (SPTEF is 1)	Transmit buffer is written to on generation of 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)
Transmit buffer empty detection	Supported				
Receive buffer full detection	Supported*2				
Overrun error detection	Supported*2	Supported*2, *4	Supported*2, *4	Supported*2	Supported*2
Parity error detection	Supported*2,*3				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported
Underrun error detection	Supported	Not supported	Not supported	Supported	Not supported

Note 1. This function is not supported in this mode.
Note 2. When the SPCR.TXMD bit is 1, detection of receiver buffer full, overrun error, and parity error is 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.

30.3.2 Controlling the SPI Pins

The SPI can switch pin states based on the MSTR, MODFEN, and SPMS bit settings in SPCR. Table 30.5 lists the relationship between the pin states and bit settings. The I/O port settings must follow this relationship.

Table 30.5 Relationship between pin states and bit settings (1 of 2)

Mode	Pin	Pin state*2
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCKn	CMOS output
	SSLn0 to SSLn3	CMOS output
	MOSIn	CMOS output
	MISOn	Input

Table 30.5 Relationship between pin states and bit settings (2 of 2)

Mode	Pin	Pin state*2
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCKn*3	CMOS output/Hi-Z
	SSLn0	Input
	SSLn1 to SSLn3*3	CMOS output/Hi-Z
	MOSIn*3	CMOS output/Hi-Z
	MISOn	Input
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCKn	Input
	SSLn0	Input
	SSLn1 to SSLn3*5	Hi-Z*1
	MOSIn	Input
	MISOn*4	CMOS output/Hi-Z
Master mode (clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCKn	CMOS output
	SSLn0 to SSLn3*5	Hi-Z*1
	MOSIn	CMOS output
	MISOn	Input
Slave mode (clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCKn	Input
	SSLn0 to SSLn3*5	Hi-Z*1
	MOSIn	Input
	MISOn	CMOS output

Note 1. This function is not supported in this mode.
Note 2. SPI settings are not reflected in multiplexed pins for which the SPI function is not selected.
Note 3. When SSLn0 is at the active level, the pin state is Hi-Z.
Note 4. When SSLn0 is at the non-active level or the SPCR.SPE bit is 0, the pin state is Hi-Z.
Note 5. These pins are available for use as I/O port pins.

The SPI in single-master mode (SPI operation) or multi-master mode (SPI operation) determines the MOSI signal values during the SSL negation period based on the MOIFE and MOIFV bit settings in SPPCR, as listed in Table 30.6.

Table 30.6 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

30.3.3 SPI System Configuration Examples

30.3.3.1 Single master and single slave with the MCU as a master

Figure 30.5 shows a single-master and single-slave SPI system configuration example where the MCU is the master. In the single-master and single-slave configuration, the SSLn0 to SSLn3 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 signal.

Note 1. In the transfer format used when SPCMD0.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.

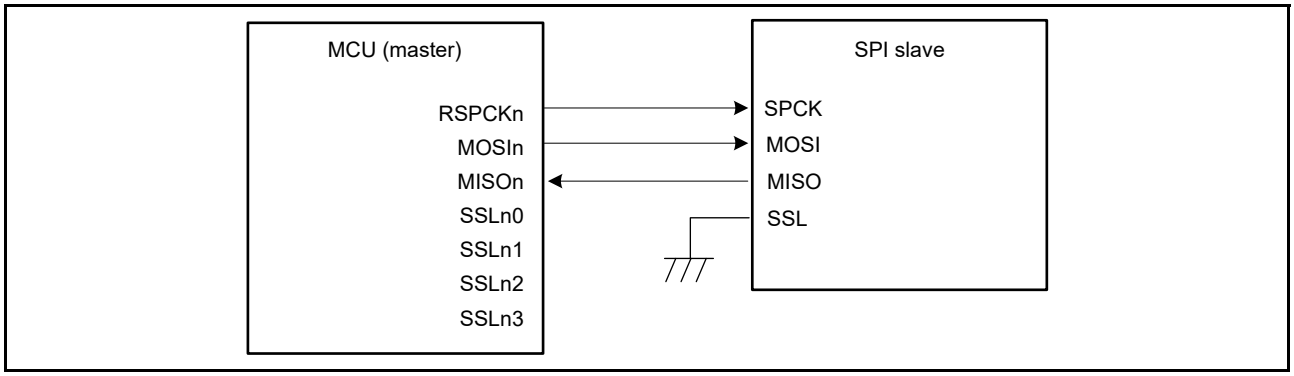


Figure 30.5 Single-master/single-slave configuration example with the MCU as the master

30.3.3.2 Single master and single slave with the MCU as a slave

Figure 30.6 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 signals.*1

In the single-slave configuration in which the SPCMD0.CPHA bit is set to 1, the SSLn0 input of the MCU (slave) is fixed to the low level, and the MCU (slave) stays selected. This enables serial transfer (Figure 30.7).

Note 1. When SSLn0 is at a non-active level, the pin state is Hi-Z.

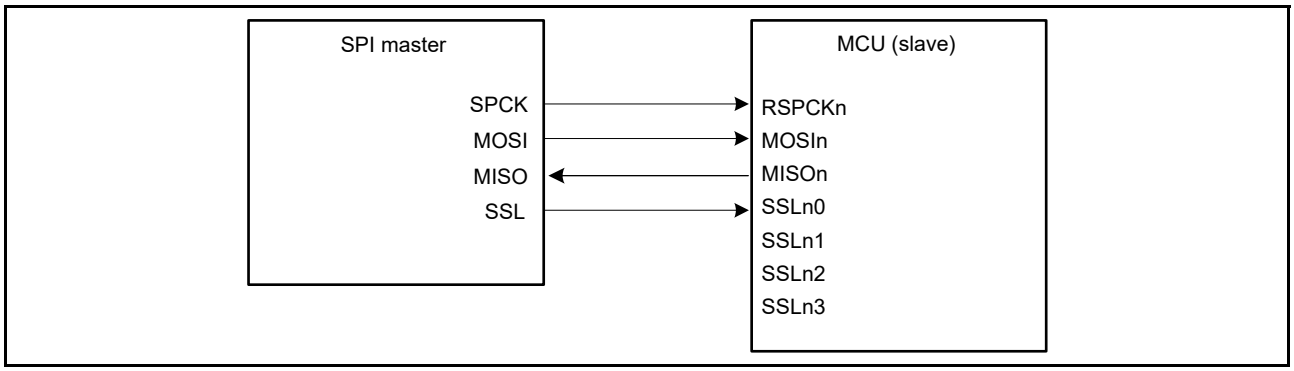


Figure 30.6 Single-master/single-slave configuration example with the MCU as a slave and CPHA = 0

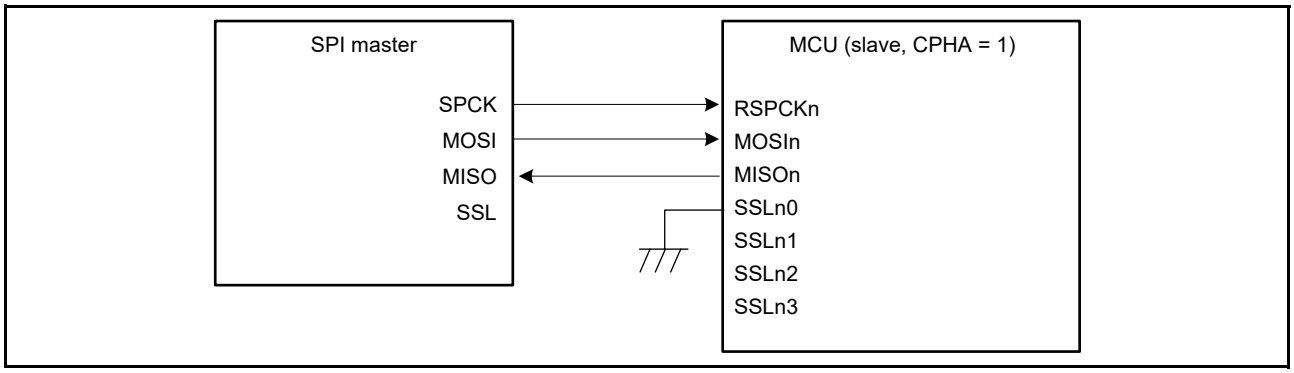


Figure 30.7 Single-master/single-slave configuration example with the MCU as a slave and CPHA = 1

30.3.3.3 Single master and multi slave with the MCU as a master

Figure 30.8 shows a single-master/multi-slave SPI system configuration example where the MCU is a master. In the example, 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 RSPCK 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 RSPCKn, MOSIn, 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.

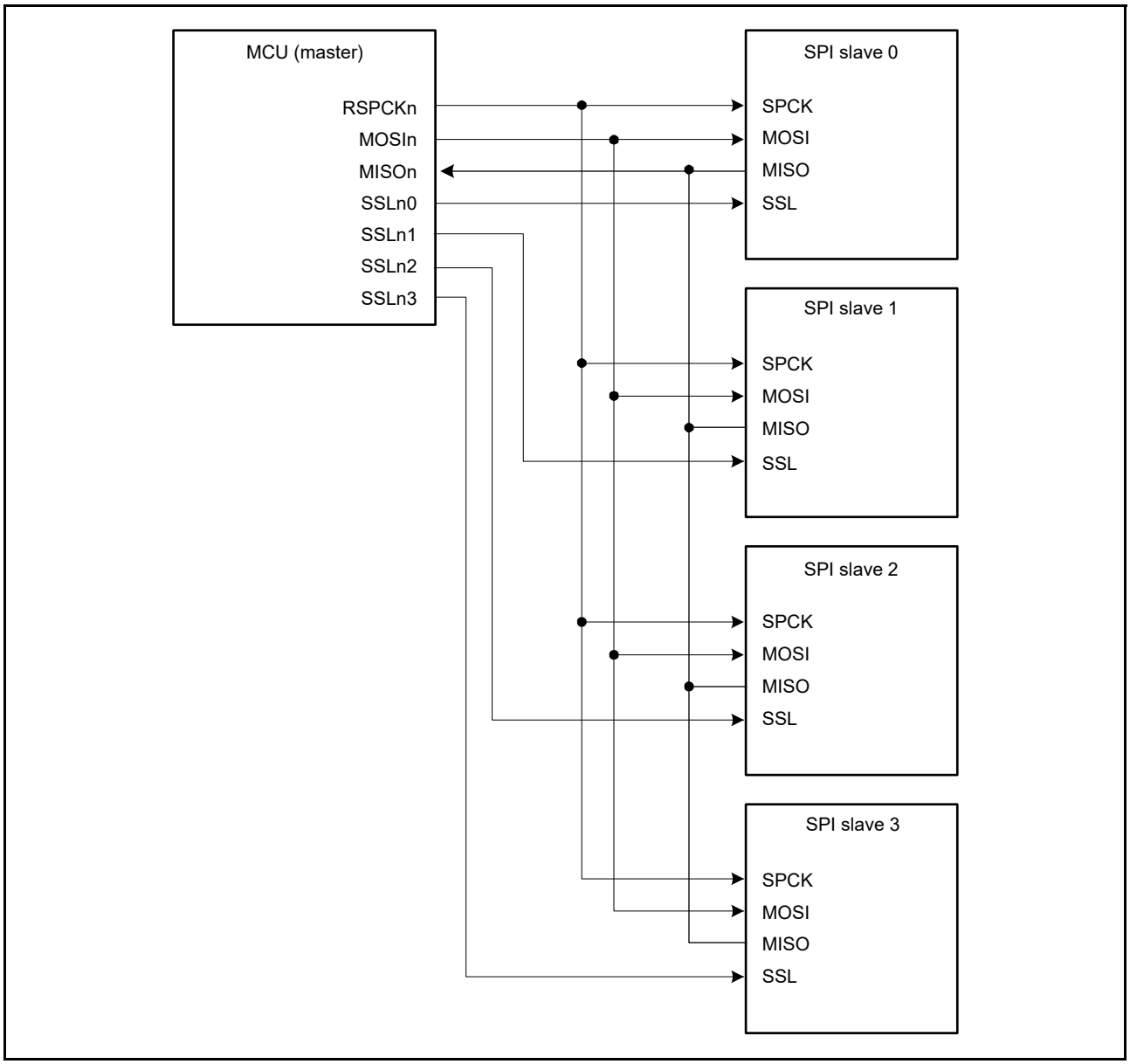


Figure 30.8 Single-master/multi-slave configuration example with the MCU as a master

30.3.3.4 Single master and multi-slave with the MCU as a slave

Figure 30.9 shows a single-master/multi-slave SPI system configuration example where the MCU is a slave. In this example, 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 the SPCK, MOSI, SSLX, and SSLY signals. Of the MCUs (slave X or slave Y), the slave that receives low-level input into the SSLn0 input drives the MISOn signal.

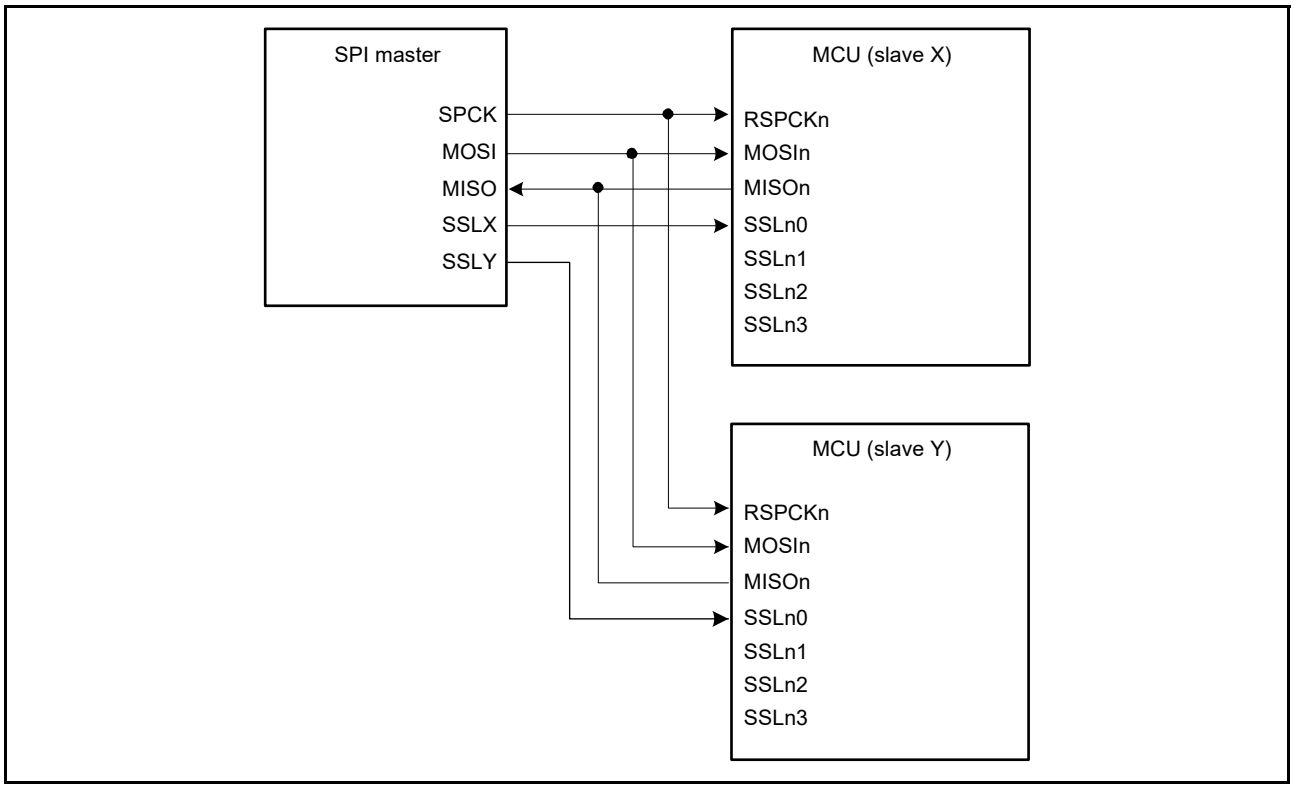


Figure 30.9 Single-master/multi-slave configuration example with the MCU as a slave

30.3.3.5 Multi-master and multi-slave with the MCU as a master

Figure 30.10 shows a multi-master/multi-slave SPI system configuration example where the MCU is a master. In this example, 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 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.

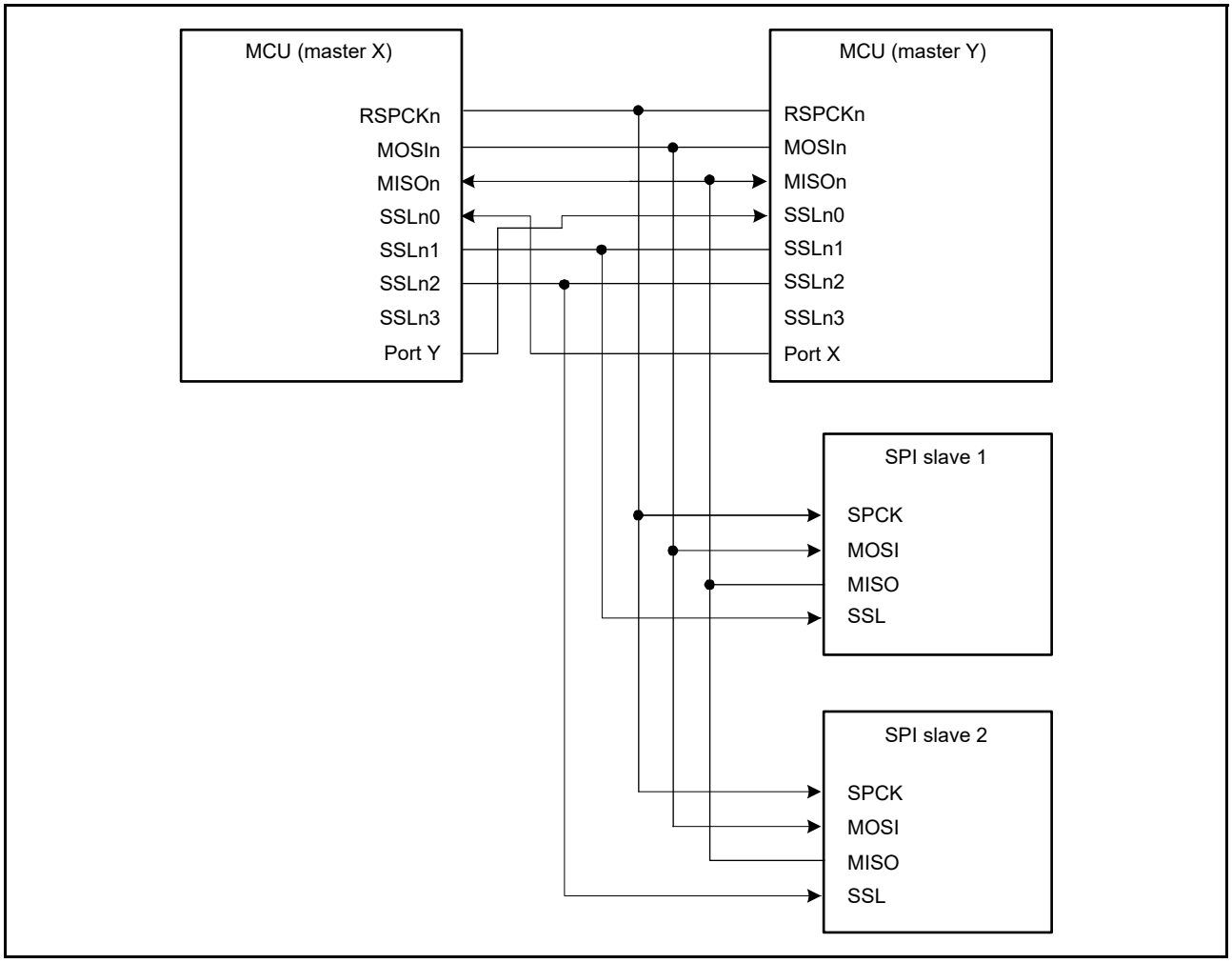


Figure 30.10 Multi-master/multi-slave configuration example with the MCU as a master

30.3.3.6 Master and slave in clock synchronous mode with the MCU as a master

Figure 30.11 shows a master and slave in clock synchronous mode where the MCU is a master. In this 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.

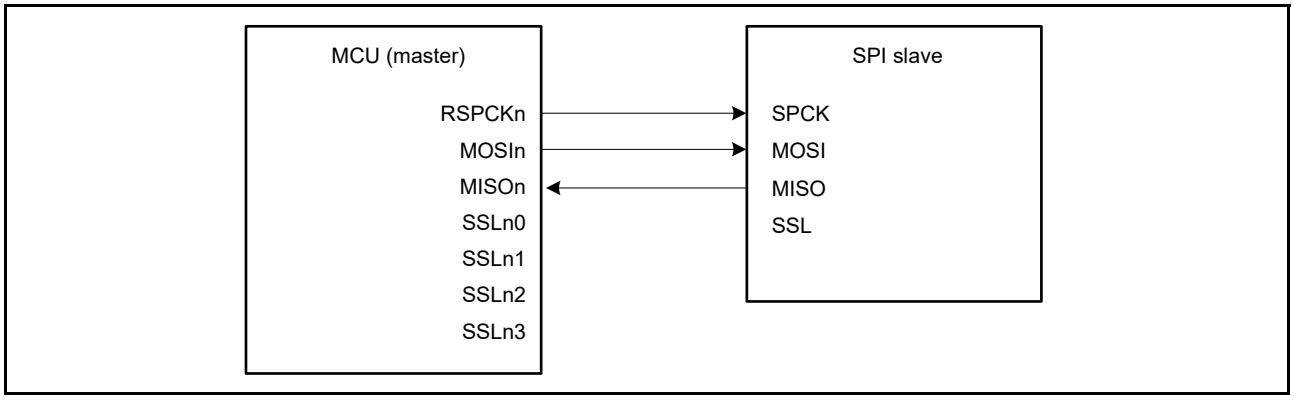


Figure 30.11 Configuration example of master/slave in clock synchronous mode with the MCU as a master

30.3.3.7 Master and slave in clock synchronous mode with the MCU as a slave

Figure 30.12 shows a master and slave in clock synchronous mode configuration where the MCU is a slave. When the MCU operates as a slave in clock synchronous mode, the MCU (slave) drives the MISOn signal and the SPI master drives the SPCK and MOSI signals. SSLn0 to SSLn3 of the MCU (slave) are not used.

The MCU (slave) can only execute serial transfer in the single-slave configuration when SPCMD0.CPHA is set to 1.

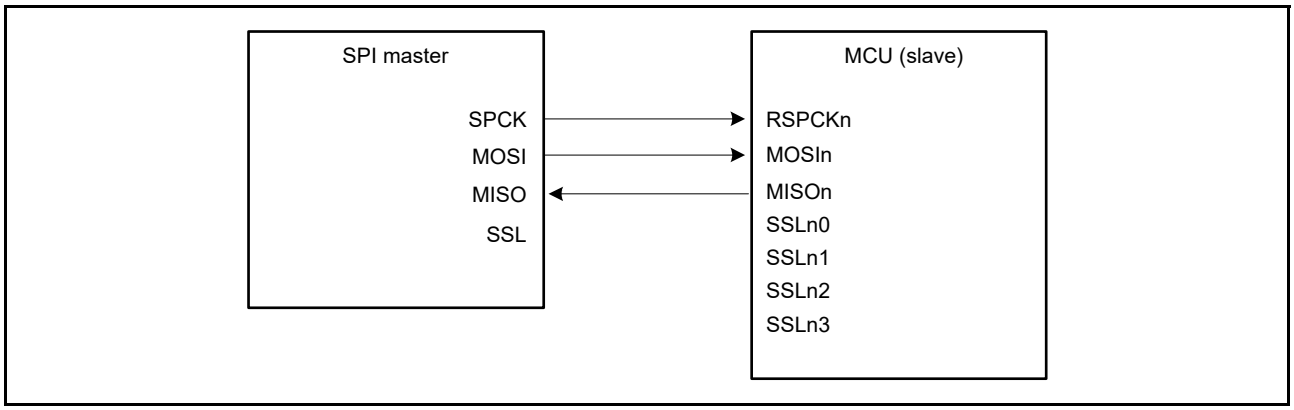


Figure 30.12 Configuration example of master and slave in clock synchronous mode with the MCU as a slave and CPHA = 1

30.3.4 Data Format

The data format of the SPI depends on the settings in the SPI Command Register 0 (SPCMD0) 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 0 (SPCMD0.SPB[3:0]).

(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 the SPI Command Register 0 (SPCMD0.SPB[3:0]). In this case, however, the last bit is a parity bit.

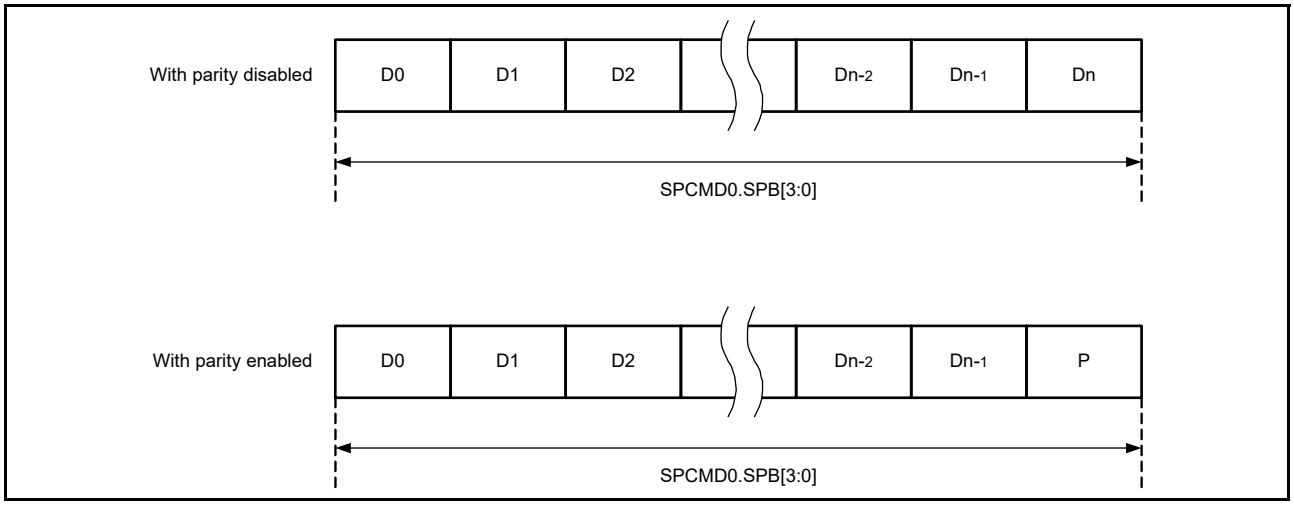


Figure 30.13 Data format with parity disabled and enabled

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

When parity is disabled, data for transmission is copied to the shift register with no pre-processing. This section describes the connection between the SPI Data Register (SPDR/SPDR_HA) and the shift register in terms of the combination of MSB- or LSB-first order and data length.

(1) MSB-first transfer with 32-bit data

Figure 30.14 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 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 bits R31 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.

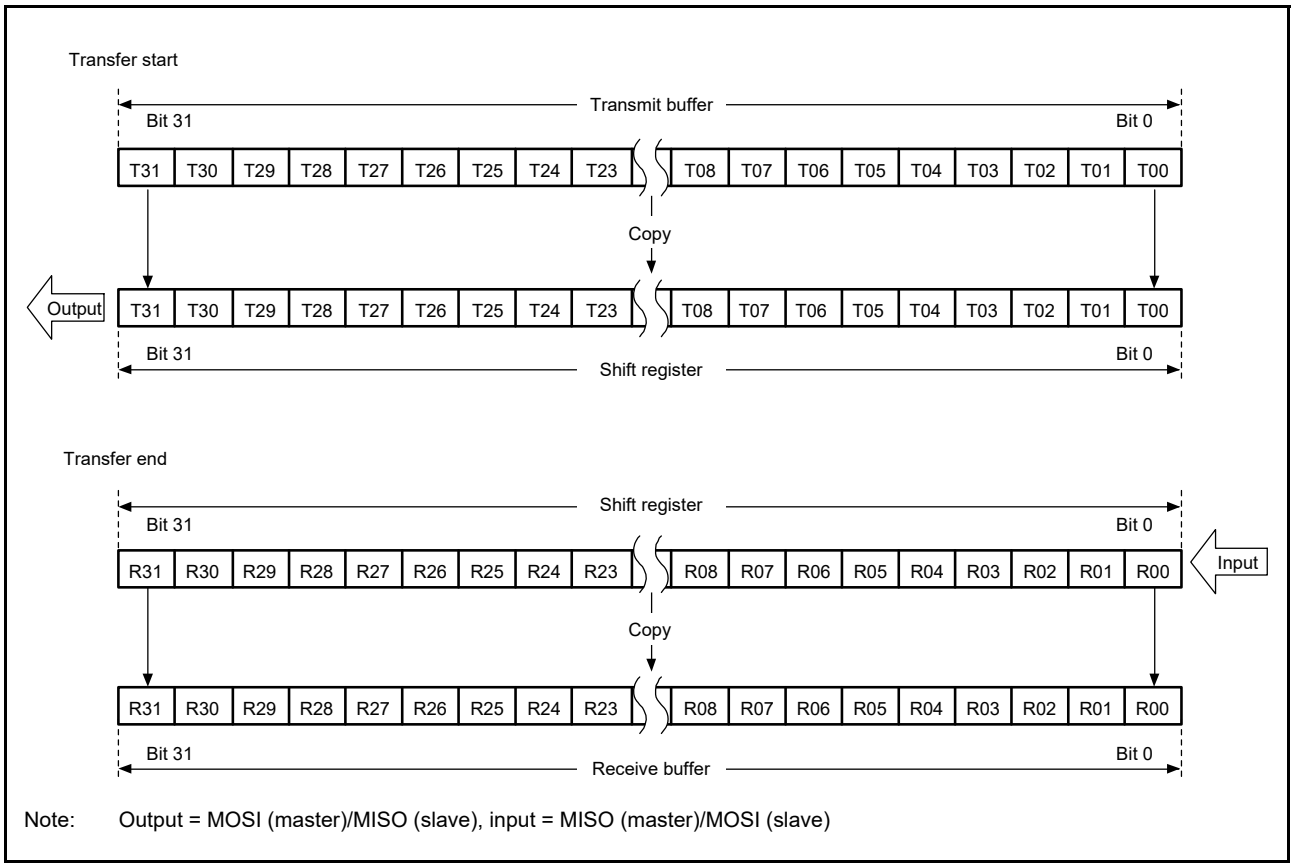


Figure 30.14 MSB-first transfer with 32-bit data and parity disabled

(2) MSB-first transfer with 24-bit data

Figure 30.15 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 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 during transmission results in 0 being inserted in the upper 8 bits of the receive buffer.

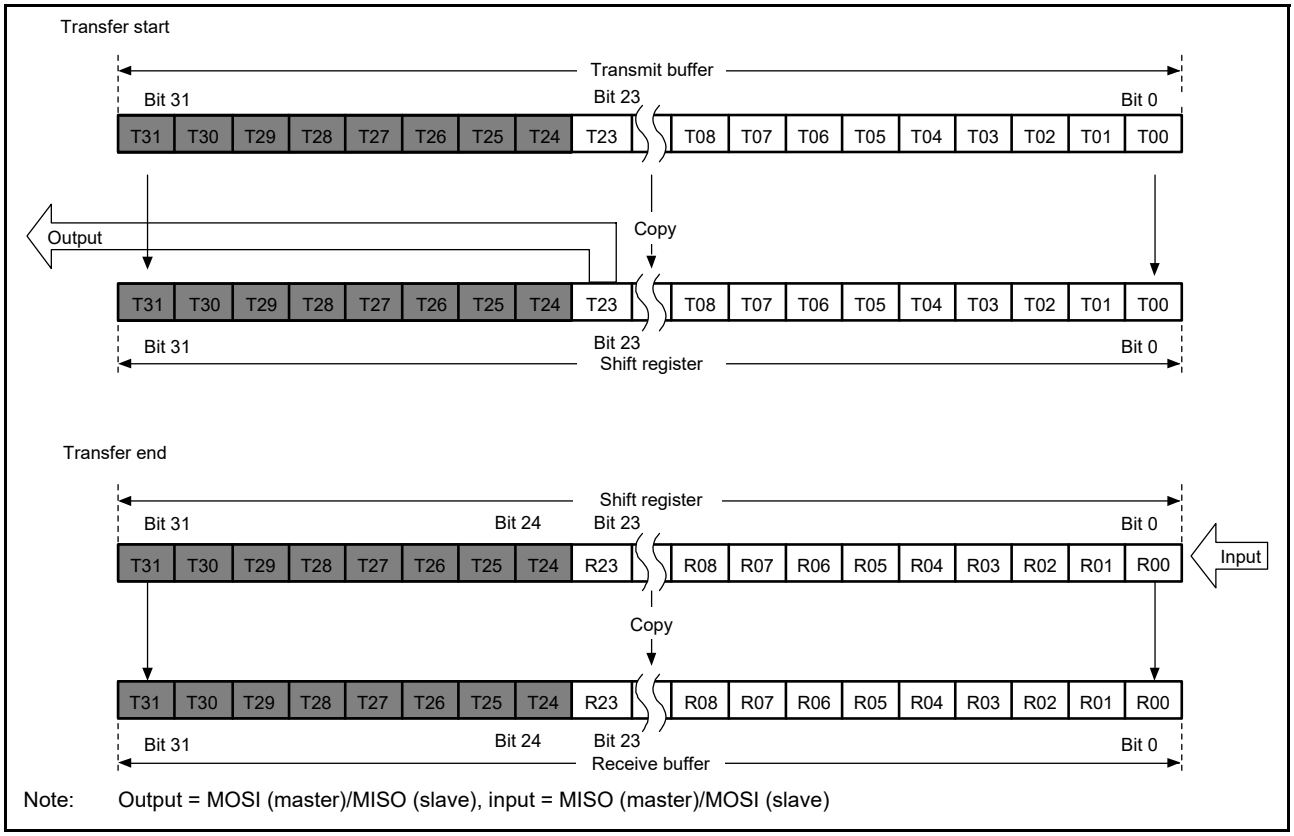


Figure 30.15 MSB-first transfer with 24-bit data and parity disabled

(3) LSB-first transfer with 32-bit data

Figure 30.16 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.

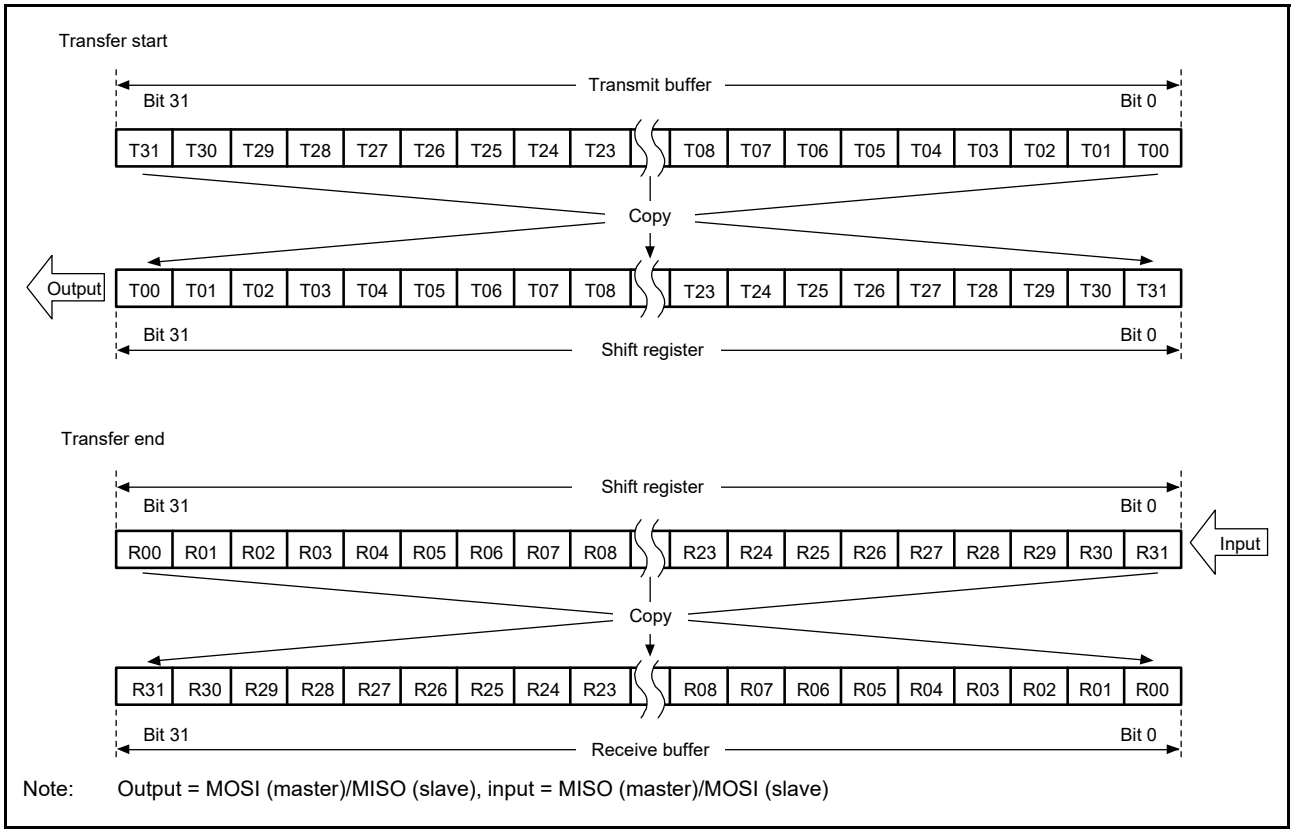


Figure 30.16 LSB-first transfer with 32-bit data and parity disabled

(4) LSB-first transfer with 24-bit data

Figure 30.17 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 the R00 to R23 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.

The upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 during transmission results in 0 being inserted in the upper 8 bits of the receive buffer.

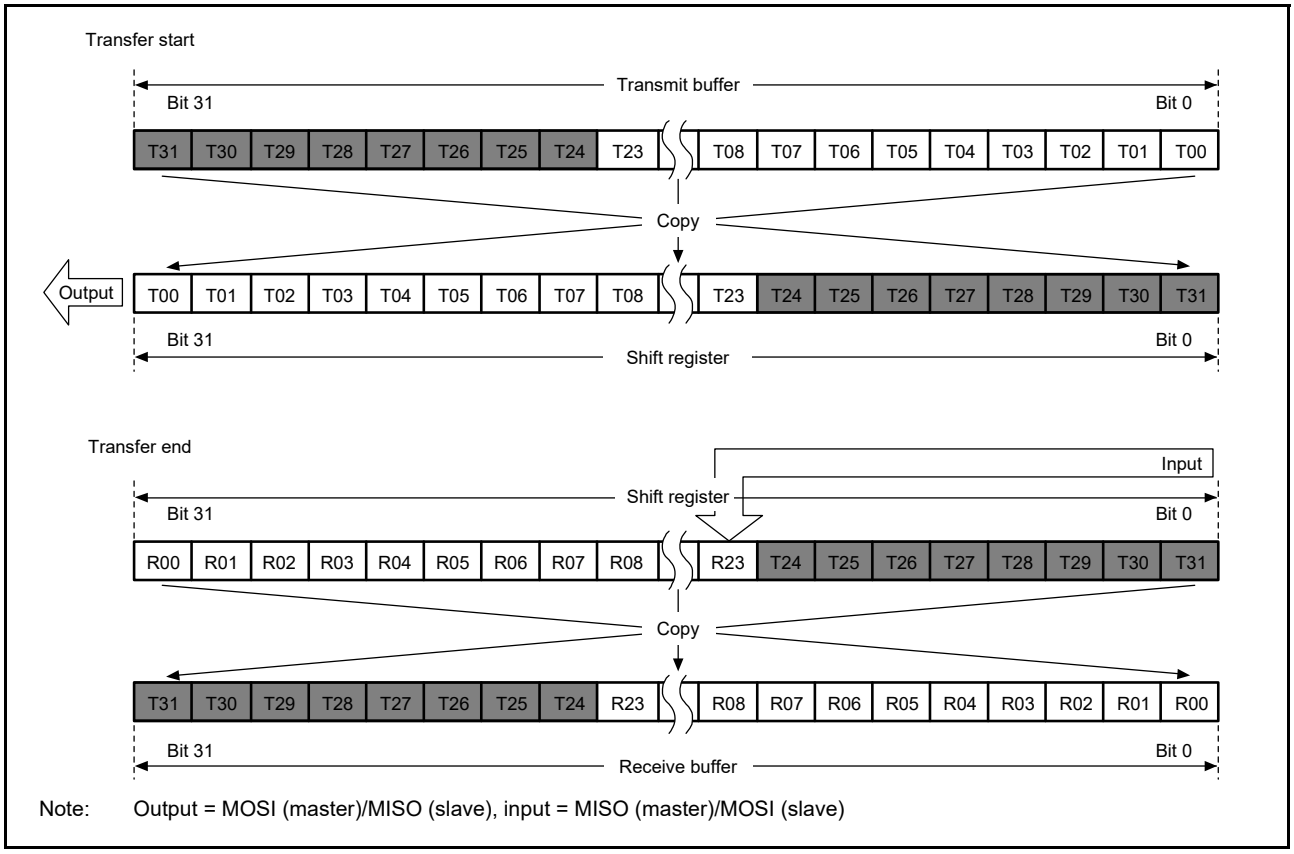


Figure 30.17 LSB-first transfer with 24-bit data and parity disabled

30.3.4.2 Operation when parity is enabled (SPCR2.SPPE = 1)

When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

(1) MSB-first transfer with 32-bit data

Figure 30.18 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 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. On copying of data to the shift register, the data from R31 to P is checked for parity.

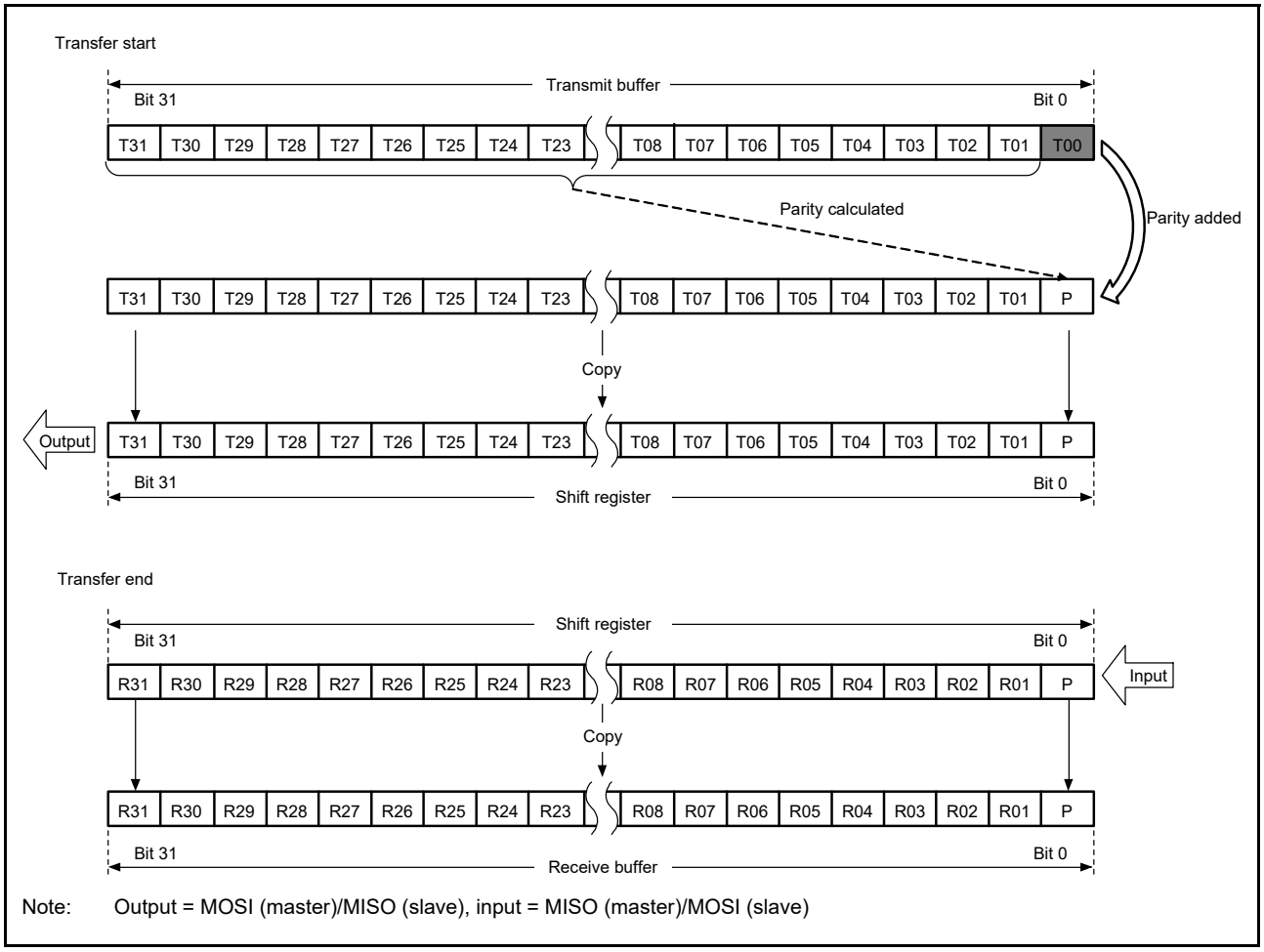


Figure 30.18 MSB-first transfer with 32-bit data and parity enabled

(2) MSB-first transfer with 24-bit data

Figure 30.19 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an SPI data length of 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 is copied to the shift register. Data is transmitted in the order T23, T22, …, T01, and P.

In reception, received data is shifted in bit-by-bit through bit [0] of the shift register. When bits R23 to P are collected after input of the required number of 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.

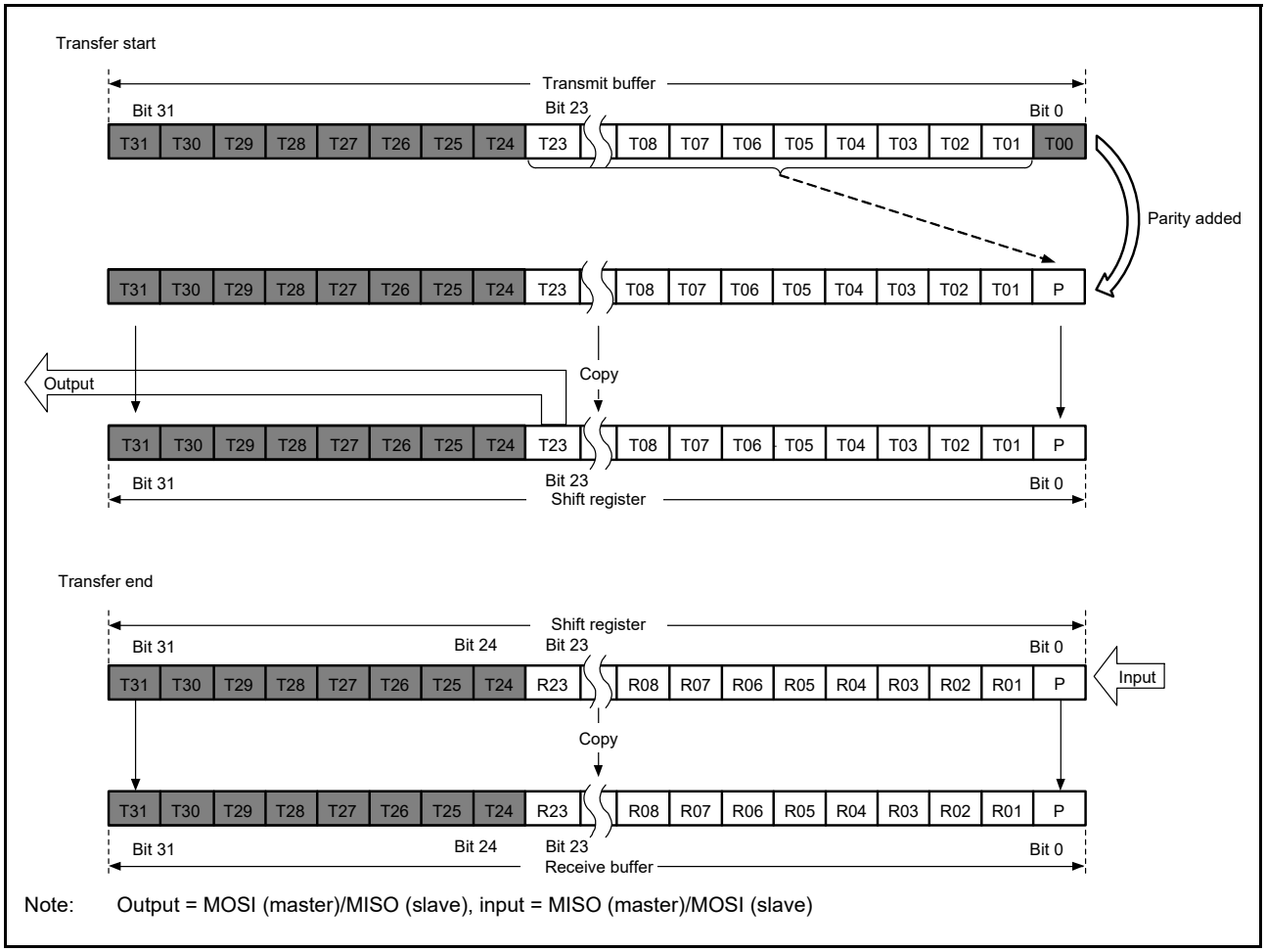


Figure 30.19 MSB-first transfer with 24-bit data and parity enabled

(3) LSB-first transfer with 32-bit data

Figure 30.20 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an SPI data length of 32 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole value is copied to the shift register. Data is transmitted in the order T00, T01, …, T30, and P.

In reception, received data is shifted in bit by bit through bit [0] of the shift register. When 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.

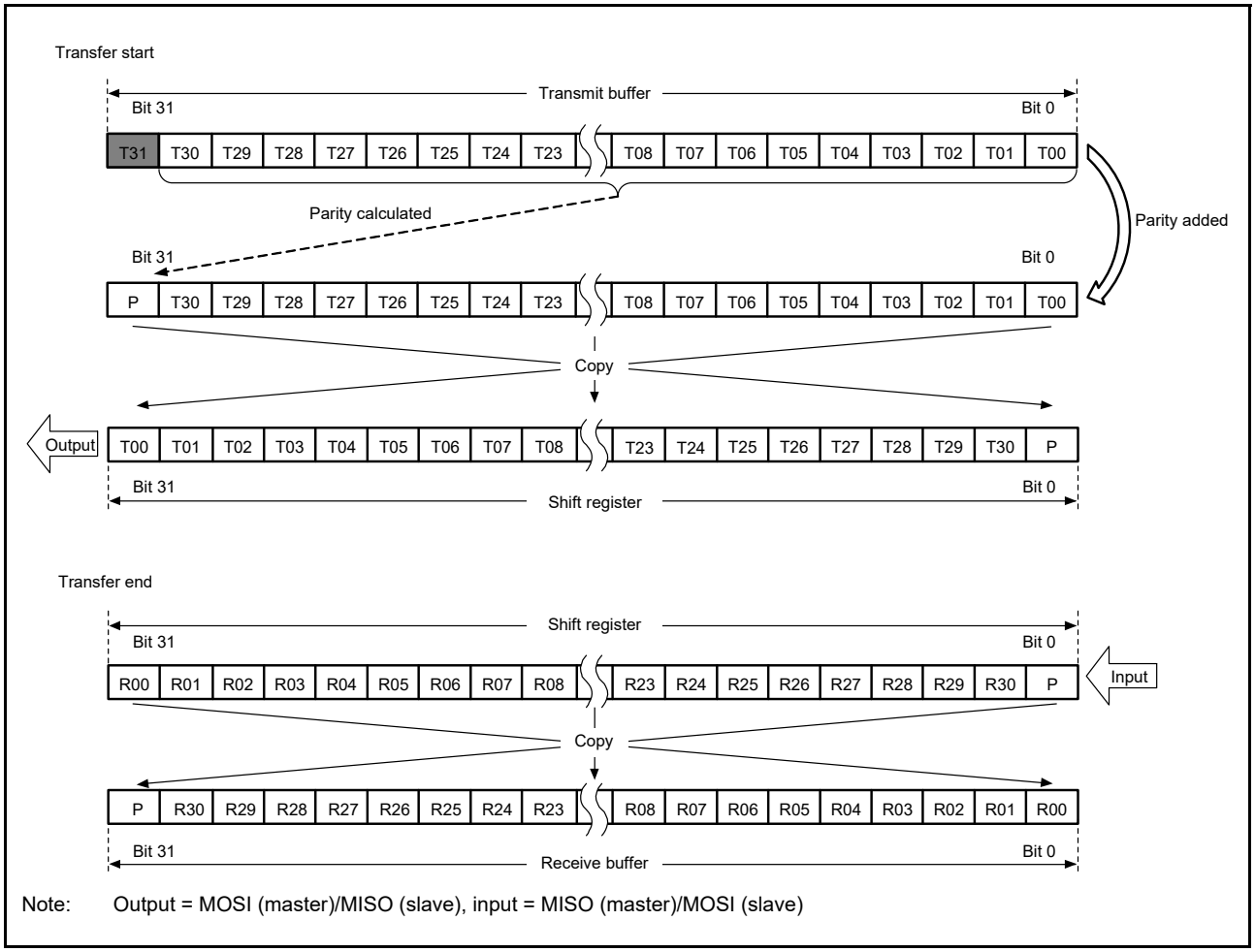


Figure 30.20 LSB-first transfer with 32-bit data and parity enabled

(4) LSB-first transfer with 24-bit data

Figure 30.21 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an 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. On copying of data to the shift register, the data from R00 to P is checked for parity. The upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 during transmission results in 0 being inserted in the upper 8 bits of the receive buffer.

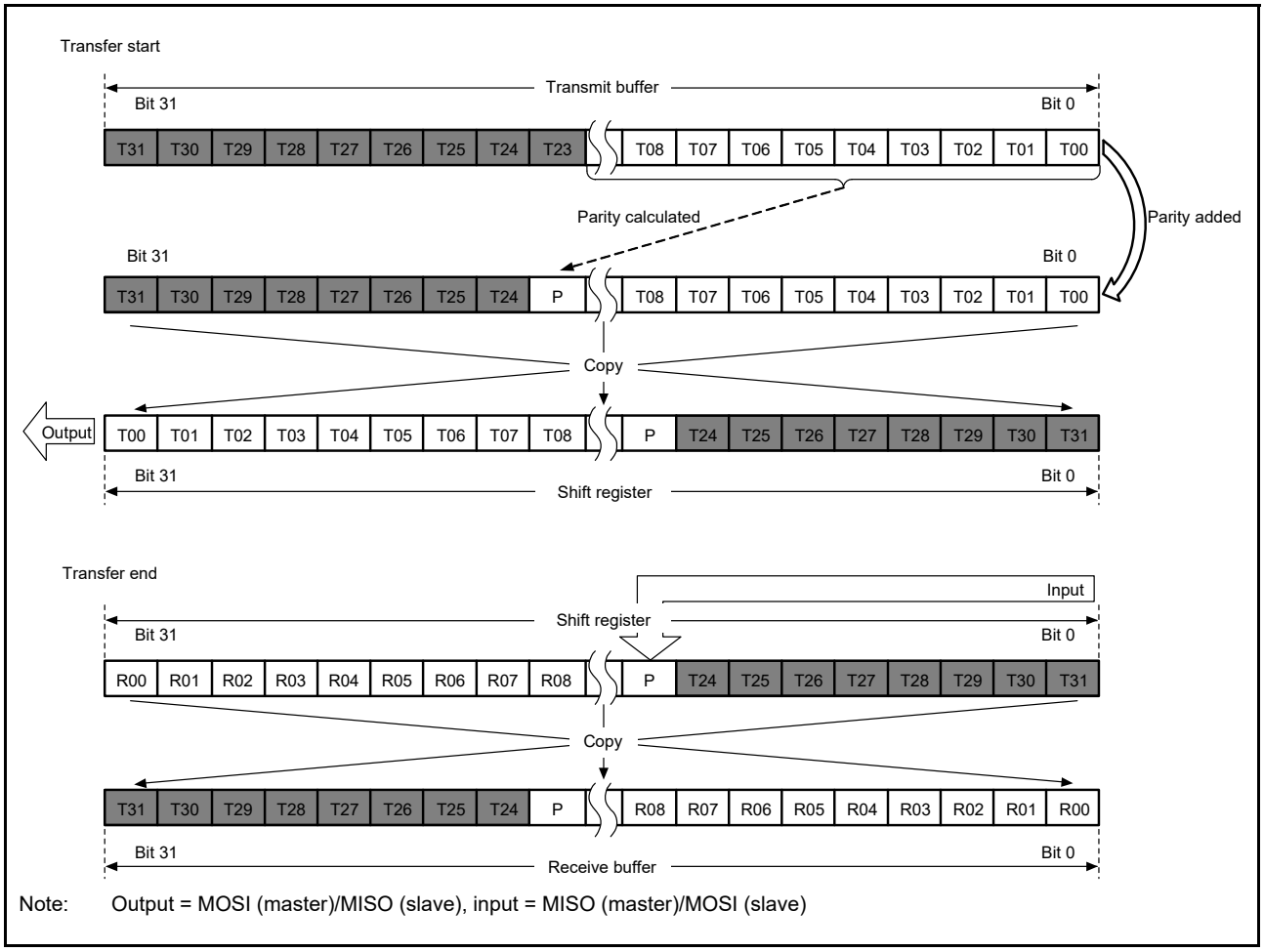


Figure 30.21 LSB-first transfer with 24-bit data and parity enabled

30.3.5 Transfer Format

30.3.5.1 Transfer format when CPHA = 0

Figure 30.22 shows an example transfer format for the serial transfer of 8-bit data when the SPCMD0.CPHA bit is 0. Do not perform clock synchronous operation (SPCR.SPMS bit is 1) when the SPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 30.22, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMD0.CPOL bit is 0 and RSPCKn (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI settings. For details, see section 30.3.2, Controlling the SPI Pins.

When the SPCMD0.CPHA bit is 0, the driving of valid data to the MOSIn and MISOn signals begins at an SSLni signal assertion. The first RSPCKn signal change that occurs after the SSLni signal assertion becomes the first transfer data fetch. After this, data is sampled on every 1 RSPCK cycle. The change timing for MOSIn and MISOn signals is 1/2 RSPCK cycle after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCKn 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 30.3.10.1, Master mode operation.

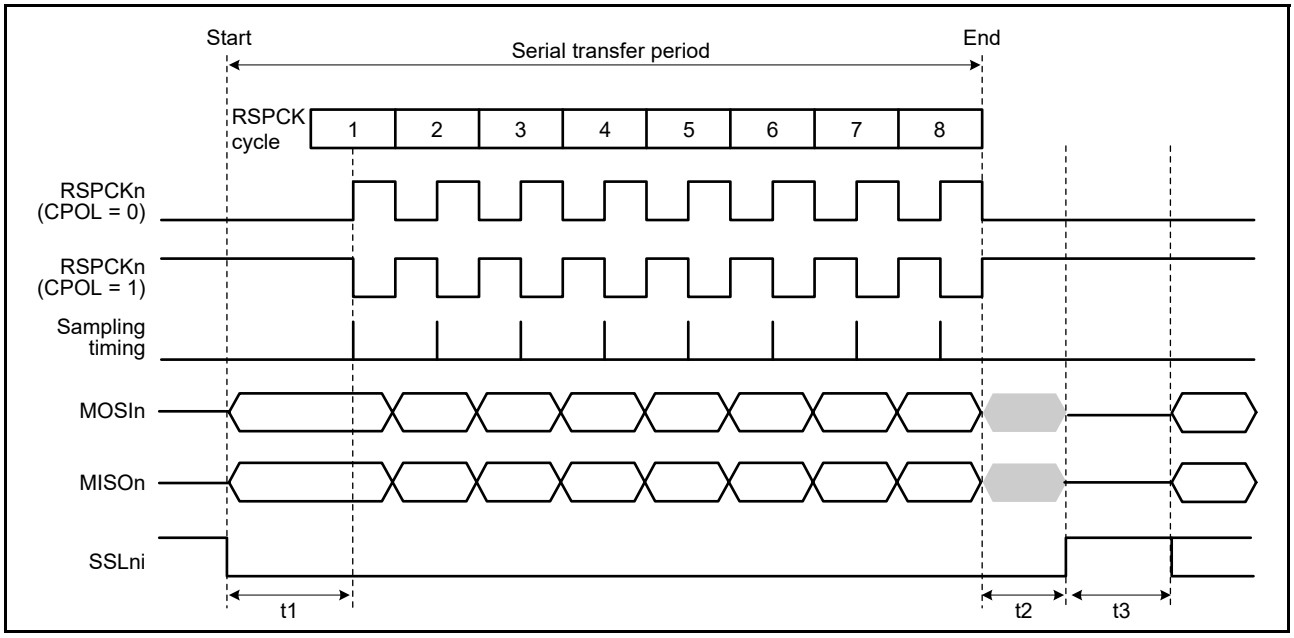


Figure 30.22 SPI transfer format when CPHA = 0

30.3.5.2 When CPHA = 1

Figure 30.23 shows an example transfer format for the serial transfer of 8-bit data when the SPCMD0.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLni signals are not used, and only the three signals RSPCKn, MOSIn, and MISOn handle communications. In Figure 30.23, RSPCK (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMD0.CPOL bit is 0 and RSPCK (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI mode (master or slave). For details, see section 30.3.2, Controlling the SPI Pins.

When the SPCMD0.CPHA bit is 1, the driving of invalid data to the MISOn signal begins at an SSLni signal assertion. The output of valid data to the MOSIn and MISOn signals begins at the first RSPCKn signal change that occurs after the SSLni signal assertion. After this, data is updated every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycle after the data update timing. The SPCMD0.CPOL bit setting does not affect the RSPCKn signal operation timing. It only affects the signal polarity.

t1, t2, and t3 are the same as those when CPHA = 0. For a description of t1, t2, and t3 when the SPI of the MCU is in master mode, see section 30.3.10.1, Master mode operation.

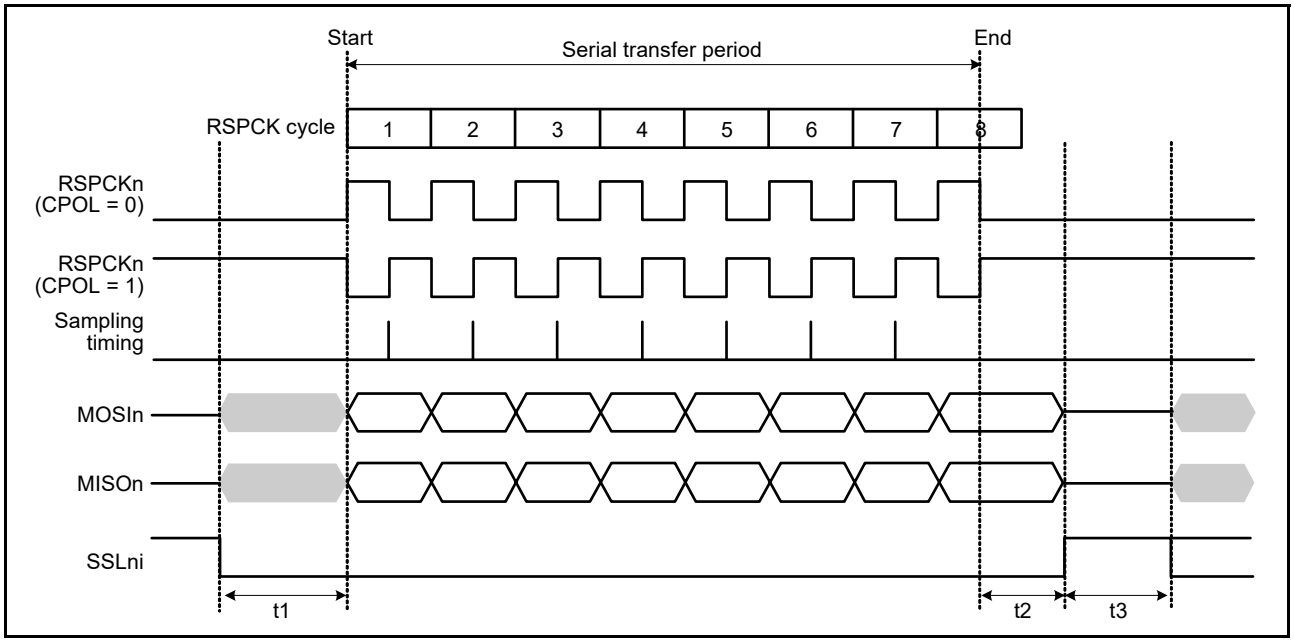


Figure 30.23 SPI transfer format when CPHA = 1

30.3.6 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 30.24 and Figure 30.25 indicate the condition of access to the SPDR/SPDR_HA register, where W denotes a write cycle.

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

Figure 30.24 shows an example 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 SPCMD0.CPHA bit is 1 and the SPCMD0.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

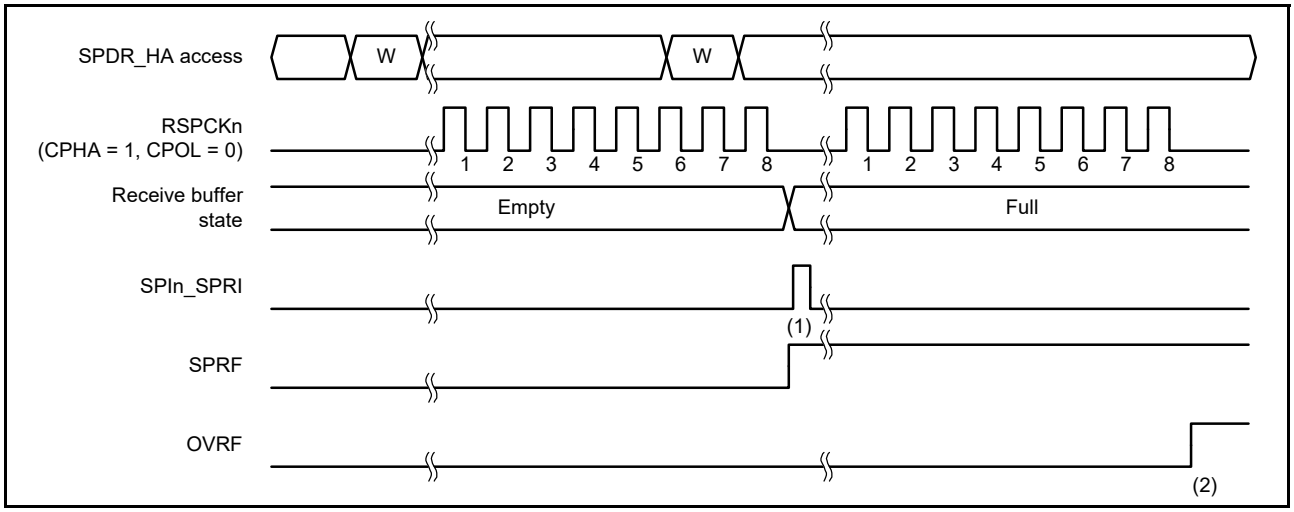


Figure 30.24 Operation example when SPCR.TXMD = 0

The operation of the flags at timings (1) and (2) in Figure 30.24 is as follows:

1. When a serial transfer ends with the SPDR/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/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.

30.3.6.2 Transmit-only operations (SPCR.TXMD = 1)

Figure 30.25 shows an operation example where the Communications Operating Mode Select bit (SPCR.TXMD) is set to 1. In this example, the SPI performs an 8-bit serial transfer in which the SPCMD0.CPHA bit is 1 and the SPCMD0.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

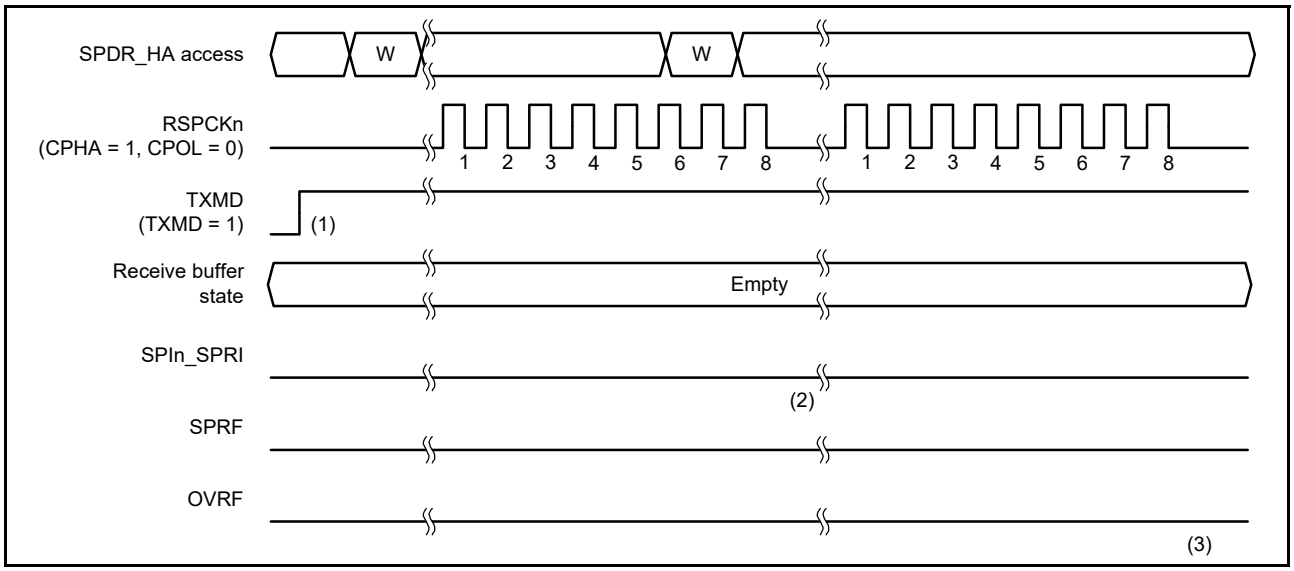


Figure 30.25 Operation example when SPCR.TXMD = 1

The operation of the flags at timings (1) to (3) in Figure 30.25 is as follows:

(1) Make sure there is no data left in the receive buffer (SPSR.SPRF flag is 0) and the SPSR.OVRF flag is 0 before entering the transmit-only mode (SPCR.TXMD = 1).

(2) When a serial transfer ends with the SPDR/SPDR_HA receive buffer 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 SPDR/SPDR_HA receive buffer 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).

30.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts

Figure 30.26 and Figure 30.27 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/SPDR_HA register, where W denotes a write cycle and R a read cycle. In Figure 30.26, the SPI performs an 8-bit serial transfer when the SPCR.TXMD bit is 0, the SPCMD0.CPHA bit is 0, and the SPCMD0.CPOL bit is 0.

In Figure 30.27, the SPI performs an 8-bit serial transfer in which the SPCR.TXMD bit is 0, the SPCMD0.CPHA bit is 1, and the SPCMD0.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

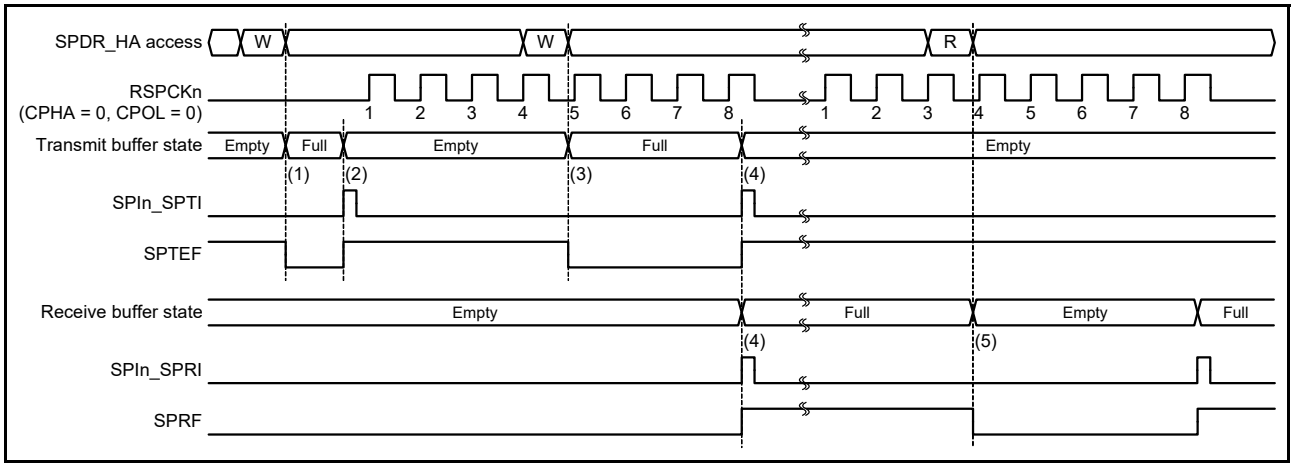


Figure 30.26 Operation example of SPIn_SPTI and SPIn_SPRI interrupts when CPHA = 0 and CPOL = 0

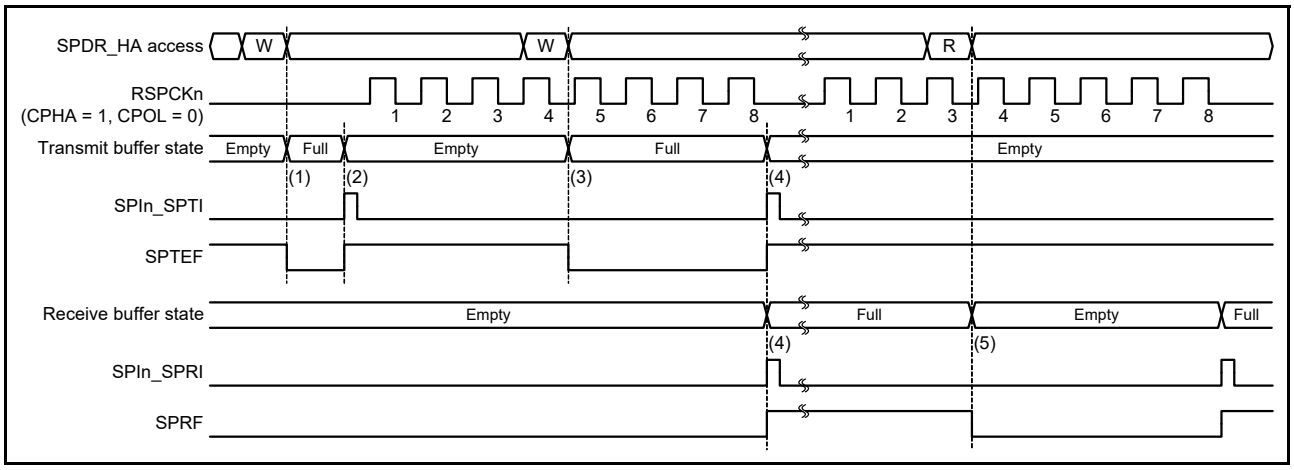


Figure 30.27 Operation example of SPIn_SPTI and SPIn_SPRI interrupts when CPHA = 1 and CPOL = 0

The operation of the SPI at timings (1) to (5) in Figure 30.27 is as follows:

(1) When transmit data is written to SPDR/SPDR_HA with the transmit buffer of SPDR/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 SPI mode. For details, see section 30.3.10, SPI Operation, and section 30.3.11, Clock Synchronous Operation.

(3) When transmit data is written to SPDR/SPDR_HA either by the transmit buffer empty interrupt routine, or by the processing of the transmit buffer empty using the SPTEF flag, the SPI writes data to the transmit buffer and clears the SPTEF flag to 0. Because the 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 SPDR/SPDR_HA receive buffer empty, the SPI copies the receive data in the shift register to the receive buffer, 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, when the transmit buffer was full before the serial transfer ended, the SPI sets the SPTEF flag to 1 and copies data in the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, on completion of the serial transfer, the SPI determines that the shift register is empty, so data transfer from the transmit buffer to the shift register is enabled.

(5) When SPDR/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/SPDR_HA is written to when the transmit buffer holds untransmitted data (SPTEF flag is 0), the SPI does not update data in the transmit buffer. When writing to SPDR/SPDR_HA, make sure to use either 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 bit is 0), set the SPTIE bit to 0.

When serial transfer ends with the receive buffer full (SPRF flag is 1), the SPI does not copy data from the shift register to the receive buffer, and detects an overrun error (see section 30.3.8, 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 IELSRm.IR flags in the ICU, where m is the interrupt vector number, can be used to confirm the states of the transmit and receive buffers. Similarly, the SPTEF and SPRF flags can be used to confirm the states of the transmit and receive buffers. See section 13, Interrupt Controller Unit (ICU) for the interrupt vector numbers.

30.3.8 Error Detection

In normal SPI serial transfer, data written to the SPDR/SPDR_HA transmit buffer 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 30.7 lists the relationship between non-normal transfer operations and the SPI error detection function.

Table 30.7 Relationship between non-normal transfer operations and SPI error detection function

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
8	The SSLn0 input signal is negated during serial transfer in slave mode	• Serial transfer is suspended • Transmit or receive data is missing • Driving of the MISOn output signal is stopped • SPI function is disabled.	Mode fault error

In operation 1 described in Table 30.7, the SPI does not detect an error. To prevent data omission during writes to SPDR/SPDR_HA, the writes to SPDR/SPDR_HA must be executed using a transmit buffer empty interrupt request (when 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 the other errors, see the following sections:

- Underrun errors, indicated in operation 3, see section 30.3.8.4, Underrun errors
- Overrun errors, indicated in operation 4, see section 30.3.8.1, Overrun errors
- Parity errors, indicated in operation 5, see section 30.3.8.2, Parity errors
- Mode fault errors indicated in operations 6 to 8, see section 30.3.8.3, Mode fault errors.

For the transmit and receive interrupts, see section 30.3.7, Transmit Buffer Empty and Receive Buffer Full Interrupts.

30.3.8.1 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 30.28 shows an example operation of the OVRF and SPRF flags. The SPSR and SPDR/SPDR_HA accesses shown in Figure 30.28 indicate the condition of accesses to SPSR and SPDR/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 SPCMD0.CPHA bit is 1 and the SPCMD0.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

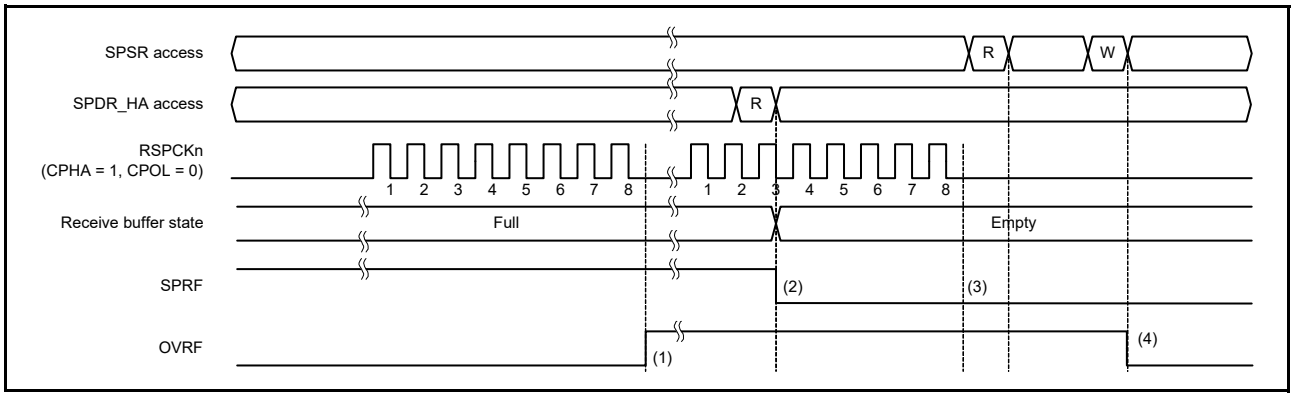


Figure 30.28 Operation example of OVRF and SPRF flags

The operation of the flags at timings (1) to (4) in Figure 30.28 is as follows:

(1) If a serial transfer terminates with the SPRF flag set to 1 (receive buffer full), the SPI detects an overrun error, and sets the OVRF flag to 1. The SPI does not copy data in the shift register to the receive buffer. Even when the SPPE bit is 1, parity errors are not detected.

(2) When SPDR/SPDR_HA is read, the SPI outputs the data in the receive buffer. The SPRF flag is then set to 0. The receive buffer becoming empty does not set the OVRF flag to 0.

(3) If the serial transfer ends with the OVRF flag set to 1 (an overrun error occurs), the SPI does not copy 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 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. If an overrun error occurs and the OVRF flag is set to 1, normal reception cannot be performed until OVRF is set to 0.

When the RSPCK auto-stop function is enabled in master mode, an overrun error does not occur. Figure 30.29 and Figure 30.30 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

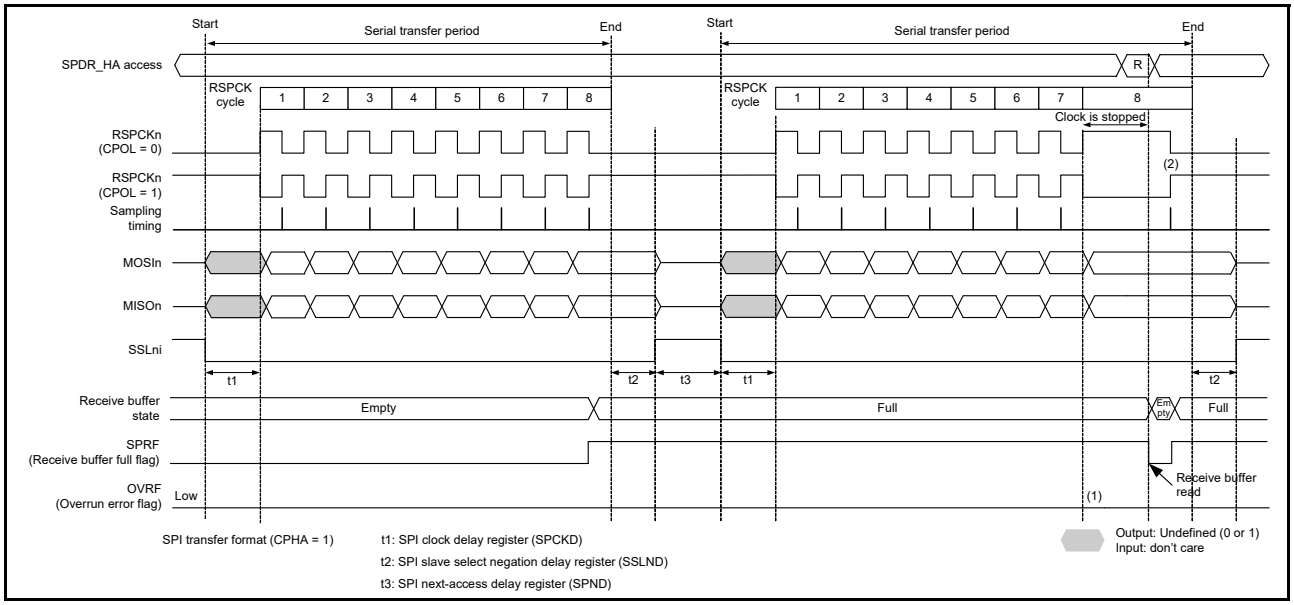


Figure 30.29 Clock stop waveform when serial transfer continues while receive buffer is full in master mode with CPHA = 1

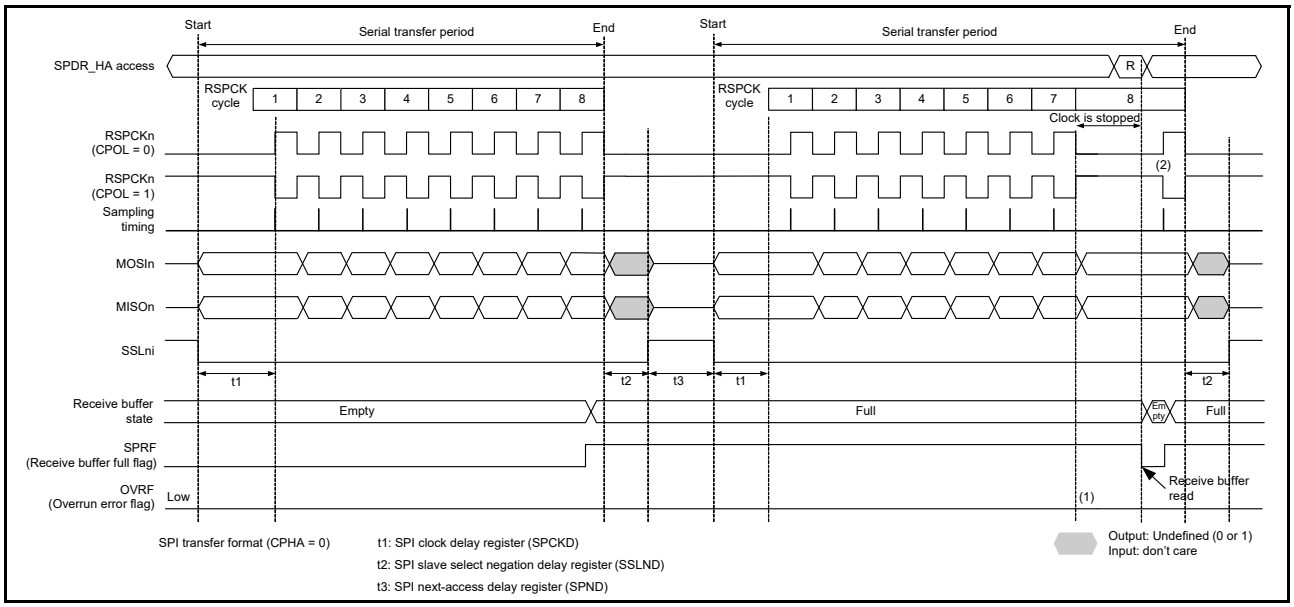


Figure 30.30 Clock stop waveform when serial transfer continues while receive buffer is full in master mode with CPHA = 0

The operation of the flags at timings (1) and (2) in Figure 30.29 and Figure 30.30 is as follows:

(1) When the receive buffer is full, an overrun error does not occur because the RSPCK clock is stopped.

(2) If SPDR/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 is set to 0).

30.3.8.2 Parity errors

When full-duplex synchronous serial communications is performed with the SPCR.TXMD bit set to 0 and the SPCR2.SPPE bit set to 1, 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 30.31 shows an example operation of the OVRF and PERF flags. The SPSR access shown in Figure 30.31 indicates the condition of access to the SPSR register, where W denotes a write cycle, and R a read cycle. In the example, full-duplex synchronous serial communications is performed while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1. The SPI performs an 8-bit serial transfer in which the SPCMD0.CPHA bit is 1 and the SPCMD0.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

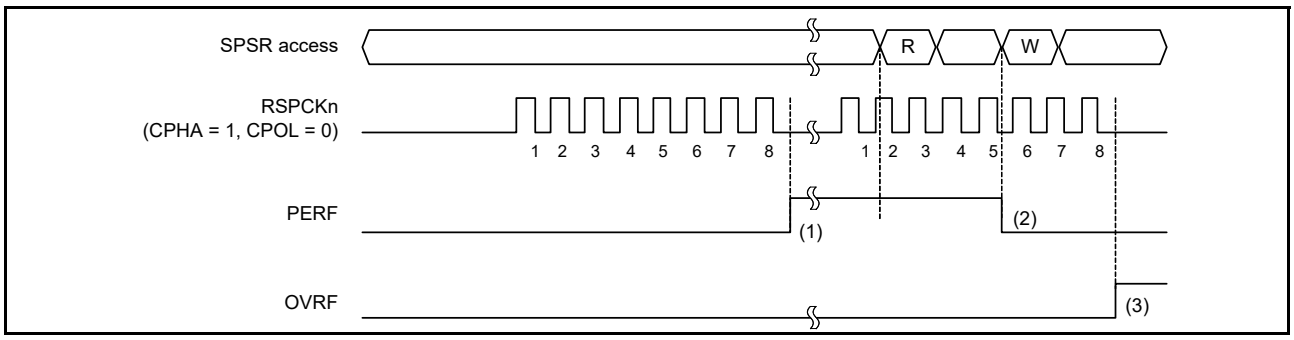


Figure 30.31 Operation example of the OVRF and PERF flags

The operation of the flags at timing (1) to (3) in Figure 30.31 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 time and sets the PERF flag to 1 if a parity error is detected.

(2) If 0 is written to the PERF flag after the SPSR register is read when the PERF flag is 1, the PERF flag is set to 0.

(3) When the SPI detects an overrun error and serial transfer is terminated, 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 by reading SPSR or by using an SPI error interrupt and reading SPSR. When executing a serial transfer, make sure that parity errors are detected early, for instance by reading SPSR.

30.3.8.3 Mode fault errors

The SPI operates in multi-master mode when the SPCR.MSTR bit is 1, the SPCR.SPMS bit is 0, and the SPCR.MODFEN bit is 1. If the active level is input for the SSLn0 input signal of the SPI in multi-master mode, the SPI detects a mode fault error regardless of the status of the serial transfer, and sets the SPSR.MODF flag to 1. The active level of the SSLn0 signal is determined by the SSLP.SSL0P bit.

When the MSTR bit is 0, the SPI operates in slave mode. The SPI detects a mode fault error if the MODFEN bit of the SPI in slave mode is 1, and the SPMS bit is 0, and if the SSLn0 input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

On detecting a mode fault error, the SPI stops driving the output signals and clears the SPCR.SPE bit to 0 (see section 30.3.9, Initializing the SPI). For multi-master configuration, detection of a mode fault error is used to stop the driving of output signals and the SPI function, which allows the master to be released.

The occurrence of a mode fault error can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. Detecting mode-fault errors without using the SPI error interrupt requires polling of SPSR.

When the MODF flag is 1, writing 1 to the SPE bit is ignored by the SPI. To enable the SPI function after the detection of a mode fault error, the MODF flag must be set to 0.

30.3.8.4 Underrun errors

When a serial transfer begins with the SPCR.MSTR bit set to 0 (slave mode), the SPCR.SPE bit is set to 1, and the transmission data not prepared, the SPI detects an underrun error. The SPI then sets SPSR.MODF flag and SPSR.UDRF flag to 1. On detecting an underrun error, the SPI stops driving the output signals and clears the SPCR.SPE bit to 0 (see section 30.3.9, Initializing the SPI).

The occurrence of an underrun error can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. Detecting underrun errors without using the SPI error interrupt requires polling of SPSR.

When the MODF flag is 1, writing 1 to the SPE bit is ignored by the SPI. To enable the SPI function after the detection of an underrun error, the MODF flag must be set to 0.

30.3.9 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 occurs, the SPI initializes all of the module functions. This section describes initialization by clearing of the SPCR.SPE bit, and by a system reset.

30.3.9.1 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. 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.

30.3.9.2 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 30.3.9.1, Initialization by clearing the SPE bit.

30.3.10 SPI Operation

30.3.10.1 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 30.3.8, 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, 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 30.3.5, Transfer Format.

(2) Terminating serial transfer

Regardless of the SPCMD0.CPHA bit setting, the SPI terminates a serial transfer after transmitting an RSPCKn edge associated with the final sampling timing. If free space is available in the receive buffer (SPRX) (SPSR.SPRF flag is 0), on termination of serial transfer, the SPI copies data from the shift register to the receive buffer of the SPDR/SPDR_HA register.

The final sampling timing varies depending on the bit length of the transfer data. In master mode, the SPI data length depends on the SPCMD0.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 30.3.5, Transfer Format.

(3) RSPCK delay (t1)

The RSPCK delay value of the SPI in master mode depends on the SPCMD0.SCKDEN bit setting and the SPCKD register setting. The SPI determines an RSPCK delay during serial transfer using the SPCMD0.SCKDEN bit and SPCKD, as listed in Table 30.8. For a definition of RSPCK delay, see section 30.3.5, Transfer Format.

Table 30.8 Relationship between the SCKDEN bit, SPCKD, and RSPCK delay

SPCMD0.SCKDEN bit	SPCKD.SCKDL[2:0] bits	RSPCK delay
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(4) SSL negation delay (t2)

The SSL negation delay value of the SPI in master mode depends on the SPCMD0.SLNDEN bit setting and the SSLND register setting. The SPI determines an SSL negation delay during serial transfer using the SPCMD0.SLNDEN bit and SSLND, as listed in Table 30.9. For a definition of SSL negation delay, see section 30.3.5, Transfer Format.

Table 30.9 Relationship between the SLNDEN bit, SSLND, and SSL negation delay

SPCMD0.SLNDEN bit	SSLND.SLNDL[2:0] bits	SSL negation delay
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(5) Next-access delay (t3)

The next-access delay value of the SPI in master mode depends on the SPCMD0.SPNDEN bit setting and the SPND register setting. The SPI determines a next-access delay during serial transfer using the SPCMD0.SPNDEN bit and SPND, as listed in Table 30.10. For a definition of next-access delay, see section 30.3.5, Transfer Format.

Table 30.10 Relationship between the SPNDEN bit, SPND, and next-access delay (1 of 2)

SPCMD0.SPNDEN bit	SPND.SPNDL[2:0] bits	Next-access delay
0	000b to 111b	1 RSPCK + 2 PCLKB

Table 30.10 Relationship between the SPNDEN bit, SPND, and next-access delay (2 of 2)

SPCMD0.SPNDEN bit	SPND.SPNDL[2:0] bits	Next-access delay
1	000b	1 RSPCK + 2 PCLKB
	001b	2 RSPCK + 2 PCLKB
	010b	3 RSPCK + 2 PCLKB
	011b	4 RSPCK + 2 PCLKB
	100b	5 RSPCK + 2 PCLKB
	101b	6 RSPCK + 2 PCLKB
	110b	7 RSPCK + 2 PCLKB
	111b	8 RSPCK + 2 PCLKB

(6) Initialization flow

Figure 30.32 shows an example of SPI initialization flow when the SPI is in master mode. For information on how to set up the ICU, DTC, and I/O ports, see the individual block descriptions.

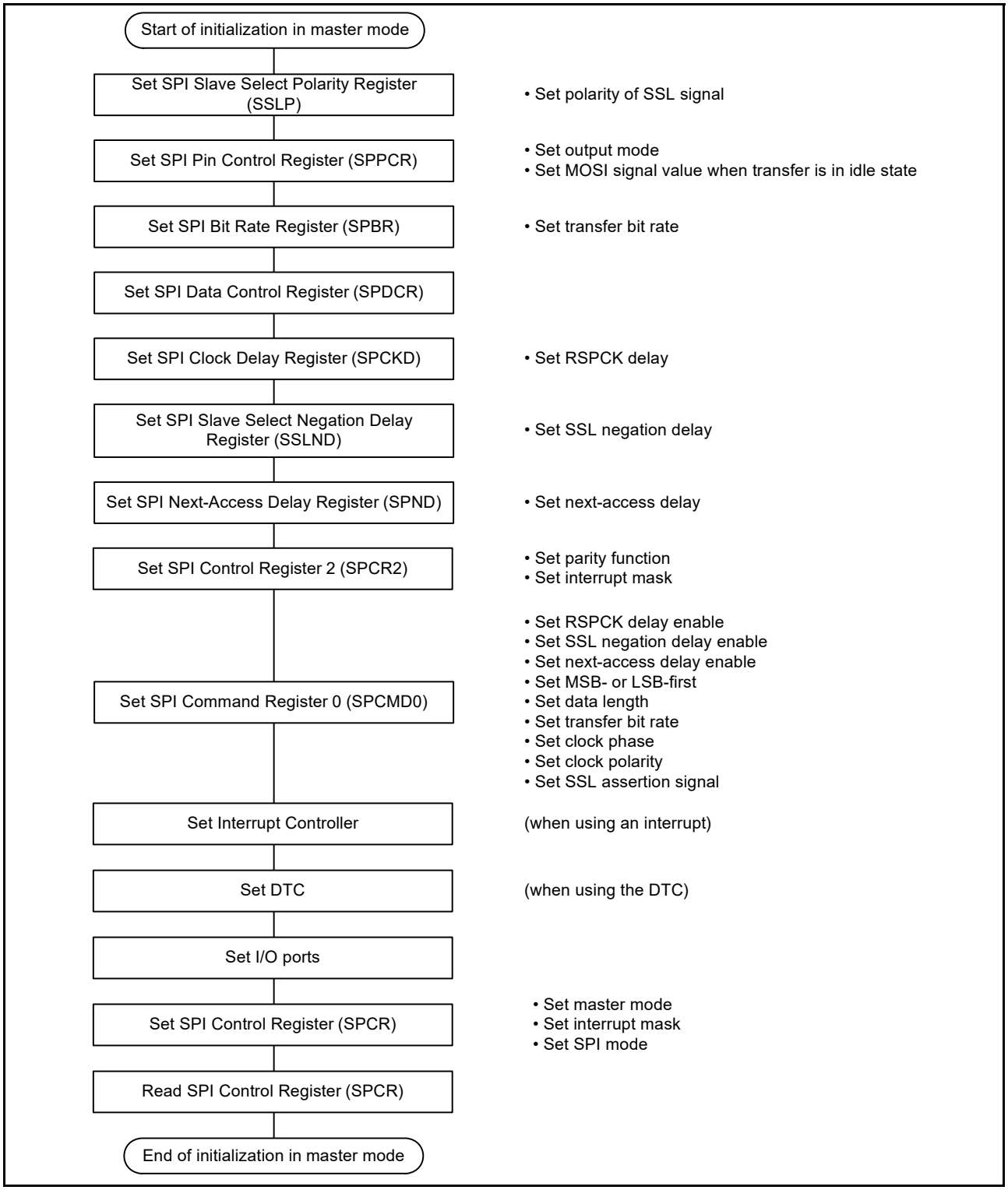


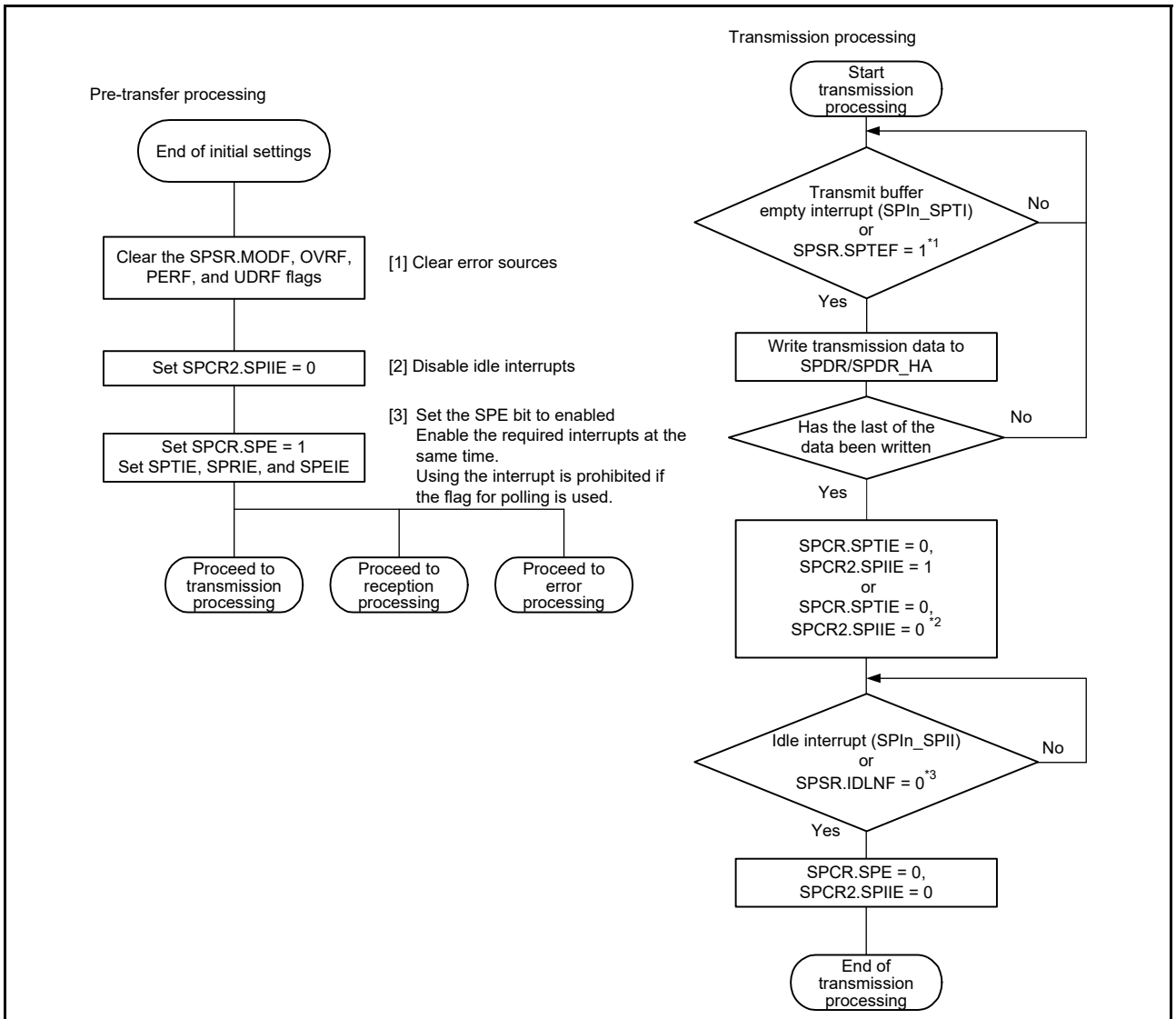
Figure 30.32 Example of initialization flow in master mode for SPI operation

(7) Software processing flow

Figure 30.33 to Figure 30.35 show examples of the software processing flow.

(a) Transmit processing flow

When transmitting data, with the SPIn_SPII interrupt enabled, the CPU is notified of the completion of data transmission after the last data write for transmission.



Note 1. Before writing data for transmission to SPDR/SPDR_HA, check the transmit buffer empty interrupt 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 PCLKB after writing data for transmission to SPDE and before starting to poll PSR.IDLNF, if the flag for polling is used.

Figure 30.33 Transmission flow in master mode

(b) Receive processing flow

The SPI does not handle receive-only operations, therefore processing for transmission is required.

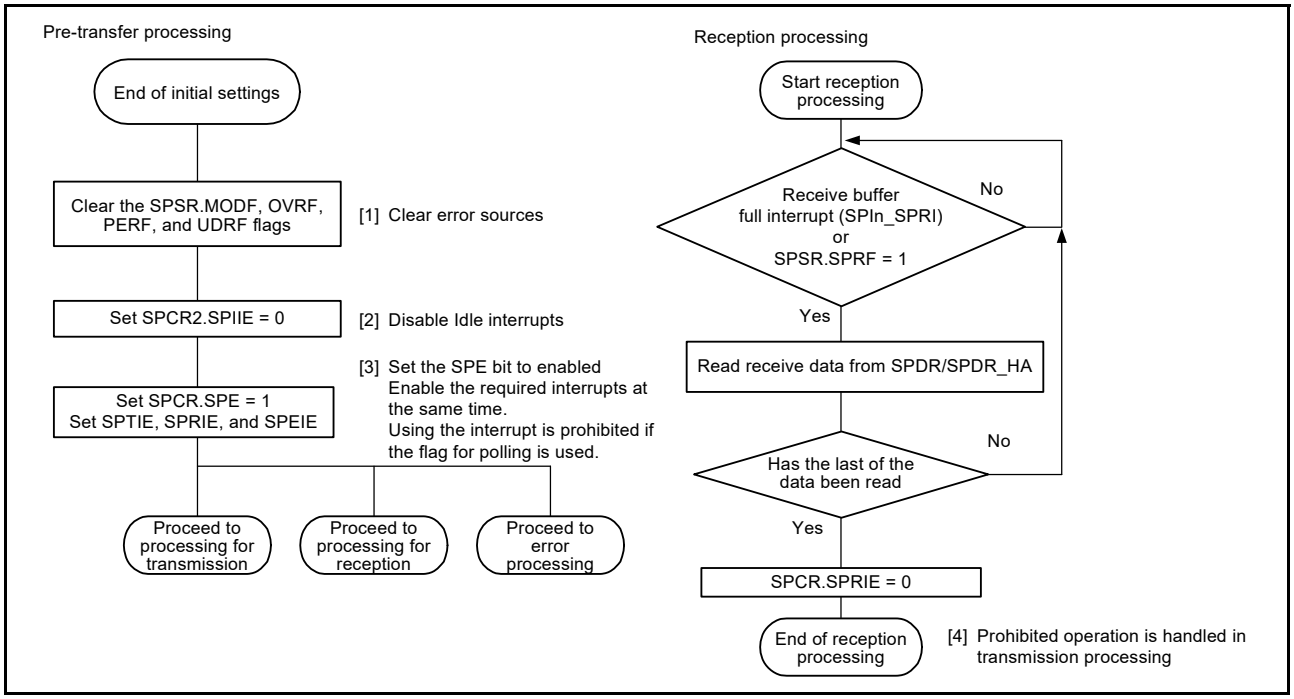


Figure 30.34 Reception flow in master mode

(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.

When an error is detected using an interrupt, clear the ICU.IELSRm.IR flag in the error processing routine. If this is not done, the ICU.IELSRm.IR flag might continue to indicate a transmit buffer empty or a receive buffer full interrupt request. If an SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

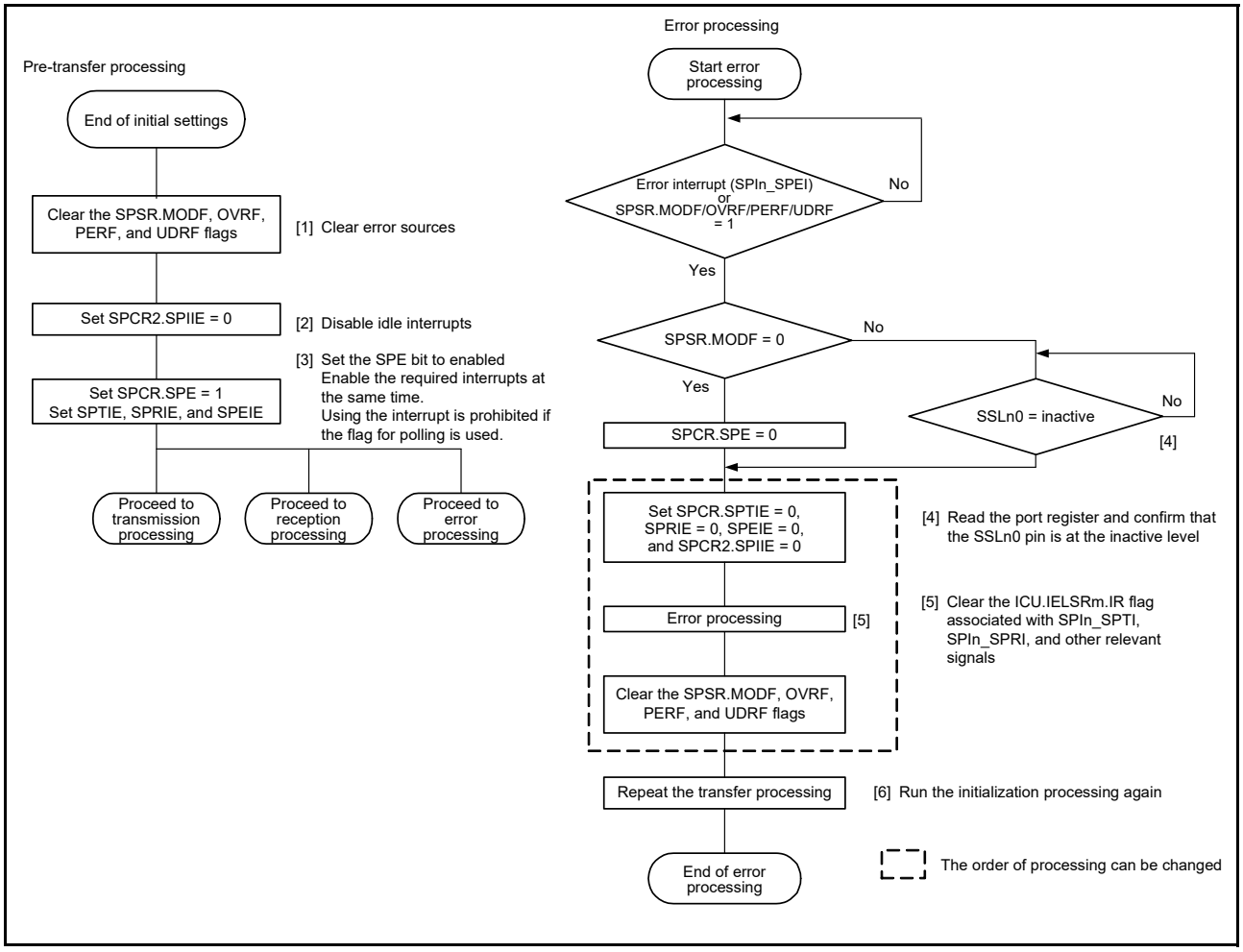


Figure 30.35 Error processing flow in master mode

30.3.10.2 Slave mode operation

(1) Starting a serial transfer

When the SPCMD0.CPHA bit is 0, if the SPI detects an SSLn0 input signal assertion, it must drive valid data to the MISOn output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLn0 input signal triggers the start of a serial transfer.

When the CPHA bit is 1, if the SPI detects the first RSPCKn edge in an SSLn0 signal asserted condition, it must drive valid data to the MISOn output signal. For this reason, when the CPHA bit is 1, the first RSPCKn edge in an SSLn0 signal asserted condition triggers the start of a serial transfer.

Regardless of the CPHA bit setting, the SPI drives the MISOn output signal on SSLn0 signal assertion. The data that is output by the SPI is either valid or invalid, depending on the CPHA bit setting.

The polarity of the SSLn0 input signal depends on the SSLP.SSL0P setting. For details on the SPI transfer format, see section 30.3.5, Transfer Format.

(2) Terminating a serial transfer

Regardless of the SPCMD0.CPHA bit setting, 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 (the SPSR.SPRF flag is 0), on termination of a serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR/SPDR_HA register. On termination of a serial transfer, the SPI changes the status of the shift register to empty, regardless of the receive buffer state. A mode fault error occurs if the SPI detects an SSLn0 input signal negation from the beginning of the serial transfer to the end of the serial transfer (see section 30.3.8, Error Detection).

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length is determined by the SPCMD0.SPB[3:0] bit setting. The polarity of the SSLn0 input signal is determined by the SSLP.SSL0P bit setting. For details on the SPI transfer format, see section 30.3.5, 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 30.7, 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) Initialization flow

Figure 30.36 shows an example of initialization flow for SPI operation when the SPI is in slave mode. For information on how to set up the ICU, DTC, and I/O ports, see the individual block descriptions.

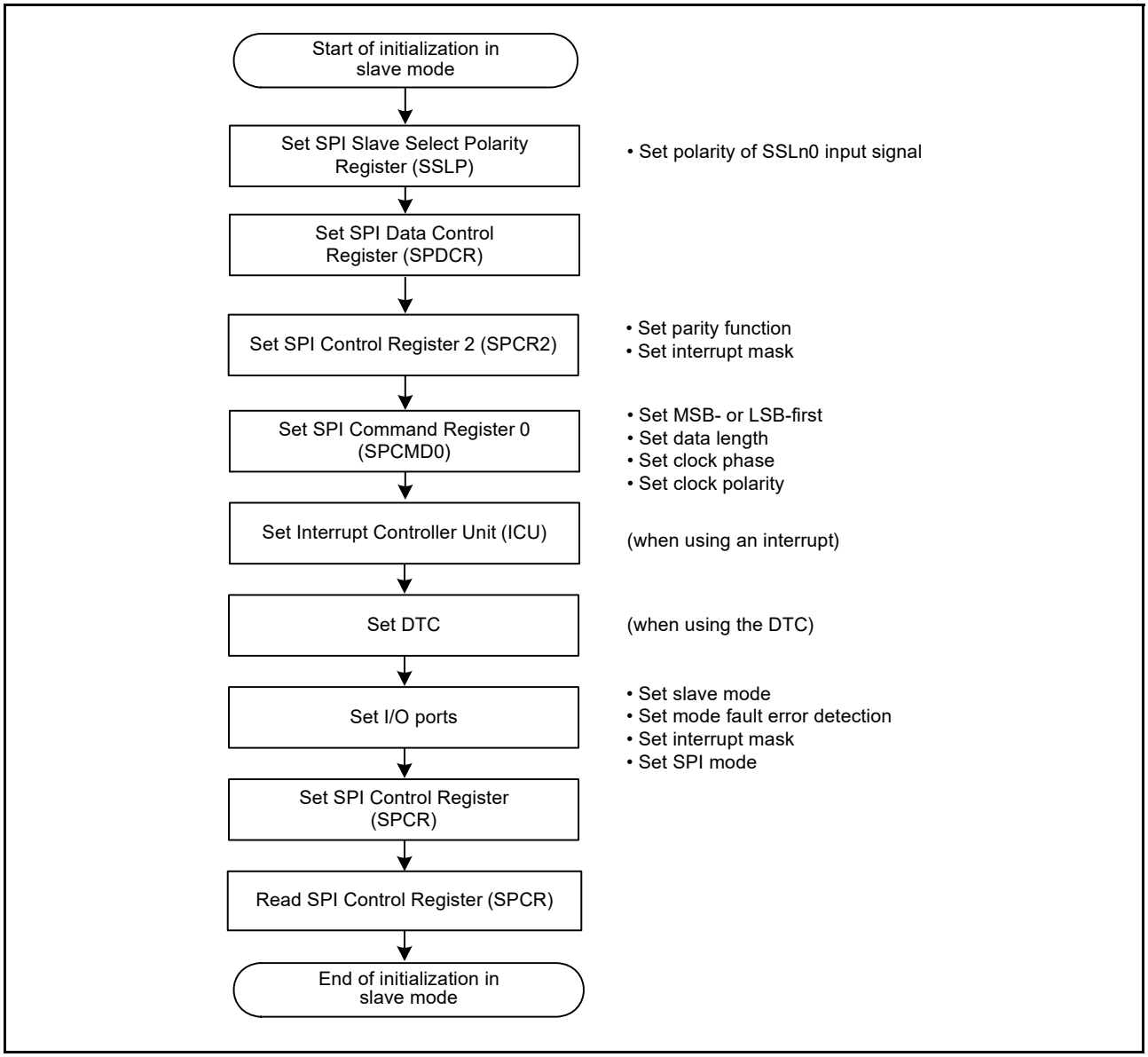
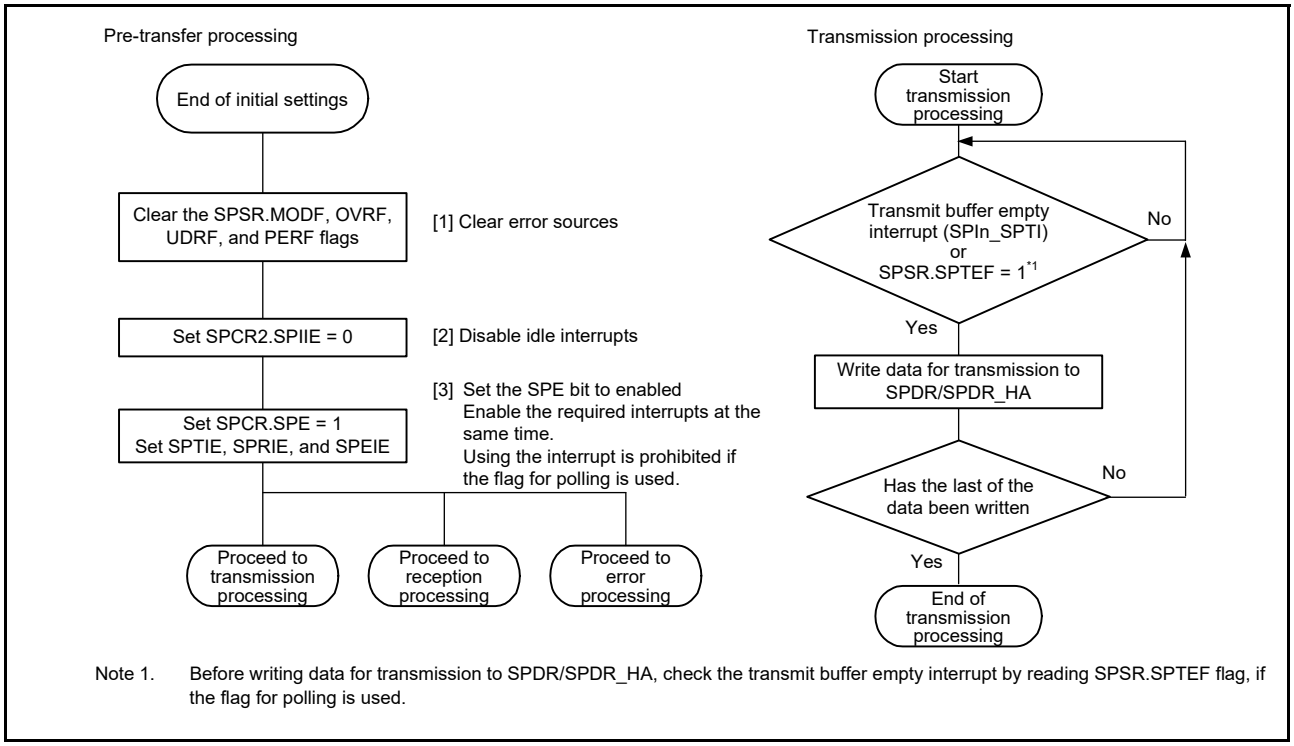


Figure 30.36 Example initialization flow in slave mode for SPI operation

(5) Software processing flow

Figure 30.37 to Figure 30.39 show examples of the software processing flow.

(a) Transmit processing flow



Note 1. Before writing data for transmission to SPDR/SPDR_HA, check the transmit buffer empty interrupt by reading SPSR.SPTEF flag, if the flag for polling is used.

Figure 30.37 Transmission flow in slave mode

(b) Receive processing flow

The SPI does not handle receive-only operation, therefore processing for transmission is required.

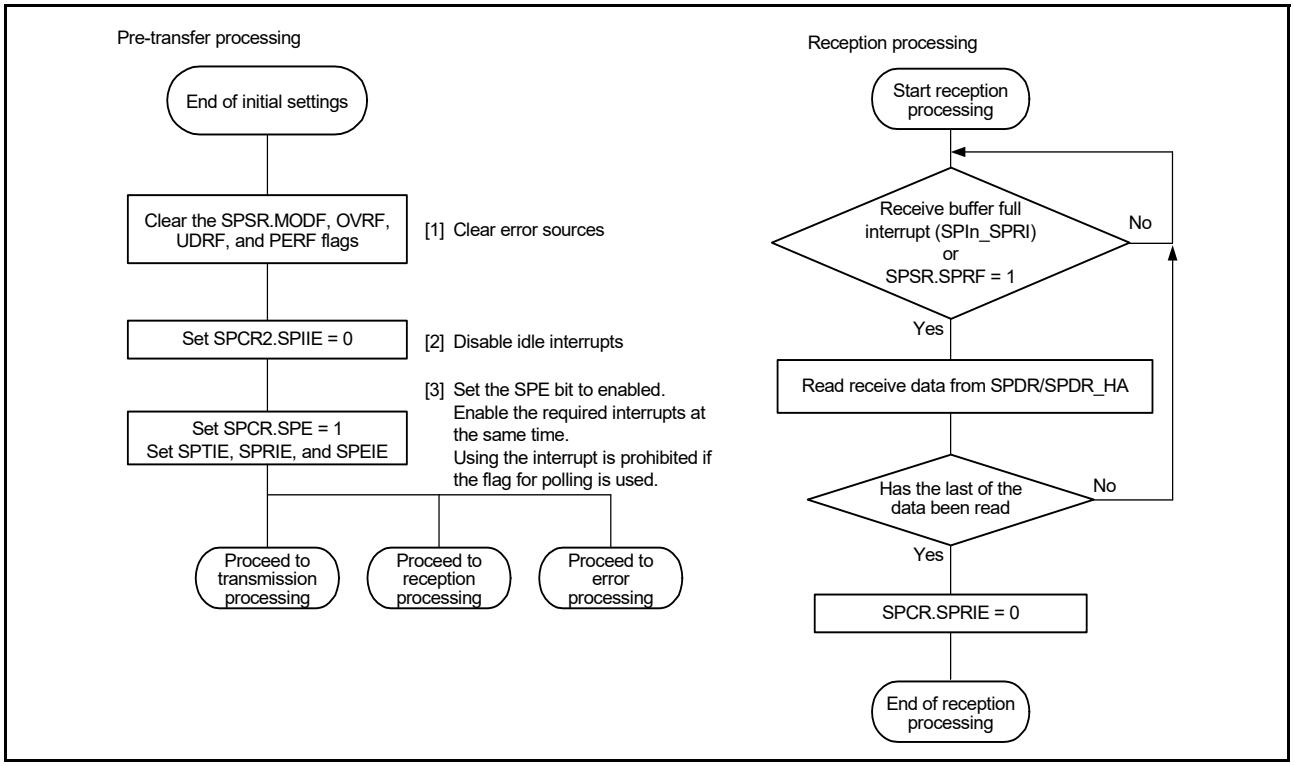


Figure 30.38 Reception flow in slave mode

(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.IELSRm.IR flag in the error processing routine. If this is not done, the ICU.IELSRm.IR flag might continue to indicate the transmit buffer empty or receive buffer full interrupt request. If the receive buffer full request is indicated, read the receive buffer and initialize the sequencer in the SPI.

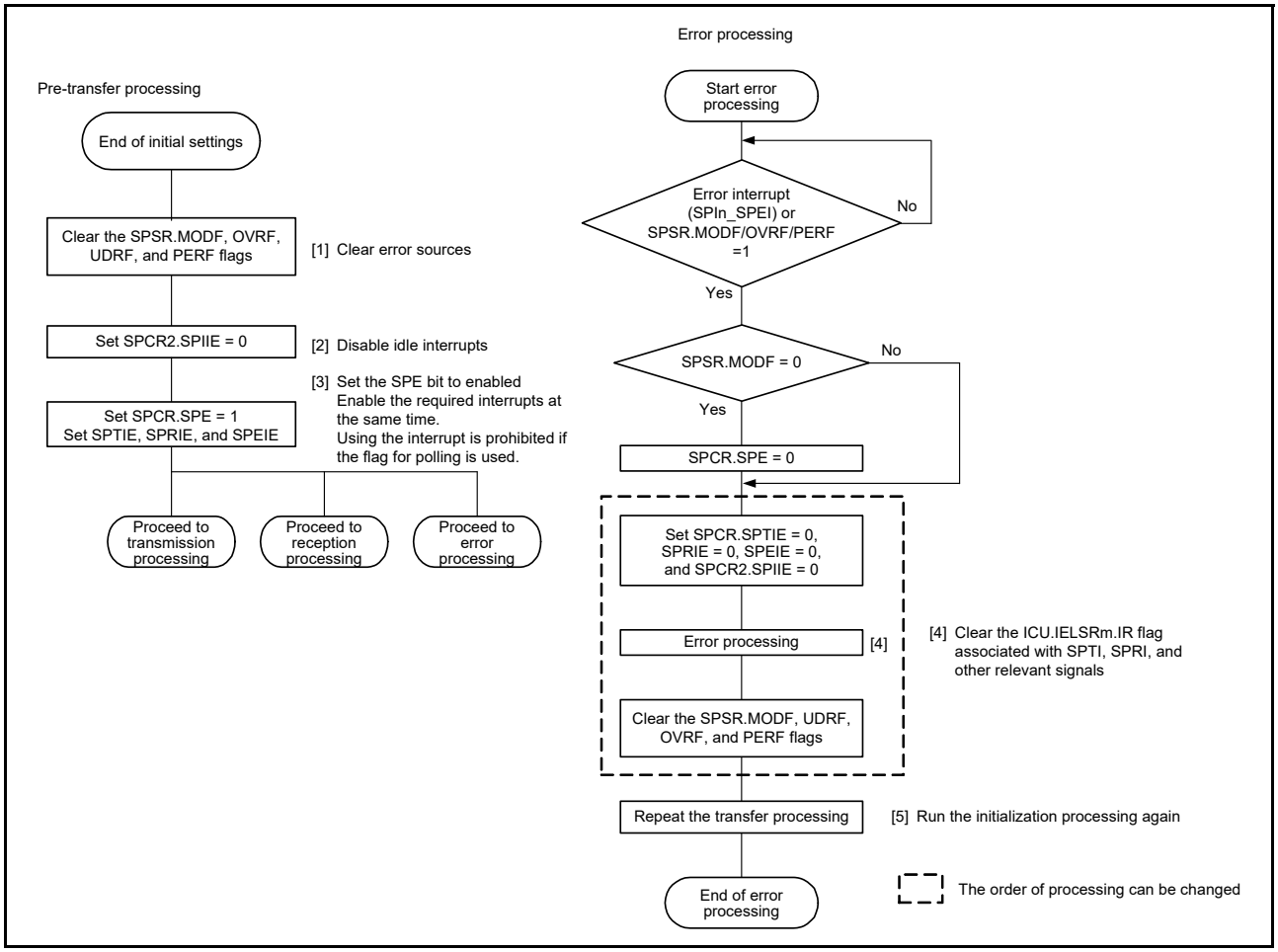


Figure 30.39 Error processing flow for slave mode

30.3.11 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 SPCMD0.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

30.3.11.1 Master mode operation

(1) Starting serial transfer

The SPI updates the data in the transmit buffer (SPTX) of SPDR/SPDR_HA when data is written to the SPDR/SPDR_HA register with the transmit buffer being empty, that is, data for the next transfer is not set, and the SPSR.SPTEF flag is 1. When the shift register is empty after data is written to 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 30.3.5, 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 (the SPSR.SPRF flag is 0), on termination of serial transfer, the SPI copies data from the shift register to the receive buffer of the SPI Data Register (SPDR/SPDR_HA).

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMD0.SPB[3:0] bit setting. Transfer in clock synchronous operation is conducted without the SSLn0 output signal. For details on the SPI transfer format, see section 30.3.5, Transfer Format.

(3) Initialization flow

Figure 30.40 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, DTC, and I/O Ports, see the individual block descriptions.

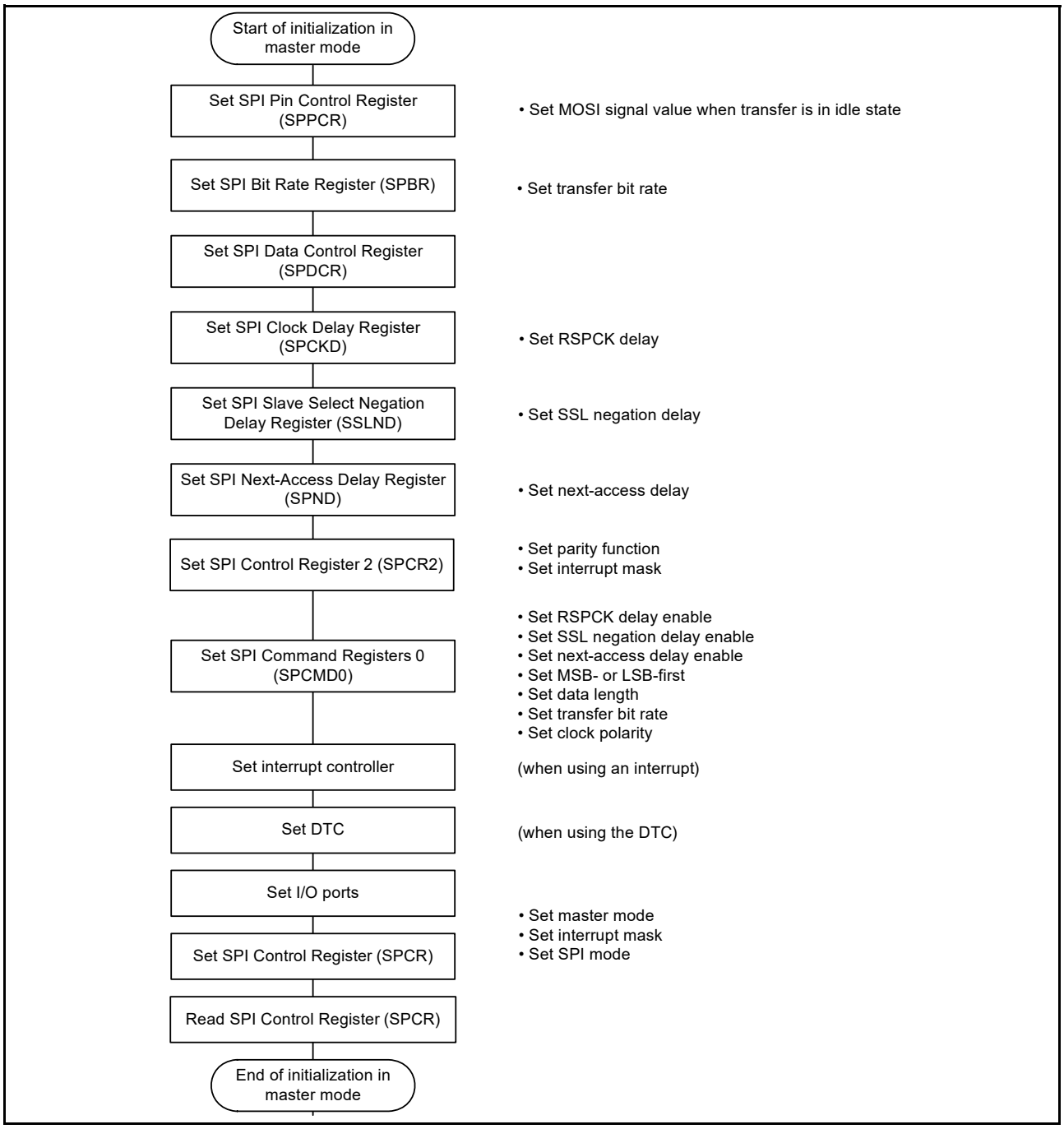


Figure 30.40 Example of initialization flow in master mode for clock synchronous operation

(4) Software processing flow

Software processing during clock synchronous master operation is the same as that for SPI master operation. For details, see section 30.3.10.1, (7) Software processing flow.

Note: Mode fault errors are not generated in clock synchronous operation.

30.3.11.2 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 30.3.5, 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 (the SPSR.SPRF flag is 0), on termination of a serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR/SPDR_HA register. On termination of a serial transfer, the SPI changes the status of the shift register to empty regardless of the receive buffer. The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length depends on the SPCMD0.SPB[3:0] bit setting.

For details on the SPI transfer format, see section 30.3.5, Transfer Format.

(3) Initialization flow

Figure 30.41 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, DTC, and I/O Ports, see the individual block descriptions.

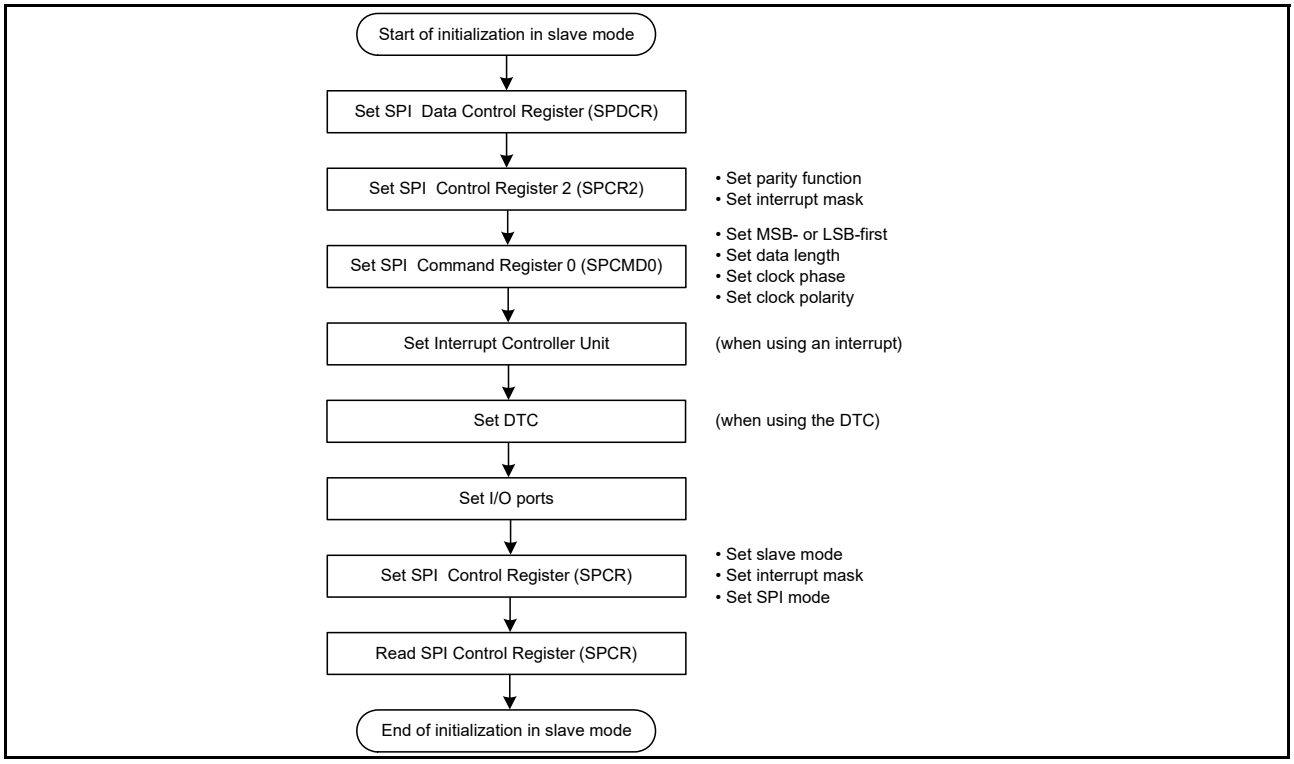


Figure 30.41 Example of initialization flow in slave mode for clock synchronous operation

(4) Software processing flow

Software processing during clock synchronous slave operation is the same as that for SPI slave operation. For details, see section 30.3.10.2, (5) Software processing flow.

Note: Mode fault errors are not generated in clock synchronous operation.

30.3.12 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the SPI shuts off the path between the MISOn pin and the shift register if the SPCR.MSTR bit is 1, or between the MOSIn pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path of the shift register, establishing a loopback mode. The SPI does not shut off the path between the MOSIn pin and the shift register if the SPCR.MSTR bit is 1, or between the MISOn pin and the shift register if the SPCR.MSTR bit is 0. 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 30.11 lists the relationship between the SPLP2 and SPLP bits and the received data. Figure 30.42 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 30.11 SPLP2 and SPLP bit settings and received data

SPPCR.SPLP2 bit	SPPCR.SPLP bit	Received data
0	0	Input data from the MOSIn or MISOn pin
0	1	Inverted transmit data
1	0	Transmit data
1	1	Transmit data

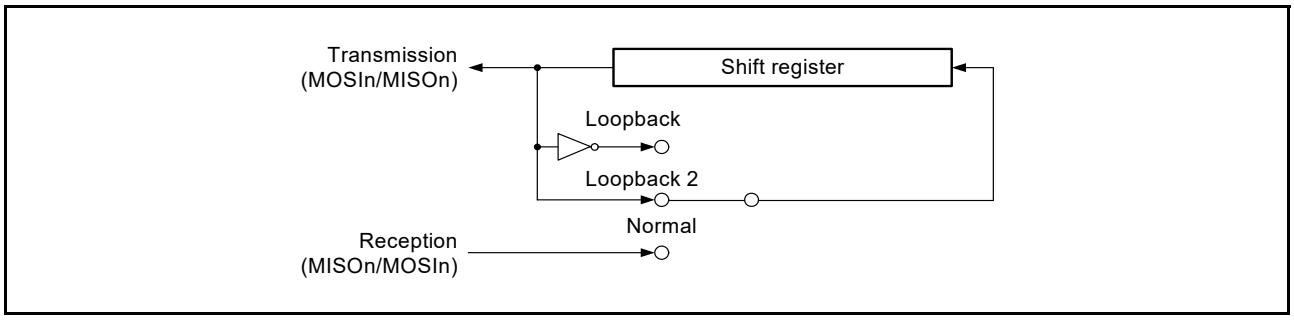


Figure 30.42 Configuration of shift register I/O paths in loopback mode for master mode

30.3.13 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. To detect defects in these units, the parity circuit performs self-diagnosis as shown in Figure 30.43.

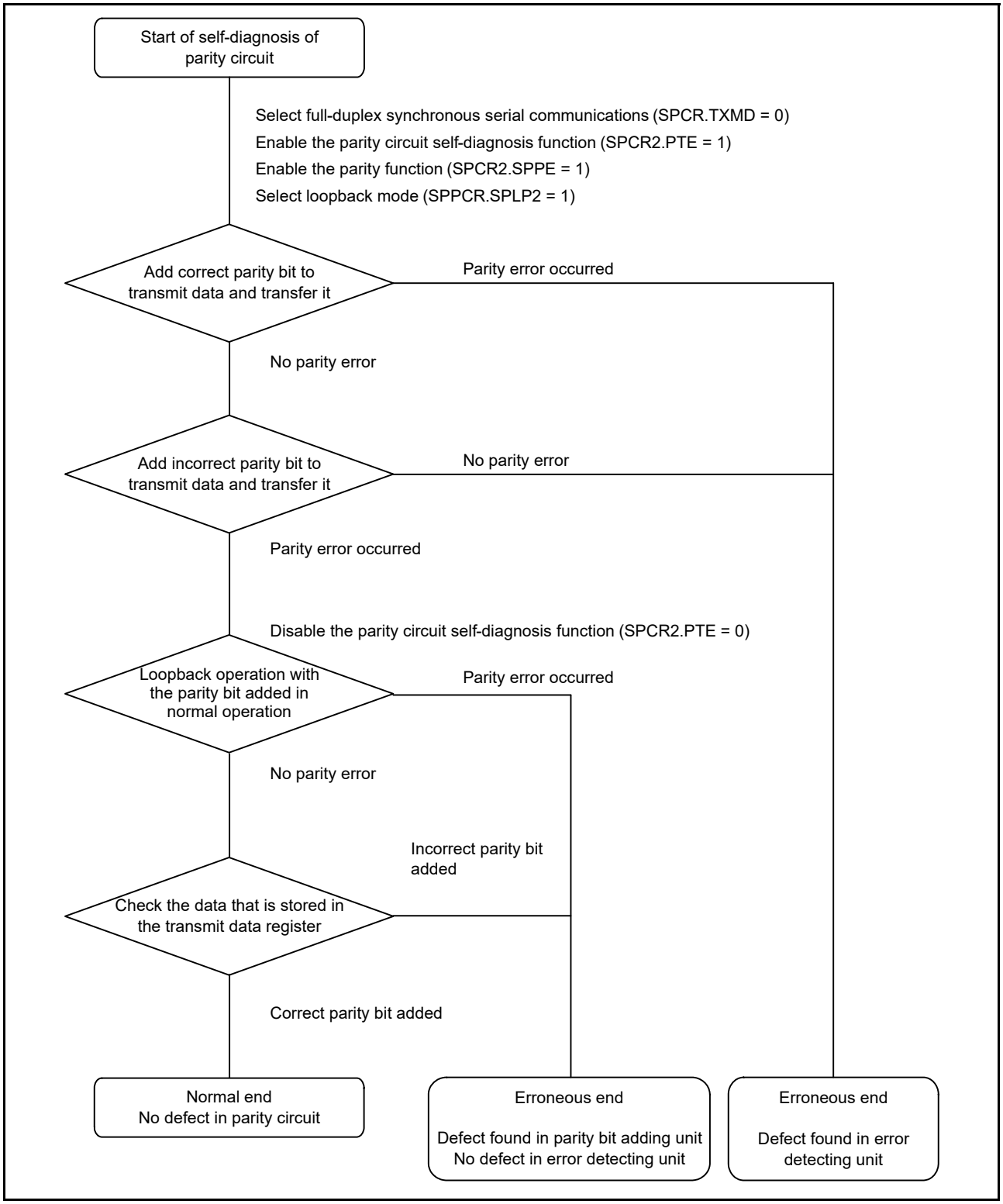


Figure 30.43 Self-diagnosis flow for parity circuit

30.3.14 Interrupt Sources

The SPI has eight interrupt sources:

- Receive buffer full
- Transmit buffer empty
- Transmission completed
- Mode fault
- Underrun
- Overrun
- Parity error
- SPI idle.

The DTC 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 30.12 lists the flags associated with the interrupt sources for the SPI. An interrupt is generated on satisfaction of an interrupt condition in Table 30.12. Clear the receive buffer full and transmit buffer empty sources through a data transfer.

When using the DTC to perform data transmission or reception, you must first set up the DTC to be in a transfer-enabled status before setting the SPI. For information on how to set the DTC, see section 16, Data Transfer Controller (DTC).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt occur while the ICU.IELSRm.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.IELSRm.IR flag becomes 0. A saved interrupt request is automatically discarded when it is output as an actual interrupt request. The interrupt enable bit (SPCR.SPTIE or SPCR.SPRIE) for an internally saved interrupt request can also be set to 0.

Table 30.12 SPI interrupt sources

Interrupt source	Symbol	Interrupt condition	DTC activation
Receive buffer full	SPIn_SPRI	The receive buffer becomes full (SPSR.SPRF flag is 1) while the SPCR.SPRIE bit is 1	Possible
Transmit buffer empty	SPIn_SPTI	The transmit buffer becomes empty (SPSR.SPTEF flag is 1) while the SPCR.SPTIE bit is 1	Possible
SPI errors (mode fault, underrun, overrun, and parity error)	SPIn_SPEI	The SPSR.MODF, OVRF, PERF, and 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 30.14.	Impossible

30.4 Event Link Controller Event Output

The Event Link Controller (ELC) can produce the following event output signals:

- Receive buffer full event output
- Transmit buffer empty event output
- Mode fault, underrun, overrun, or parity event output
- SPI idle event output
- Transmission-completed event output.

The event link output signal is output regardless of the interrupt enable bit setting.

30.4.1 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.

30.4.2 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.

30.4.3 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 30.5.4, Restrictions on Mode Fault, Underrun, Overrun, or Parity Error Event Output if using this event signal.

(1) Mode fault

Table 30.13 lists the conditions for occurrence of a mode fault event.

Table 30.13 Conditions for occurrence of mode fault

Condition	SPCR.MODFEN bit	SSLn0 pin	Remark
SPI operation (SPMS = 0) Slave (SPCR.MSTR = 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 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 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 in the TXMD bit in SPCR is 0 and the SPPE bit in SPCR2 is 1.

30.4.4 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).

30.4.5 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 30.14 lists the conditions for occurrence of a transmission-completed event in slave mode.

Table 30.14 Conditions for generation of transmission-completed event in slave mode

Mode of operation	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.

30.5 Usage Notes

30.5.1 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.

30.5.2 Restrictions 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.

30.5.3 Restrictions on Starting Transfer

If the ICU.IELSRm.IR flag is 1 when transfer starts, the interrupt request is internally saved after transfer starts, which can lead to unanticipated behavior of the ICU.IELSRm.IR flag.

To prevent this, use the following procedure to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1):

1. Confirm that transfer stopped (SPCR.SPE is 0).
2. Set the associated interrupt enable bit (SPCR.SPTIE or SPCR.SPRIE) to 0.
3. Read the associated interrupt enable bit (SPCR.SPTIE or SPCR.SPRIE) and confirm that its value is 0.
4. Set the ICU.IELSRm.IR flag to 0.

30.5.4 Restrictions 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, the SPCR.MSTR bit is 1, and the SPCR.MODFEN bit is 1).

30.5.5 Restrictions on 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.

31. Cyclic Redundancy Check (CRC) Calculator

31.1 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 31.1 lists the CRC calculator specifications and Figure 31.1 shows a block diagram.

Table 31.1 CRC specifications

Parameter	Specifications	
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.

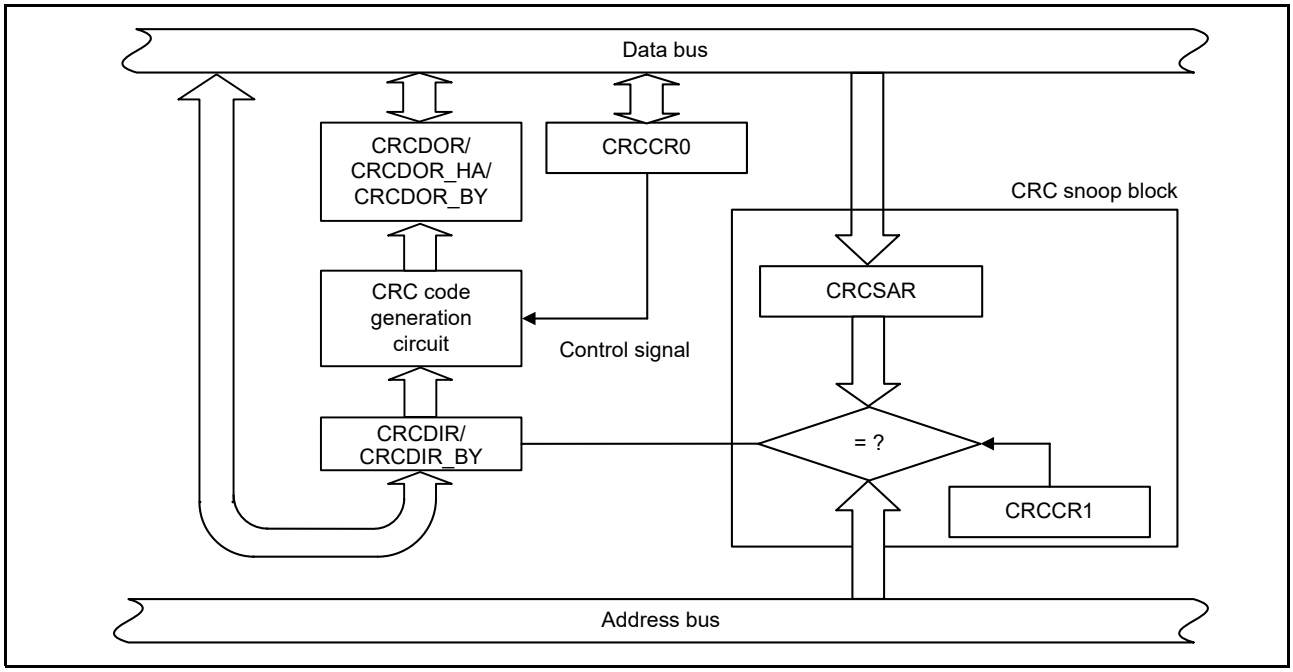


Figure 31.1 CRC block diagram

31.2 Register Descriptions

31.2.1 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.	R/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 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 31.3, Operation.

DORCLR bit (CRCDOR/CRCDOR_HA/CRCDOR_BY Register Clear)

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.

31.2.2 CRC Control Register 1 (CRCCR1)

Address(es): CRC.CRCCR1 4007 4001h

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

Bit	Symbol	Bit name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CRCSWR	Snoop-On-Write/Read Switch	0: Snoop-on-read 1: Snoop-on-write.	R/W
b7	CRCSEN	Snoop Enable	0: Disabled 1: Enabled.	R/W

CRCSWR bit (Snoop-On-Write/Read Switch)

The CRCSWR bit selects the direction of the 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.

31.2.3 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 (CRCDIR[31:24]) is a read/write 8-bit register to write data for CRC-8, CRC-16, or CRC-CCITT calculation.

31.2.4 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 (CRCDOR[31:16]) register is a read/write 16-bit register for CRC-16 or CRC-CCITT calculation.

The CRCDOR_BY (CRCDOR[31:24]) 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 perform the calculations using a value other than the initial value.

Data written to the CRCDIR/CRCDIR_BY register is CRC calculated and the result is stored in the CRCDOR/CRCDOR_HA/CRCDOR_BY register. If the CRC code is calculated following transferred data and the result is 0000 0000h, there is no CRC error.

31.2.5 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-bit 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.

31.3 Operation

31.3.1 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 31.2 and Figure 31.3 show the LSB-first and MSB-first data transmission examples respectively. Figure 31.4 and Figure 31.5 show the LSB-first and MSB-first data reception examples.

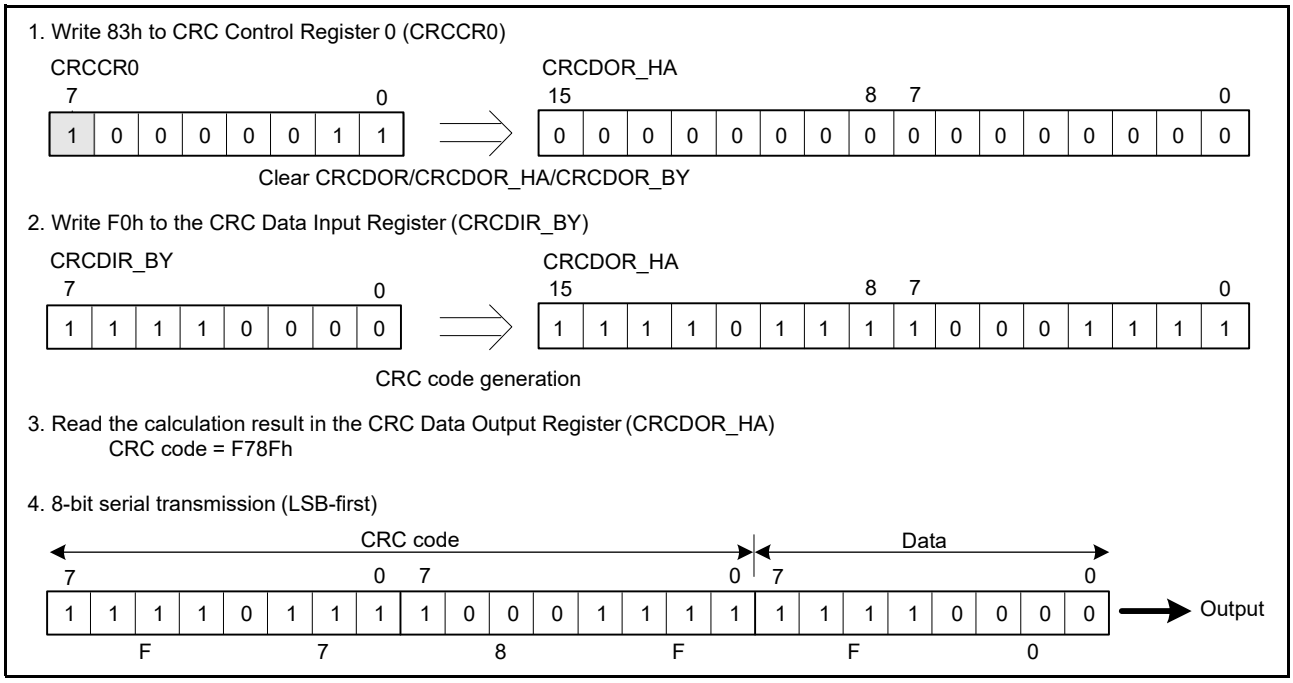


Figure 31.2 LSB-first data transmission

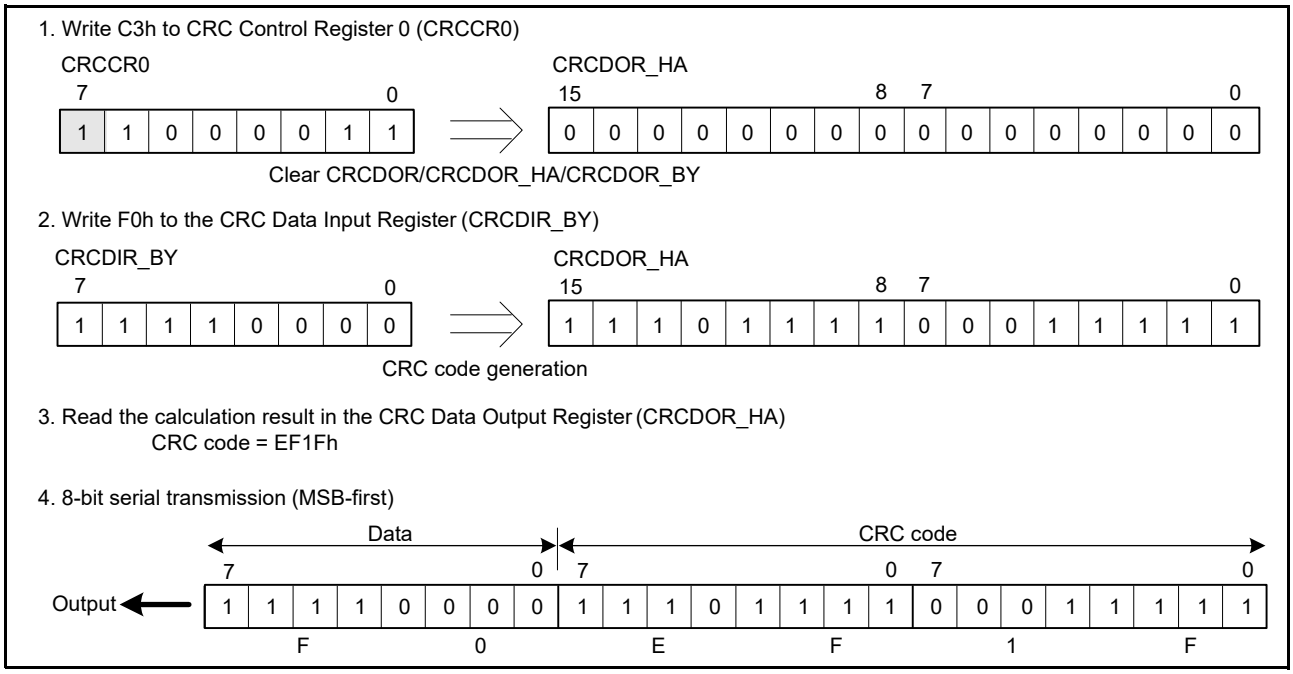


Figure 31.3 MSB-first data transmission

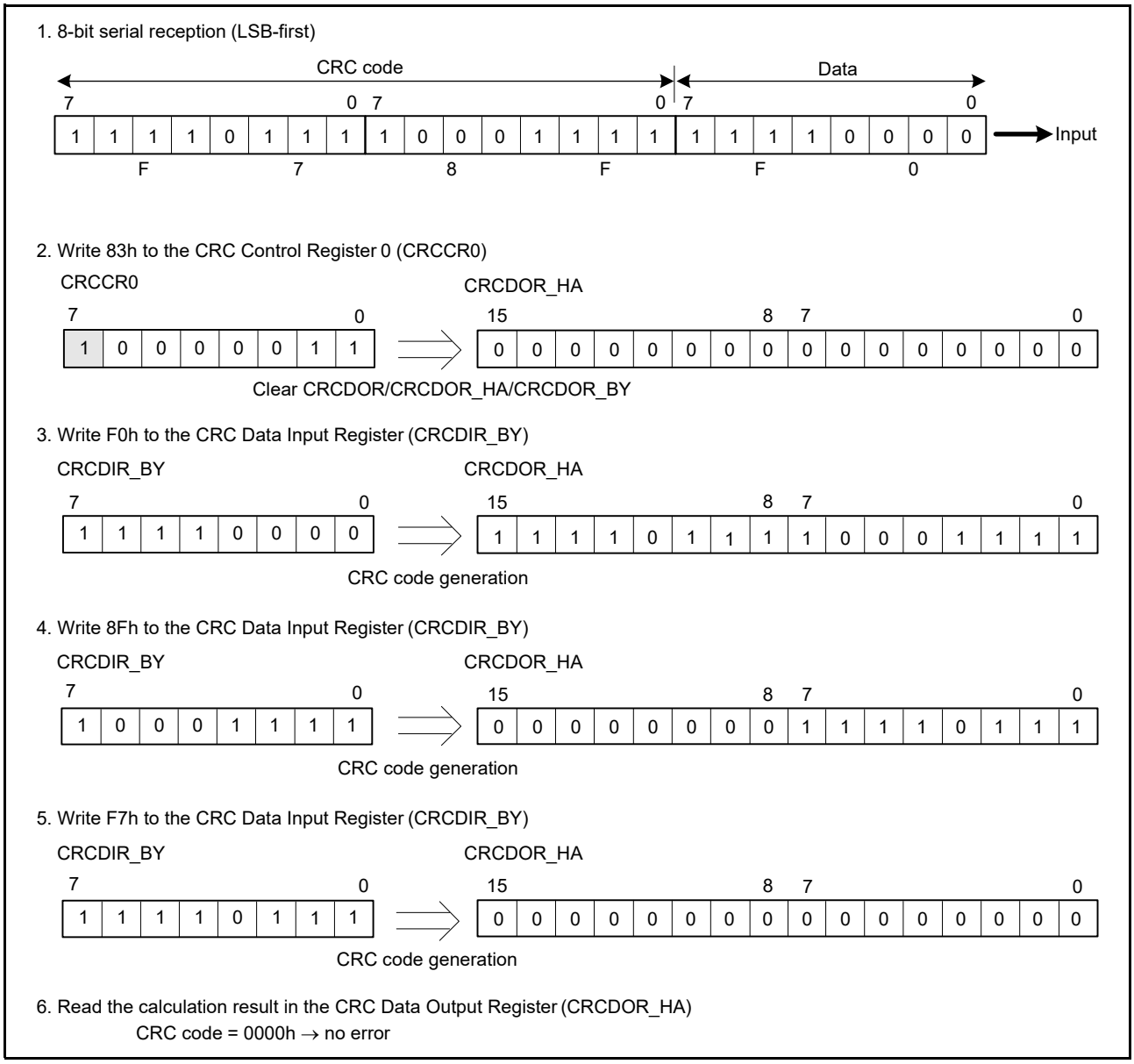


Figure 31.4 LSB-first data reception

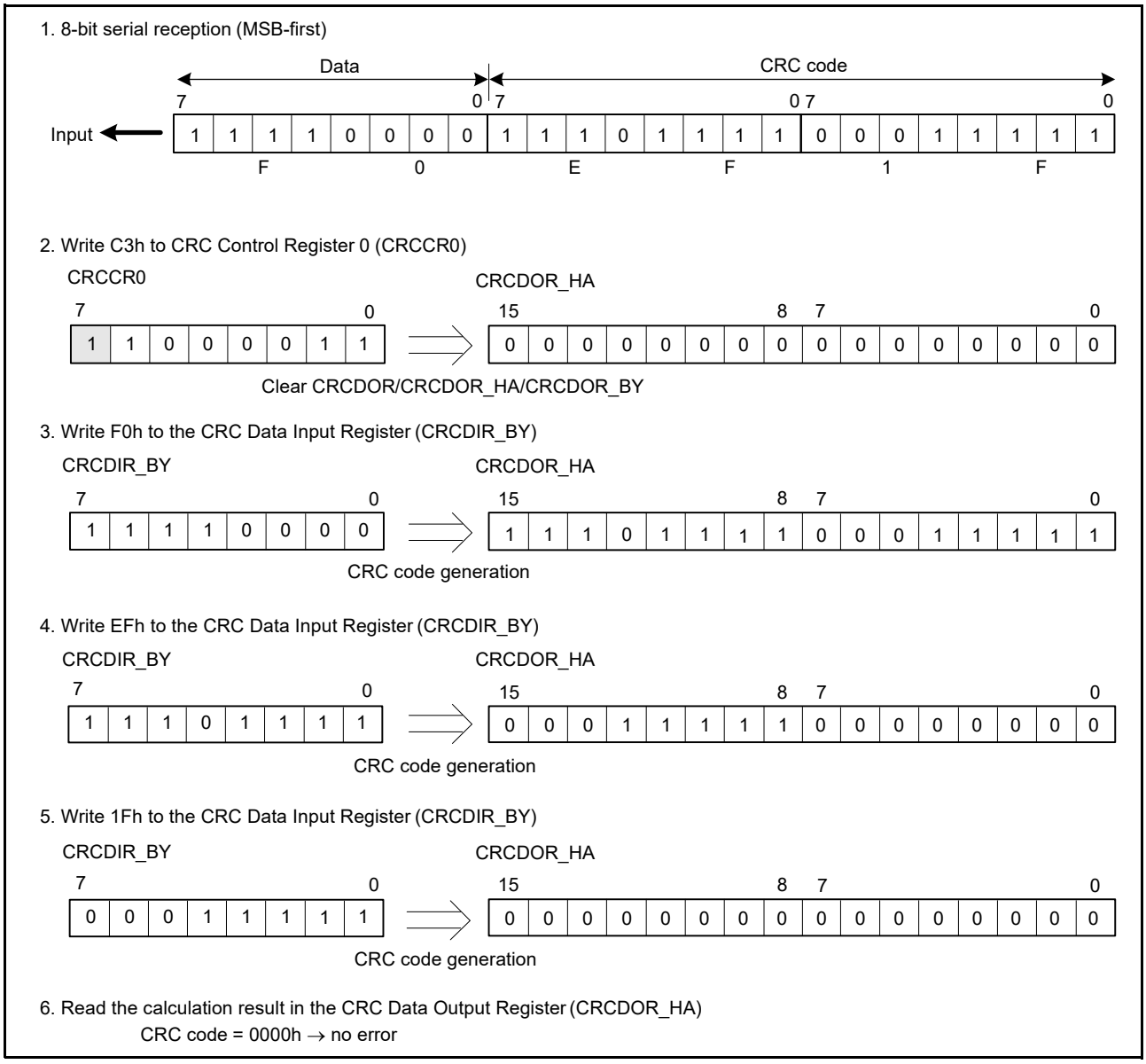


Figure 31.5 MSB-first data reception

31.3.2 CRC Snoop

The CRC snoop function monitors reads from and writes to a specific register address and performs CRC calculation on the data read from and written to that register address automatically. Because the CRC snoop recognizes writes to and reads from a specific register address as a trigger to automatically perform CRC calculation, there is no need to write data to the CRCDIR_BY register. All I/O register addresses specified in the Snoop Address Register (CRCSAR) are subject to the CRC snoop. The CRC snoop is useful in monitoring writes to the serial transmit buffer, and reads from the serial receive buffer.

To use this function, write a target I/O register address to bits CRCSA13 to CRCSA0 in the CRCSAR register, and set CRCSEN bit in the CRCCR1 register to 1. Then, set the CRCSWR bit in the CRCCR1 register to 1 to enable snooping on writes to the target address, or set the CRCSWR bit in the CRCCR1 register to 0 to enable snooping on reads from the target address.

When setting the CRCSEN bit to 1, CRCSWR bit to 1 and writing data to a target I/O register address in a bus master module such as CPU and DTC, the CRC calculator stores the data in the CRCDIR_BY register and performs CRC calculation. Similarly, when setting the CRCSEN bit to 1, CRCSWR bit to 0 and reading data in a target I/O register address in a bus master module such as CPU 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.

31.4 Usage Notes

31.4.1 Settings for the Module-Stop State

The Module Stop Control Register C (MSTPCRC) can enable or disable CRC calculator operation. The CRC is initially stopped after a reset. The registers become accessible on releasing from the module-stop state. For details, see section 11, Low Power Modes.

31.4.2 Notes on Transmission

The sequence of transmission for the CRC code differs based on whether the transmission is LSB-first or MSB-first.

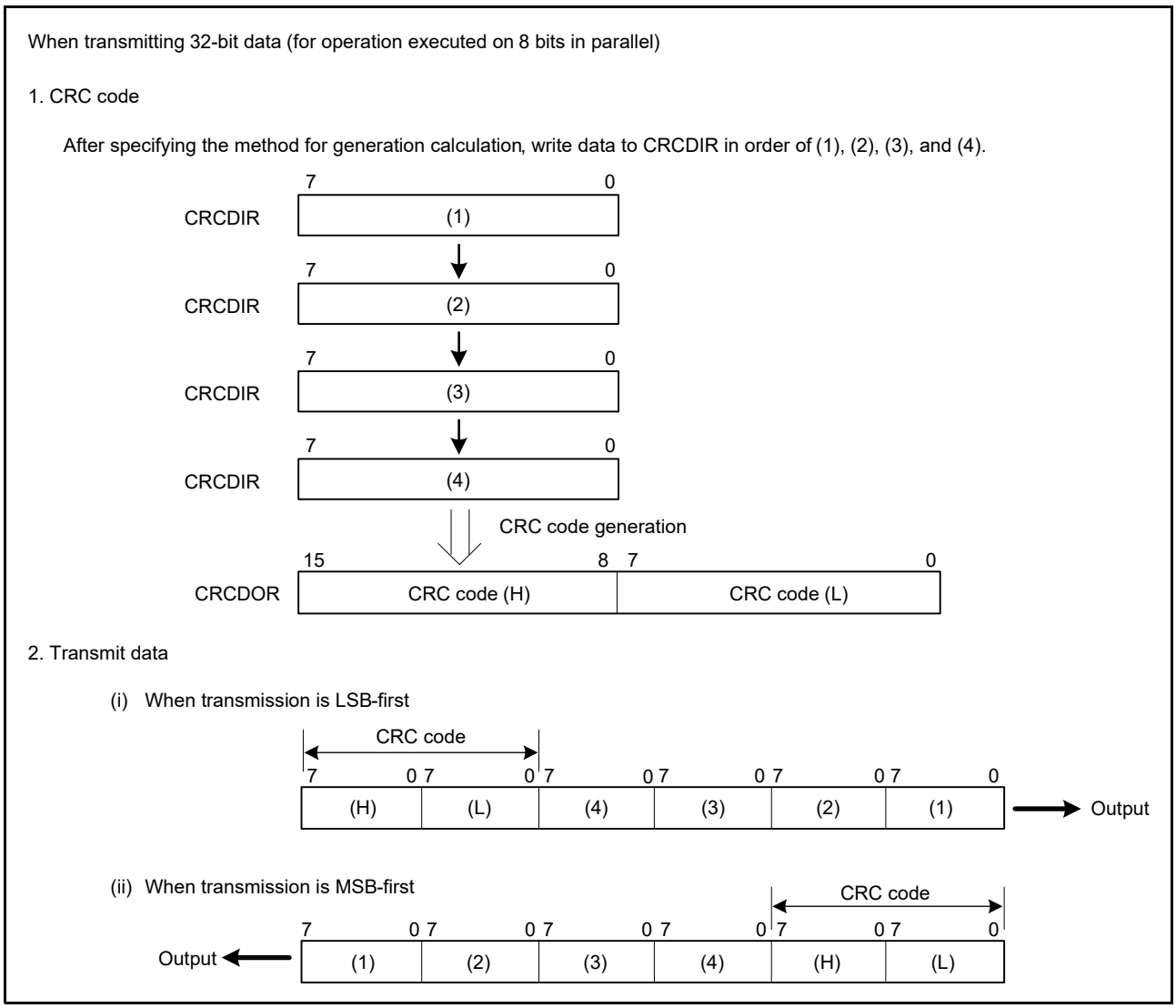


Figure 31.6 LSB-first and MSB-first data transmission

32. 16-Bit A/D Converter (ADC16)

32.1 Overview

The MCU provides a 16-bit successive approximation A/D converter (ADC16) unit. Up to 17 analog input channels are selectable. Reference voltage of SDADC24*1, temperature sensor output and internal reference voltage can be selected for conversion. The A/D conversion accuracy is 16-bit conversion.

Analog inputs of channels AN000 to AN007 are used for selection of single-ended input and differential input, and analog inputs of channels AN008 and AN016 to AN023*1 are used only as single-ended inputs. Up to 17 channels of single-ended inputs can be used. Differential analog inputs of channels 0 to 7 can be used as differential inputs of up to four channels. For differential inputs, even channels and odd channels (even channel + 1), such as channel 0 and channel 1 or channel 2 and channel 3, are used as differential inputs for A/D conversion.

Note 1. The reference voltage of SDADC24 is SBIAS/VREFI. When not using SDADC24, SBIAS/VREFI cannot be A/D converted. For SBIAS/VREFI setting, see section 33, 24-Bit Sigma-Delta A/D Converter (SDADC24).

The ADC16 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 ADC16 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 ADC16 is A/D-converted.

The temperature sensor output and internal reference voltage cannot be selected for conversion simultaneously. The temperature sensor output and the internal reference voltage are converted independently.

The ADC16 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, and outputs an interrupt when the A/D-converted value of the selected channel meets the comparison conditions.

The reference power supply pin (VREFH0) or internal reference voltage for ADC16 (VREFADC) is selectable as the high potential reference voltage. The low potential reference voltage is the reference power supply ground pin (VREFL0).

Calibration allows high-precision A/D conversion by obtaining the Capacitor Array DAC (C-DAC) linearity error correction value and gain (offset) error correction value under the conditions of use. Normal A/D conversion during calibration or calibration during normal A/D conversion is disabled.

Table 32.1 lists the ADC16 specifications, Table 32.2 lists the functions, and Figure 32.1 shows a block diagram.

Table 32.1 ADC16 specifications (1 of 2)

Parameter	Specifications
Number of units	One unit
Input channels	• Single-ended input mode: Up to 17 channels (AN000 to AN008, AN016 to AN023) • Differential input mode: Up to 4 channels (AN000 to AN007)
Extended input	Temperature sensor output, internal reference voltage, reference voltage of SDADC24*4
A/D conversion method	Successive approximation method
Resolution	16 bits
Conversion time*6	0.82 μs/channel when A/D conversion clock PCLKD (ADCLK) operates at 32 MHz
A/D conversion clock	Peripheral module clock PCLKB*1 and A/D conversion clock PCLKD (ADCLK)*1 can be set with the following division ratios: PCLKB to PCLKD (ADCLK) frequency ratio = 1:1, 1:2, 1:4
Data registers	• 17 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 reference voltage of SDADC24 • 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 • 16-bit accuracy output for A/D conversion results • 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 reference voltage of SDADC24, on the temperature sensor output, and on the internal reference voltage • Continuous scan mode*5: - A/D conversion is performed repeatedly on the analog inputs of arbitrarily selected channels, on the reference voltage of SDADC24, on the temperature sensor output, and on the internal reference voltage. • Group scan mode*5: - 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*5 (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.
Functions	• Variable sampling state count • Self-diagnosis of A/D converter • Selectable A/D-converted value average mode • Analog input disconnection detection function (discharge and precharge functions) • Double trigger mode (duplication of A/D conversion data) • Automatic clear function for A/D data registers • Digital comparison of values in the comparison register and the data register, and between values in the data registers • Analog characteristics can be corrected by calibration (C-DAC, gain (offset) error).

Table 32.1 ADC16 specifications (2 of 2)

Parameter	Specifications
Interrupt sources	• In single scan mode (double trigger deselected), an A/D scan end interrupt request and ELC event signal (ADC160_ADI*3) can be generated on completion of a single scan. - A compare interrupt request (ADC160_CMPAI/ADC160_CMPBI) can be generated in response to matches with a digital comparison condition. - A window compare ELC event signal (ADC160_WCMPM) can be generated in response to matches with a condition for digital comparison. - A window compare ELC event signal (ADC160_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 (ADC160_ADI*3) can be generated on completion of two scans • In continuous scan mode, an A/D scan end interrupt request and ELC event signal (ADC160_ADI*3) 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 (ADC160_ADI*3) can be generated on completion of group A scan, whereas an A/D scan end interrupt request for group B (ADC160_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 (ADC160_ADI*3) can be generated on completion of two group A scans, and an A/D scan end interrupt request for group B (ADC160_GBADI) can be generated on completion of group B scan • The ADC160_ADI*3, ADC160_GBADI, ADC160_WCMPM, and ADC160_WCMPUM can activate the Data Transfer Controller (DTC) • A calibration end interrupt request (ADC160_ADI*3) can be generated on completion of calibration.
ELC interface	Scan can be started by a trigger from the ELC
Reference voltage	• VREFH0 or VREFADC can be selected as the high potential reference voltage • VREFL0 is the low potential reference voltage.
Module-stop function	Module-stop state can be specified*2

Note 1. Peripheral module clock PCLKB is set in the SCKDIVCR.PCKB[2:0] bits and A/D conversion clock ADCLK is set in the SCKDIVCR.PCKD[2:0] bits. The maximum frequency of PCLKB is 32 MHz and the maximum frequency of PCLKD (ADCLK) is 32 MHz when ADC16 is in use.

Note 2. For details, see section 11, Low Power Modes.

Note 3. When using a calibration end interrupt request (ADC160_ADI), set the A/D Interrupt Control Register (ADICR) to 03h. When using an A/D scan end interrupt request and ELC event signal (ADC160_ADI), set the A/D Interrupt Control Register (ADICR) to 00h.

Note 4. The reference voltage of SDADC24 is SBIAS/VREFI. When not using SDADC24, SBIAS/VREFI cannot be A/D converted. For SBIAS/VREFI setting, see section 33, 24-Bit Sigma-Delta A/D Converter (SDADC24).

Note 5. When selecting the temperature sensor output or the internal reference voltage, do not use continuous scan mode or group scan mode.

Note 6. A/D conversion processing time under the fastest condition.

Table 32.2 ADC16 functions (1 of 2)

Parameter			ADC160
Analog input channel			AN000 to AN008, AN016 to AN023 Reference voltage of SDADC24*3 Internal reference voltage Temperature sensor output
Conditions for A/D conversion start	Software	Software trigger	Enabled
	External trigger	Trigger input pin	ADTRG0
	Synchronous trigger (trigger from ELC)	ELC trigger	ELC_AD00 ELC_AD01
Interrupt			ADC160_ADI ADC160_GBADI ADC160_CMPAI ADC160_CMPBI
Output to ELC			ADC160_ADI ADC160_WCMPM ADC160_WCMPUM

Table 32.2 ADC16 functions (2 of 2)

Parameter	ADC160
Setting of module-stop function*1, *2	MSTPCRD.MSTPD16 bit

Note 1. For details, see section 11, Low Power Modes.
Note 2. Wait for settling time to start A/D conversion after release from the module-stop state. For details on the settling time, see section 47, Electrical Characteristics.
Note 3. The reference voltage of SDADC24 is SBIAS/VREFI. When not using SDADC24, SBIAS/VREFI cannot be A/D converted. For SBIAS/VREFI setting, see section 33, 24-Bit Sigma-Delta A/D Converter (SDADC24).

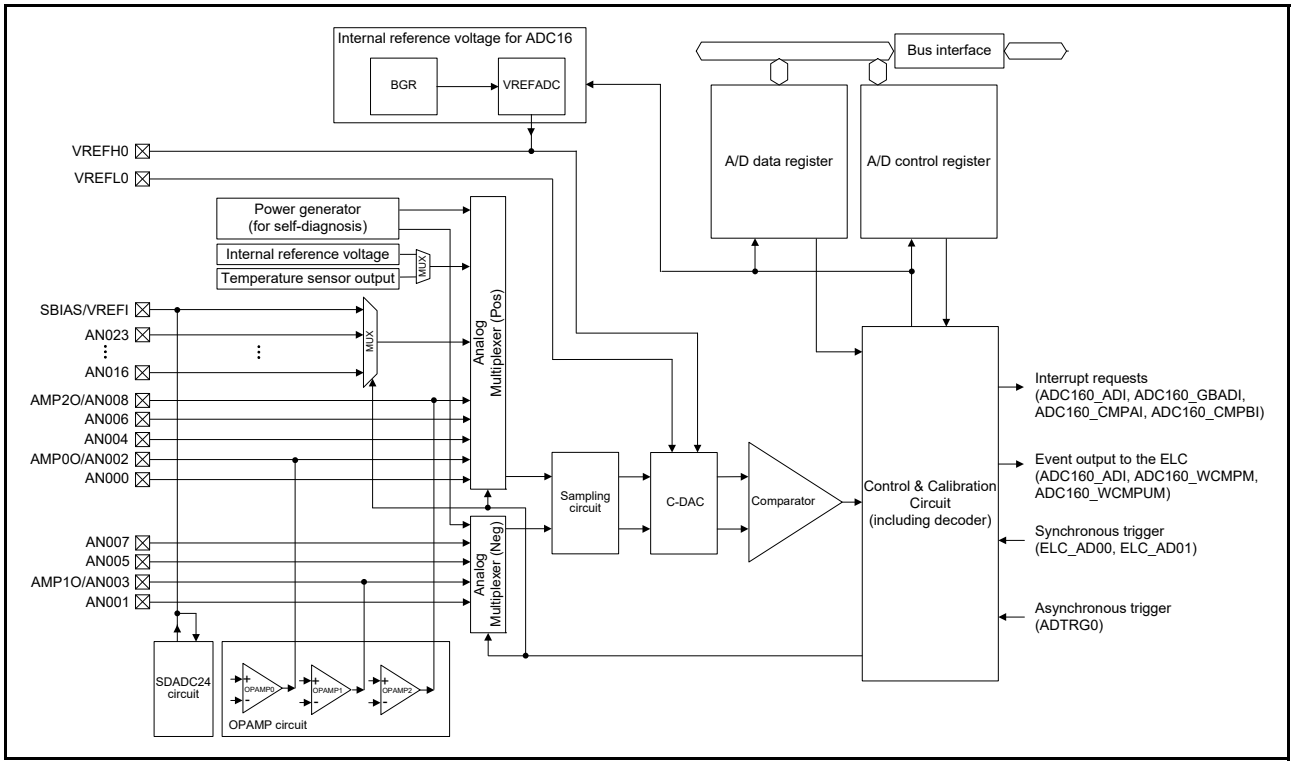


Figure 32.1 ADC16 block diagram

Table 32.3 lists the ADC16 I/O pins.

Table 32.3 ADC16 I/O pins

Pin name	I/O	Function
AVCC0	Input	Analog block power supply pin
AVSS0	Input	Analog block power supply ground pin
VREFH0	Input*1	Reference power supply pin
	Output*2	VREFADC output voltage
VREFL0	Input	Reference power supply ground pin
AN000 to AN008, AN016 to AN023	Input	Analog input pins 00 to 08, 16 to 23
ADTRG0	Input	External trigger input pin for starting A/D conversion
SBIAS/VREFI*3	I/O	Power supply pin for sensor or external VREF input pin for the SDADC24.

Note 1. When VREFH0 is applied to the high potential reference voltage.
Note 2. When VREFADC is applied to the high potential reference voltage.
Note 3. For details, see section 33, 24-Bit Sigma-Delta A/D Converter (SDADC24).

32.2 Register Descriptions

32.2.1 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 = 0 to 8, 16 to 24*1): 16-bit read-only registers for storing the A/D conversion results. When a differential input is selected, the A/D conversion result is stored in the even-numbered ADDRy registers (y = 0 to 7). A/D data register value of odd-numbered ADDRy registers (y = 0 to 7) are not guaranteed.
- ADDBLDR register: 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 registers: 16-bit read-only registers for storing the A/D conversion results in response to the triggers during extended operation in double trigger mode.
- ADTSDR register: 16-bit read-only register for storing the A/D conversion result of temperature sensor output.
- ADOCDR register: 16-bit read-only register for storing the A/D conversion result of internal reference voltage.

Note 1. ADDR24 is the register for storing the A/D conversion result of the reference voltage of SDADC24 (SBIAS/VREFI).

The following conditions determine the formats for data in the ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, and ADOCDR registers:

- The setting of the Average Count Select bits (ADADC.ADC[2:0]) (once, twice, three times, four times, or 16 times setting).

(1) When A/D-converted value average mode is not selected

Address(es): ADC160.ADDR0 4005 C020h to ADC160.ADDR8 4005 C030h, ADC160.ADDR16 4005 C040h to ADC160.ADDR24 4005 C050h, ADC160.ADDBLDR 4005 C018h, ADC160.ADDBLDRA 4005 C084h, ADC160.ADDBLDRB 4005 C086h, ADC160.ADTSDR 4005 C01Ah, ADC160.ADOCDR 4005 C01Ch

	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]	Converted Value 15 to 0	16-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, 4, 8 or 16 times is specified in the A/D-converted value average mode. This register indicates the mean of A/D-converted values on a specific channel.

32.2.2 A/D Self-Diagnosis Data Register (ADRD)

The ADRD register is a 16-bit read-only register that holds the A/D conversion results based on the self-diagnosis of ADC16. The self-diagnosis status is stored in the ADRST.DIAGST[1:0] bits, see section 32.2.3, A/D Self-Diagnostic Status Register (ADRST).

A/D-converted value average mode cannot be applied to the A/D self-diagnosis function. For details on self-diagnosis, see section 32.2.12, A/D Control Extended Register (ADCER).

Address(es): ADC160.ADRD 4005 C0F8h

	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]	Converted Value 15 to 0	16-bit A/D-converted value	R

32.2.3 A/D Self-Diagnostic Status Register (ADRST)

ADRST register is an 8-bit read-only register that stores self-diagnostic status information.

Address(es): ADC160.ADRST 4005 C0FAh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DIAGST[1:0]	
Value after reset:	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 under a condition that an ideal value of the A/D conversion result is 8000h 1 0: Self-diagnosis was executed under a condition that an ideal value of the A/D conversion result is 0000h 1 1: Self-diagnosis was executed under a condition that an ideal value of the A/D conversion result is 7FFFh. For details on self-diagnosis, see section 32.2.12, A/D Control Extended Register (ADCER).	R
b7 to b2	—	Reserved	These bits are read as 0	R

32.2.4 A/D Control Register (ADCSR)

Address(es): ADC160.ADCSR 4005 C000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADST	ADCS[1:0]		—	—	—	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
b6	GBADIE	Group B Scan End Interrupt Enable	0: Disable ADC160_GBADI interrupt generation on group B scan completion 1: Enable ADC160_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
b12, b10	—	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 PCLKB clock cycles.

DBLANS[4:0] bits (Double Trigger Channel Select)

The DBLANS[4:0] bits select one of the channels for A/D conversion data duplication in double trigger mode. When duplicating the channels selected for differential input, select the associated even-numbered channels. The A/D conversion results from the specified analog input channel in the DBLANS[4:0] bits 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 32.4 and Table 32.5 show selection of the channel for double-triggered operation.

A/D-converted value average mode in double trigger mode can be set by selecting the channel using the DBLANS[4:0] bits of the ADADS0/1 registers. In double trigger mode, the channels selected in the ADANSA0 and ADANSA1 registers are invalid, and the channel selected by 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 multiple channel analog input can be selected for group B even in double trigger mode.

Only set the DBLANS[4:0] bits when the ADST bit is 0. Do not set these bits at the same time you write 1 to the ADST bit.

To enter A/D-converted value average mode when in double trigger mode, select the channel using the DBLANS[4:0] bits in the ADANSA0 and ADANSA1 registers.

Table 32.4 Relationship between DBLANS bit settings and double-trigger enabled channels (single-ended input mode)

Single-ended input mode			
DBLANS[4:0]	**Duplication channel**	**DBLANS[4:0]**	**Duplication channel**
00000	AN000	10000	AN016
00001	AN001	10001	AN017
00010	AN002	10010	AN018
00011	AN003	10011	AN019
00100	AN004	10100	AN020
00101	AN005	10101	AN021
00110	AN006	10110	AN022
00111	AN007	10111	AN023
01000	AN008	11000	SBIAS/VREFI*1
01001	—	11001	—
01010	—	11010	—
01011	—	11011	—
01100	—	11100	—
01101	—	11101	—
01110	—	11110	—
01111	—	11111	—

Note: A/D converted data from the self-diagnosis function, temperature sensor output, and internal reference voltage cannot be used in double trigger mode.

Note 1. The reference voltage of SDADC24 is SBIAS/VREFI. When not using SDADC24, SBIAS/VREFI cannot be A/D converted. For SBIAS/VREFI setting, see section 33, 24-Bit Sigma-Delta A/D Converter (SDADC24).

Table 32.5 Relationship between DBLANS bit settings and double-trigger enabled channels (differential input mode)

Differential input mode	
DBLANS[4:0]	**Duplication channel**
00000	AN000, AN001
00001	—
00010	AN002, AN003
00011	—
00100	AN004, AN005
00101	—
00110	AN006, AN007

GBADIE bit (Group B Scan End Interrupt Enable)

The GBADIE bit enables or disables group B scan end interrupt (ADC160_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 ADC160_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 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 the DBLE bit is set (double trigger mode is selected), the channels specified in the ADANSA0 and ADANSA1 registers are invalid. Double trigger mode is deselected by setting DBLE bit to 0. When DBLE bit is set to 1 again, the double-trigger mode operation is the same as the first time when scanning with the first trigger.

Do not select double trigger mode in continuous scan mode. Software triggering cannot be used in double trigger mode. Always set the ADST bit to 0 before setting the DBLE bit. In other words, do not set the DBLE 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.

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 (AN000 to AN008, AN016 to AN023, SBIAS/VREFI) selected in the ADANSA0 and ADANSA1 registers in ascending order of channel number, for a maximum of 18 channels. When 1 cycle of A/D conversion completes for all the selected channels, the scan conversion stops.

In continuous scan mode, when the ADST bit in ADCSR register 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 ADST bit in ADCSR register is set to 0 during continuous scan, A/D conversion stops even when scanning is in progress.

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 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 of a maximum of 21 channels selected with the ADANSB0 and ADANSB1 registers, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion stops.

If the conversion processes in group A and 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 a priority to group A conversion.

In group scan mode, select different channels and triggers for group A and group B.

When selecting temperature sensor output or internal reference voltage, select single scan mode and perform A/D conversion after deselecting all analog input channels in the ADANSA0 and ADANSA1 registers. When A/D conversion of the temperature sensor output or internal reference voltage completes, A/D conversion stops.

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 32.6 Selectable targets for A/D Conversion depending on settings of scan mode and double trigger mode

Scan mode setting	Double trigger mode setting	Targets for A/D conversion				
		Self-diagnosis	Analog input (including group A)	Analog input (group B)	Temperature sensor output	Internal reference voltage
Single scan	DBLE = 0	✓	✓	x	✓	✓
	DBLE = 1	x	✓(1 ch only)	x	x	x
Continuous scan	DBLE = 0	✓	✓	x	x	x
	DBLE = 1	x	x	x	x	x
Group scan	DBLE = 0	✓	✓	✓	x	x
	DBLE = 1	x	✓(1 ch only)	✓	x	x

✓: Selectable. x: 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 the ADCSR.EXTRG bit is 0 and the ADCSR.TRGE bit is 1
- When the synchronous trigger (ELC) selected in the ADSTRGR.TRSB[5:0] bits is detected when the ADCSR.TRGE bit 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] = 01b and ADGSPCR.PGS = 1), the ADGSPCR.GBRP bit is set to 1, and A/D conversion of group B starts
- When 1 is written to the ADCALEXE.CALEXE bit.

[Clearing conditions]

- When 0 is written by software
- When 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 is complete in group scan mode
- When group B scan is complete in group scan mode
- When group A priority control operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), the ADGSPCR.GBRSCN bit is set to 1, and group B scanning completes
- When calibration is complete.

Note: When group A priority control operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), do not set the ADST bit to 1.

Note: When group A priority control operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), do not set the ADST bit to 0. When forcing A/D conversion to terminate, follow the procedure for clearing the ADST bit.

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 set to 1.

32.2.5 A/D Channel Select Register A0 (ADANSA0)

Address(es): ADC160.ADANSA0 4005 C004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ANSA08	ANSA07	ANSA06	ANSA05	ANSA04	ANSA03	ANSA02	ANSA01	ANSA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	ANSA08 to ANSA00	A/D Conversion Channels Select	0: Associated input channel not selected 1: Associated input channel selected. Bit [8] (ANSA08) corresponds to AN008 and bit [0] (ANSA00) corresponds to AN000.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ANSAn bits (n = 00 to 08) (A/D Conversion Channels Select)

The ANSAn.ADANSA0 bits select the analog input channels for A/D conversion from AN000 to AN008. The selected channels and their number can be set arbitrarily. The ANSA00 bit corresponds to AN000 and the ANSA08 bit corresponds to AN008.

When performing A/D conversion of 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).

When the ADANIM.ANIM[n] bit is set to 1 to select a differential input, select an even channel associated with ANIM[n].

Example: When ADANIM.ANIM[0] = 1, set the ADANSA0[1:0] bits to 01b.
When ADANIM.ANIM[1] = 1, set the ADANSA0[3:2] bits to 01b.
When ADANIM.ANIM[2] = 1, set the ADANSA0[5:4] bits to 01b.
When ADANIM.ANIM[3] = 1, set the ADANSA0[7:6] bits to 01b.

Only set the ADANSA0 register when the ADCSR.ADST bit is 0.

32.2.6 A/D Channel Select Register A1 (ADANSA1)

Address(es): ADC160.ADANSA1 4005 C006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ANSA24	ANSA23	ANSA22	ANSA21	ANSA20	ANSA19	ANSA18	ANSA17	ANSA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	ANSA24 to ANSA16	A/D Conversion Channels Select	0: Associated input channel not selected 1: Associated input channel selected. Bit [8] (ANSA24) corresponds to SBIAS/VREFI, bit [7] (ANSA23) corresponds to AN023 and bit [0] (ANSA16) corresponds to AN016.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ANSAn bits (n = 16 to 24) (A/D Conversion Channels Select)

The ADANSA1.ANSAn bits select the analog input channels for A/D conversion from AN016 to AN023 and SBIAS/VREFI. The selected channels and their number can be set arbitrarily. The ANSA16 bit corresponds to AN016, the ANSA23 bit corresponds to AN023, and the ANSA24 bit corresponds to SBIAS/VREFI.

When performing A/D conversion of 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.

32.2.7 A/D Channel Select Register B0 (ADANSB0)

Address(es): ADC160.ADANSB0 4005 C014h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ANSB08	ANSB07	ANSB06	ANSB05	ANSB04	ANSB03	ANSB02	ANSB01	ANSB00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	ANSB08 to ANSB00	A/D Conversion Channels Select	0: Associated input channel not selected 1: Associated input channel selected. Bit 8 (ANSB08) corresponds to AN008 and bit 0 (ANSB00) corresponds to AN000.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ANSBn bits (n = 00 to 08) (A/D Conversion Channels Select)

The ADANSB0.ANSBn bits select the analog input channels for A/D conversion from AN000 to AN008 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 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 ANSB00 bit corresponds to AN000, the ANSB04 bit corresponds to AN004, and the ANSB08 bit corresponds to AN008.

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.

When the ADANIM.ANIM[n] bit is set to 1 to select differential input, select the even channel corresponding to ANIM[n].

Example: When ADANIM.ANIM[0] = 1, set the ADANSB0[1:0] bits to 01b.
When ADANIM.ANIM[1] = 1, set the ADANSB0[3:2] bits to 01b.
When ADANIM.ANIM[2] = 1, set the ADANSB0[5:4] bits to 01b.
When ADANIM.ANIM[3] = 1, set the ADANSB0[7:6] bits to 01b.

Only set the ADANSB0 register when the ADCSR.ADST bit is 0.

32.2.8 A/D Channel Select Register B1 (ADANSB1)

Address(es): ADC160.ADANSB1 4005 C016h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ANSB24	ANSB23	ANSB22	ANSB21	ANSB20	ANSB19	ANSB18	ANSB17	ANSB16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	ANSB24 to ANSB16	A/D Conversion Channels Select	0: Associated input channel not selected 1: Associated input channel selected. Bit [8] (ANSB24) corresponds to SBIAS/VREFI, bit [7] (ANSB23) corresponds to AN023, and bit [0] (ANSB16) corresponds to AN016.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ANSBn bits (n = 16 to 24) (A/D Conversion Channels Select)

The ADANSB1.ANSBn bits select the analog input channels for A/D conversion from AN016 to AN023 and SBIAS/VREFI 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 ANSB16 bit corresponds to AN016, the ANSB23 bit corresponds to AN023, and the ANSB24 bit corresponds to SBIAS/VREFI.

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 ADCSR.ADST bit is 0.

32.2.9 A/D-Converted Value Average Channel Select Register 0 (ADADS0)

Address(es): ADC160.ADADS0 4005 C008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ADS08	ADS07	ADS06	ADS05	ADS04	ADS03	ADS02	ADS01	ADS00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	ADS08 to ADS00	A/D-Converted Value Average Channel Select	0: Associated input channel not selected 1: Associated input channel selected. Bit [8] (ADS08) corresponds to AN008 and bit [0] (ADS00) corresponds to AN000.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSn bits (n = 00 to 08) (A/D-Converted Value Average Channel Select)

The ADSn bits determine which A/D-converted channel from AN000 to AN008 is subject to A/D-converted value averaging. When an ADSn bit associated with a channel selected for A/D conversion is set to 1, A/D conversion of the analog input of the respective channel is performed successively 1 to 16 times, as specified in the ADC[2:0] bits in the ADADC register.

The average value is stored in the A/D Data Register.

The ADSn bits apply only to channels that are selected for A/D conversion in:

- The ANSAn bits (n = 00 to 08) in the ADANSA0 register or the DBLANS[4:0] bits in the ADCSR register
- The ANSBn bits (n = 00 to 08) in the ADANSB0 register.

For channels on which the A/D conversion is performed and for which average mode is not selected, a normal 1-time conversion is executed and the conversion result is stored in the A/D Data Register.

When the ADANIM.ANIM[n] bit is set to 1 to select differential input, average mode operation based on the setting of even channel associated with ADANIM.ANIM[n] bit is enabled.

Example: When ADANIM.ANIM[0] = 1, set the ADADS0[1:0] bits to 01b.
When ADANIM.ANIM[1] = 1, set the ADADS0[3:2] bits to 01b.
When ADANIM.ANIM[2] = 1, set the ADADS0[5:4] bits to 01b.
When ADANIM.ANIM[3] = 1, set the ADADS0[7:6] bits to 01b.

Only set the ADADS0 register bits when the ADCSR.ADST bit is 0.

32.2.10 A/D-Converted Value Average Channel Select Register 1 (ADADS1)

Address(es): ADC160.ADADS1 4005 C00Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ADS24	ADS23	ADS22	ADS21	ADS20	ADS19	ADS18	ADS17	ADS16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	ADS24 to ADS16	A/D-Converted Value Average Channel Select	0: Associated input channel not selected 1: Associated input channel selected. Bit [8] (ADS24) corresponds to SBIAS/VREFI, bit [7] (ADS23) corresponds to AN023, and bit [0] (ADS16) corresponds to AN016.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSn bits (n = 16 to 24) (A/D-Converted Value Average Channel Select)

The ADSn bits determine which A/D-converted channels from AN016 to AN023 and SBIAS/VREFI pins are subject to A/D-converted value averaging. When an ADSn bit associated with a channel selected for A/D conversion is set to 1, A/D conversion of the analog input of the selected channels is performed successively 1 to 16 times, as specified in the ADC[2:0] bits in the ADADC register.

The average value is stored in the A/D Data Register.

The ADSn bits apply only to channels that are selected for A/D conversion in:

- The ANSAn bits (n = 16 to 24) in the ADANSA1 register or the DBLANS[4:0] bits in the ADCSR register
- The ANSBn bits (n = 16 to 24) in the ADANSB1 register.

For channels on which the A/D conversion is performed and for which 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 32.2 shows a scanning operation sequence in which both the ADADS0.ADS02 and ADADS0.ADS06 bits are set to 1. For this example:

- The number of conversions is set to 4 (ADADC.ADC[2:0] = 011b)
- Channels AN000 to AN007 are selected (ADANSA0.ANSA0[15:0] = 00FFh) in continuous scan mode (ADCSR.ADCS[1:0] = 10b).

The conversion process begins with AN000. The AN002 conversion is performed successively 4 times and the averaged value is returned to A/ D Data Register 2 (ADDR2). Next, the AN003 conversion process is started. The AN006 conversion is performed successively 4 times and the averaged value is returned to A/D Data Register.

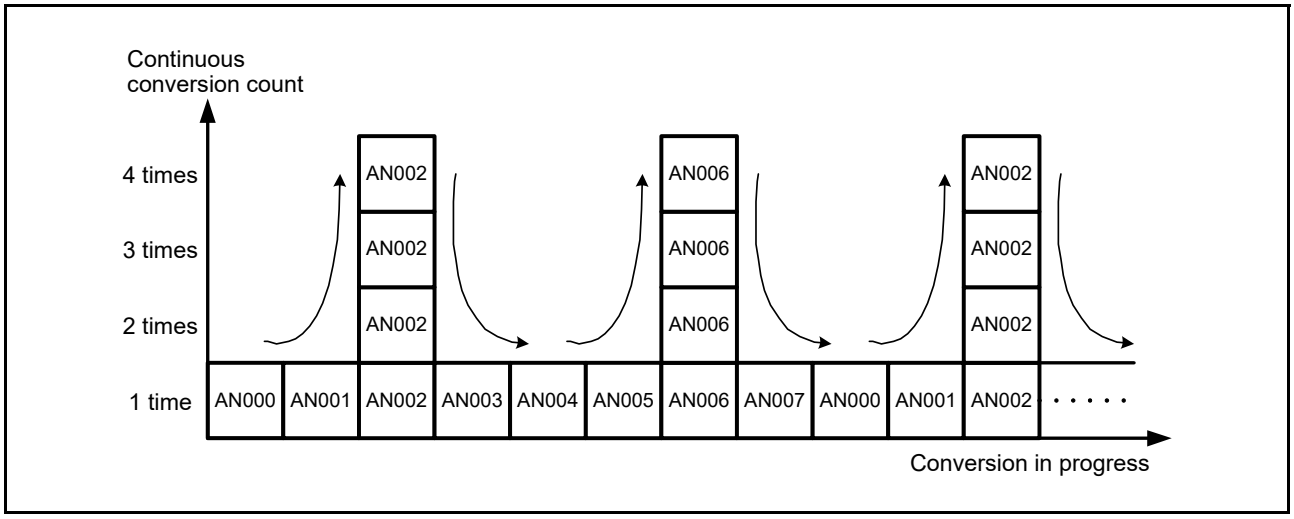


Figure 32.2 Scan conversion sequence with ADADC.ADC[2:0] = 011b, ADADS0.ADS02 = 1, and ADADS0.ADS06 = 1

32.2.11 A/D-Converted Value Average Count Select Register (ADADC)

Address(es): ADC160.ADADC 4005 C00Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	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 (same as normal conversion) 0 0 1: 2-time conversion (acquire the average of 2-time conversion) 0 1 1: 4-time conversion (acquire the average of 4-time conversion) 1 0 0: 8-time conversion (acquire the average of 8-time conversion) 1 0 1: 16-time conversion (acquire the average of 16-time conversion). Other settings are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADC[2:0] bits (Count Select)

The ADC[2:0] bits set the addition count for all channels for which A/D conversion and A/D-converted value 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 reference voltage of SDADC24, the temperature sensor output, and internal reference voltage.

The following restriction applies to the setting of the ADC[2:0] bits:

- When self-diagnosis is executed (ADCER.DIAGM = 1), do not set the ADC[2:0] bits to any value other than 000b.

Only set the ADC[2:0] bits when the ADCSR.ADST bit is 0.

32.2.12 A/D Control Extended Register (ADCER)

Address(es): ADC160.ADCER 4005 C00Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ADINV	—	—	DIAGM	DIAGLD	DIAGVAL[1:0]		—	—	ACE	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	—	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: -VREFH0 (ideal value of A/D conversion result is 8000h) 1 0: VREFL0 (ideal value of A/D conversion result is 0000h) 1 1: VREFH0 (ideal value of A/D conversion result is 7FFFh).	R/W
b10	DIAGLD	Self-Diagnosis Mode Select	0: Rotation mode for self-diagnosis voltage 1: Fixed mode for self-diagnosis voltage.	R/W
b11	DIAGM	Self-Diagnosis Enable	0: Self-diagnosis of ADC16 disabled 1: Self-diagnosis of ADC16 enabled.	R/W
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	ADINV	Single-Ended Input A/D Converted Data Inversion Select	This bit selects the data display range of A/D-converted data of single-ended mode for odd-channels input: 0: Data is stored in a range of -32768 to 0 1: Data is stored in a range of 0 to 32767.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

ACE bit (A/D Data Register Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all 0) of the ADDRy, ADRD, ADRST, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, or ADOCDR register after any of these registers is read by the CPU or DTC. Automatic clearing of the A/D data registers enables detection of failures that are not updated in the A/D data registers. For details, see section 32.3.9, Usage Example of A/D Data Register Automatic Clearing Function.

DIAGVAL[1:0] bits (Self-Diagnosis Conversion Voltage Select)

The DIAGVAL[1:0] bits select the voltage value used in self-diagnosis fixed voltage mode. For details, see the ADCER.DIAGLD bit description.

Do not execute self-diagnosis by setting the ADCER.DIAGLD bit to 1 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 the DIAGLD bit to 0 selects conversion of the voltages in rotation mode where -VREFH0, VREFL0, and VREFH0 are converted, in that order. After reset and when the self-diagnosis voltage rotation mode is selected, self-diagnosis is executed from -VREFH0. The self-diagnosis voltage value does not return to -VREFH0 when scan conversion completes. When scan conversion is restarted, rotation starts at the voltage value following the previous value.

Setting the DIAGLD bit to 1 selects fixed voltage, in which the fixed voltage specified in the ADCER.DIAGVAL[1:0] bits is converted. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value.

Only set the DIAGLD bit when the ADCSR.ADST bit is 0.

DIAGM bit (Self-Diagnosis Enable)

The DIAGM bit enables or disables self-diagnosis.

Self-diagnosis is used to detect a failure of the ADC16. In self-diagnosis mode, one of the three voltage values (-VREFH0, VREFL0, or VREFH0) 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). The ADRD register can be read out 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 group B.

Only set the DIAGM bit when the ADCSR.ADST bit is 0.

ADINV bit (Single-Ended Input A/D Converted Data Inversion Select)

This bit is used to select "-32768 to 0" or "0 to 32767" as the A/D conversion result data display range when storing the A/D conversion result of the odd AN000 to AN007 analog inputs in register ADDRy (y = 1, 3, 5, 7), ADDBLDR, ADDBLDRA, or ADDBLDRB. A/D conversion of the even AN000 to AN007 analog inputs, analog inputs AN016 to AN023, differential input, the reference voltage of SDADC24 (SBIAS/VREFI), temperature sensor, or internal reference voltage is not affected by this bit.

Only set the ADINV bit when the ADCSR.ADST bit is 0.

The following shows the data inversion method according to the ADINV bit setting:

- When A/D conversion result data = 0000h: 0000h
- When A/D conversion result data = 8000h: All bits inverted
- When A/D conversion result data ≠ 8000h, 0000h: All bits inverted + 0001h.

Table 32.7 Single-ended input A/D converted data inversion select

ADINV = 0		ADINV = 1	
Hex.	**Dec.**	**Hex.**	**Dec.**
0000h	0	0000h	0
FFFFh	-1	0001h	1
FFFEh	-2	0002h	2
:	:	:	:
8002h	-32766	7FFEh	32766
8001h	-32767	7FFFh	32767
8000h	-32768	7FFFh	32767

32.2.13 A/D Conversion Start Trigger Select Register (ADSTRGR)

Address(es): ADC160.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 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 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 (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by the trigger might have no effect.

When the GPT module is selected as an A/D conversion start trigger, a delay for synchronization processing occurs. For details, see section 32.3.8, Analog Input Sampling and Scan Conversion Time.

Table 32.8 lists the A/D conversion startup sources selected in the TRSB[5:0] bits.

Table 32.8 Selection of A/D activation 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 a synchronous trigger (ELC), set the TRGE bit in the ADCSR register to 1 and set the EXTRG bit in the ADCSR register to 0
- When using an asynchronous trigger (ADTRG0), do not use a software trigger or an asynchronous trigger
- 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 (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by a trigger might have no effect. For details, see section 32.3.8, Analog Input Sampling and Scan Conversion Time.

Table 32.9 lists the A/D conversion start sources selected in the TRSA[5:0] bits.

Table 32.9 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

32.2.14 A/D Conversion Extended Input Control Register (ADEXICR)

Address(es): ADC160.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 Average Mode Select	0: Temperature sensor output A/D-converted value average mode not selected 1: Temperature sensor output A/D-converted value average mode selected.	R/W
b1	OCSAD	Internal Reference Voltage A/D-Converted Value Average Mode Select	0: Internal reference voltage A/D-converted value average mode not selected 1: Internal reference voltage A/D-converted value average mode selected.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TSSA	Temperature Sensor Output A/D Conversion Select	0: A/D conversion of temperature sensor output disabled 1: A/D conversion of temperature sensor output enabled.	R/W
b9	OCSA	Internal Reference Voltage A/D Conversion Select	0: A/D conversion of internal reference voltage disabled 1: A/D conversion of internal reference voltage enabled.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TSSAD bit (Temperature Sensor Output A/D- Converted Value 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 ADADC. The mean value is returned to the ADTSDR register.

Only set the TSSAD bit when the ADCSR.ADST bit is 0.

OCSAD bit (Internal Reference Voltage A/D- Converted Value 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 in the ADC[2:0] bits in the ADADC register. The mean value is returned to the ADOCDR register.

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, ADANSB0/1 registers, the ADCSR.DBLE and ADESICR.OCSA bits 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 ADC16 executes discharge (15 ADCLK) before executing sampling. The minimum sampling time is 5 µs. The ADC16 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, ADANSB0/1 registers, the ADCSR.DBLE and ADESICR.TSSA bits 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 ADC16 executes discharge (15 ADCLK) before executing sampling. The minimum sampling time is 5 µs. The ADC16 executes discharge each time it executes A/D conversion of the internal reference voltage.

Only set the OCSA bit when the ADCSR.ADST bit is 0.

32.2.15 A/D Sampling State Register n (ADSSTRn) (n = 00 to 08, L, T, O)

Address(es): ADC160.ADSSTR00 4005 C0E0h to ADC160.ADSSTR08 4005 C0E8h, ADC160.ADSSTRL 4005 C0DDh, ADC160.ADSSTRT 4005 C0DEh, ADC160.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 32 MHz, then one state is 31.25 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.

When the ADANIM.ANIM[n] bit is set to 1 to select differential input, select the even channel associated with ANIM[n] bit.

Example: When ADANIM.ANIM[0] = 1, ADSSTR00.SST[7:0] bits are used.
When ADANIM.ANIM[1] = 1, ADSSTR02.SST[7:0] bits are used.
When ADANIM.ANIM[2] = 1, ADSSTR04.SST[7:0] bits are used.
When ADANIM.ANIM[3] = 1, ADSSTR06.SST[7:0] bits are used.

The lower limit of the sampling time setting depends on the frequency ratio:

- If the frequency ratio of PCLKB to PCLKD (ADCLK) = 1:1, the sampling time must be set to a value of more than 5 states
- If the frequency ratio of PCLKB to PCLKD (ADCLK) = 1:2 or 1:4, the sampling time must be set to a value of more than 6 states.

Table 32.10 shows the relationship between the A/D Sampling State Register n and the associated channels. For details, see section 32.3.8, Analog Input Sampling and Scan Conversion Time.

Only set the SST[7:0] bits when the ADCSR.ADST bit is 0.

Table 32.10 Relationship between A/D Sampling State Register n and associated channels

Bit name	Associated channels
ADSSTR00.SST[7:0] bits*1	AN000
ADSSTR01.SST[7:0] bits	AN001
ADSSTR02.SST[7:0] bits	AN002
ADSSTR03.SST[7:0] bits	AN003
ADSSTR04.SST[7:0] bits	AN004
ADSSTR05.SST[7:0] bits	AN005
ADSSTR06.SST[7:0] bits	AN006
ADSSTR07.SST[7:0] bits	AN007
ADSSTR08.SST[7:0] bits	AN008
ADSSTRL.SST[7:0] bits	AN016 to AN023, SBIAS/VREFI
ADSSTRT.SST[7:0] bits	Temperature sensor output*2
ADSSTRO.SST[7:0] bits	Internal reference voltage*2

Note 1. When the self-diagnosis function is selected, the sampling time set in the ADSSTR00.SST[7:0] bits is applied.
Note 2. When the temperature sensor output or the internal reference voltage is converted, set the sampling time to more than 5 μs. Because the maximum SST[7:0] value is 255 states, the ADCLK frequency must be such that the resulting sampling time is 32 MHz or at least 5 μs.

32.2.16 A/D Disconnection Detection Control Register (ADDISCR)

Address(es): ADC160.ADDISCR 4005 C07Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PCHG[1:0]		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
b5, b4	PCHG[1:0]	Precharge/discharge select	b5 PCHG[1]: Precharge/discharge selection: 0: The analog input path 1*1 voltage is discharged 1: The analog input path 1*1 voltage is precharged. b4 PCHG[0]: Precharge/discharge selection: 0: The analog input path 2*2 voltage is discharged 1: The analog input path 2*2 voltage is precharged.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Even analog input channels from AN000 to AN008 or from AN016 to AN023.
Note 2. Odd analog input channels from AN000 to AN007.

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
- Reference voltage of SDADC24 (SBIAS/VREFI)
- Differential input mode.

ADNDIS[3:0] bits (Precharge/discharge period)

The ADNDIS[3:0] bits specify the period of precharge or discharge. When the 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 assist function is enabled.

PCHG[1:0] bits (Precharge/discharge select)

Setting the PCHG[1] bit to 1 selects precharge and setting the PCHG[1] bit to 0 selects discharge. Setting the PCHG[0] bit to 1 selects precharge and setting the PCHG[0] bit to 0 selects discharge. Table 32.11 shows the operation of the disconnection detection assist function.

Execution of precharge or discharge in the disconnection detection assist function is automatically controlled by the PCHG[1:0] setting, and the analog input channel (odd analog input channels from AN000 to AN007 in single-ended mode, even analog input channels from AN000 to AN008 in single-ended mode, or analog input channels from AN016 to AN023).

Table 32.11 Operation of the disconnection detection assist function

PCHG[1:0]	ADNDIS[3:0]	Analog channel input mode	Next A/D conversion	Analog input path 1	Analog input path 2
—	0000b	—	—	No connection	No connection
—	0001b	Function not guaranteed			
11b	1111b to 0010b	Single-ended	Even analog input channels from AN000 to AN008, analog input channels from AN016 to AN023, temperature sensor or internal reference voltage	Precharge	No connection
01b				Discharge	
10b				Precharge	
00b				Discharge	
11b			Odd analog input channels AN000 to AN008	No connection	Precharge
01b					Precharge
10b					Discharge
00b					Discharge

32.2.17 A/D Interrupt Control Register (ADICR)

Address(es): ADC160.ADICR 4005 C07Dh

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

Bit	Symbol	Bit name	Description	R/W
b1, b0	ADIC[1:0]	A/D Interrupt Control	b1 b0 0 0: ADC160_ADI is generated at the end of A/D scan 1 1: ADC160_ADI is generated at the end of calibration. Other settings are prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

32.2.18 A/D Group Scan Priority Control Register (ADGSPCR)

Address(es): ADC160.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
b1	GBRSCN	Group B Restart Setting	Enabled only when PGS = 1, ignored 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, ignored 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 the PGS bit 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*1)

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 32.7.2, Notes on Stopping A/D Conversion. When the PGS bit is set to 1, use the settings as described in section 32.3.6.3, 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 is 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 group A conversion.

When the GBRSCN bit is set to 0, triggers input during A/D conversion are ignored. Only set the GBRSCN bit when 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*2)

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 automatically restarts 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 of the GBRSCN bit. Only set the GBRP bit while the ADCSR.ADST is 0.

The setting of the GBRP bit is valid when the PGS bit is 1.

32.2.19 A/D Compare Function Control Register (ADCMPCR)

Address(es): ADC160.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 ADC160_WCMPM when window A OR window B comparison conditions are met. Otherwise, output ADC160_WCMPUM. 0 1: Output ADC160_WCMPM when window A EXOR window B comparison conditions are met. Otherwise, output ADC160_WCMPUM 1 0: Output ADC160_WCMPM when window A AND window B comparison conditions are met. Otherwise, output ADC160_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 ADC160_WCMPM and ADC160_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 ADC160_WCMPM and ADC160_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: ADC160_CMPBI interrupt disabled when comparison conditions (window B) are met 1: ADC160_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

Bit	Symbol	Bit name	Description	R/W
b15	CMPAIE	Compare A Interrupt Enable	0: ADC160_CMPAI interrupt disabled when comparison conditions (window A) are met 1: ADC160_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 bit. 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 the CMPBE 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 A/D Compare Function 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)
- A/D Compare Function Window A Channel Select Registers 0/1 (ADCMPANSR0, ADCMPANSR1)
- A/D Compare Function Window A Extended Input Select Register (ADCMPANSER).

CMPBIE bit (Compare B Interrupt Enable)

The CMPBIE bit enables or disables the ADC160_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 ADC160_CMPAI interrupt output when the comparison conditions (window A) are met.

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

Address(es): ADC160.ADCMPANSR0 4005 C094h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPCHA08	CMPCHA07	CMPCHA06	CMPCHA05	CMPCHA04	CMPCHA03	CMPCHA02	CMPCHA01	CMPCHA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	CMPCHA08 to CMPCHA00	Compare Window A Channel Select	0: Compare function disabled for the associated input channel 1: Compare function enabled for the associated input channel. Bit [8] (CHPCHA08) corresponds to AN008 and bit [0] (CMPCHA00) corresponds to AN000.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPCHAn bits (n = 00 to 08) (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 = 00 to 08) and the ADANSB0.ANSBn bits (n = 00 to 08).

When the ADANIM.ANIM[n] bit is set to 1 to select differential input, select the even channel associated with ANIM[n] bit.

Example: When ADANIM.ANIM[0] = 1, set the ADCMPANSR0[1:0] bits to 01b.
When ADANIM.ANIM[1] = 1, set the ADCMPANSR0[3:2] bits to 01b.
When ADANIM.ANIM[2] = 1, set the ADCMPANSR0[5:4] bits to 01b.
When ADANIM.ANIM[3] = 1, set the ADCMPANSR0[7:6] bits to 01b.

Only set the CMPCHAn bits when the ADCSR.ADST bit is 0.

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

Address(es): ADC160.ADCMPANSR1 4005 C096h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPCHA24	CMPCHA23	CMPCHA22	CMPCHA21	CMPCHA20	CMPCHA19	CMPCHA18	CMPCHA17	CMPCHA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	CMPCHA24 to CMPCHA16	Compare Window A Channel Select	0: Compare function disabled for the associated input channel 1: Compare function enabled for the associated input channel. Bit [8] (CMPCHA24) corresponds to SBIAS/VREFI, bit [7] (CMPCHA23) corresponds to AN023, and bit [0] (CMPCHA16) corresponds to AN016.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPCHAn bits (n = 16 to 24) (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 = 16 to 24) and the ADANSB1.ANSBn bits (n = 16 to 24).

Only set the CMPCHAn bits when the ADCSR.ADST bit is 0.

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

Address(es): ADC160.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.

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

Address(es): ADC160.ADCMPLR0 4005 C098h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPLCHA08	CMPLCHA07	CMPLCHA06	CMPLCHA05	CMPLCHA04	CMPLCHA03	CMPLCHA02	CMPLCHA01	CMPLCHA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	CMPLCHA08 to CMPLCHA00	Compare Window A Comparison Condition Select	These bits set comparison conditions for channels AN000 to AN008 to which window A comparison conditions are applied. Comparison conditions are shown in Figure 32.3. • 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
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPLCHAn bits (n = 00 to 08) (Compare Window A Comparison Condition Select)

The CMPLCHAn bits specify the comparison conditions for channels AN000 to AN008 to which window A comparison conditions are applied. These bits can be set for each analog input to be compared. CMPLCHA00, CMPLCHA04, and

CMPLCHA08 correspond to AN000, AN004, and AN008, 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 (ADC160_CMPAI) is generated.

When the ADANIM.ANIM[n] bit is set to 1 to select differential input, select the even channel associated with ANIM[n] bit.

Example: When ADANIM.ANIM[0] = 1, set the ADCMPLR0[1:0] bits to 01b.
When ADANIM.ANIM[1] = 1, set the ADCMPLR0[3:2] bits to 01b.
When ADANIM.ANIM[2] = 1, set the ADCMPLR0[5:4] bits to 01b.
When ADANIM.ANIM[3] = 1, set the ADCMPLR0[7:6] bits to 01b.

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 ≤ ADCMPDR1value	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 32.3 Comparison conditions for compare function window A

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

Address(es): ADC160.ADCMPLR1 4005 C09Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPLCHA24	CMPLCHA23	CMPLCHA22	CMPLCHA21	CMPLCHA20	CMPLCHA19	CMPLCHA18	CMPLCHA17	CMPLCHA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	CMPLCHA24 to CMPLCHA16	Compare Window A Comparison Condition Select	These bits set comparison conditions for channels AN016 to AN023 and SBIAS/VREFI to which window A comparison conditions are applied. Comparison conditions are shown in Figure 32.3. • 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
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPLCHAn bits (n = 16 to 24) (Compare Window A Comparison Condition Select)

The CMPLCHAn bits specify the comparison conditions for channels AN016 to AN023 and SBIAS/VREFI to which window A comparison conditions are applied. These bits can be set for each analog input to be compared. CMPLCHA16, CMPLCHA23, and CMPLCHA24 correspond to AN016, AN023, and SBIAS/VREFI, 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 (ADC160_CMPAI) is generated.

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

Address(es): ADC160.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 32.3. • When the window A function is disabled (ADCMPCR.WCMPE = 0): 0: ADCMPDR0 value > A/D-converted value 1: ADCMPDR0 value < A/D-converted value. • When the window A function is enabled (ADCMPCR.WCMPE = 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

Bit	Symbol	Bit name	Description	R/W
b1	CMPLOCA	Compare Window A Internal Reference Voltage Comparison Condition Select	Comparison conditions are shown in Figure 32.3. • When the window A function is disabled (ADCMPCR.WCMPE = 0): 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value. • When the window A function is enabled (ADCMPCR.WCMPE = 1): 0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPLTSA bit (Compare Window A Temperature Sensor Output Comparison Condition Select)

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 (ADC160_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 (ADC160_CMPAI) is generated.

32.2.26 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): ADC160.ADCMPDR0 4005 C09Ch, ADC160.ADCMPDR1 4005 C09Eh, ADC160.ADWINLLB 4005 C0A8h, ADC160.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 ADCMPDR0/1 register specifies the reference data when the compare window A function is used. ADCMPDR0 register sets the lower reference for window A, and ADCMPDR1 register 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 register sets the lower reference for window B, and ADWINULB register sets the upper reference for window B.

ADCMPDR0/1, ADWINULB, and ADWINLLB are read/write registers.

ADCMPDR0/1, 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 references 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 32.4. 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.

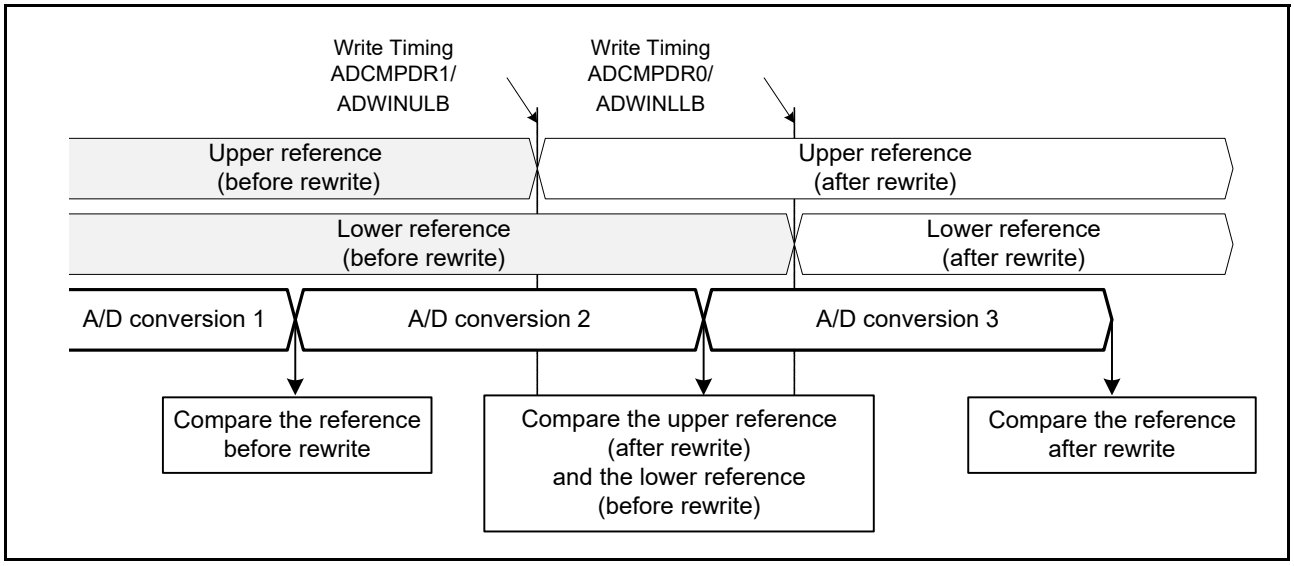


Figure 32.4 Comparison between upper reference and lower reference before and after a rewrite

The ADCMPDR0/1, ADWINLLB, and ADWINULB registers use different formats depending on the following conditions:

- The value of the A/D-Converted Value Average Channel Select bits (A/D-converted value average mode selected or not selected)
- Set ADCMPDR0, ADCMPDR1, ADWINLLB, and ADWINULB in the two's complement format. Values within a range of 8000h to 7FFFh (-32768 to 32767) can be set. However, the data output range varies with analog inputs to be A/D-converted. Therefore, set these registers according to the output range shown in Table 32.13.

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

Address(es): ADC160.ADCMPSR0 4005 C0A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPSTCHA08	CMPSTCHA07	CMPSTCHA06	CMPSTCHA05	CMPSTCHA04	CMPSTCHA03	CMPSTCHA02	CMPSTCHA01	CMPSTCHA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	CMPSTCHA08 to CMPSTCHA00	Compare Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE =1), these bits indicate the comparison result of channels AN000 to AN008 to which window A comparison conditions are applied: 0: Comparison conditions not met 1: Comparison conditions met.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPSTCHAn flags (n = 00 to 08) (Compare Window A Flag)

CMPSTCHAn are comparison result status flags of channels AN000 to AN008 to which window A comparison conditions are applied. When the comparison condition set by ADCMPLR0.CMPLCHAn is met at the end of A/D conversion, the associated flag is set to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt (ADC160_CMPAI) request is generated when this bit is set to 1. CMPSTCHA00, CMPSTCHA04, and CMPSTCHA08 correspond to AN000, AN004, and AN008, respectively.

Writing 1 to the CMPSTCHAn flags is invalid.

[Setting condition]

- The condition set in ADCMPLR0.CMPLCHAn is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

When the ADANIM.ANIM[n] bit is set to 1 to select differential input, select the even channel associated with ANIM[n] bit.

Example: When ADANIM.ANIM[0] = 1, set the ADCMPSR0[1:0] bits to 01b.
When ADANIM.ANIM[1] = 1, set the ADCMPSR0[3:2] bits to 01b.
When ADANIM.ANIM[2] = 1, set the ADCMPSR0[5:4] bits to 01b.
When ADANIM.ANIM[3] = 1, set the ADCMPSR0[7:6] bits to 01b.

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

Address(es): ADC160.ADCMPSR1 4005 C0A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPSTCHA24	CMPSTCHA23	CMPSTCHA22	CMPSTCHA21	CMPSTCHA20	CMPSTCHA19	CMPSTCHA18	CMPSTCHA17	CMPSTCHA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	CMPSTCHA24 to CMPSTCHA16	Compare Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1), these bits indicate the comparison result of channels AN016 to AN023 and SBIAS/VREFI to which window A comparison conditions are applied. 0: Comparison conditions not met 1: Comparison conditions met.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPSTCHAn flags (n = 16 to 24) (Compare Window A Flag)

CMPSTCHAn are comparison result status flags of channels AN016 to AN023 and SBIAS/VREFI 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 flag is set to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt (ADC160_CMPAI) request is generated when CMPSTCHAn is set to 1. CMPSTCHA16, CMPSTCHA23, and CMPSTCHA24 correspond to AN016, AN023, and SBIAS/VREFI, respectively.

Writing 1 to the CMPSTCHAn flags is invalid.

[Setting condition]

- The condition set in ADCMPLR1.CMPLCHAn is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

32.2.29 A/D Compare Function Window A Extended Input Channel Status Register (ADCMPSER)

Address(es): ADC160.ADCMPSER 4005 C0A4h

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

Bit	Symbol	Bit name	Description	R/W
b0	CMPSTTSA	Compare Window A Temperature Sensor Output Compare Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1), this bit indicates the temperature sensor output comparison result. 0: Comparison conditions 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 flag (Compare Window A Temperature Sensor Output Compare Flag)

CMPSTTSA 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 flag is set to 1. When ADCMPCR.CMPAIE is 1, a compare interrupt (ADC160_CMPAI) request is generated when this flag is set to 1.

Writing 1 to the CMPSTTSA flag is invalid.

[Setting condition]

- The condition set in ADCMPLER.CMPLTSA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

CMPSTOCA flag (Compare Window A Internal Reference Voltage Compare Flag)

CMPSTOCA is a status flag that indicates the internal reference voltage comparison result. When the comparison condition set by ADCMPLER.CMPLOCA is met at the end of A/D conversion, this flag is set to 1. When ADCMPCR.CMPAIE is 1, a compare interrupt (ADC160_CMPAI) request is generated when this flag is set to 1.

Writing 1 to the CMPSTOCA flag is disabled.

[Setting condition]

- The condition set in ADCMPLER.CMPLOCA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

32.2.30 A/D Compare Function Window B Channel Select Register (ADCMPBNSR)

Address(es): ADC160.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. The maximum channel is AN023. b5 b0 0 0 0 0 0 0: AN000 0 0 0 0 0 1: AN001 0 0 0 0 1 0: AN002 : 0 0 1 0 0 0: AN008 0 1 0 0 0 0: AN016 : : 0 1 0 1 0 1: AN021 0 1 0 1 1 0: AN022 0 1 0 1 1 1: AN023 0 1 1 0 0 0: Reference voltage of SDADC24 (SBIAS/VREFI) 1 0 0 0 0 0: Temperature sensor 1 0 0 0 0 1: Internal reference voltage 1 1 1 1 1 1: Not select. Other settings are prohibited.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	CMPLB	Compare Window B Comparison Condition Setting	This bit sets comparison conditions of channels for window B. The comparison conditions are shown in Figure 32.5. • 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 from AN000 to AN008, AN016 to AN023, the reference voltage of SDADC24 (SBIAS/VREFI), 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 = 00 to 08)
- ADANSA1.ANSAn bits (n = 16 to 24)
- ADANSB0.ANSBn bits (n = 00 to 08)
- ADANSB1.ANSBn bits (n = 16 to 24).

When the ADANIM.ANIM[n] bit is set to 1 to select differential input, select the even channel associated with ANIM[n] bit.

Example: When ADANIM.ANIM[0] = 1, set the ADCMPBNSR.CMPCHB[5:0] bits to 000000b.
When ADANIM.ANIM[1] = 1, set the ADCMPBNSR.CMPCHB[5:0] bits to 000010b.
When ADANIM.ANIM[2] = 1, set the ADCMPBNSR.CMPCHB[5:0] bits to 000100b.
When ADANIM.ANIM[3] = 1, set the ADCMPBNSR.CMPCHB[5:0] bits to 000110b.

Set CMPCHB[5:0] bits when 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 ADCMPBSR0.CMPSTB bit is set to 1 and a compare interrupt ADC160_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 32.5 Compare conditions for compare function window B

32.2.31 A/D Compare Function Window B Status Register (ADCMPBSR)

Address(es): ADC160.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 AN000 to AN008, AN016 to AN023, reference voltage of SDADC24 (SBIAS/VREFI), 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 flag (Compare Window B Flag)

CMPSTB is a status flag that indicates the comparison result for channels AN000 to AN008, AN016 to AN023, the reference voltage of SDADC24 (SBIAS/VREFI), 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 flag is set to 1. When ADCMPCR.CMPBIE is 1, a compare interrupt ADC160_CMPBI request is generated when this flag is set to 1.

Writing 1 to the CMPSTB flag is invalid.

[Setting condition]

- The condition set in ADCMPBNSR.CMPLB is met when ADCMPCR.CMPBE = 1.

[Clearing condition]

- Writing 0 after reading 1.

32.2.32 A/D Compare Function Window A/B Status Monitor Register (ADWINMON)

Address(es): ADC160.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 the ADCMPLR0/ADCMPLR1 and ADCMPLER registers. Otherwise, it is read as 0.

[Setting condition]

- The A/D-converted value meets the condition set in the ADCMPLR0/ADCMPLR1 and ADCMPLER registers when ADCMPCR.CMPAE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set in the ADCMPLR0/ADCMPLR1 and ADCMPLER registers when ADCMPCR.CMPAE = 1
- ADCMPCR.CMPAE = 0 (automatically cleared when the ADCMPCR.CMPAE value changes from 1 to 0).

MONCMPB bit (Comparison Result Monitor B)

The read-only MONCMPB bit is read as 1 when the A/D converted value of the window B target channel meets the condition set in the ADCMPBNSR.CMPLB bit. Otherwise, it is read as 0.

[Setting condition]

- The A/D-converted value meets the condition set in ADCMPBNSR.CMPLB when ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set in ADCMPBNSR.CMPLB when ADCMPCR.CMPBE = 1
- ADCMPCR.CMPBE = 0 (automatically cleared when the ADCMPCR.CMPBE value changes from 1 to 0).

32.2.33 A/D Dedicated Reference Voltage Circuit Control Register (VREFAMPCNT)

Address(es): ADC160.VREFAMPCNT 4005 C0F4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	BGREN	VREFADCEN	VREFADCG[1:0]		OLDETEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	OLDETEN	OLDET Enable	0: Over current detection disabled 1: Over current detection enabled.	R/W
b2, b1	VREFADCG[1:0]	VREFADC Output Voltage Control	Bits for selecting the internal reference voltage for the A/D converter. b2 b1 0 x: 1.5 V 1 0: 2.0 V 1 1: 2.5 V.	R/W

Bit	Symbol	Bit name	Description	R/W
b3	VREFADCEN	VREFADCG Enable	0: VREFADC output is disabled and VREFH0 is selected as the high-potential reference voltage 1: VREFADC output is enabled and VREFADC is selected as the high-potential reference voltage.	R/W
b4	BGREN	BGR Enable	0: Turn off the BGR power 1: Turn on the BGR power.	R/W
b7, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

The VREFADC output voltage is output to the VREFH0 pin. When using VREFADC, do not input voltage to VREFH0. To stabilize the VREFADC output voltage, connect the VREFH0 pin to VREFL0 pin through a capacitor (1µF).

OLDETEN bit (OLDET Enable)

The OLDETEN bit enables or disables over current detection.

VREFADC output is controlled by combinations of VREFADCG[1:0], VREFADCEN, and BGREN bits as shown in Table 32.12.

Table 32.12 VREFADC output voltage control list

MODE	BGREN	VREFADCEN	VREFADCG[1]	VREFADCG[0]	OLDETEN	VREFADC Output voltage
After reset	0	0	0	0	0	Hi-Z
BGR only	1	0	x	x	x	Hi-Z
VREFADC 1.5 V output	1	1	0	x	0 or 1	1.5 V
VREFADC 2.0 V output	1	1	1	0	0 or 1	2.0 V
VREFADC 2.5 V output	1	1	1	1	0 or 1	2.5 V

x: Don't care

32.2.34 A/D Channel Input Mode Select Register (ADANIM)

Address(es): ADC160.ADANIM 4005 C0F0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	ANIM[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	ANIM[3:0]	Analog Channel Input Mode Select	Bits for selecting the analog input A/D conversion mode: 0: Single-ended mode 1: Differential mode.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ANIM[3:0] bits (Analog Channel Input Mode Select)

The ANIM[3:0] bits select single-ended mode or differential mode for A/D conversion of AN000 to AN007.

ANIM[0] bit corresponds to AN000 and AN001, ANIM[1] bit corresponds to AN002 and AN003, and ANIM[3] bit corresponds to AN006 and AN007.

When differential mode is used, only the following combinations of analog channels are available:

AN000 and AN001, AN002 and AN003, AN004 and AN005, AN006 and AN007.

After A/D conversion is made in differential mode, the A/D conversion result is stored in an even-numbered A/D data

register ADDRy (y = 0 to 7).

Example: When A/D conversion is made with ANIM[0] = 1, the A/D conversion result is stored in ADDR0 register.
When A/D conversion is made with ANIM[1] = 1, the A/D conversion result is stored in ADDR2 register.
When A/D conversion is made with ANIM[2] = 1, the A/D conversion result is stored in ADDR4 register.
When A/D conversion is made with ANIM[3] = 1, the A/D conversion result is stored in ADDR6 register.

Set the ANIM[3:0] bits when the ADCSR.ADST bit is 0.

32.2.35 A/D Calibration Execution Register (ADCALEXE)

Address(es): ADC160.ADCALEXE 4005 C0F2h

	b7	b6	b5	b4	b3	b2	b1	b0
	CALEXE	CALMON	—	—	—	—	—	—
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	CALMON	Calibration Status Flag	0: Calibration not in progress 1: Calibration in progress.	R
b7	CALEXE	Calibration Start	0: Calibration not started 1: Calibration started.	R/W

CALMON flag (Calibration Status Flag)

When the CALEXE bit is set to 1 and calibration starts, this flag is set to 1. When calibration ends (normal end or writing 0 to the ADCSR.ADST bit), this flag is set to 0.

This flag indicates the execution status of the calibration.

[Setting condition]

- When the calibration starts.

[Clearing condition]

- When the calibration ends.

CALEXE bit (Calibration Start)

When 1 is written to the CALEXE bit, calibration starts. When starting calibration, the following conditions must be satisfied:

- All trigger inputs are disabled (ADCSR.TRGE = 0)
- All scan groups are stopped (ADCSR.ADST = 0).

The ADCSR.ADST and ADCALEXE.CALMON bits are 1 during calibration.

If 1 is written to CALEXE bit while A/D conversion is in progress, calibration start is disabled.

When the ADICR.ADIC[1:0] bits are set to 11b, an ADC160_ADI interrupt request is generated at the end of calibration.

Calibration end can be confirmed by any of the following methods:

- Read the ADCSR.ADST bit to confirm that calibration has ended. The value 0 indicates that calibration has ended.
- Read the ADCALEXE.CALMON bit to confirm that calibration has ended. The value 0 indicates that calibration has ended.
- Check that an ADC160_ADI interrupt request is generated.

32.3 Operation

32.3.1 Selection of Analog Input Channels

Analog input channel 0 (AN000) to 7 (AN007) are used to select single-ended input or differential input. Analog input channel 8 (AN008) and analog input channel 16 (AN016) to 23 (AN023) are used only for single-ended inputs.

Up to 17 channels can be used for single-ended inputs and up to 4 channels of analog input channels 0 to 7 can be used for differential inputs.

Selection of single-ended input or differential input is set in the A/D Channel Input Mode Select Register (ADANIM).

For A/D conversion of single-ended inputs, set the ANIM[n] (n = 0 to 3) bits in the A/D Channel Input Mode Select Register (ADANIM) to 0.

For A/D conversion of differential inputs, set the ANIM[n] (n = 0 to 3) bits in the A/D Channel Input Mode Select Register (ADANIM) to 1 and set the bit for selecting the even channel associated with ANIM[n] to 1.

Example: When ADANIM.ANIM[0] = 1, set the ADANSA0[1:0] bits to 01b.
When ADANIM.ANIM[1] = 1, set the ADANSA0[3:2] bits to 01b.
When ADANIM.ANIM[2] = 1, set the ADANSA0[5:4] bits to 01b.
When ADANIM.ANIM[3] = 1, set the ADANSA0[7:6] bits to 01b.

32.3.2 Results of A/D Conversion

A/D conversion results are output in two's complement format. The A/D conversion result output range varies with A/D converted analog inputs.

Table 32.13 shows the A/D conversion result output ranges of each A/D conversion.

Set the compare window setting registers (ADCMPDR0, ADCMPDR1, ADWINULB, and ADWINLLB) according to the output range shown in Table 32.13.

Table 32.13 A/D conversion result output ranges of each A/D conversion

A/D conversion	Input mode	ADCER.ADINV (data inversion)	Output range (hexadecimal)	Output range (decimal)
Temperature sensor	Single-ended	x	0000h to 7FFFh	0 to 32767
Internal reference voltage	Single-ended	x	0000h to 7FFFh	0 to 32767
Self-diagnosis	-	x	8000h to 7FFFh	-32768 to 32767
AN000, AN002, AN004, AN006	Single-ended	x	0000h*1 to 7FFFh	0*1 to 32767
	Differential	x	8000h to 7FFFh	-32768 to 32767
AN001, AN003, AN005, AN007	Single-ended	1	0000h*1 to 7FFFh	0*1 to 32767
		0	8000h to 0000h*2	-32768 to 0*2
	Differential	x	8000h to 7FFFh	-32768 to 32767
AN008, AN016 to AN023, SBIAS/VREFI	Single-ended	x	0000h to 7FFFh	0 to 32767

x: Don't care

Note: A/D conversion result of odd channels AN000 to AN007 in single-ended mode can be inverted according to the ADCER.ADINV bit setting value. Therefore, the A/D conversion results can be stored in the A/D data registers in the same output range as even channels AN000 to AN008 or AN016 to AN023.

Note 1. If the analog input voltage is lower than VREFL0 by swing of the input voltage, the data output from the ADC is a negative value.

Note 2. If the analog input voltage is lower than VREFL0 by swing of the input voltage, the data output from the ADC is a positive value.

The relationship between A/D conversion result range and compare window setting range is shown with the analog inputs in Figure 32.6 to Figure 32.8.

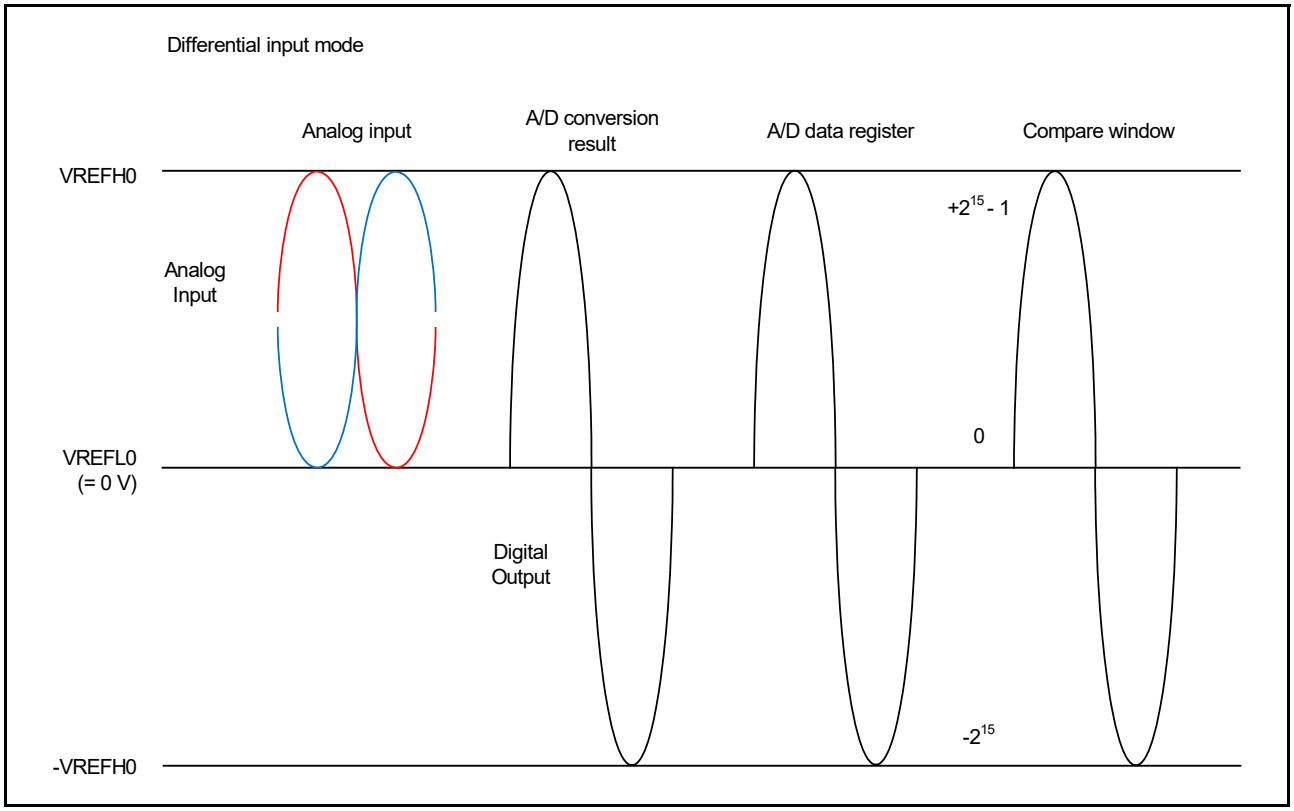


Figure 32.6 Relationship between A/D output range and compare window setting range in differential input mode

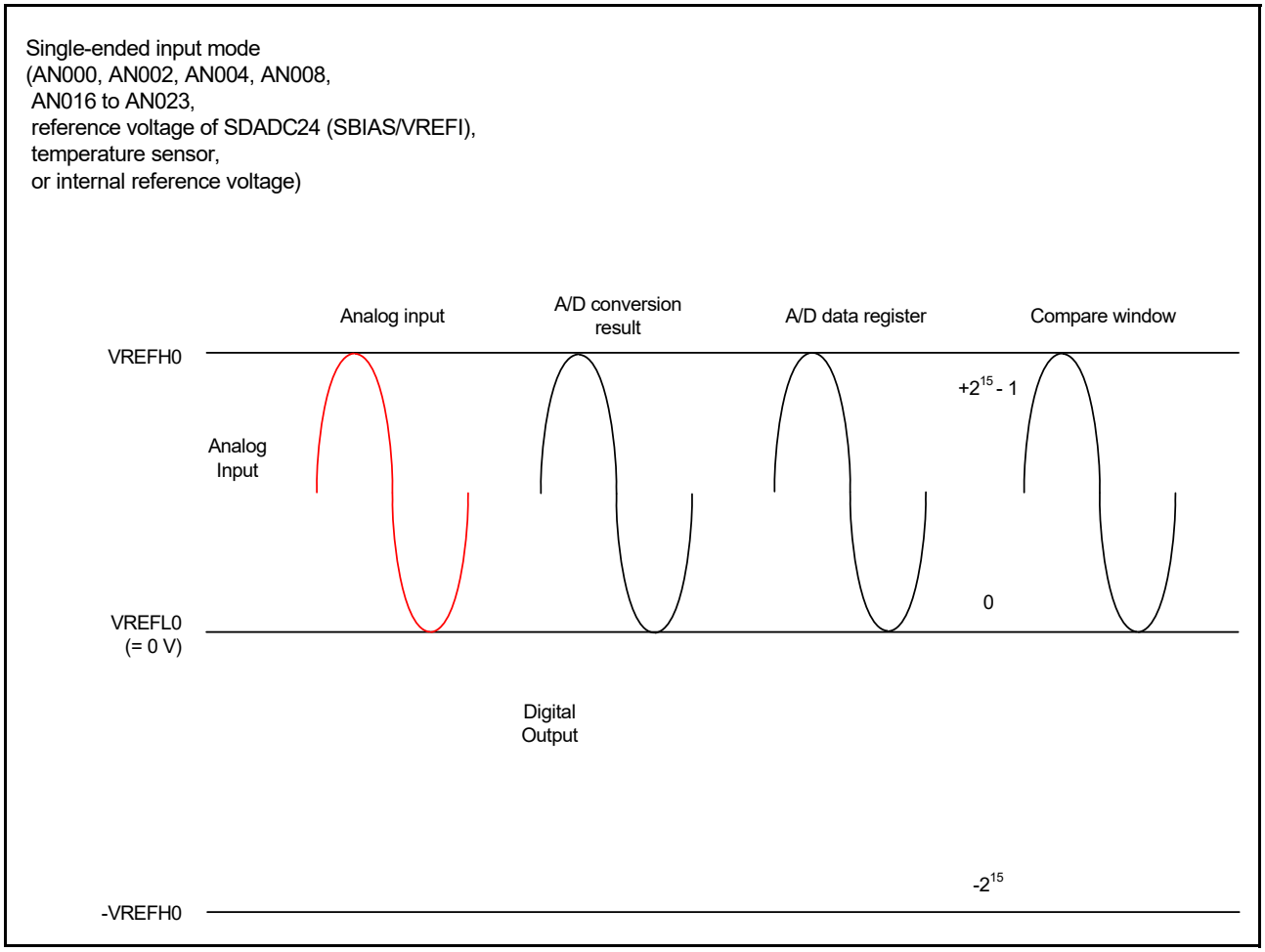


Figure 32.7 Relationship between A/D output range and compare window setting range in single-ended input mode (excluding AN001, AN003, AN005, and AN007)

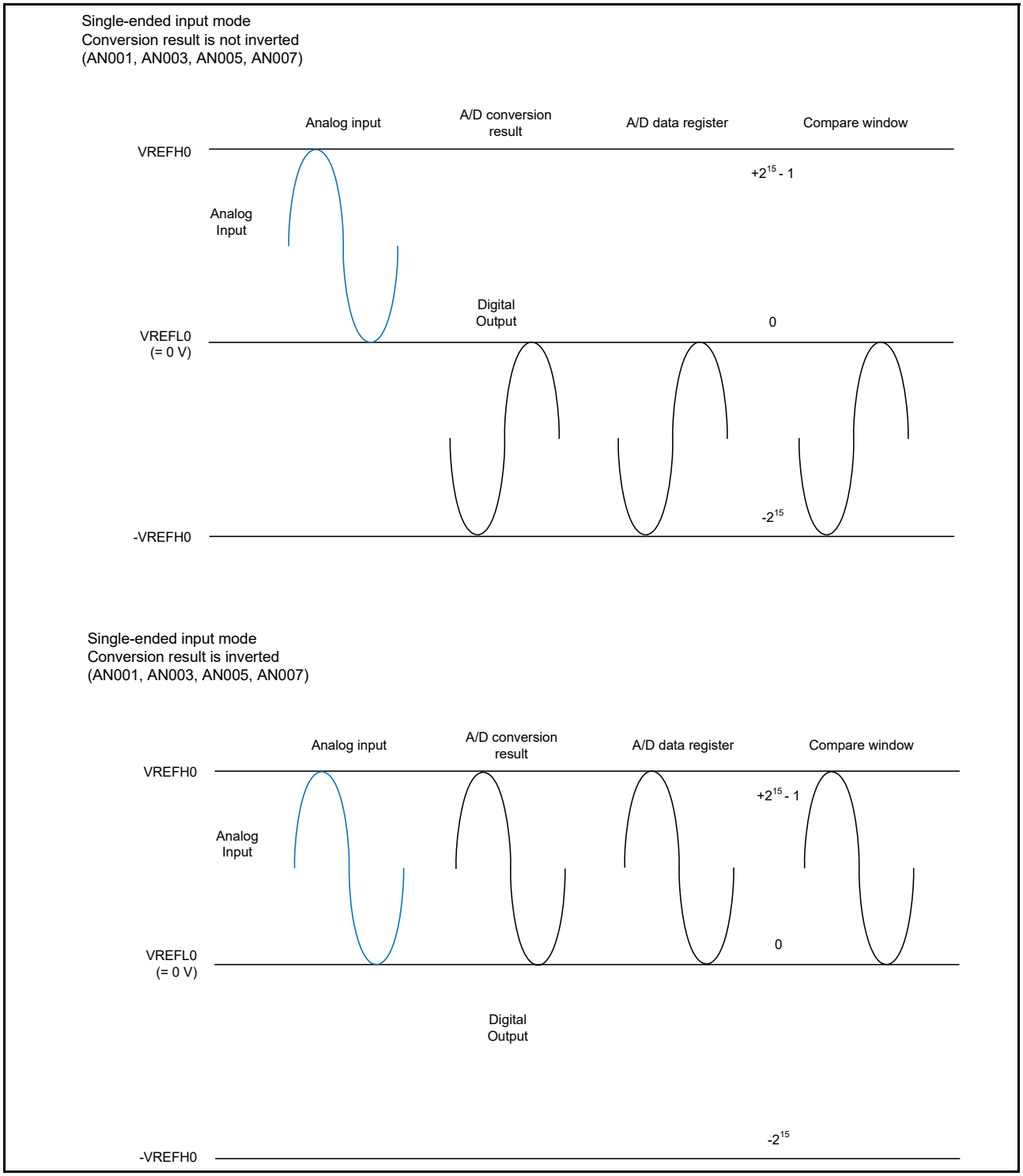


Figure 32.8 Relationship between A/D output range and compare window setting range in single-ended input mode (AN001, AN003, AN005, and AN007)

32.3.3 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.

A scan conversion is performed in any of the three operating modes:

- Single scan mode
- Continuous scan mode
- Group scan 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 ADST bit in ADCSR register to 0. In group scan mode, the selected channels in group A and the selected channels in 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 reference voltages 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 for A/D conversion, single scan mode should be used.

Double trigger mode can be used with single scan mode or group scan mode. With double trigger mode enabled (ADCSR.DBLE bit is 1), A/D conversion data of a channel selected in the DBLANS[4:0] bits in ADCSR register is duplicated only if the conversion is started by the synchronous trigger (ELC) selected in the TRSA[5:0] bits in ADSTRGR register. In group scan mode, only group A can use double trigger mode.

In the extended operation of double trigger mode, the A/D conversion operation is generated from the synchronous trigger combination selected in the ADSTRGR.TRSA[5:0] bits. In addition to normal double trigger mode operation, A/D conversion data with odd number trigger (ELC_AD00) is stored in A/D Data Duplexing Register A (ADDBLDRA), and A/D conversion data with even number trigger (ELC_AD01) is stored in A/D Data Duplexing Register B (ADDBLDRB). In the extended operation of double trigger mode, when 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 in A/D Data Duplexing Register B (ADDBLDRB). The ADC16 ignores a synchronous trigger that occurs during the A/D conversion started by another synchronous trigger.

Calibration allows high-precision A/D conversion by obtaining the C-DAC linearity error correction value and gain (offset) error correction value under usage conditions. Before starting calibration, all scans must be stopped. Do not input any
A/D conversion start trigger during calibration (ADCSR.TRGE = 0). Also, do not start calibration during A/D conversion. Calibration stops when calibration has finished or the ADCSR.ADST bit is cleared to 0 (from 1) by software.

32.3.4 Single Scan Mode

32.3.4.1 Basic operation

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as follows:

1. When the ADST bit in ADCSR register 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. Each time A/D conversion and the gain correction complete in single-ended mode, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
   Each time A/D conversion and the gain correction completes in differential input mode, A/D conversion result is stored in the associated even A/D Data Register y (ADDRy). The same storing operation is also performed in continuous scan mode and group scan mode.
3. When A/D conversion and the gain correction of all the selected channels completes, an ADC160_ADI interrupt request is generated.
4. The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically set to 0 when A/D conversion and the gain correction of all the selected channels completes. The ADC16 then enters a wait state.

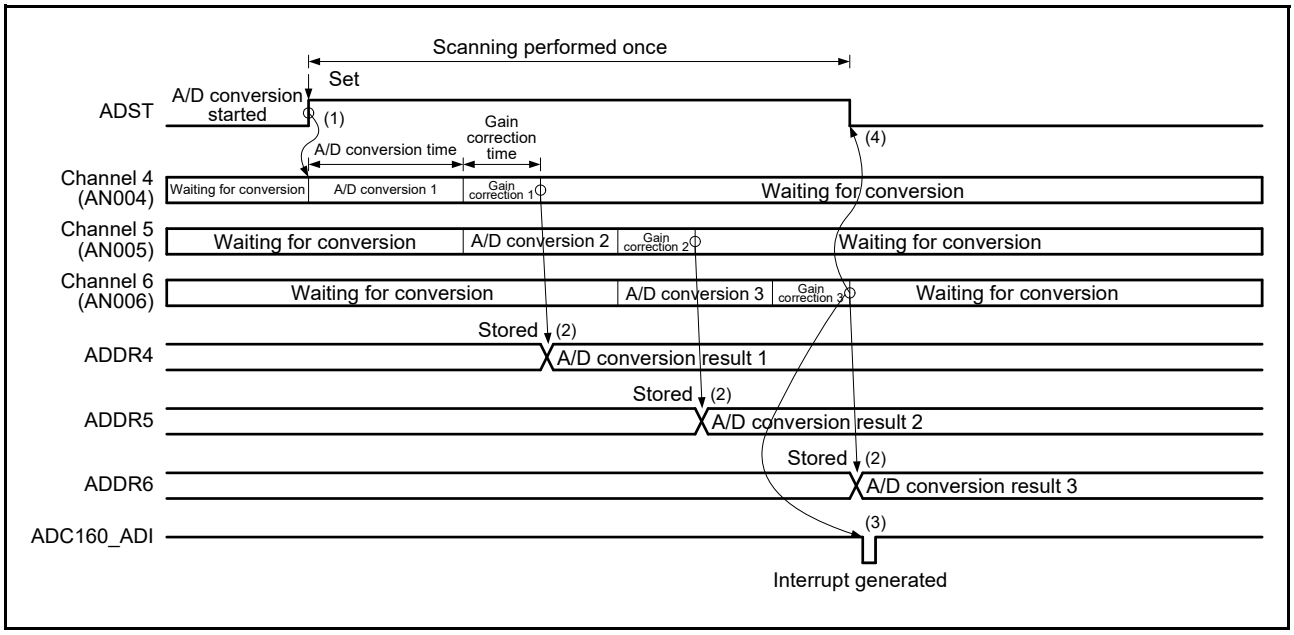


Figure 32.9 Example of single scan mode operation when AN004 to AN006 are selected (AN004 to AN006: single-ended mode)

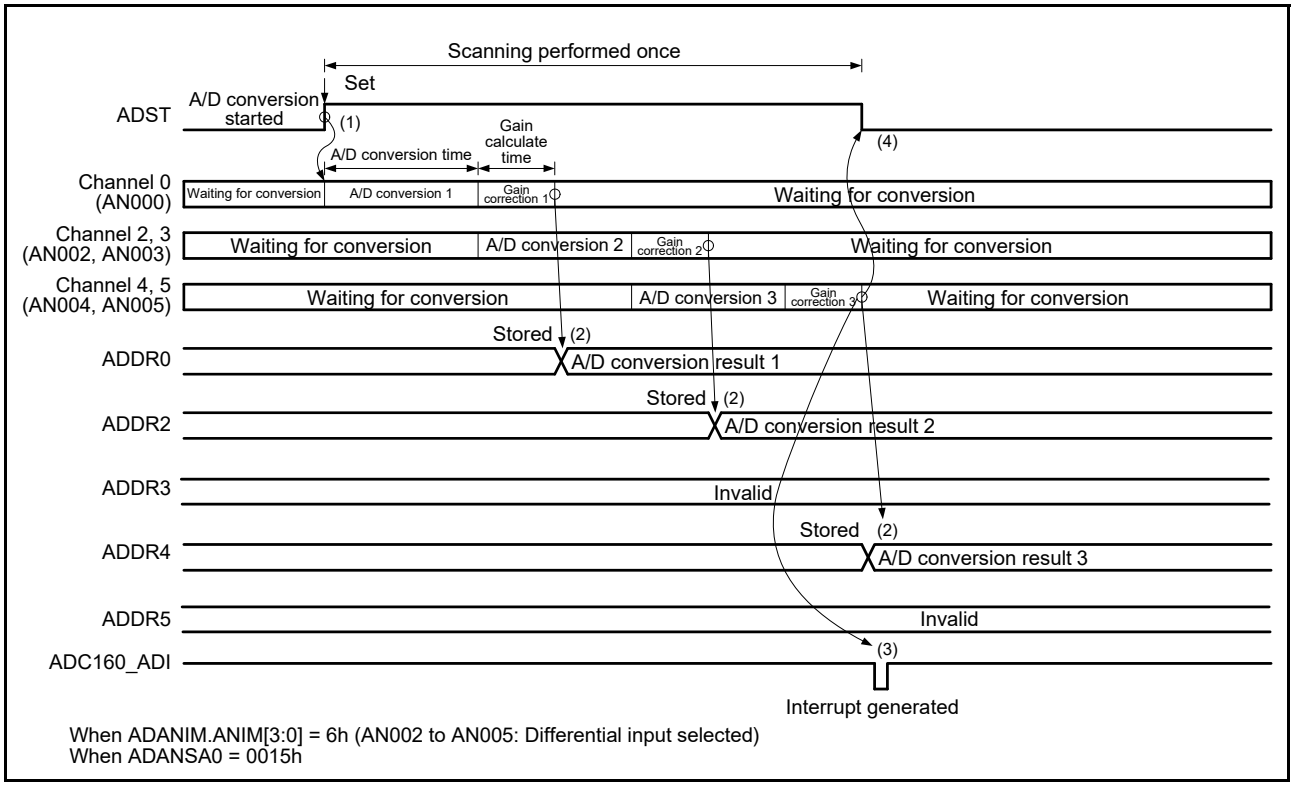


Figure 32.10 Example of single scan mode operation when AN000, AN002 to AN005 are selected (AN000: single-ended mode, AN002 to AN005: differential input mode)

32.3.4.2 Channel selection and self-diagnosis

When channels and self-diagnosis are selected, A/D conversion is first performed on the reference voltage (-VREFH0, VREFL0, or VREFH0) supplied to the ADC16. 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 ADST bit in ADCSR register 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 and the gain correction 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 and the gain correction of a single channel complete, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
4. When A/D conversion and the gain correction of all the selected channels completes, an ADC160_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 and the gain correction of all the selected channels completes. The ADC16 then enters a wait state.

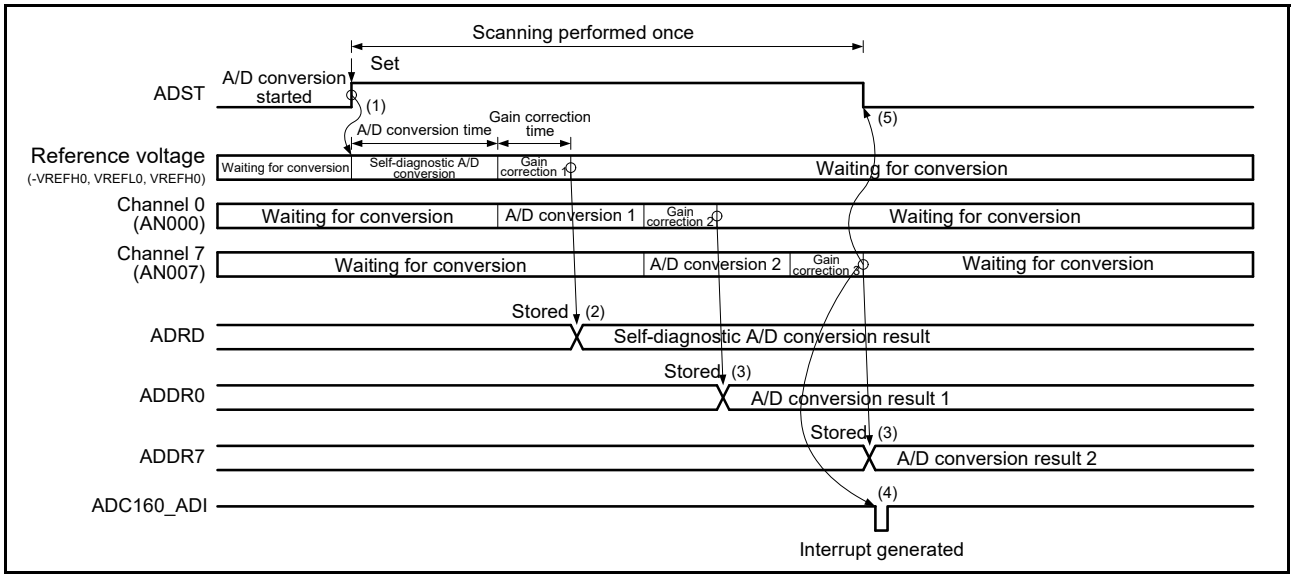


Figure 32.11 Example of basic operation in single scan mode when AN000 and AN007 are selected with self-diagnosis

32.3.4.3 A/D conversion of temperature sensor output/internal reference voltage

A/D conversion is performed on the temperature sensor output or the internal reference voltage in single scan mode as described in this section.

When selecting A/D conversion of the temperature sensor output or the internal reference voltage, deselect all analog input channels by setting the ADANSA0 and 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 and 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 and the gain correction, the result is stored in the associated Temperature Sensor Data Register (ADTSDR) or A/D Internal Reference Voltage Data Register (ADOCDR), and an ADC160_ADI interrupt request is generated.
4. The ADST bit remains 1 during A/D conversion and is automatically set to 0 on completion of A/D conversion and the gain correction. The ADC16 then enters a wait state.

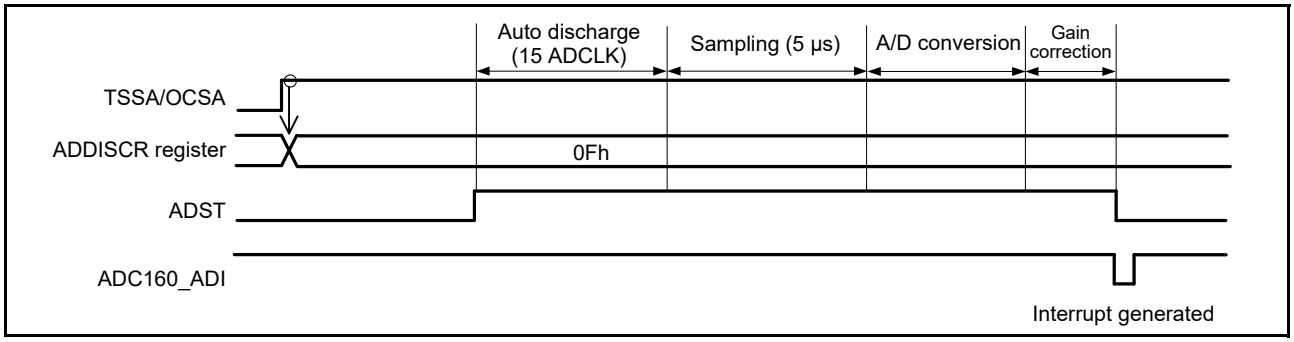


Figure 32.12 Example of basic operation in single scan mode when temperature sensor output or internal reference voltage is selected

32.3.4.4 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.

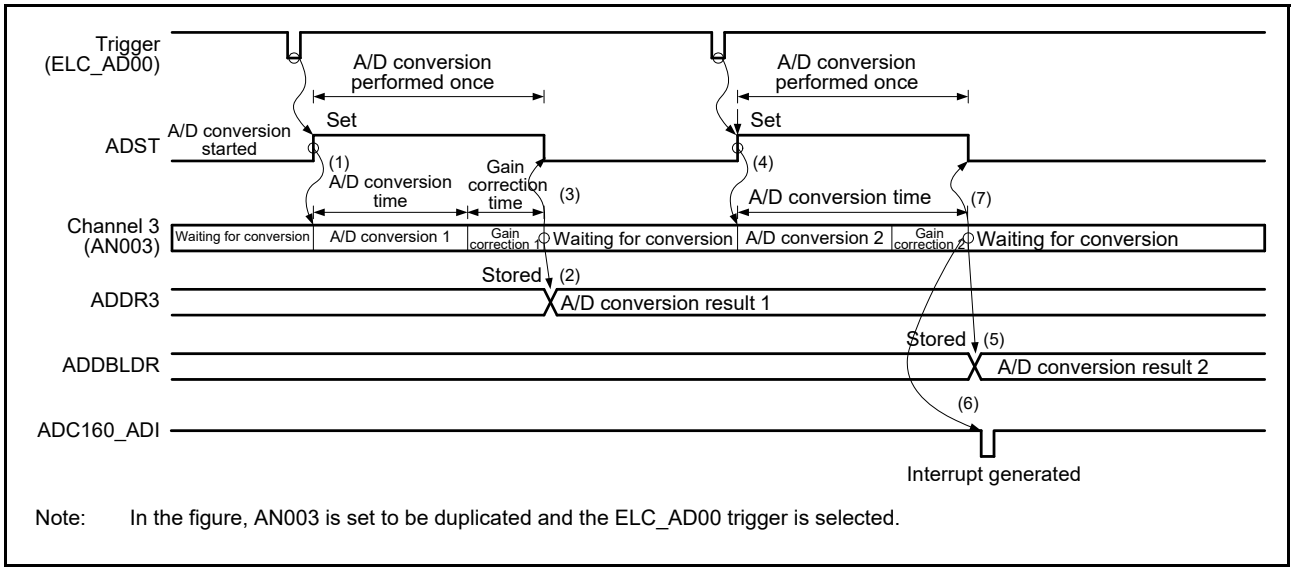
Deselect self-diagnosis and deselect the temperature sensor output A/D conversion and the internal reference voltage A/D conversion by setting the ADEXICR.TSSA and ADEXICR.OCSA bits to 0.

Duplication of A/D conversion data is enabled by setting the channel number to be duplicated in the DBLANS[4:0] bits in ADCSR register and setting the DBLE bit in ADCSR register to 1. When the DBLE bit in ADCSR register 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 ADST bit in ADCSR register 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 DBLANS[4:0] bits in ADCSR register.
2. Each time A/D conversion and the gain correction of a single channel complete, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
3. The ADST bit is automatically set to 0 and the ADC16 enters a wait state. An ADC160_ADI interrupt request is not generated.
4. When the ADST bit in ADCSR register is set to 1 (A/D conversion start) by the second trigger input, A/D conversion starts on the single channel selected in the DBLANS[4:0] bits in ADCSR register.
5. When A/D conversion and the gain correction completes, the A/D conversion result is stored in the A/D Data Duplexing Register (ADDBLDR), which is only used in double trigger mode.
6. An ADC160_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 and the gain correction completes. The ADC16 then enters a wait state.



Note: In the figure, AN003 is set to be duplicated and the ELC_AD00 trigger is selected.

Figure 32.13 Example of operation in single scan mode with double trigger mode selected when AN003 is duplicated

32.3.4.5 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 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.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated in the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid.

In 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 and the gain correction completes, the A/D conversion result is stored in the associated A/D Data Register y (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 ADC16 enters a wait state. An ADC160_ADI interrupt is not generated.
4. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger (ELC_AD00/ELC_AD01), A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
5. When A/D conversion and the gain correction complete, the A/D conversion result is stored in the A/D Data Duplexing Register (ADDBLDR) and in A/D Data Duplexing Register A (ADDBLDRA) or A/D Data Duplexing Register B (ADDBLDRB) when the trigger of ELC_AD00 or ELC_AD01 is input, respectively.
6. An ADC160_ADI interrupt request is generated.
7. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion and the gain correction completes. The ADC16 then enters a wait state.

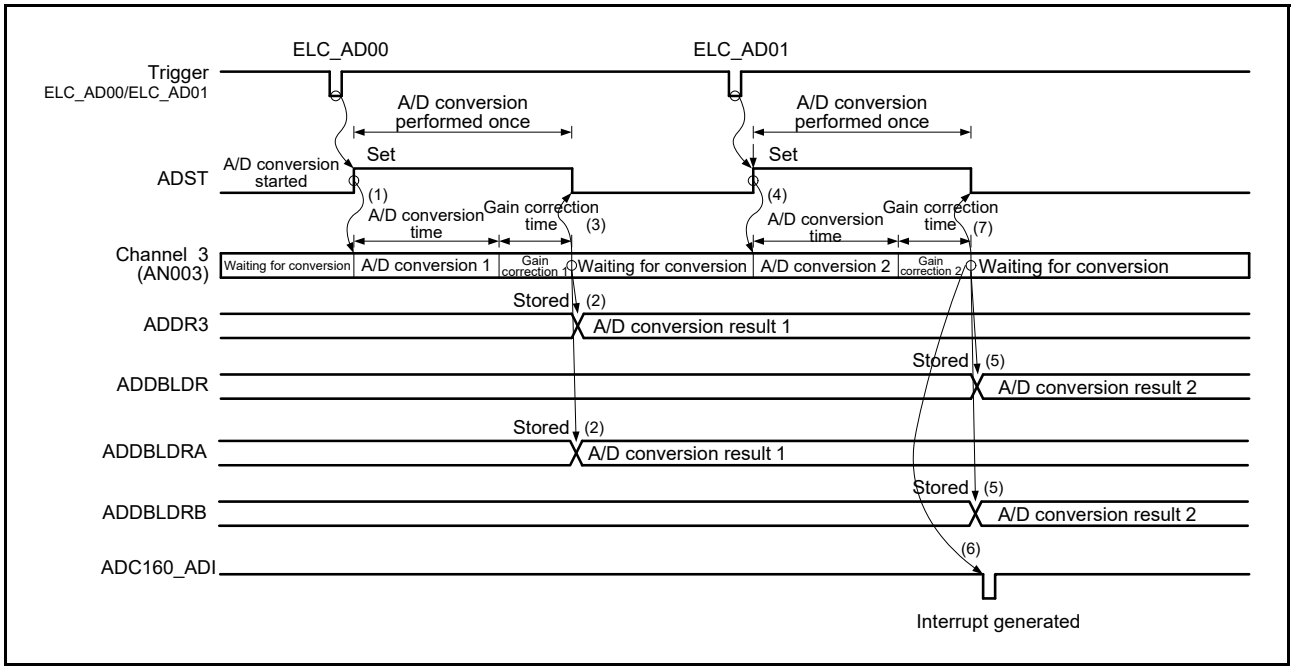


Figure 32.14 Example of extended operation in double trigger mode (1) with duplication selected for AN003, and ELC_AD00/ELC_AD01 selected

32.3.5 Continuous Scan Mode

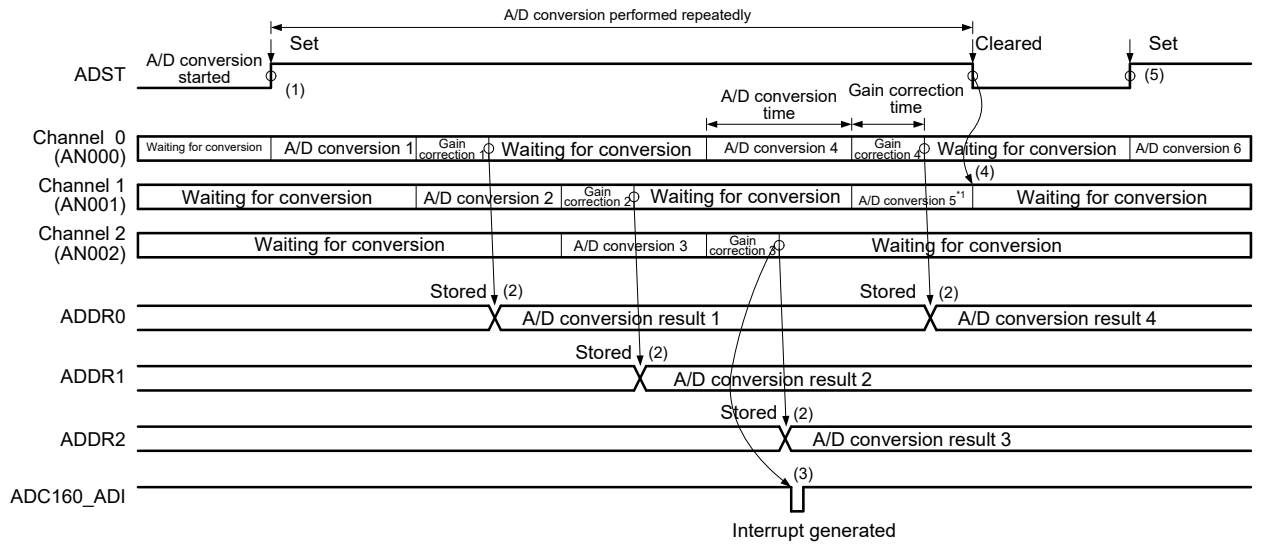
32.3.5.1 Basic operation

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 ADST bit in ADCSR register 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. Each time A/D conversion and the gain correction of a single channel complete, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
3. When A/D conversion and the gain correction of all the selected channels complete, an ADC160_ADI interrupt request is generated. The ADC16 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 ADST bit in ADCSR register 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 and the gain correction stop and the ADC16 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.



Note 1. Data for A/D conversion 5 is ignored.

Figure 32.15 Example of basic operation in continuous scan mode with AN000 to AN002 selected

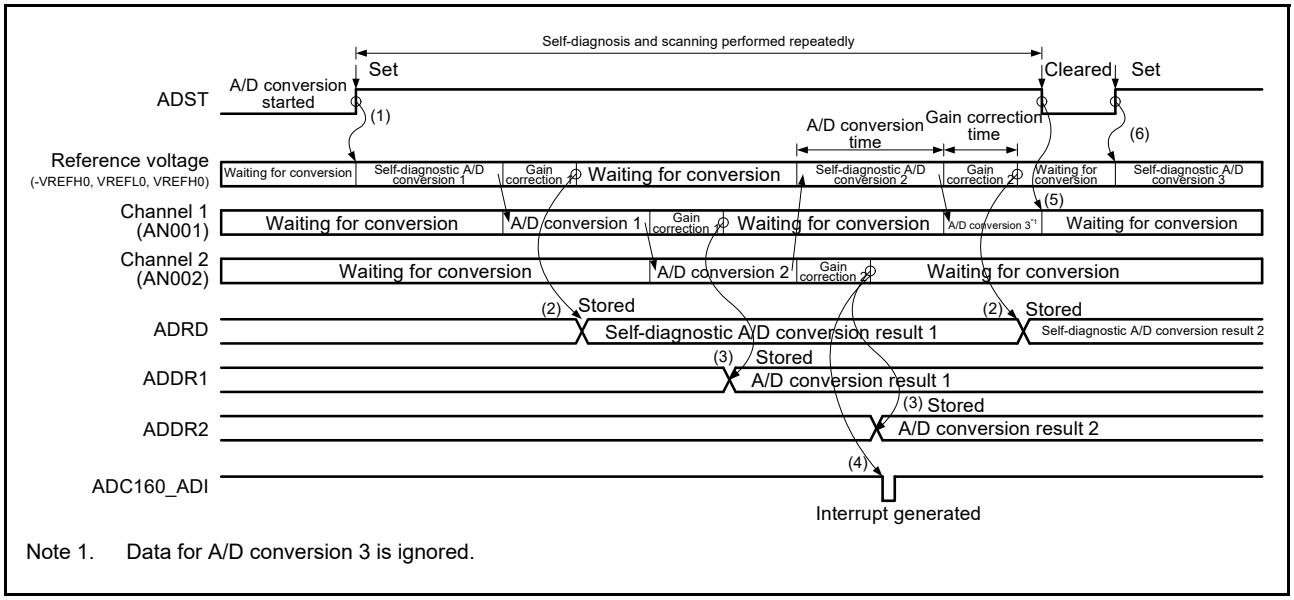
32.3.5.2 Channel selection and self-diagnosis

When channels are selected together with self-diagnosis, A/D conversion is first performed on the reference voltage (-VREFH0, VREFL0, or VREFH0) supplied to the ADC16, then A/D conversion is performed on the analog input of the selected channels. This sequence is repeated as described in the 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 ADST bit in ADCSR register 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 and the gain correction for self-diagnosis complete, 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 and the gain correction of a single channel complete, the A/D conversion result is stored into the associated A/D Data Register y (ADDRy).
4. When A/D conversion and the gain correction of all the selected channels complete, an ADC160_ADI interrupt request is generated. At the same time, the ADC16 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 and the gain correction stop and the ADC16 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.



Note 1. Data for A/D conversion 3 is ignored.

Figure 32.16 Example of basic operation in continuous scan mode when AN001 and AN002 are selected with self-diagnosis

32.3.6 Group Scan Mode

32.3.6.1 Basic operation

In group scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in group A or group B after scanning is started by a synchronous trigger (ELC). The scan operation of each group is similar to the scan operation in single scan mode.

The synchronous triggers of group A and B can be selected 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, deselect the temperature sensor output A/D conversion and the internal reference voltage A/D conversion by setting the ADEXICR.TSSA and ADEXICR.OCSA bits to 0. When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed for group A and 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 and the gain correction complete, an ADC160_ADI interrupt is generated.
3. Scanning of group B is started by ELC_AD01.
4. When group B scanning and the gain correction complete, an ADC160_GBADI interrupt is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).

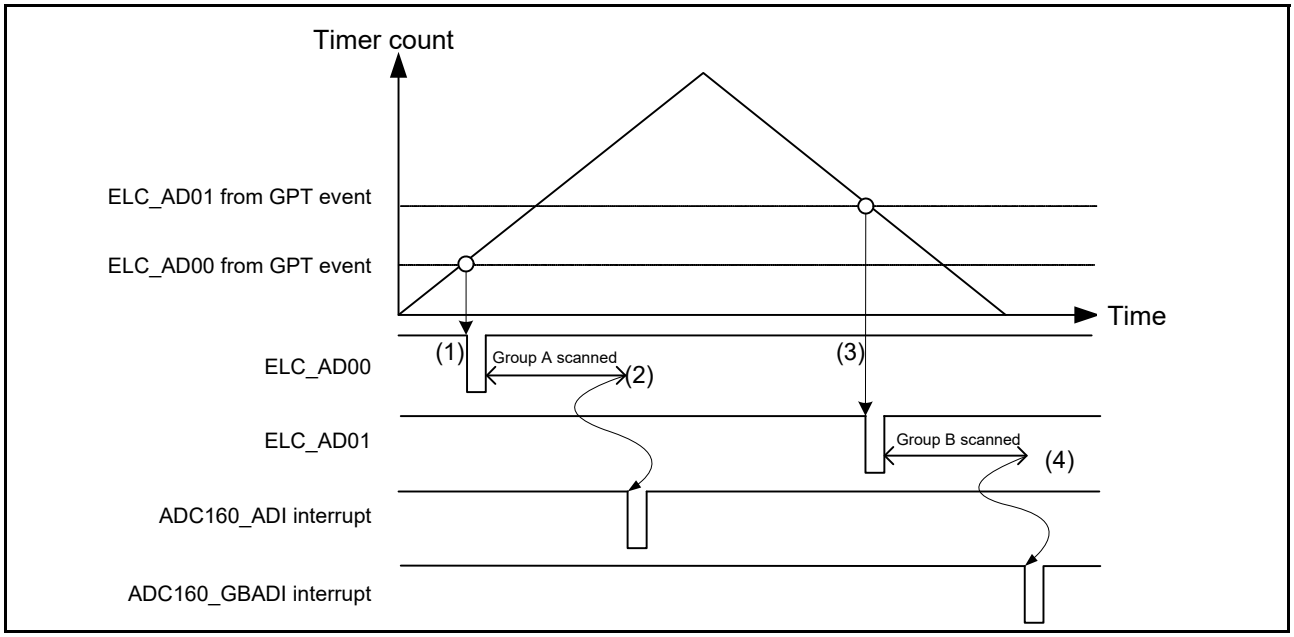


Figure 32.17 Example of basic operation in group scan mode with synchronous triggers from ELC

32.3.6.2 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 an ELC_AD00/ELC_AD01 is selected as group A synchronous triggers 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 in the DBLANS[4:0] bits in ADCSR and setting the DBLE bit in ADCSR register 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. 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 B is started by the ELC_AD00 trigger from the ELC.
2. When group B scanning and the gain correction completes, an ADC160_GBADI interrupt is generated if the GBADIE bit in ADCSR register is 1 (group B scan end interrupt is enabled).
3. The first scanning of group A is started by the first ELC_AD01 trigger.
4. When the first scanning of group A and the gain correction complete, the conversion result is stored in the associated A/D Data Register y (ADDRy). An ADC160_ADI interrupt request is not generated.

5. The second scanning of group A is started by the second ELC_AD01 trigger.
6. When the second scanning of group A and the gain correction complete, the conversion result is stored in the ADDBLDR register. An ADC160_ADI interrupt is generated.

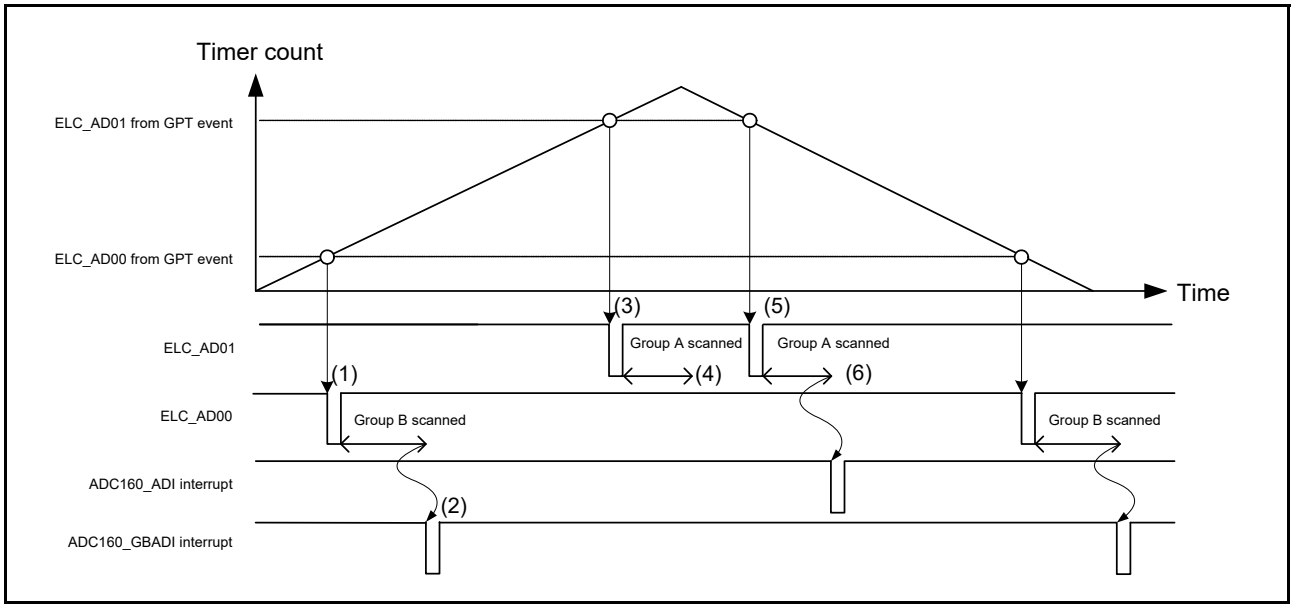


Figure 32.18 Example of operation in group scan mode with double trigger mode using synchronous triggers from the ELC

32.3.6.3 Operation with group A priority control

Setting the ADGSPCR.PGS bit to 1 in group scan mode allows the operation to proceed with group A priority control. When setting the PGS bit in the ADGSPCR register to 1, follow the procedure described in Figure 32.19. 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 setting of the ADGSPCR.GBRSCN bit is 0, the ADC16 enters wait state on completion of the A/D conversion for group A. If the setting of the ADGSPCR.GBRSCN bit is 1, the ADC16 automatically restarts group B scanning from the head of the group after completion of the A/D conversion for group A. Table 32.14 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. For group B, select channels different from those for group A, using the ADANSB0 and ADANSB1 registers.

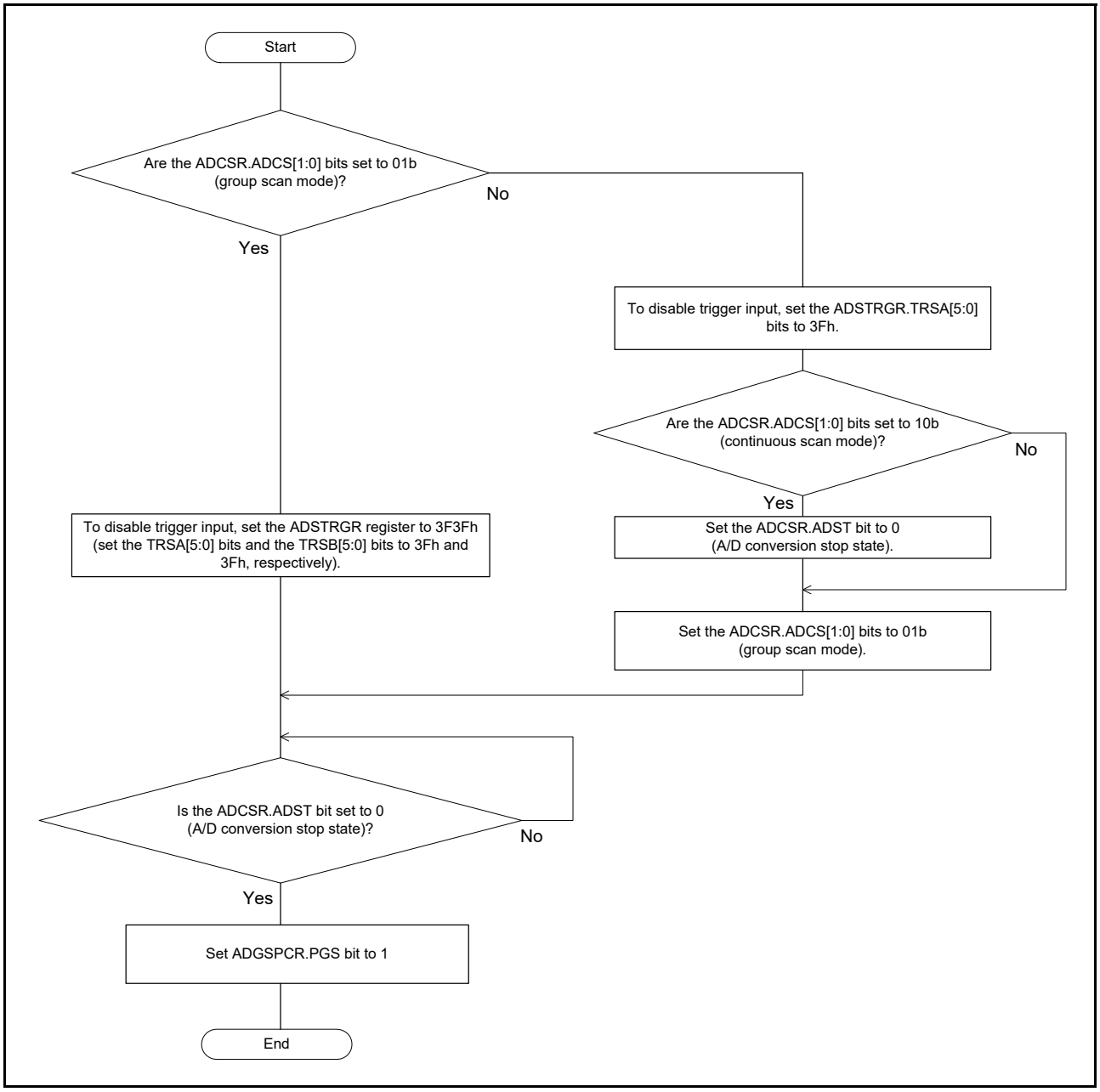


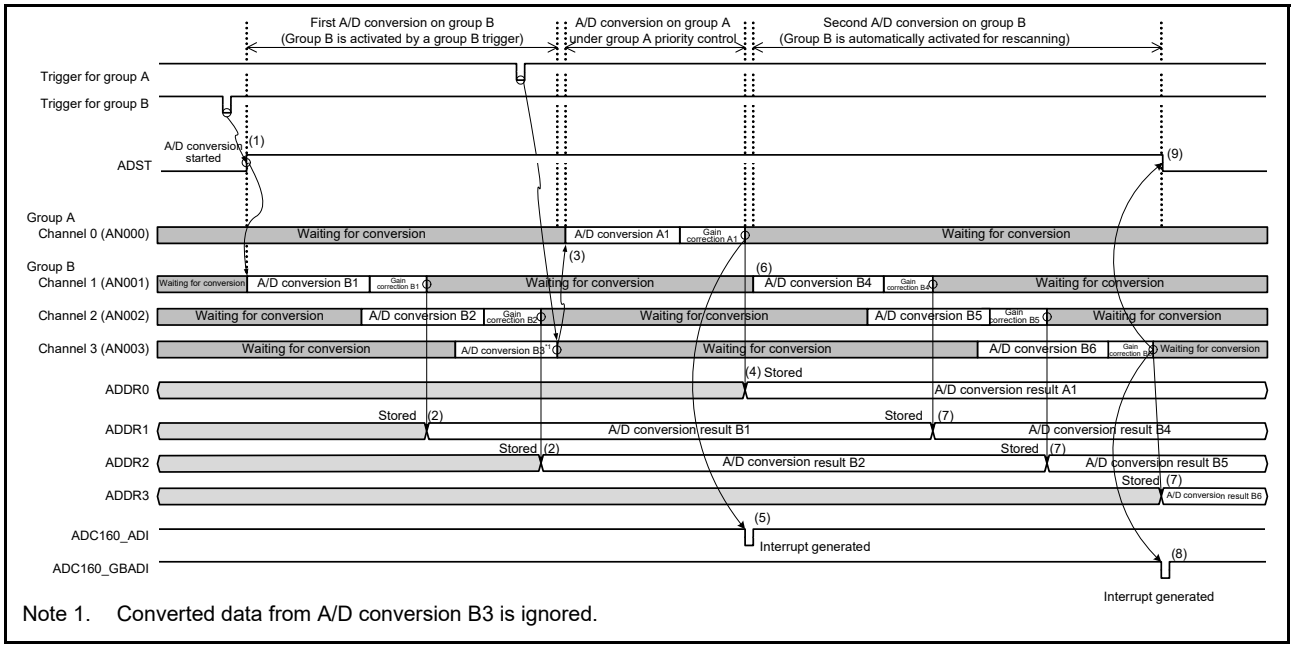
Figure 32.19 Flow for ADGSPCR.PGS bit setting

Table 32.14 Control of A/D conversion operations based on the ADGSPCR.GBRSCN bit settings

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
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 the operations in group scan mode with group A priority control (for example, (ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

1. When input of a trigger for group B sets the ADCSR.ADST bit to 1 (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 and the gain correction, 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 and gain correction are not completed 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 and the gain correction on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
5. An ADC160_ADI interrupt request is generated.
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 and the gain correction on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
8. An ADC160_GBADI interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).
9. The ADCSR.ADST bit is automatically cleared and the 16-bit A/D converter enters the wait state when A/D conversion and gain correction are complete.



Note 1. Converted data from A/D conversion B3 is ignored.

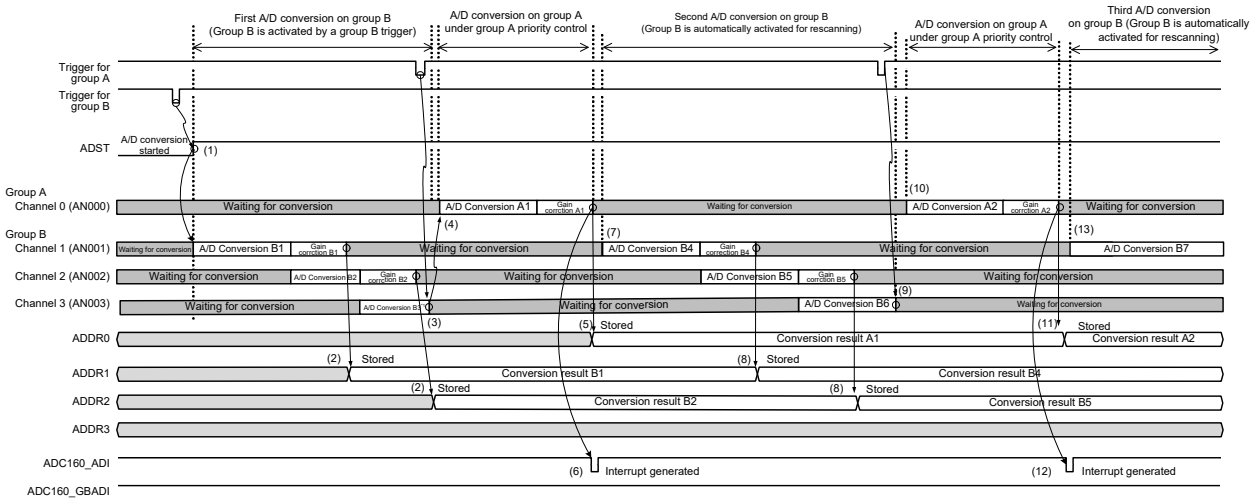
Figure 32.20 Example operation with group A priority control (1), when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0

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 0 is selected for group A and channels 1 to 3 are selected for group B when operation on group A is given priority (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0).

1. When a group B trigger input sets the ADCSR.ADST bit to 1 (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 and the gain correction on a single 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, A/D conversion for group B is discontinued with the ADCSR.ADST bit remains 1. If A/D conversion and gain correction are 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. 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 and the gain correction on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
6. An ADC160_ADI interrupt request is generated.
7. If the ADGSPCR.GBRSCN bit is 1, when the A/D conversion and the gain correction of group A are 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 and the gain correction on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
9. When a group A trigger is input while A/D conversion on group B is rescanning, A/D conversion for group B is discontinued with the ADCSR.ADST bit remains 1.
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 and the gain correction on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
12. An ADC160_ADI interrupt request is generated.

13. If the ADGSPCR.GBRSCN bit is 1 when the A/D conversion and the gain correction of group A are 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 and the gain correction of A/D conversion on group B and the ADC16 enters a wait state.



Note 1. Converted data from A/D conversion B3 and B6 are ignored.

Figure 32.21 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 1 to 3 are selected for group A and channel 0 is selected for group B when operation on group A is given priority (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0).

1. When input of a 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 and the gain correction 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 and gain correction complete. However, if group A triggers are input continuously, the scan operation on group B is canceled by group A and is not performed.

4. On completion of the group A conversion and gain correction, an ADC160_ADI interrupt request is generated without register setting.

5. On completion of the group A conversion and gain correction, the ADCSR.ADST bit remains 1 and group B is rescanned. 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 and the gain correction on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).

7. On completion of the rescanning operation and the gain correction on group B, an ADC160_GBADI interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).

8. The ADCSR.ADST bit is automatically cleared and the 16-bit A/D converter enters the wait state when A/D conversion and gain correction are complete.

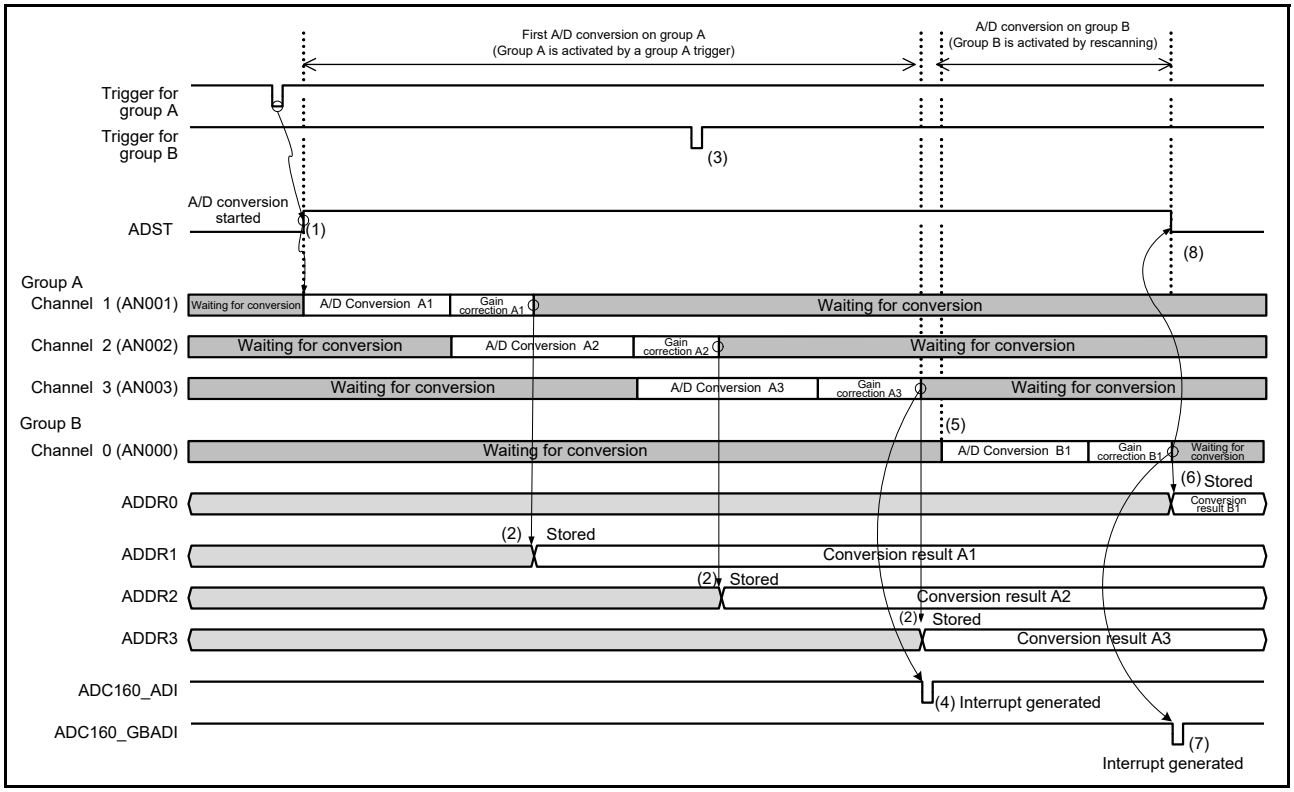


Figure 32.22 Example operation with group A priority control (3), when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0

The following sequence is an example of operation with group A priority control in which channel 0 is selected for group A and channels 1 to 3 are selected for group B (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0).

1. When input of a 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 and the gain correction 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, A/D conversion for group B is discontinued with the ADCSR.ADST bit remains 1. Next, 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 and the gain correction on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
5. An ADC160_ADI interrupt request is generated.
6. The ADCSR.ADST bit is automatically cleared and the 16-bit A/D converter enters the wait state when A/D conversion and gain correction are complete.

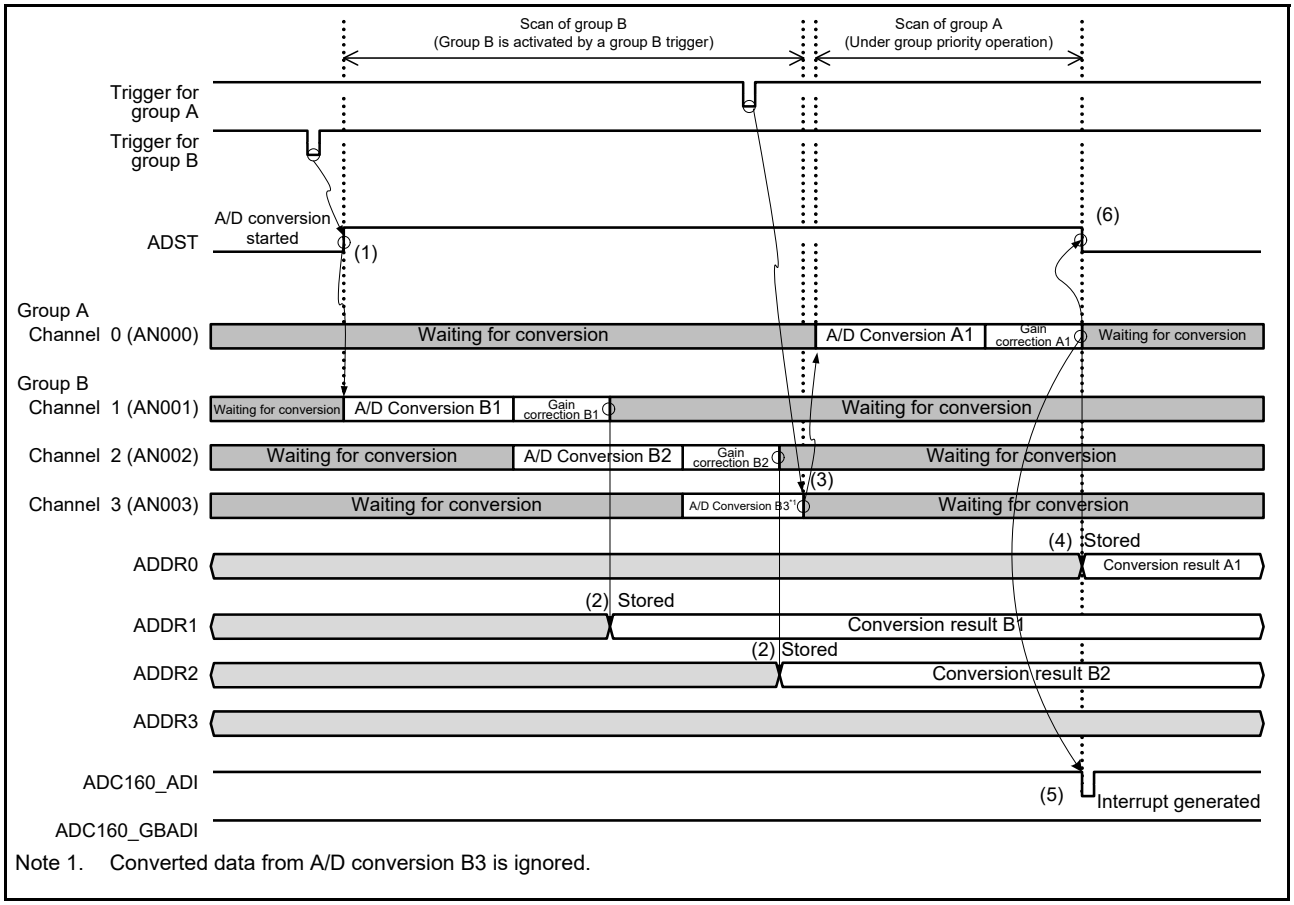
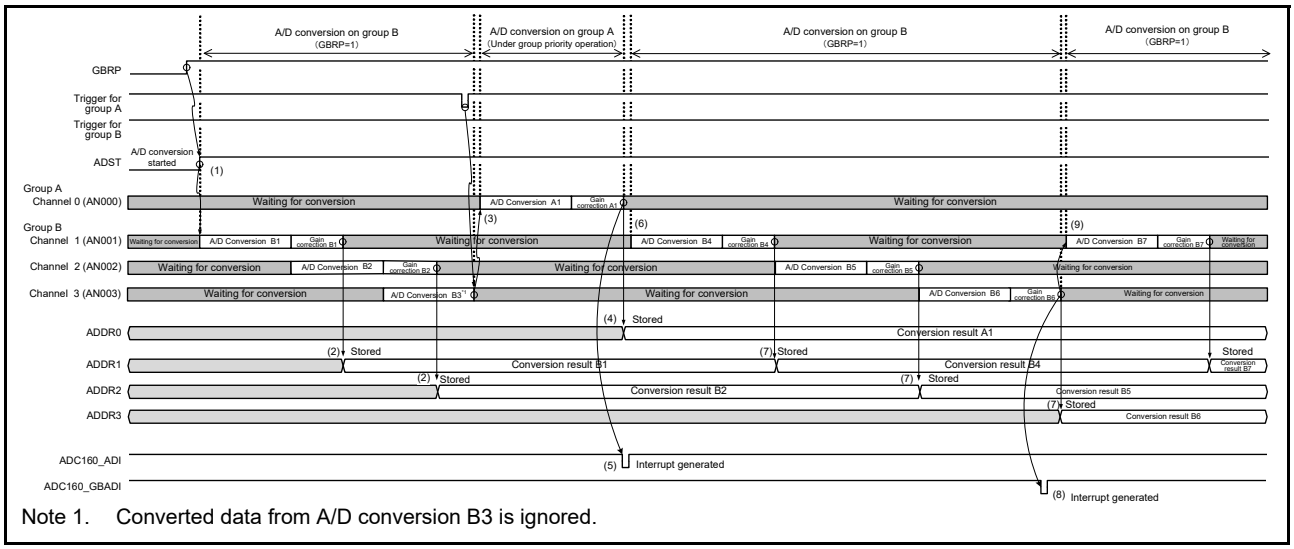


Figure 32.23 Example operation with group A priority control (4), when ADGSPCR.GBRSCN = 0 and ADGSPCR.GBRP = 0

The following sequence is an example of operation with group A priority control in which channel 0 is selected for group A and channels 1 to 3 are selected for group B (ADGSPCR.GBRP = 1).

1. The ADCSR.ADST bit is set to 1 (A/D conversion start) when ADGSPCR.GBRP bit 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 and the gain correction on a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
3. If a group A trigger is input while A/D conversion for group B is in progress, A/D conversion for group B is discontinued with the ADCSR.ADST bit remains 1. Next, 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 and the gain correction on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
5. An ADC160_ADI interrupt request is generated.
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 and the ADCSR.ADST bit remains 1.
7. On completion of A/D conversion and the gain correction on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
8. An ADC160_GBADI interrupt request is generated if the ADCSR.GBADIE bit is 1.
9. 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 and the ADCSR.ADST bit remains 1. 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 when ADGSPCR.GBRP = 1, follow the

procedure shown in Figure 32.38.



Note 1. Converted data from A/D conversion B3 is ignored.

Figure 32.24 Example operation with group A priority control (5) when ADGSPCR.GBRP = 1

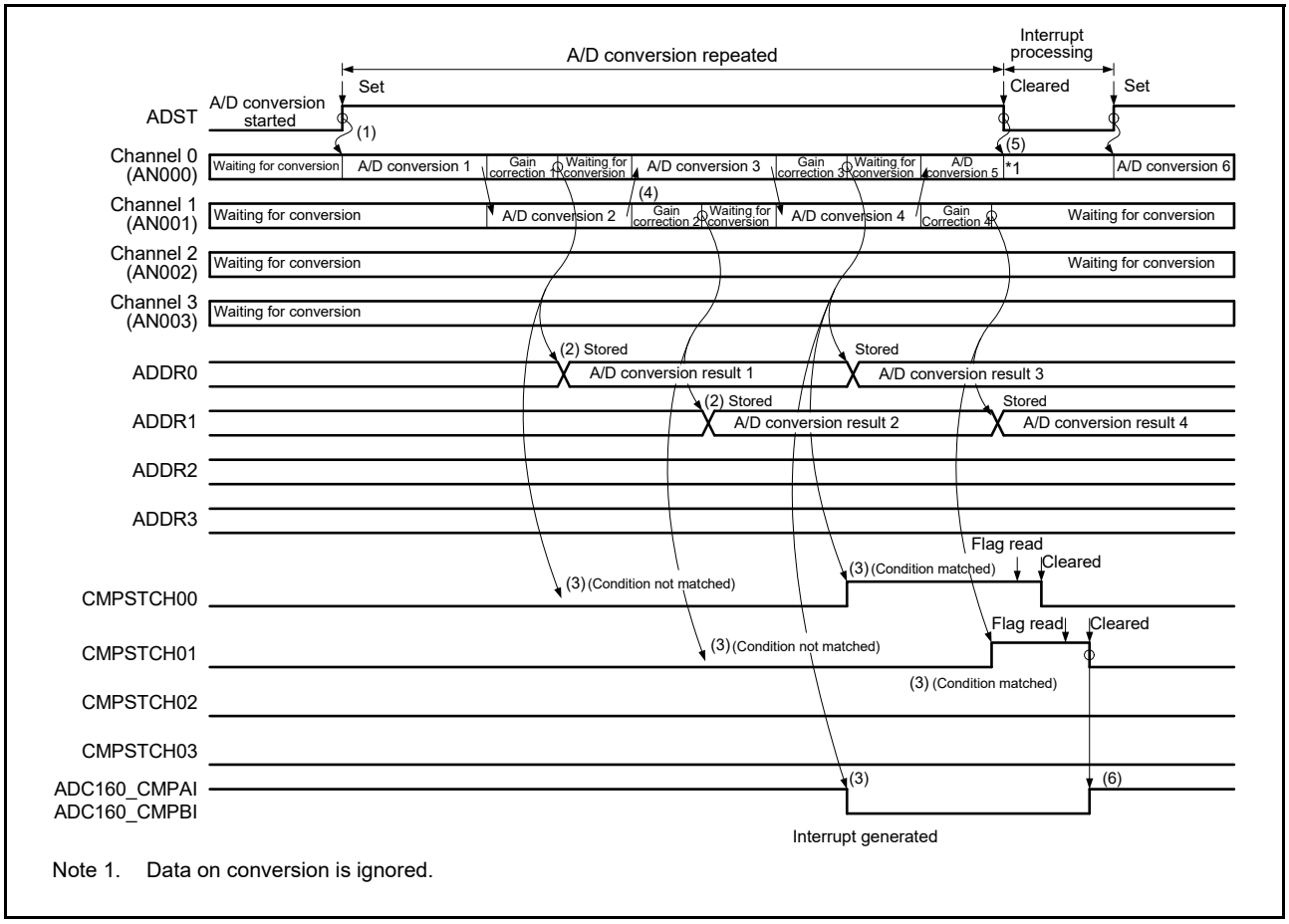
32.3.7 Compare Function for Window A and Window B

32.3.7.1 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 in the order of 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 and the gain correction, the A/D conversion result is stored in the associated A/D Data Register (ADDRy, ADTSDR, or ADOCDR). When ADCMPCR.CMPAE bit is 1, if bits in the ADCMPANSR0/1 register or the ADCMPANSER register are set for window A, the A/D conversion result is compared with the ADCMPDR0/1 registers value. When ADCMPCR.CMPBE bit is 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 registers, 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 ADC160_CMPAI interrupt request (level) is generated. In the same way, when window B meets the condition set in ADCMPBNSR.CMPLB, the Compare Window B flag (ADCMPBSR.CMPSTB) is set to 1. If the ADCMPCR.CMPBIE bit is 1, an ADC160_CMPBI interrupt request (level) is generated.
4. On completion of all selected A/D conversions, gain correction, and comparisons, scan restarts.
5. After the ADC160_CMPAI and ADC160_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, the ADC160_CMPAI interrupt request is canceled. In the same way, when all compare flags of window B are cleared, the ADC160_CMPBI interrupt request is canceled. To perform comparison again, restart the A/D conversion.



Note 1. Data on conversion is ignored.

Figure 32.25 Example of compare function operation when AN000 to AN003 are compared

32.3.7.2 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 reference voltage value, and outputs the ADC160_WCMPM/ADC160_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 when one channel in window A meets the comparison condition, the comparison result of window A is met. When using this function, perform A/D conversion in single scan mode.

Any channels from AN000 to AN008, AN016 to AN023, reference voltage of SDADC24, internal reference voltage, and temperature sensor output are selectable for window A. However, neither the internal reference voltage nor the temperature sensor output can be selected together with any other channel. Additionally, if the internal reference voltage is selected as the high-potential reference voltage of the ADC16, the internal reference voltage or the temperature sensor output cannot be A/D converted.

A single channel from AN000 to AN008, AN016 to AN023, reference voltage of SDADC24, internal reference voltage, and temperature sensor output is selectable for window B. Additionally, if the internal reference voltage is selected as the high-potential reference voltage, the internal reference voltage or the temperature sensor output cannot be A/D converted.

The following sequence describes the setting procedure and an example when using the 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 the ADCMPANSR0/1 and ADCMPANSER registers. Set the window comparison conditions in the ADCMPLR0/1 and 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/ADWINLLB registers.
4. Set the composite conditions for window A/B, window A/B operation enable, and interrupt output enable in the ADCMPCR register.

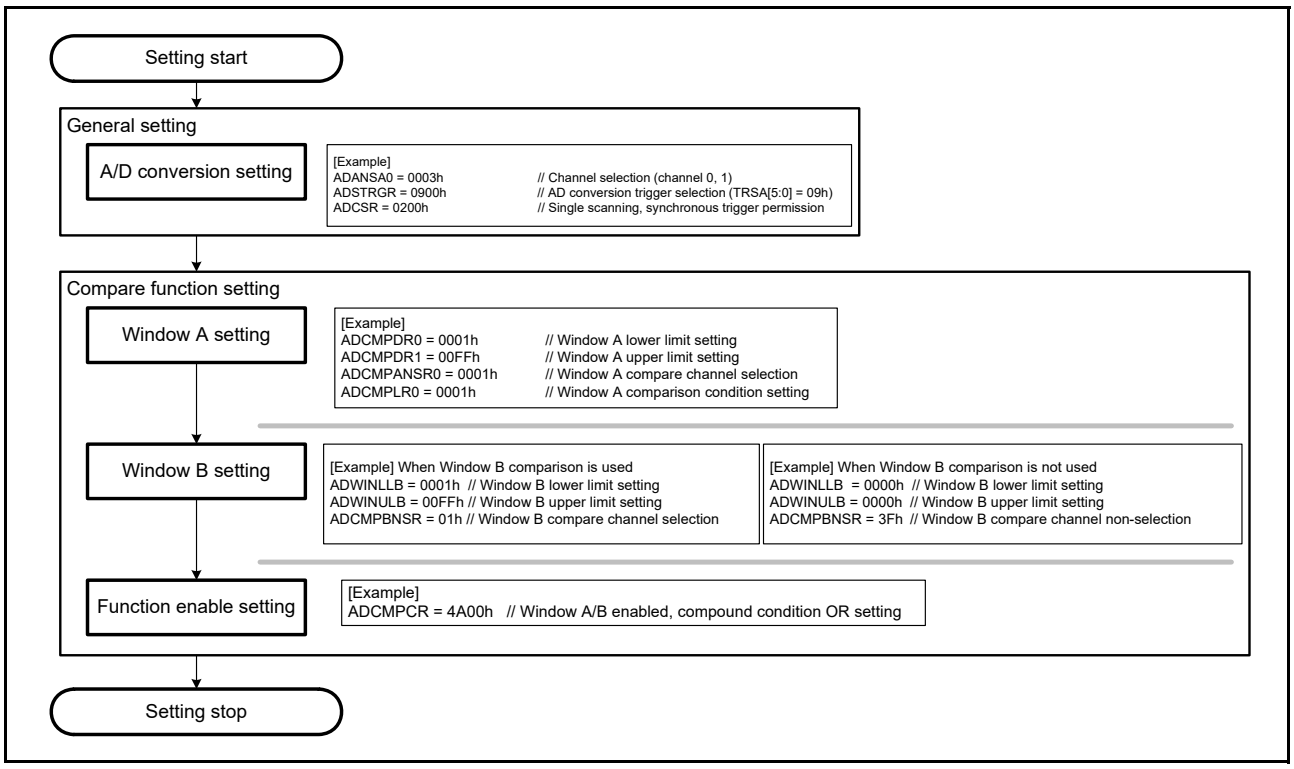


Figure 32.26 Setting example when using event output of the compare function

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 the 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 always mismatch (ADCMPCR.WCMPE = 1, ADWINLLB[15:0] = ADWINULB[15:0] = 0000h, and ADCMPBNSR.CMPLB = 1).

Figure 32.27 shows an example event output operation of the compare function.

A scan end event (ADC160_ADI) is output at the same time as single scan and the gain correction completion. A match or mismatch event (ADC160_WCMPM/ADC160_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.

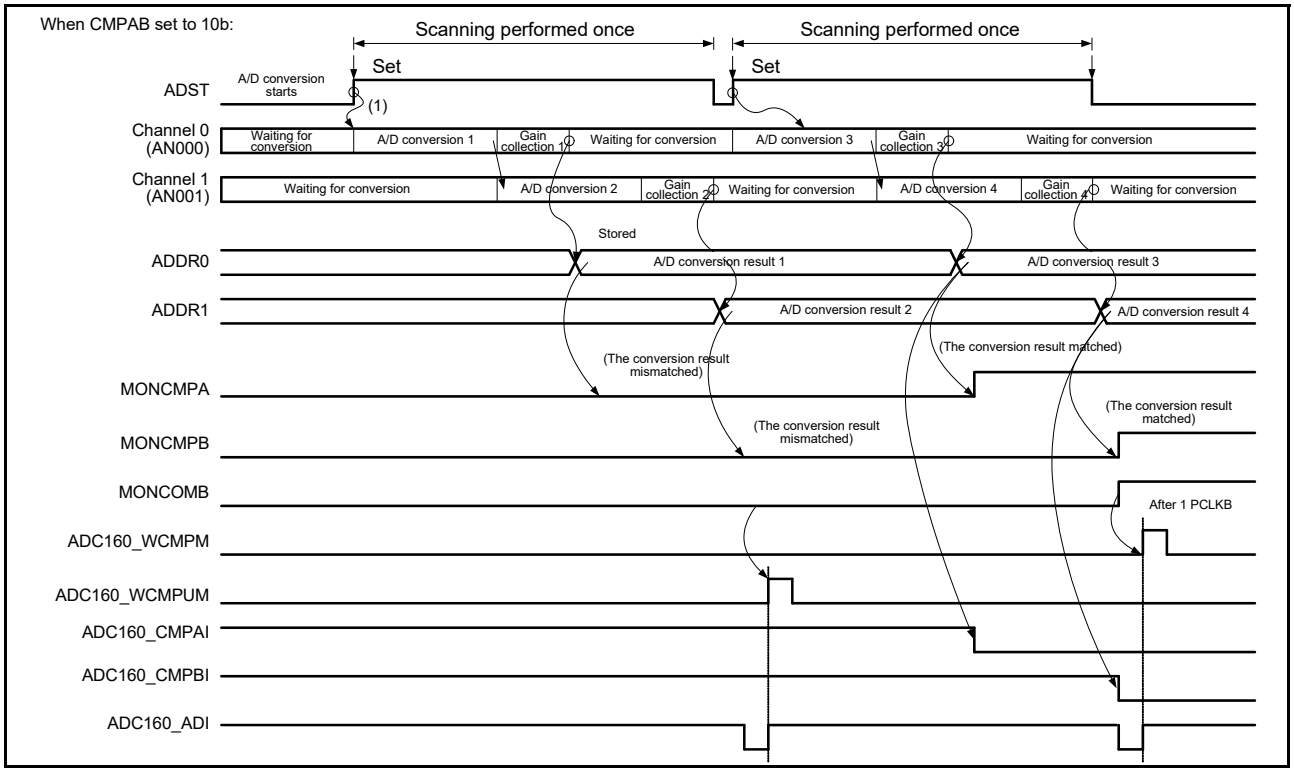


Figure 32.27 Event output operation example of the compare function when AN000 and AN001 are compared

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 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 bit to 0.

32.3.7.3 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.

32.3.8 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 32.28 shows the scan conversion timing, in which scan conversion is activated by a software trigger or a synchronous trigger (ELC). Figure 32.29 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})*2, A/D conversion processing time (t_{CONV}), gain correction time (t_{GAIN}) 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}) and gain correction time (t_{GAIN}). However, because gain correction occurs concurrently with sampling, the gain correction time affects the A/D conversion processing time only in the first cycle of single scan, continuous scan, and group scan. Gain correction time (t_{GAIN}) does not affect the second and later cycles. The sampling time (t_{SPL}) is used to charge sample-and-hold circuits in the A/D converter. If there is not sufficient sampling time due to the high impedance of an analog input signal source, or if the A/D conversion clock (ADCLK) is slow, sampling time can be adjusted using the ADSSTRn register.

The time for conversion by successive approximation (t_{SAM}) is 18 ADCLK states. Table 32.15 shows the scan conversion time.

The scan conversion time (t_{SCAN}) in single scan mode for which the number of selected channels is *n* can be determined as follows:

$$t_{SCAN} = t_D + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV}{}^{*3} \times n) + t_{GAIN} + t_{ED}$$

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 $(t_{DIS} \times n) + t_{DIAG} + (t_{CONV}{}^{*3} \times n)$.

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, $t_{DIAG} = 0$.

Note 3. When input sampling time (t_{SPL}) of all selected channels are the same, this element equals $t_{CONV} \times n$. If each channel has a different sampling time, this element equals the sum of t_{SPL} and t_{SAM} for each selected channel.

Table 32.15 shows the times for conversion during scanning.

Table 32.15 Times for conversion during scanning (in numbers of ADCLK and PCLKB cycles)

Parameter			Symbol		Type/Conditions: Synchronous trigger*5	Type/Conditions: Asynchronous trigger	Type/Conditions: Software trigger	Unit
Scan start processing time*1, *2	A/D conversion on group A under group A priority control	Group B is to be stopped (Group A is activated after group B is stopped due to an A/D conversion source of group A)	t_D		2 PCLKB + 6 ADCLK, 5 PCLKB + 3 ADCLK*6	—	—	Cycle
		Group B is not to be stopped (activation by an A/D conversion source of group A)			2 PCLKB + 4 ADCLK	—	—	
	A/D conversion when self-diagnosis is enabled	A/D conversion for self-diagnosis is to be started			2 PCLKB + 6 ADCLK	4 PCLKB + 6 ADCLK	6 ADCLK	
	Other than above				2 PCLKB + 4 ADCLK	2 PCLKB + 4 ADCLK	4 ADCLK	
Disconnection detection assistance processing time*7			t_{DIS}		The setting of ADNDIS[3:0] (initial value = 0h) × ADCLK*3			
Self-diagnosis conversion processing time*1	Sampling time		t_{DIAG}	t_{SPL}	The setting of ADSSTR00*4 (initial value = 0Dh) × ADCLK			
	Time for conversion by successive approximation			t_{SAM}	18 ADCLK			
A/D conversion processing time*1	Sampling time		t_{CONV}	t_{SPL}	The setting of ADSSTRn*4 (n = 00 to 08, L, T, O) (initial value = 0Dh) × ADCLK			
	Time for conversion by successive approximation			t_{SAM}	18 ADCLK			
Scan end processing time*1			t_{ED}		1 PCLKB + 3 ADCLK, 2 PCLKB + 2 ADCLK*6			
Gain correction time*1			tGAIN		10 ADCLK			

Note 1. See Figure 32.27 and Figure 32.28 for example of times t_D, t_{DIAG}, t_{CONV}, t_{GAIN} and t_{ED}. t_D and t_{ED} are the maximum time.

Note 2. This is the maximum time required from software writing or trigger input to A/D conversion start.

Note 3. The value is fixed to Fh (15 ADCLK) when the temperature sensor output or internal reference voltage is A/D-converted.

Note 4. The ADSSTRn register setting should satisfy the sampling time of electrical characteristics.

Note 5. This does not include the time consumed in the path from timer output to trigger input.

Note 6. If ADCLK is faster than PCLKB (PCLKB to ADCLK frequency ratio = 1:2 or 1:4).

Note 7. See Figure 32.30 for example of times t_{DIS}.

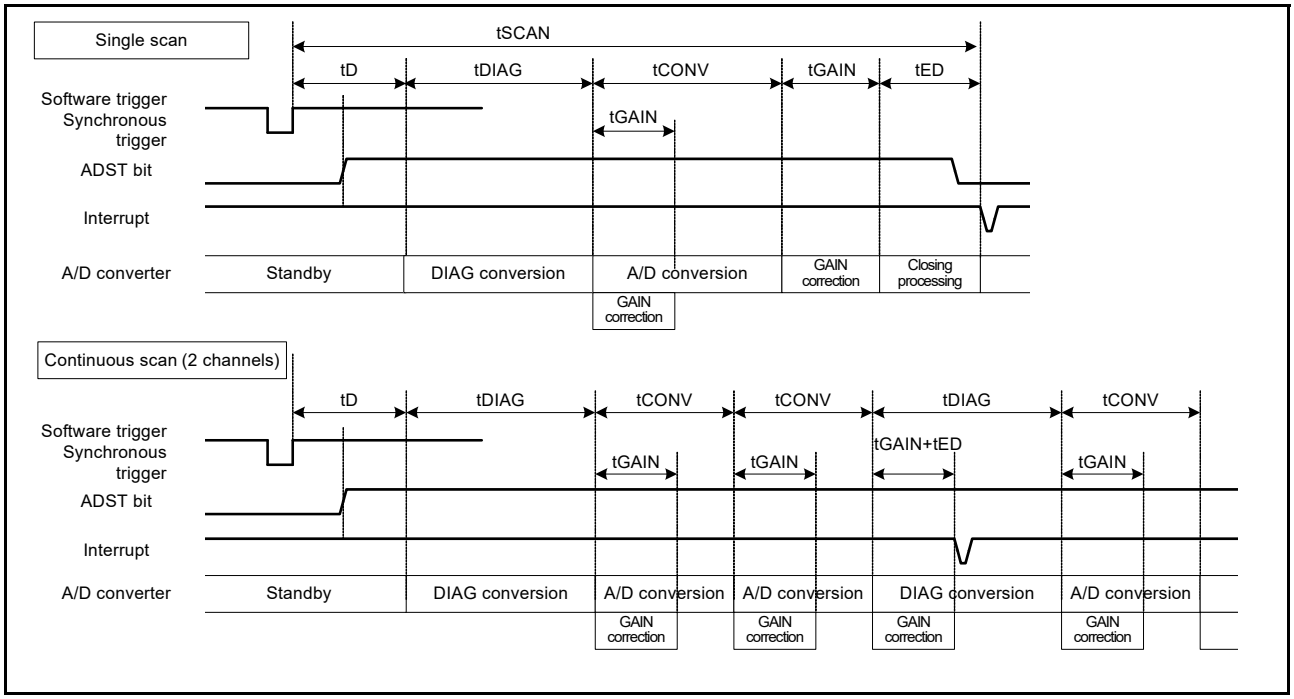


Figure 32.28 Scan conversion timing when activated by software or synchronous trigger input (ELC)

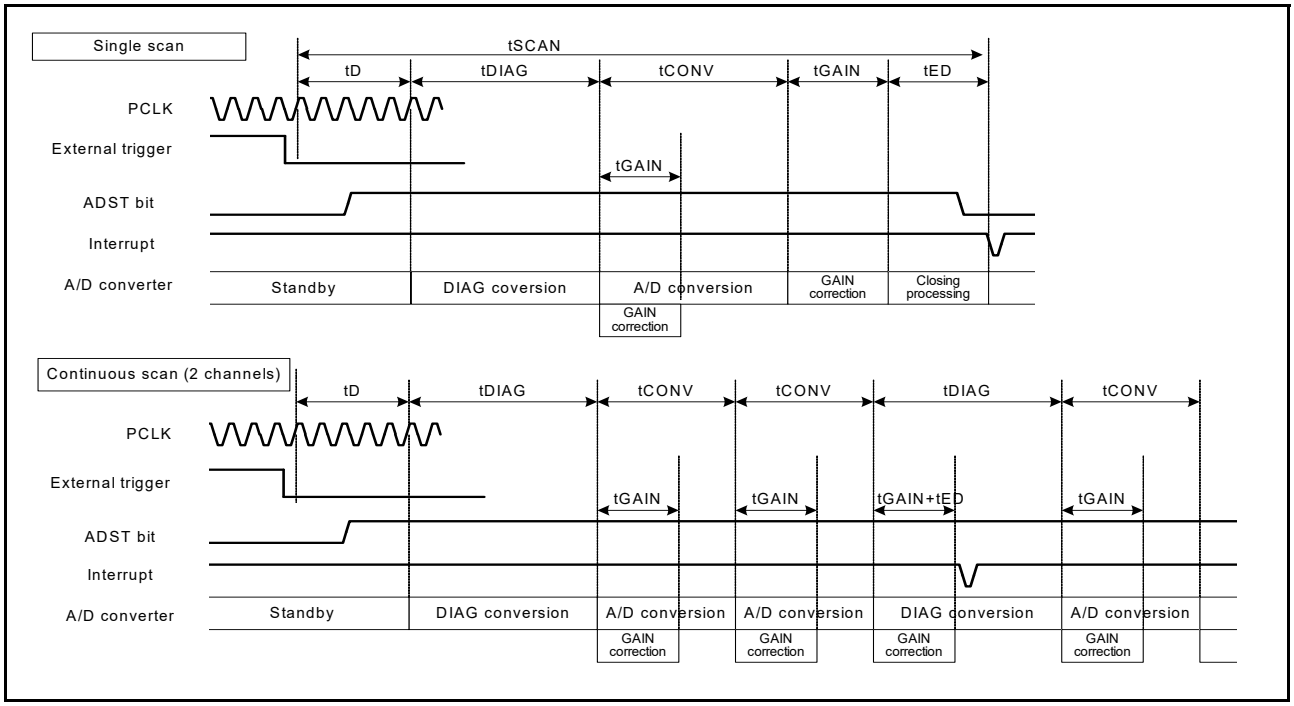


Figure 32.29 Scan conversion timing when activated by asynchronous trigger input (ADTRG0)

32.3.9 Usage Example of A/D Data Register Automatic Clearing Function

Setting the ACE bit in the ADCER register 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 or DTC.

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 register is 0 (automatic clearing is 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 register is 1 (automatic clearing is enabled), if ADDRy = 0111h is read by the CPU or DTC, 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.

32.3.10 A/D-Converted Value Average Mode

A/D-converted value average mode can be used when A/D conversion of the analog input of the selected channels, A/D conversion of the reference voltage of SDADC24, A/D conversion of the temperature sensor output, or A/D conversion of the internal reference voltage is selected.

In A/D-converted value average mode, the same channel is A/D-converted 2, 4, 8 or 16 consecutive times and the mean of the converted values is stored in the data register. The use of the average of these results can improve the accuracy of A/D conversion, depending on the types of noise components that are present. This function, however, cannot always guarantee an improvement in A/D conversion accuracy.

32.3.11 Disconnection Detection Assist Function

The ADC16 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
- The reference voltage of SDADC24 (SBIAS/VREFI)
- Differential input mode.

Figure 32.30 shows the A/D conversion operation when the disconnection detection assist function is used. Figure 32.31 shows an example of disconnection detection when precharge is selected. Figure 32.32 shows an example of disconnection detection when discharge is selected.

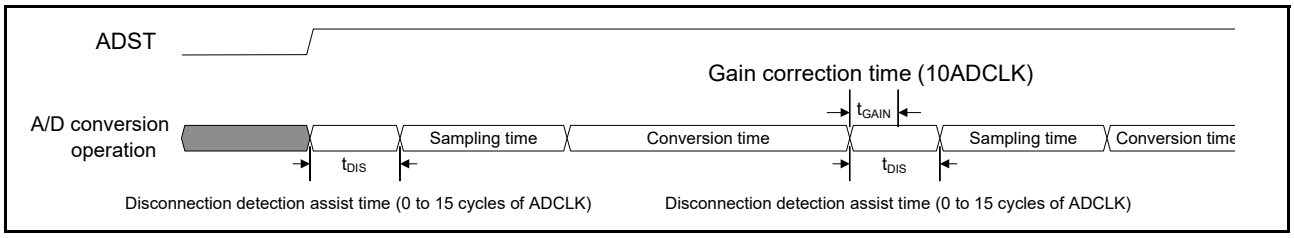
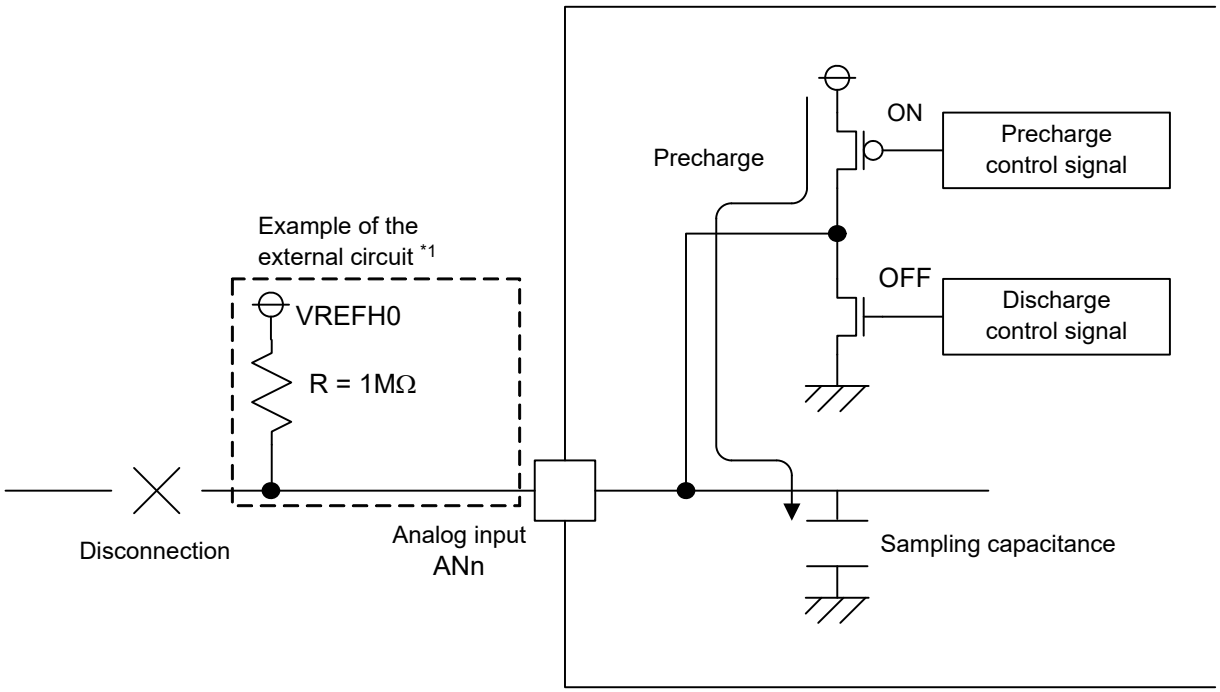
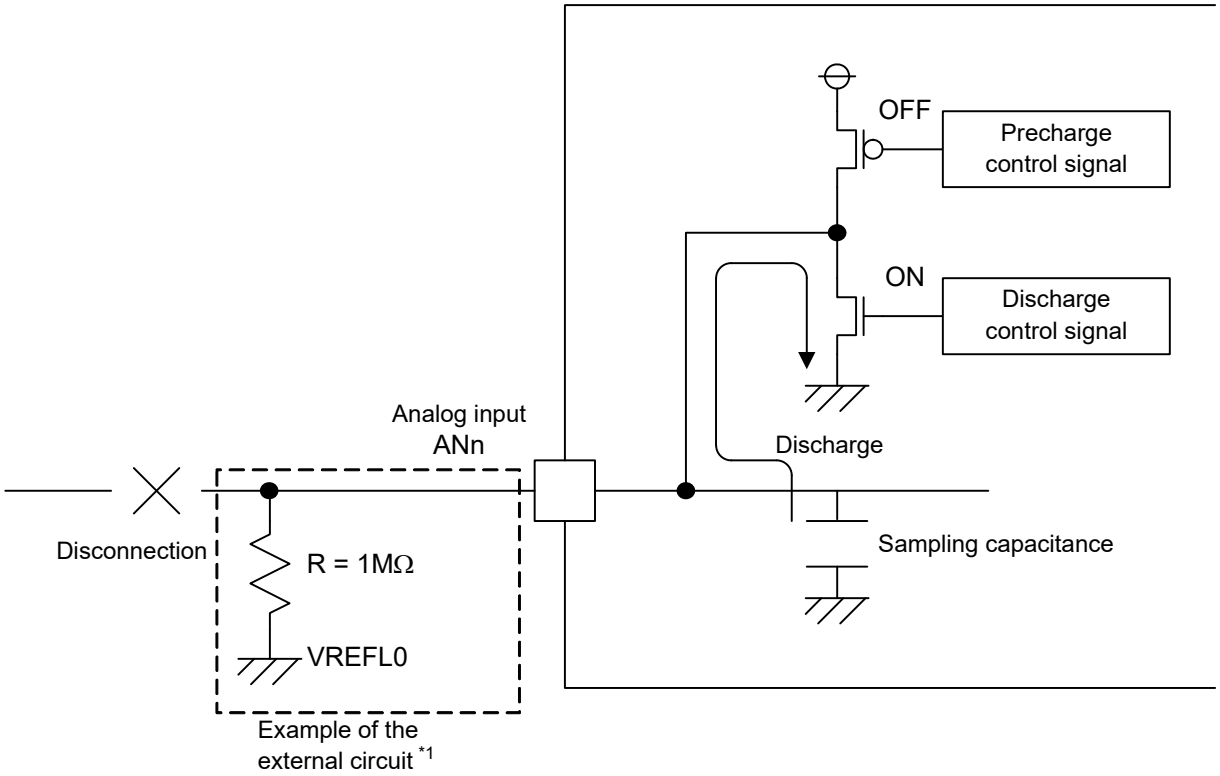


Figure 32.30 A/D conversion operation when disconnection detection assist function is used



Note 1. The converted result should be used after full evaluation because the resulting data on disconnection varies depending on the external circuit.

Figure 32.31 Example of disconnection detection when precharge is selected



Note 1. The converted result should be used after full evaluation because the resulting data on disconnection varies depending on the external circuit.

Figure 32.32 Example of disconnection detection when discharge is selected

32.3.12 Starting A/D Conversion with Asynchronous Trigger

The A/D conversion can be started by the input of an asynchronous trigger. To start the A/D 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 both the ADCSR.TRGE and ADCSR.EXTRG bits to 1.

Figure 32.33 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 18, I/O Ports.

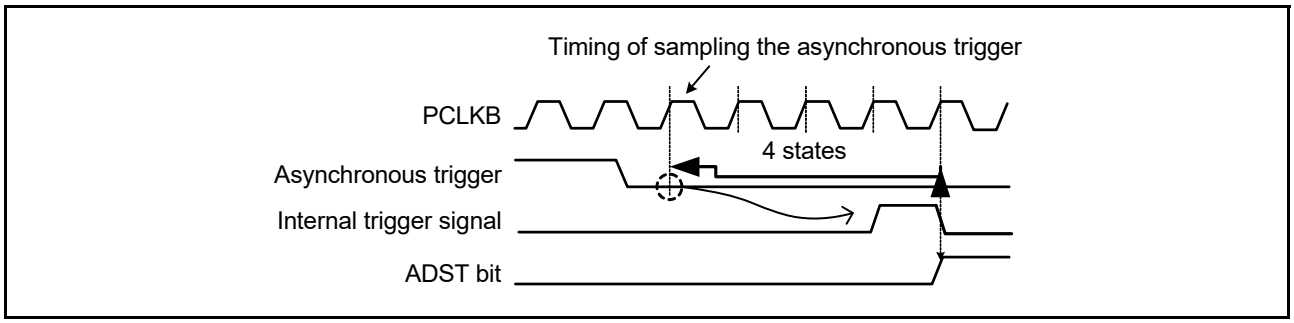


Figure 32.33 Asynchronous trigger input timing

32.3.13 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.

32.3.14 Calibration Function

Calibration allows high-precision A/D conversion by obtaining the C-DAC linearity error correction value and gain (offset) error correction value from the internally generated analog input under usage conditions. Calibration is started when 1 is written to the ADCALEXE.CALEXE bit.

Ongoing calibration does not end until the ADCSR.ADST bit is cleared to 0 (from 1) or calculation of all errors is complete. When the calibration ends, the ADCSR.ADST bit is cleared to 0 and an interrupt (ADC16_ADI) is output when the ADICR.ADIC[1:0] bits are set to 11b. Because all scans are inactive after the calibration ends, to start a single scan, continuous scan, or group scan, perform the required scanning procedure.

If calibration is stopped by clearing the ADCSR.ADST bit to 0 before calibration ends, the correction value is not guaranteed. In this case, the correction value must be updated by re-executing calibration. The correction value after reset is the correction value calculated at shipment.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by writing 1 to ADCALEXE.CALEXE bit with software, the correction values for C-DAC linearity error and GAIN error are calculated.
2. When calculation of all correction values completes and the ADICR.ADIC[1:0] bits are set to 11b (ADC16_ADI interrupt due to calibration end enabled), an ADC16_ADI interrupt request is generated.
3. The ADCSR.ADST bit remains 1 (A/D conversion start) during calibration, and is automatically cleared to 0 when calculation of all correction values completes and the A/D converter enters the standby state.

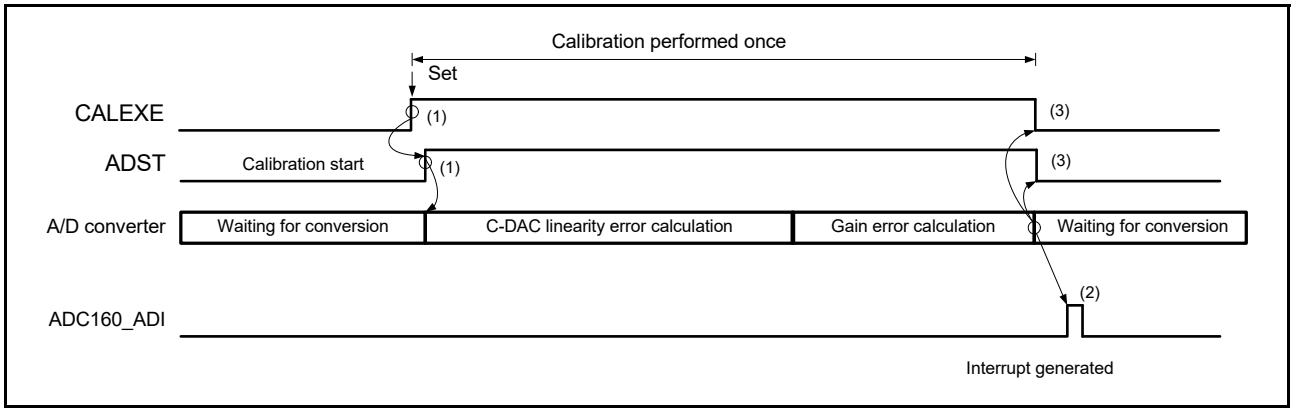


Figure 32.34 Example of calibration operation (C-DAC linearity error calculation and gain error calculation)

Figure 32.35 shows the software flow and an operation example.

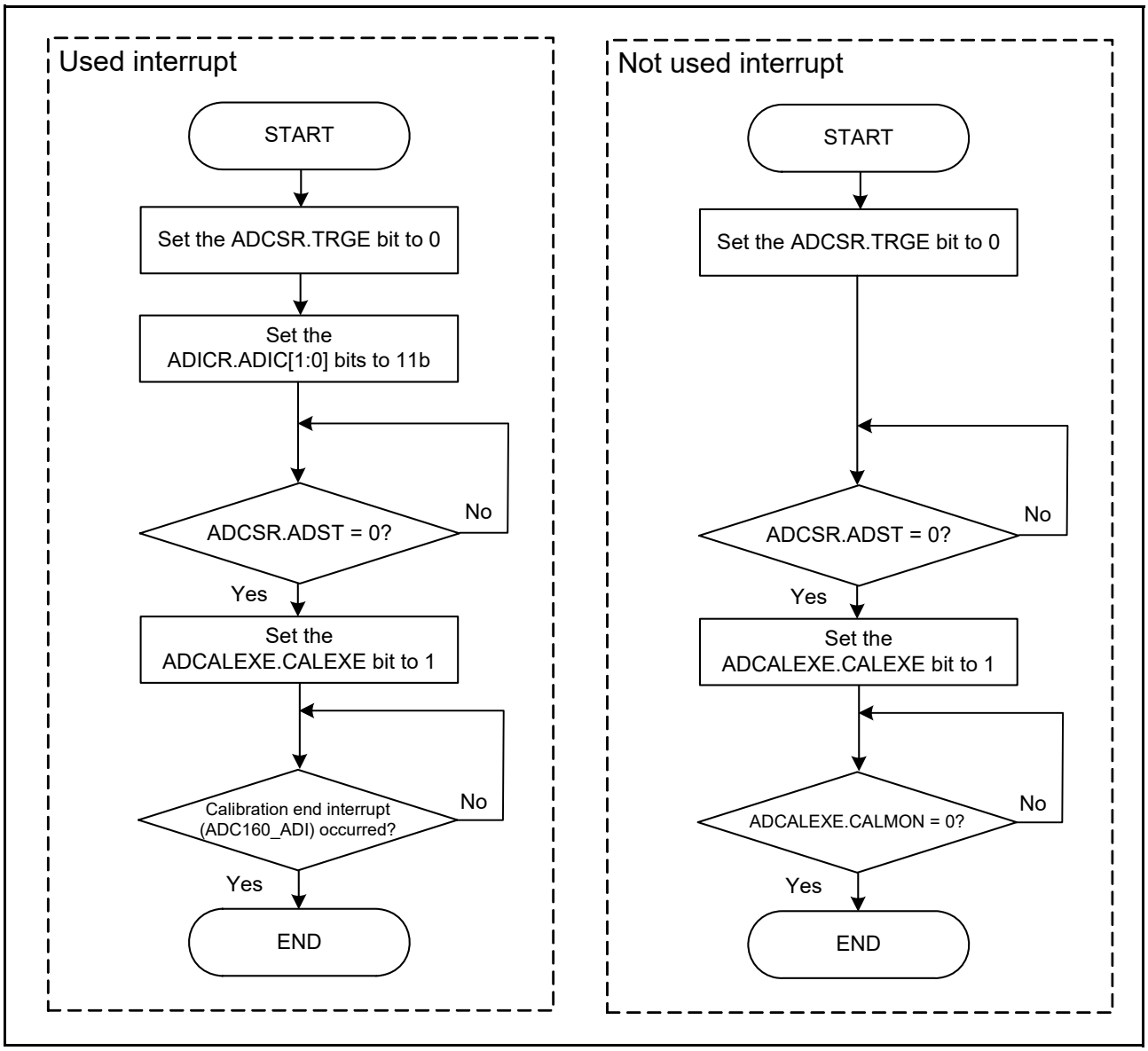


Figure 32.35 Software flow and operation example of calibration operation.

32.3.15 Calibration Time

In the calibration operation, C-DAC linearity error calculation (t_{CDAC}) and gain error calculation (t_{GAIN}) are performed after the calibration start delay time (t_{SDCAL}). When the ADICR.ADIC[1:0] bits are set to 11b after gain error is calculated, an interrupt is output (t_{EDCAL}) and the calibration is complete.

The calibration time (t_{CAL}) is the total of calibration start delay time, C-DAC linearity error calculation time, gain error calculation time, and calibration end delay time, which is expressed as the following expression:

$t_{CAL} = t_{SDCAL} + t_{CDAC} + t_{GAIN} + t_{EDCAL}$

Table 32.16 Required calibration time (shown as the number of ADCLK and PCLKB cycles)

Parameter	Symbol	Software trigger	Unit
Calibration start delay time	t_{SDCAL}	9 PCLKB + 3 ADCLK*1	Cycle
C-DAC linearity error calculation time	t_{CDAC}	770,048 ADCLK	
Gain error correction calculation time	t_{GAIN}	4,864 ADCLK	
Calibration end delay time	t_{EDCAL}	3 PCLKB + 15 ADCLK*2	
Calibration time	t_{CAL}	Approx. 24.22*3	ms

Note 1. Maximum time from software write to A/D conversion start.
Note 2. Maximum time from calibration end to interrupt output.
Note 3. When ADCLK = PCLKB = 32 MHz.

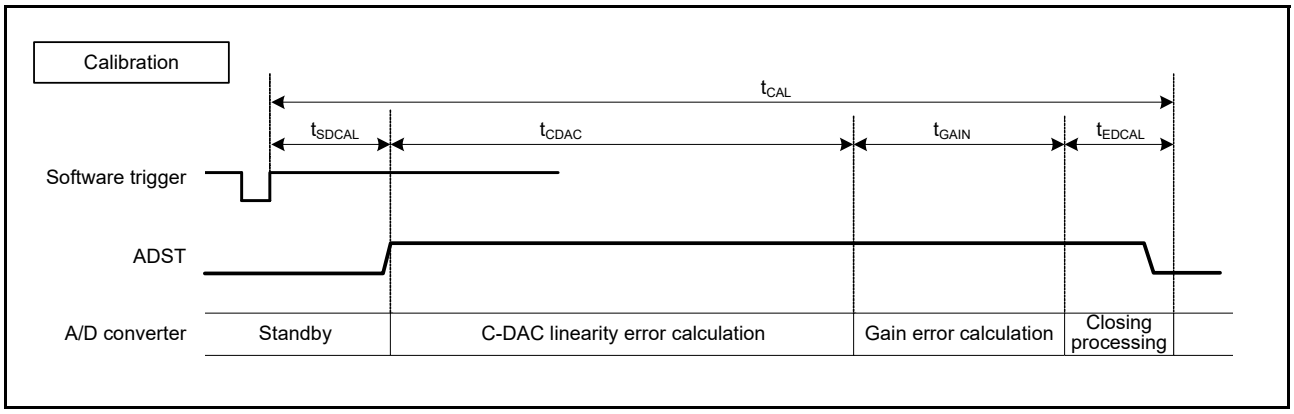


Figure 32.36 Calibration timing

32.4 Interrupt Sources and DTC Transfer Requests

32.4.1 Interrupt Requests

The ADC16 can send scan end interrupt requests (ADC160_ADI and ADC160_GBADI), a compare interrupt requests (ADC160_CMPAI and ADC160_CMPBI), and a calibration end interrupt request (ADC160_ADI) to the CPU.

An ADC160_ADI interrupt is always generated and an ADC160_GBADI interrupt can be generated by setting the ADCSR.GBADIE bit to 1. Similarly, ADC160_CMPAI and ADC160_CMPBI interrupts can be generated by setting the ADCMPCR.CMPAIE and ADCMPCR.CMPBIE bits to 1.

When using a calibration end interrupt request (ADC160_ADI), set the A/D Interrupt Control bits (ADICR.ADIC[1:0]) to 11b. When using an A/D scan end interrupt request and ELC event signal (ADC160_ADI), set the A/D Interrupt Control bits (ADICR.ADIC[1:0]) to 00b.

In addition, the DTC can be started when an ADC160_ADI or an ADC160_GBADI interrupt is generated. Using an ADC160_ADI or an ADC160_GBADI interrupt to allow the DTC to read the converted data enables continuous conversion without a burden on software.

Table 32.17 ADC16 interrupt sources and ELC events

Operation			Interrupt request or ELC event	Interrupt request	DTC activation	ELC event request	Function
Scan mode	Double trigger mode	Compare Function Window A/B					
Single scan mode	Deselect	Deselect	ADC160_ADI*1	✓	✓	✓	ADC160_ADI is generated at the end of single scan
		Select	ADC160_ADI*1	✓	✓	✓	ADC160_ADI is generated at the end of single scan
			ADC160_CMPAI	✓	x	x	ADC160_CMPAI is generated when window A comparison conditions are met
			ADC160_CMPBI	✓	x	x	ADC160_CMPBI is generated when window B comparison conditions are met
			ADC160_WCMPM	x	✓	✓	ADC160_WCMPM is generated when window A/B comparison conditions are met
			ADC160_WCMPUM	x	✓	✓	ADC160_WCMPUM is generated when window A/B comparison conditions are not met
	Select	Deselect	ADC160_ADI*1	✓	✓	✓	ADC160_ADI is generated at the end of scans in the even-numbered times
Continuous scan mode	Deselect	Deselect	ADC160_ADI*1	✓	✓	✓	ADC160_ADI is generated at the end of all the selected channels scan
		Select	ADC160_CMPAI	✓	x	x	ADC160_CMPAI is generated when window A comparison conditions are met
			ADC160_CMPBI	✓	x	x	ADC160_CMPBI is generated when window B comparison conditions are met
Group scan mode	Deselect	Deselect	ADC160_ADI*1	✓	✓	✓	ADC160_ADI is generated at the end of group A scan
			ADC160_GBADI	✓	✓	x	ADC160_GBADI dedicated to group B is generated at the end of group B scan
		Select	ADC160_ADI*1	✓	✓	✓	ADC160_ADI is generated at the end of group A scan
			ADC160_GBADI	✓	✓	x	ADC160_GBADI dedicated to group B is generated at the end of group B scan
			ADC160_CMPAI	✓	x	x	ADC160_CMPAI is generated when window A comparison conditions are met
			ADC160_CMPBI	✓	x	x	ADC160_CMPBI is generated when window B comparison conditions are met
	Select	Deselect	ADC160_ADI*1	✓	✓	✓	ADC160_ADI is generated at the end of Group A scans in the even-numbered times
			ADC160_GBADI	✓	✓	x	ADC160_GBADI dedicated to group B is generated at the end of group B scan
Calibration mode	Deselect	Deselect	ADC160_ADI*2	✓	✓	✓	ADC160_ADI is generated at the end of calibration.

Note 1. Set the A/D Interrupt Control bits (ADICR.ADIC[1:0]) to 00b.
Note 2. Set the A/D Interrupt Control bits (ADICR.ADIC[1:0]) to 11b.

For details on DTC settings, see section 16, Data Transfer Controller (DTC).

32.5 Event Link Function

32.5.1 Event Output to the ELC

The ELC uses the ADC160_ADI interrupt request signal as an event signal, enabling link operation for the preset module. The ADC160_GBADI interrupt and ADC160_CMPAI/ADC160_CMPBI interrupts cannot be used as event signals. For details, see Table 32.17, ADC16 interrupt sources and ELC events.

32.5.2 ADC16 Operation through an Event from the ELC

The ADC16 can start A/D conversion by the preset event signal specified in the ELSRn settings of the ELC as follows:

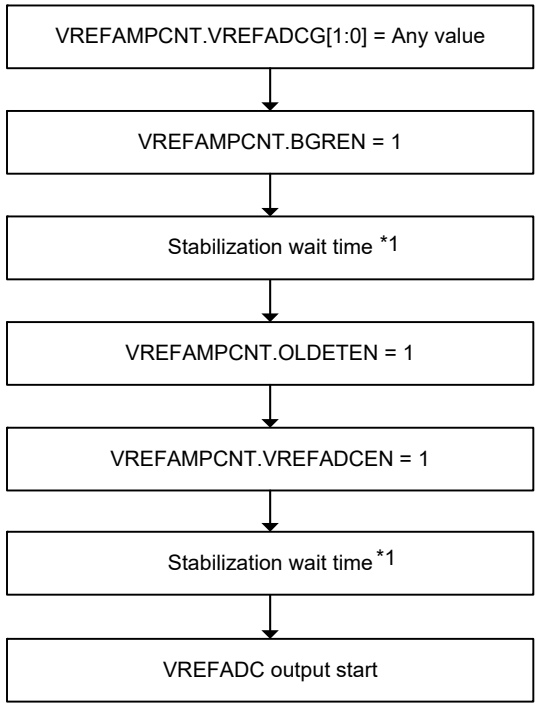
- Select the ELC_AD00 signal in the ELC.ELSR8 register
- Select the ELC_AD01 signal in the ELC.ELSR9 register.

If an ELC_AD00 or ELC_AD01 event occurs during A/D conversion, the event is disabled.

32.6 Selecting Reference Voltage

The ADC16 can select VREFH0 or VREFADC as the high-potential reference voltage, and VREFL0 as the low-potential

reference voltage. Set these reference voltages before starting A/D conversion. For details on reference voltage setting, see section 32.2.33, A/D Dedicated Reference Voltage Circuit Control Register (VREFAMPCNT). Figure 32.37 shows the startup flow of VREFADC.



Note 1. For details of stabilization wait time, see section 47, Electrical Characteristics.

Figure 32.37 VREFADC startup flow

32.7 Usage Notes

32.7.1 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.

32.7.2 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 32.38.

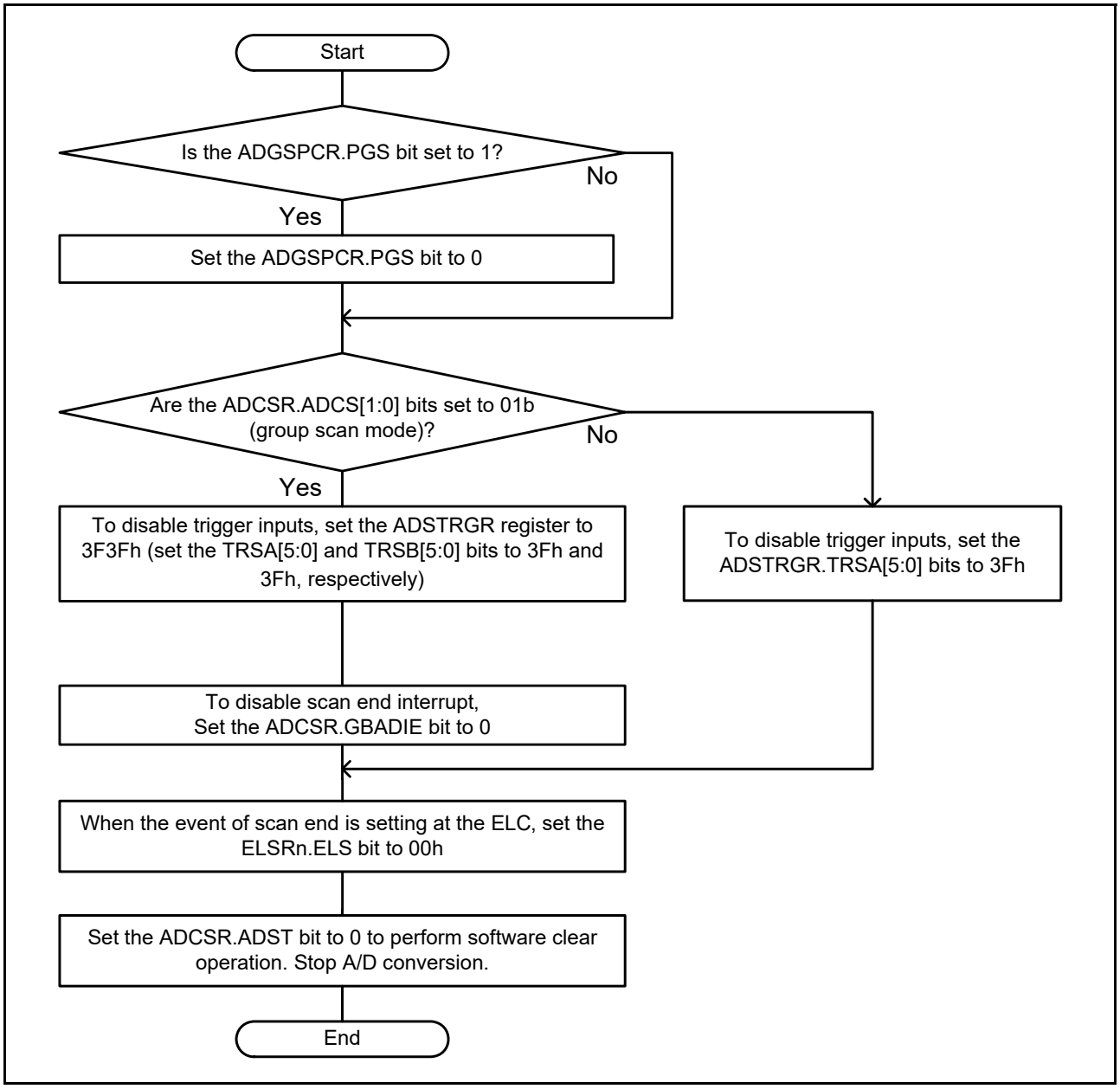


Figure 32.38 Procedure for clearing the ADCSR.ADST bit through software

32.7.3 A/D Conversion Restarting Timing and Termination Timing

A maximum of 6 ADCLK cycles is required for the idle analog unit of the ADC16 to restart on setting the ADCSR.ADST bit to 1. A maximum of 3 ADCLK cycles is required for the operating analog unit of the ADC16 to terminate on setting the ADCSR.ADST bit to 0.

32.7.4 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.

32.7.5 Settings for the Module-Stop State

The Module Stop Control Register D (MSTPCRD) can enable or disable ADC16 operation. The ADC16 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 settling times before starting A/D conversion. For details on the settling time, see section 47, Electrical

Characteristics. For details, see section 11, Low Power Modes.

32.7.6 Restrictions on Entering the Low Power States

Before entering the module-stop state or Software Standby mode, be sure to stop A/D conversion. Set the ADCSR.ADST bit to 0 and secure a period of time until the analog unit of the ADC16 stops. Follow the procedure shown in Figure 32.38 to clear the ADCSR.ADST bit through software. Then, wait for 3 ADCLK clock cycles before entering the module-stop state or Software Standby mode.

32.7.7 Error in Absolute Accuracy when Disconnection Detection Assistance is in Use

Using the disconnection detection assistance leads to an error in absolute accuracy of the ADC16. 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).

Only use disconnection detection assistance after thorough evaluation.

32.7.8 Operating Modes and Status Bits

Initialize or set again individually, if necessary, the voltage values in self-diagnosis, the value of the first scan or second scan in double trigger mode, the data buffer pointer, and status monitor in the compare function.

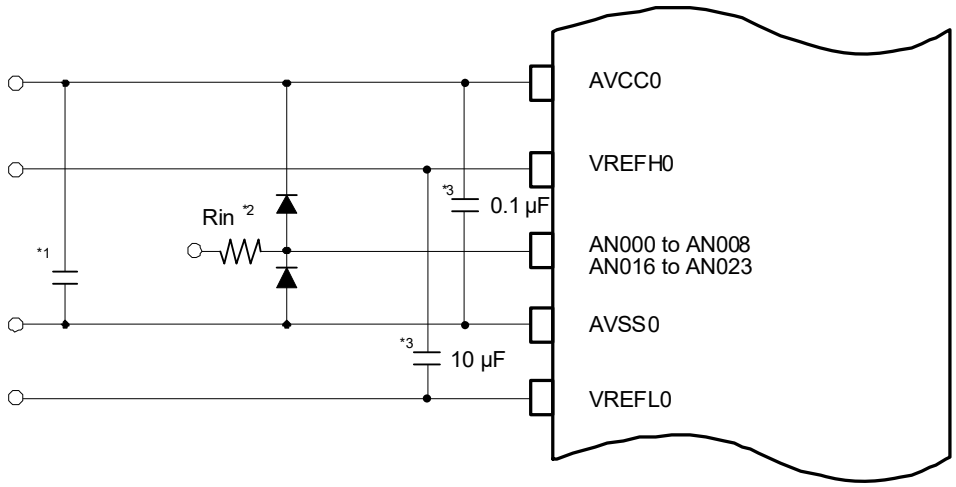
- Select the voltage values in self-diagnosis (ADCER.DIAGVAL[1:0]) after setting ADCER.DIAGLD bit to 1
- The double trigger mode operates as the first scan after setting ADCSR.DBLE bit from 0 to 1
- The status monitor bits (MONCMPA, MONCMPB, and MONCOMB) in the compare function are initialized after setting ADCMPCR.CMPAE bit and ADCMPCR.CMPBE bit to 0.

32.7.9 Notes on Board Design

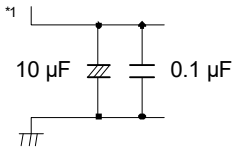
The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or placed near each other. If these rules are not followed, noise can occur on analog signals and A/D conversion accuracy is affected. The analog input pins (AN000 to AN008, AN016 to AN023), 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).

32.7.10 Notes on Noise Reduction

To prevent the analog input pins (AN000 to AN008, AN016 to AN023) 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 (AN000 to AN008, AN016 to AN023). Figure 32.39 shows an example protection circuit for analog inputs when VREFH0 is selected as the high-potential reference voltage for the ADC16.



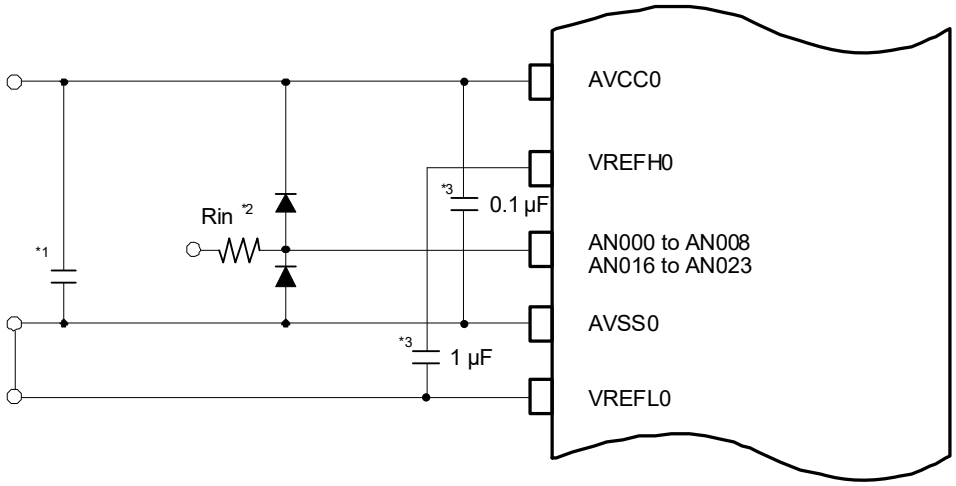
Note 1. The values shown here are reference values.



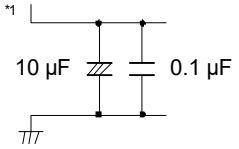
Note 2. Rin: Signal source impedance.
Note 3. Place the capacitors between the power supply pins AVCC0 and AVSS0, VREFH0 and VREFL0 as close to the pins as possible to improve the precision of A /D conversion.

Figure 32.39 Example protection circuit for analog inputs when VREFH0 is selected as the high-potential reference voltage for the ADC16

Figure 32.40 shows an example protection circuit for analog inputs when VREFADC is selected as the high-potential reference voltage for the ADC16.



Note 1. The values shown here are reference values



Note 2. Rin: Signal source impedance.
Note 3. Place the capacitors between the power supply pins AVCC0 and AVSS0, VREFH0 and VREFL0 as close to the pins as possible to improve the precision of A /D conversion.

Figure 32.40 Example protection circuit for analog inputs when VREFADC is selected as the high-potential reference voltage for the ADC16

32.7.11 Port Setting when Using the 16-bit A/D Converter Input

When using the 16-bit A/D converter, do not use PORT0 and PORT5 as general I/O, peripheral functions I/O, and IRQn inputs. Also, when using the normal-precision channels, do not use P100 to P107 as general I/O, peripheral functions I/O, and IRQn inputs.

32.7.12 Relationship between the ADC16, OPAMP, ACMPHS, and SDADC24

Table 32.18 lists the A/D conversion targets that should not be selected as OPAMP, ACMPHS, and SDADC24 inputs during A/D conversion.

Table 32.18 OPAMP, ACMPHS, and SDADC24 pins that should not be selected during A/D conversion

Target of 16-bit A/D conversion	OPAMP	ACMPHS	SDADC24
AN000	AMP0+	IVCMP0	—
AN001	AMP0-	IVREF0	—
AN004	AMP1-	IVREF1	—
AN005	AMP1+	IVCMP1	—
AN006	AMP2-	—	—
AN007	AMP2+	—	—
AN016	—	IVCMP2	ANSD0P
AN017	—	IVREF2	ANSD0N
AN018	—	—	ANSD1P
AN019	—	—	ANSD1N
AN020	—	—	ANSD2P
AN021	—	—	ANSD2N
AN022	—	—	ANSD3P
AN023	—	—	ANSD3N

32.7.13 Notes on Canceling Software Standby Mode

After transitioning from Software Standby mode to Normal mode, wait for settling times before starting A/D conversion. For details on the settling time, see section 47, Electrical Characteristics.

32.7.14 Notes on Calibration Function

The ADC16 is calibrated at the time of shipment.

When using the calibration function, be sure to perform in an environment where the analog block power supply, reference power supply, and ADCLK are stable. If calibration is performed in an unstable environment, A/D conversion accuracy might deteriorate more than at the time of shipment.

33. 24-Bit Sigma-Delta A/D Converter (SDADC24)

33.1 Overview

Table 33.1 lists the 24-bit Sigma-Delta A/D Converter (SDADC24) specifications, and Figure 33.1 shows a block diagram.

Table 33.1 SDADC24 specifications (1 of 2)

Parameter	Specifications
Input channel	Single-ended input mode: Up to 10 channels*1 (external inputs: 8 channels, input from the internal OPAMP: 2 channels) Differential input mode: Up to 5 channels (external inputs: 4 channels, input from the internal OPAMP: 1 channels)
A/D conversion method	Sigma-delta conversion method
Resolution	24 bits
Analog input	• Single-ended input - Conversion is possible with single-ended input on both positive and negative channels. • Differential input.
Oversampling frequency	• Normal A/D conversion mode: 1 MHz • Low-power A/D conversion mode: 0.125 MHz.
Power control	• Power-on/power-off can be selected for VBIAS, PGA, and sigma-delta A/D converter power • Power-on/power-off can be selected for ADBGR, SBIAS, and ADREG power • VREF reference voltage (SBIAS/VREFI) can be set (step: 0.2 V, range: 0.8 to 2.4 V) Note: 2.4 V can be set in external VREF (VREFI) mode only • Sensor reference voltage (SBIAS) can be activated independently.
Programmable gain instrumentation amplifier (PGA)	• The gain of an instrumentation amplifier can be set for each channel. (×1 to ×32 can be set by a combination of G_{SET1} and G_{SET2}.) - G_{SET1} range of the previous-stage amplifier: 1, 2, 3, 4, 8 - G_{SET2} range of the next-stage amplifier: 1, 2, 4, 8 • The offset voltage can be adjusted for each channel by using a D/A converter connected to the next-stage amplifier - Offset voltage adjustment (-164 to +164 mV, 31 levels: 5 bits) • PGA offset can be measured as self-diagnosis • Disconnection detection assist: possible on both positive and negative sides in single-ended input mode.
Data registers	• One A/D conversion result register and one A/D conversion average value register: - The channel number that corresponds to an A/D conversion result can be checked with a special register - An overflow flag is provided for A/D conversion results • Differential input mode: code is 2's complement • Single-ended input mode: straight binary • Reverse output can be selected for the conversion results of the single-ended negative channel.
Operation clock	• The 24-bit sigma-delta A/D converter reference clock is generated from the peripheral clock output by the clock generation circuit according to the SDADC24 operation mode. 1/1, 1/2, 1/3, 1/4, 1/5, 1/6, 1/8, 1/12, or 1/16 can be selected • The SDADC24 reference clock/oversampling clock changes according to the mode as follows: - Normal A/D conversion mode: 4 MHz/1 MHz - Low-power A/D conversion mode: 500 kHz/125 kHz. Note: When the A/D converter is used in low-power A/D conversion mode, the specified frequency of the SDADC24 reference clock is divided by 8 by using an internal frequency divider.
Conversion start condition	• Software trigger • Hardware trigger (ELC_SDADC24).
Operation mode	• Continuous scan mode • Single scan mode • One-shot operation.
Oversampling rate	• 64, 128, 256, 512, 1024, or 2048 can be selected • Can be set for each channel.

Table 33.1 SDADC24 specifications (2 of 2)

Parameter	Specifications
A/D conversion count	• The A/D conversion count can be set and the A/D conversion count specification mode can be selected for 1 AUTOSCAN cycle. 1. For register setting values, specify 1 to 8032 (N) $N = 32 \times (2^n - 1) + m \times 2^n$ (m and n correspond to values set in the PGAC0 to PGAC4 registers. m: b16 to b20, n: b21 to b23. If N = 00h, one-shot operation that stops when one A/D conversion ends is set.) 2. For register setting values, specify 1 to 255 (N) linearly (N corresponds to the value set in the PGAC0 to PGAC4 registers. N: b16 to b23. If N = 00h, one-shot operation that stops each time A/D conversion ends is set.) The A/D conversion count can be set for each channel.
Averaging of A/D conversion results	• The averaging operation can be selected: 1. Do not perform averaging 2. Perform averaging, and trigger an SDADC24 conversion end interrupt each time A/D conversion occurs 3. Perform averaging, and trigger an SDADC24 conversion end interrupt each time the average value is updated. • The number of data items to be averaged can be selected as 8, 16, 32, or 64. Note: The number of data items to be averaged can be set for each channel.
Interrupt cause	• A/D conversion end interrupt (SDADC_ADI) • A/D automatic scan completion interrupt (SDADC_SCANEND) • Calibration completion interrupt (SDADC_CALIEND).
SDADC24 operation	• A/D conversion of each input channel is executed on a round-robin basis • A/D conversion of a specific channel can be stopped using the permission/stop register of each channel.
Digital filter	• Down sampling of A/D conversion results is performed using the SINC3 digital filter
SDADC24 calibration	• Analog characteristics can be corrected by calibration (gain error and offset error)

Note 1. The number of channels that can simultaneously perform A/D conversion is up to 5 channels.

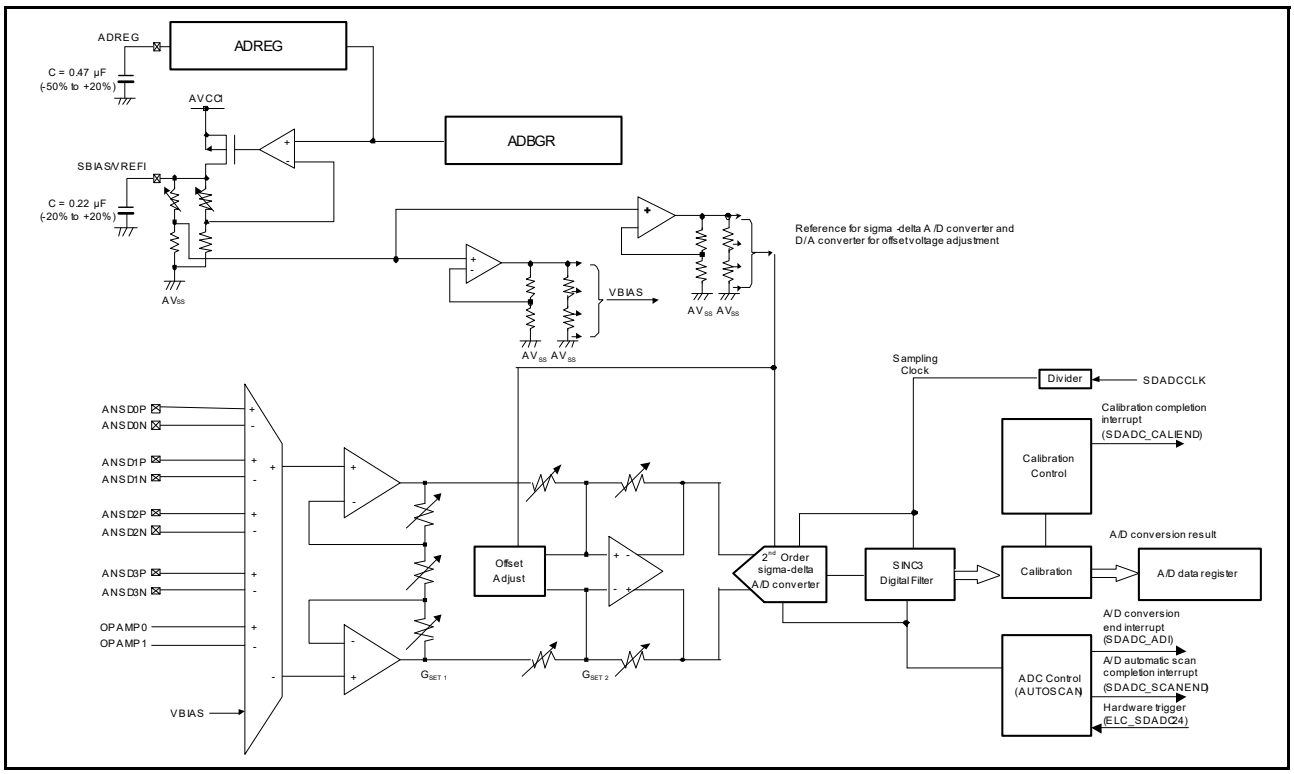


Figure 33.1 SDADC24 block diagram

Table 33.2 lists the SDADC24 pin configuration.

Table 33.2 SDADC24 I/O pins

Pin name	I/O	Function
AVCC1	Input	Analog block power supply pin
AVSS1	Input	Analog block power supply ground pin
ADREG	I/O	Power supply pins for PGA and the sigma-delta A/D converter
SBIAS/VREFI	Input	External reference voltage input pin (VREFI)
	Output	Sensor Power supply pin (SBIAS)
ANSD0P to ANSD3P, ANSD0N to ANSD3N	Input	Analog input pins

33.2 Register Descriptions

33.2.1 Startup Control Register 1 (STC1)

Address(es): SDADC24.STC1 4009 C000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VREFSEL	—	—	—	VSBIAS[3:0]				SDADLPM	—	—	—	CLKDIV[3:0]			
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	CLKDIV[3:0]*1	SDADC24 reference clock division select	b3 b2 b1 b0 0 0 0 0 : SDADCCLK (no division) 0 0 0 1 : SDADCCLK/2 (1/2) 0 0 1 0 : SDADCCLK/3 (1/3) 0 0 1 1 : SDADCCLK/4 (1/4) 0 1 0 0 : SDADCCLK/5 (1/5) 0 1 0 1 : SDADCCLK/6 (1/6) 0 1 1 0 : SDADCCLK/8 (1/8) 0 1 1 1 : SDADCCLK/12 (1/12) 1 0 0 0 : SDADCCLK/16 (1/16). 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	SDADLPM*1	A/D conversion operation mode select	0: Normal A/D conversion mode SDADC24 reference clock: 4 MHz Oversampling clock: 1 MHz 1: Low-power A/D conversion mode SDADC24 reference clock: 500 kHz Oversampling clock: 125 kHz. (1/8 of the clock in normal A/D conversion mode)	R/W
b11 to b8	VSBIAS[3:0]	Reference voltage select	b11 b10 b9 b8 0 0 0 0 : 0.8 V 0 0 0 1 : 1.0 V 0 0 1 0 : 1.2 V 0 0 1 1 : 1.4 V 0 1 0 0 : 1.6 V 0 1 0 1 : 1.8 V 0 1 1 0 : 2.0 V 0 1 1 1 : 2.2 V 1 1 1 1 : 2.4 V (this voltage can be set only if VREFSEL = 1). Other settings are prohibited.	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	VREFSEL	VREF mode select	0: Internal VREF mode 1: External VREF mode.	R/W

Note: Only set the STC1 register when ADC1.SDADTMD, ADC2.SDADST, and CLBSSR.CLBSS bits are 0.
Note 1. These bits must be set while the SDADCCLK clock is stopped (SYSTEM.SDADCCKCR.SDADCCKEN = 0) and the ADBGR is powered off (STC2.BGRPON = 0).

CLKDIV[3:0] bits (SDADC24 reference clock division select)

The SDADC24 reference clock is generated by dividing the SDADC24 clock (SDADCCLK) according to the CLKDIV[3:0] bits. These bits select the division ratio of the SDADCCLK. No division, 1/2, 1/3, 1/4, 1/5, 1/6, 1/8, 1/12, or 1/16 can be selected.

Set the CLKDIV[3:0] bits so that the SDADC24 reference clock is output at 4 MHz. When the A/D converter is used in low-power A/D conversion mode, the specified frequency of the SDADC24 reference clock is automatically divided by 8 (500 kHz) by using an internal frequency divider.

SDADLPM bit (A/D conversion operation mode select)

The SDADLPM bit selects either normal A/D conversion mode or low-power A/D conversion mode.

VSBIAS[3:0] bits (Reference voltage select)

For internal VREF mode (VREFSEL = 0), select the output voltage value for sensor reference voltage.

For external VREF mode (VREFSEL = 1), select the reference voltage input value to be input externally.

Note: 2.4 V can only be selected in external VREF mode (VREFSEL = 1).

VREFSEL bit (VREF mode select)

The VREFSEL bit controls the VREF mode of the reference voltage for the sensor.

33.2.2 Startup Control Register 2 (STC2)

Address(es): SDADC24.STC2 4009 C004h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	ADFPWDS	ADCPON	BGRPON
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	BGRPON	BGR part power control	0: Turn the power of ADBGR, SBIAS, and ADREG off 1: Turn the power of ADBGR, SBIAS, and ADREG. on.	R/W
b1	ADCPON	ADC reference supply part power control	0: Turn the power of VBIAS, PGA and sigma-delta A/D converter off 1: Turn the power of VBIAS, PGA and sigma-delta A/D converter on.	R/W
b2	ADFPWDS	ADREG forced power-down mode	0: The power of ADREG is controlled by the BGRPON setting 1: The power of only ADREG is turned off regardless of the BGRPON setting.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Only set the STC2 register when the ADC1.SDADTMD, ADC2.SDADST, and CLBSSR.CLBSS bits are 0.

BGRPON bit (BGR part power control)

The BGRPON bit controls power-on and power-off for the ADBGR, SBIAS, and ADREG power supply.

ADCPON bit (ADC reference supply part power control)

The ADCPON bit controls power-on or power-off for the VBIAS, PGA and sigma-delta A/D converter power supply.

ADFPWDS bit (ADREG forced power-down mode)

If the ADFPWDS bit is set to 1, the power of only ADREG is turned off. Setting the ADFPWDS and ADCPON bits to 1 at the same time is prohibited. For details of setting the ADFPWDS and BGRPON bits, and for the power status, see section 33.3.3.2, SBIAS independent operation.

33.2.3 Input Multiplexer n Setting Register (PGACn) (n = 0 to 4)

Address(es): SDADC24.PGAC0 4009 C008h, SDADC24.PGAC1 4009 C00Ch, SDADC24.PGAC2 4009 C010h, SDADC24.PGAC3 4009 C014h, SDADC24.PGAC4 4009 C018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PGAASN	PGACVE	—	PGAREV	PGAAVE[1:0]		PGAAVN[1:0]		PGACTN[2:0]			PGACTM[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PGASEL	PGAPOL	—	PGAOFS[4:0]					PGAOSR[2:0]			PGAGC[4:0]				
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
b4 to b0	PGAGC[4:0]	Gain selection of a programmable gain instrumentation amplifier	b4 b3 b2 b1 b0 : (G_{SET1}, G_{SET2}, G_{TOTAL}) 0 0 0 0 0 : (1, 1, 1) *1 0 0 1 0 0 : (2, 1, 2) 0 1 0 0 0 : (3, 1, 3) 0 1 1 0 0 : (4, 1, 4) 1 0 0 0 0 : (8, 1, 8) 0 0 0 0 1 : (1, 2, 2) 0 0 1 0 1 : (2, 2, 4) 0 1 0 0 1 : (3, 2, 6) 0 1 1 0 1 : (4, 2, 8) 1 0 0 0 1 : (8, 2, 16) 0 0 0 1 0 : (1, 4, 4) 0 0 1 1 0 : (2, 4, 8) 0 1 0 1 0 : (3, 4, 12) 0 1 1 1 0 : (4, 4, 16) 1 0 0 1 0 : (8, 4, 32) 0 0 0 1 1 : (1, 8, 8) 0 0 1 1 1 : (2, 8, 16) 0 1 0 1 1 : (3, 8, 24) 0 1 1 1 1 : (4, 8, 32). Other settings are prohibited.	R/W
b7 to b5	PGAOSR[2:0]	Oversampling ratio select	b7 b6 b5 0 0 0 : 64 0 0 1 : 128 0 1 0 : 256 *1 0 1 1 : 512 1 0 0 : 1024 1 0 1 : 2048. Other settings are prohibited.	R/W
b12 to b8	PGAOFS[4:0]	Offset voltage select	b12 b11 b10 b9 b8 1 0 0 0 1 : -164.06 / G_{SET1} 1 0 0 1 0 : -153.13 / G_{SET1} : 0 0 0 0 0 : 0 *1 : 0 1 1 0 1 : +142.19 / G_{SET1} 0 1 1 1 0 : +153.13 / G_{SET1} 0 1 1 1 1 : +164.06 / G_{SET1}.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	PGAPOL	Polarity select	0: Positive-side single-ended input 1: Negative-side single-ended input.	R/W
b15	PGASEL	Analog Channel Input Mode Select	0: Differential input mode 1: Single-ended input mode.*1	R/W

Bit	Symbol	Bit name	Description	R/W
b20 to b16	PGACTM[4:0]	Coefficient (m) selection of the A/D conversion count (N) in AUTOSCAN	Bits for setting the coefficient (m) of the A/D conversion count (N) in AUTOSCAN. b20 b19 b18 b17 b16 0 0 0 0 0 : 0 0 0 0 0 1 : 1 0 0 0 1 0 : 2 : 1 0 0 0 0 : 16 : 1 1 1 0 1 : 29 1 1 1 1 0 : 30 1 1 1 1 1 : 31.	R/W
b23 to b21	PGACTN[2:0]	Coefficient (n) selection of the A/D conversion count (N) in AUTOSCAN	Bits for setting the coefficient (n) of the A/D conversion count (N) in AUTOSCAN. b23 b22 b21 0 0 0 : 0 0 0 1 : 1 0 1 0 : 2 0 1 1 : 3 1 0 0 : 4 1 0 1 : 5 1 1 0 : 6 1 1 1 : 7.	R/W
b25, b24	PGAAVN[1:0]	Selection of the number of data to be averaged	b25 b24 0 0 : 8 0 1 : 16 1 0 : 32 1 1 : 64.	R/W
b27, b26	PGAAVE[1:0]	Selection of averaging processing	b27 b26 0 0 : Do not average the A/D conversion results 0 1 : Do not average the A/D conversion results 1 0 : Average the A/D conversion results and generates SDADC_ADI each time an A/D conversion occurs 1 1 : Perform averaging and generate SDADC_ADI each time the average value is output (A/D conversion is performed N times).	R/W
b28	PGAREV	Single-Ended Input A/D Converted Data Inversion Select	This bit selects data display of A/D converted data for negative side of single-ended inputs. 0: Do not invert the conversion result data 1: Invert the conversion result data.	R/W
b29	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30	PGACVE	Calibration enable	0: Do not calculate the calibration correction factor 1: Calculate the calibration correction factor.	R/W
b31	PGAASN	Selection of the mode for specifying the number of A/D conversions in ADSCAN	0: Specify 1 to 8,032 times using the value set in the PGACTN[2:0] and PGACTM[4:0] bits 1: Specify 1 to 255 times linearly using the value set in the PGACTN[2:0] and PGACTM[4:0] bits.	R/W

Note: Only set the PGACn register when ADC1.SDADTMD, ADC2.SDADST, and CLBSSR.CLBSS bits are 0.
Note 1. If single-ended input is used, only d_{OFR} = 0 mV, G_{SET1} = 1, G_{SET2} = 1 and OSR = 256 are supported.

PGAGC[4:0] bits (Gain selection of a programmable gain instrumentation amplifier)

This bit is used to set the gain of the programmable gain instrumentation amplifier.

For details about the relationship between G_{SET1}, G_{SET2}, and G_{TOTAL} and the internal amplitude of PGA, see section 33.3.6, Programmable Gain Instrumentation Amplifier (PGA).

PGAOSR[2:0] bits (Oversampling ratio select)

The PGAOSR[2:0] bits set the data rate (output frequency of the A/D conversion result) as 64, 128, 256, 512, 1024, or 2048.

PGAOFS[4:0] bits (Offset voltage select)

The PGAOFS[4:0] bits set the offset voltage.

Table 33.3 shows the offset voltage d_{OFR} (value calculated by converting the output voltage of the D/A converter for offset voltage adjustment into input).

Table 33.3 Setting of d_{OFR} by PGAOFS[4:0] bits

PGAOFS[4:0]						
[4]	[3]	[2]	[1]	[0]	d_{OFR}[mV]	Note
1	0	0	0	0	-	Setting prohibited
1	0	0	0	1	-164.06 / G_{SET1}	-
1	0	0	1	0	-153.13 / G_{SET1}	-
1	0	0	1	1	-142.19 / G_{SET1}	-
1	0	1	0	0	-131.25 / G_{SET1}	-
1	0	1	0	1	-120.31 / G_{SET1}	-
1	0	1	1	0	-109.38 / G_{SET1}	-
1	0	1	1	1	-98.44 / G_{SET1}	-
1	1	0	0	0	-87.50 / G_{SET1}	-
1	1	0	0	1	-76.56 / G_{SET1}	-
1	1	0	1	0	-65.63 / G_{SET1}	-
1	1	0	1	1	-54.69 / G_{SET1}	-
1	1	1	0	0	-43.75 / G_{SET1}	-
1	1	1	0	1	-32.81 / G_{SET1}	-
1	1	1	1	0	-21.88 / G_{SET1}	-
1	1	1	1	1	-10.94 / G_{SET1}	-
0	0	0	0	0	0.00 / G_{SET1}	Default
0	0	0	0	1	10.94 / G_{SET1}	-
0	0	0	1	0	21.88 / G_{SET1}	-
0	0	0	1	1	32.81 / G_{SET1}	-
0	0	1	0	0	43.75 / G_{SET1}	-
0	0	1	0	1	54.69 / G_{SET1}	-
0	0	1	1	0	65.63 / G_{SET1}	-
0	0	1	1	1	76.56 / G_{SET1}	-
0	1	0	0	0	87.50 / G_{SET1}	-
0	1	0	0	1	98.44 / G_{SET1}	-
0	1	0	1	0	109.38 / G_{SET1}	-
0	1	0	1	1	120.31 / G_{SET1}	-
0	1	1	0	0	131.25 / G_{SET1}	-
0	1	1	0	1	142.19 / G_{SET1}	-
0	1	1	1	0	153.13 / G_{SET1}	-
0	1	1	1	1	164.06 / G_{SET1}	-

PGAPOL bit (Polarity select)

The PGAPOL bit is used to set whether to input a signal from the positive-side channel or negative-side channel in single-ended mode.

PGASEL bit (Analog Channel Input Mode Select)

The PGASEL bit is used to set whether to use differential input mode or single-ended input mode.

PGACTM[4:0] and PGACTN[2:0] bits (Coefficient (m, n) selection of the A/D conversion count (N) in AUTOSCAN)

The PGACTM[4:0] bits are used to set the coefficient (m) that determines the A/D conversion count (N) in 1

AUTOSCAN cycle. Zero, 1, 2, ..., 29, 30, or 31 can be selected.

The PGACTN[2:0] bits are used to set the coefficient (n) that determines the A/D conversion count (N) in 1 AUTOSCAN cycle. Zero, 1, 2, 3, 4, 5, 6, or 7 can be selected.

When the PGACTN[2:0] = 0 and the PGACTM[4:0] = 0, one-shot operation is performed.

The following shows the expression that indicates the relationships between N, m, and n.

When PGAASN = 0:

$N = 32 \times (2^n - 1) + m \times 2^n$

(m and n correspond to the values set for this register).

A 256-level A/D conversion count can be selected by a combination of m and n. The following shows the correlation between the number of levels and A/D conversion count.

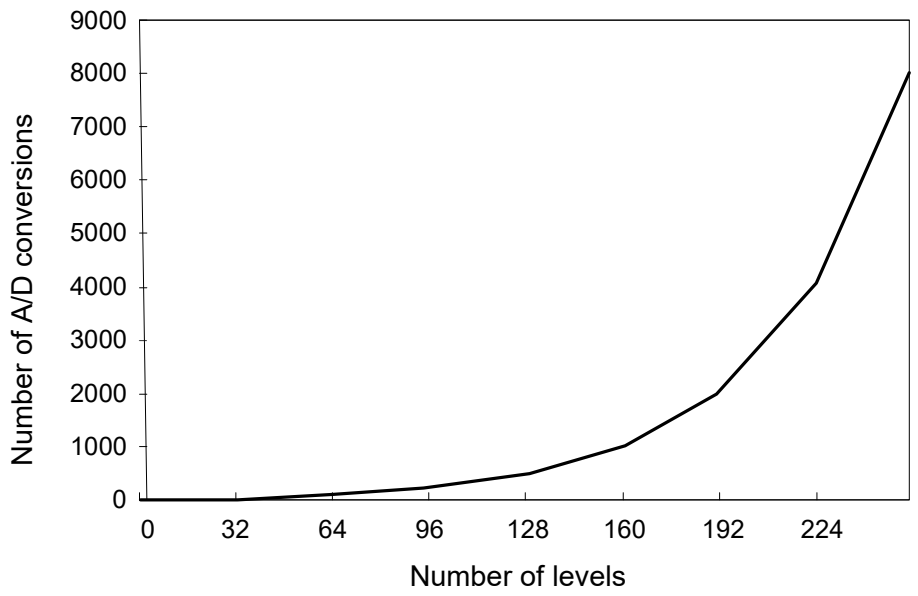


Figure 33.2 Correlation between the number of levels and A/D conversion count

When PGAASN = 1:

$N = (32 \times n) + m$

(m and n correspond to the values set for this register).

PGAAVN[1:0] bits (Selection of the number of data to be averaged)

The PGAAVN[1:0] bits are used to select the number of data items to be averaged. Eight, 16, 32, or 64 can be selected.

If averaging is to be performed, the total number of A/D conversions is calculated as follows:

A/D conversion count is determined by (count set for PGACTN[2:0] and PGACTM[4:0] bits) × (count set for the PGAAVN[1:0] bits).

PGAAVE[1:0] bits (Selection of averaging processing)

The PGAAVE[1:0] bits are used to select the averaging operation of the A/D conversion result. You can select whether to generate an interrupt for each A/D conversion or for each average value output.

PGAREV bit (Single-Ended Input A/D Converted Data Inversion Select)

The PGAREV bit is used to select whether to reverse the A/D conversion results of single-ended input (negative side). For channels that are not in negative-side single-ended input mode, the setting of this bit is ignored.

PGACVE bit (Calibration enable)

The PGACVE bit is used to set whether to calculate the calibration correction factor for each input channel during calibration. For channels in which the calibration correction factor is not calculated, calibration cannot start and the values of the offset error correction factor registers and gain error correction factor registers are not updated. Also, regardless of the value set for this bit, the correction factors are not calculated for the channels set in single-ended input mode.

For the relationship between each register and calibration correction factor calculation, see Table 33.10.

PGAASN bit (Selection of the mode for specifying the number of A/D conversions in ADSCAN)

The PGAASN bit is used to select the A/D conversion count (1 to 8,032 or 1 to 255 (linear)) in 1 AUTOSCAN cycle.

33.2.4 Sigma-Delta A/D Converter Control Register 1 (ADC1)

Address(es): SDADC24.ADC1 4009 C01Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	PGASLFT	—	—	PGADISC	PGADISA
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
	—	—	—	SDADBMP[4:0]					—	—	—	SDADTMD	—	—	—	SDADSCM
Value after 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	SDADSCM	Selection of auto scan mode	0: Continuous scan mode 1: Single scan mode.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SDADTMD	Selection of A/D conversion trigger signal	0: Software trigger (conversion is started by a write to SFR) 1: Hardware trigger (conversion is started in synchronization with the event signal selected by ELC_SDADC24).	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12 to b8	SDADBMP[4:0]	A/D conversion control of the signal from input multiplexer	0: Allow A/D conversion 1: Stop A/D conversion SDADBMP[0] = PGAC0 SDADBMP[1] = PGAC1 SDADBMP[2] = PGAC2 SDADBMP[3] = PGAC3 SDADBMP[4] = PGAC4.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	PGADISA	Control of disconnection detection	0: Normal operation 1: State of disconnection detection.	R/W
b17	PGADISC	Disconnection detection assist setting	0: Discharge 1: Precharge.	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	PGASLFT	PGA offset self-diagnosis enable	0: PGA offset self-diagnosis disabled 1: PGA offset self-diagnosis enabled.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Only set bits in the ADC1 register except the SDADTMD bit, when ADC1.SDADTMD, ADC2.SDADST, and CLBSSR.CLBSS bits are 0. Only set the SDADTMD bit when ADC2.SDADST and CLBSSR.CLBSS bits are 0.

SDADSCM bit (Selection of auto scan mode)

The SDADSCM bit is used to select the automatic scan mode. Either continuous scan mode or single scan mode can be

selected. When the PGACn.PGACTN[2:0] = 000b and the PGACn.PGACTM[4:0] = 00000b, one-shot mode takes precedence (n = 0 to 4).

SDADTMD bit (Selection of A/D conversion trigger signal)

The SDADTMD bit is used to select the A/D conversion start signal. Either software trigger or hardware trigger can be selected. If hardware trigger is selected, the product always operates in single scan mode. If the SDADTMD bit is changed from 1 to 0 during A/D conversion using a hardware trigger, A/D conversion stops as the conversion result is not guaranteed.

During the A/D conversion period:

- When software trigger is input:
  - From setting the ADC2.SDADST = 1 to A/D conversion end interrupt output
  - From setting the ADC2.SDADST = 1 to A/D conversion stop with the ADC2.SDADST = 0.
- When hardware trigger is input:
  - From hardware trigger is input to A/D conversion end interrupt output.

SDADBMP[4:0] bits (A/D conversion control of the signal from input multiplexer)

The SDADBMP[4:0] bits are used to allow or stop A/D conversion of signals from the input multiplexers for the respective bits.

Table 33.4 shows the SDADBMP[n] bit and PGACn register associated with each input channel.

Table 33.4 SDADBMP[n] bit and PGACn register associated with each input channel

Analog input pin		A/D conversion control bit of signal from the input multiplexer	Input Multiplexer n Setting register
Positive side	Negative side		
ANSD0P	ANSD0N	SDADBMP[0]	PGAC0
ANSD1P	ANSD1N	SDADBMP[1]	PGAC1
ANSD2P	ANSD2N	SDADBMP[2]	PGAC2
ANSD3P	ANSD3N	SDADBMP[3]	PGAC3
Internal OPAMP 0 (AMP0O)	Internal OPAMP 1 (AMP1O)	SDADBMP[4]	PGAC4

PGADISA bit (Control of disconnection detection)

The PGADISA bit is used to select whether to enable the disconnection detection assist mode.

Note: Disconnection detection of the input multiplexer 4 (internal OPAMP) cannot be performed. When setting SDADBMP[4] = 1, set PGADISA = 0.

PGADISC bit (Disconnection detection assist setting)

The PGADISC bit is used to select the disconnection detection charge mode (precharge/discharge).

PGASLFT bit (PGA offset self-diagnosis enable)

The PGASLFT bit is used to set the PGA offset self-diagnosis mode for input channel. For details, see Figure 33.26.

33.2.5 Sigma-Delta A/D Converter Control Register 2 (ADC2)

Address(es): SDADC24.ADC2 4009 C020h

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

Bit	Symbol	Bit name	Description	R/W
b0	SDADST	Control of A/D conversion	0: Stop A/D conversion 1: Start A/D conversion.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Only set the ADC2 register when ADC1.SDADTMD and CLBSSR.CLBSS bits are 0.

SDADST bit (Control of A/D conversion)

The SDADST bit controls whether to start or stop A/D conversion (AUTOSCAN) when ADC1.SDADTMD = 0 (software trigger).

This bit does not control A/D conversion when the ADC1.SDADTMD = 1 (hardware trigger).

Note: When SDADC24-related registers are accessed after the SDADST bit is written, register access must be made after waiting at least 2 SDADC24 reference clock cycles. Normal A/D conversion mode is 4 MHz, low-power A/D conversion mode is 500 kHz.

33.2.6 Sigma-Delta A/D Converter Conversion Result Register (ADCR)

Address(es): SDADC24.ADCR 4009 C024h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	SDADCRC[2:0]			SDADCRS	SDADCRD[23:16]							
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
	SDADCRD[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
b23 to b0	SDADCRD[23:0]	A/D Converter Conversion Result	The 24-bit A/D conversion result is displayed	R
b24	SDADCRS	Status of an A/D Conversion Result	0: Normal status (within the range) 1: Overflow occurred.*1	R
b27 to b25	SDADCRC[2:0]	Channel number for an A/D Conversion Result	0 0 0: Reset value (conversion result is invalid) 0 0 1: Input multiplexer 0 (ANSD0P / ANSD0N) 0 1 0: Input multiplexer 1 (ANSD1P / ANSD1N) 0 1 1: Input multiplexer 2 (ANSD2P / ANSD2N) 1 0 0: Input multiplexer 3 (ANSD3P / ANSD3N) 1 0 1: Input multiplexer 4 (AMP0O / AMP1O).	R
b31 to b28	—	Reserved	These bits are read as 0	R

Note 1. The maximum or minimum value of the register becomes the conversion result.

SDADCRD[23:0] bits (A/D Converter Conversion Result)

The SDADCRD[23:0] bits are used to display the value of the A/D conversion result.

SDADCRS bit (Status of an A/D Conversion Result)

The SDADCRS bit is used to display whether an overflow occurred as a result of A/D conversion. The maximum or minimum value of the register becomes the conversion result.

SDADCRC[2:0] bits (Channel number for an A/D Conversion Result)

The SDADCRC[2:0] bits are used to display the channel number of the input multiplexer for an A/D conversion result.

Note: The channel number for the A/D Conversion Result bits (SDADCRC[2:0]), the status of the A/D Conversion Result bit (SDADCRS), and the A/D Converter Conversion Result bits (SDADCRD[23:0]) are not updated during internal and external calibration. For details, see section 33.3.10, Calibration function.

33.2.7 Sigma-Delta A/D Converter Average Value Register (ADAR)

Address(es): SDADC24.ADAR 4009 C028h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	SDADMVC[2:0]			SDAD MVS	SDADMVD[23:16]							
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
	SDADMVD[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
b23 to b0	SDADMVD[23:0]	A/D Converter Conversion Result	The 24-bit A/D average value is displayed*1	R
b24	SDADMVS	Status of an A/D Conversion Result	0: Normal status (within the range) 1: Overflow occurred.*2	R
b27 to b25	SDADMVC[2:0]	Channel number for an A/D Conversion Result	0 0 0: Reset value (conversion result is invalid) 0 0 1: Input multiplexer 0 (ANSD0P / ANSD0N) 0 1 0: Input multiplexer 1 (ANSD1P / ANSD1N) 0 1 1: Input multiplexer 2 (ANSD2P / ANSD2N) 1 0 0: Input multiplexer 3 (ANSD3P / ANSD3N) 1 0 1: Input multiplexer 4 (AMP0O / AMP1O).	R
b31 to b28	—	Reserved	These bits are read as 0	R

Note 1. If the averaging function is not used, SDADMVD[23:0] = 000000h.
Note 2. The maximum or minimum value of the register becomes the conversion result.

SDADMVD[23:0] bits (A/D Converter Conversion Result)

The SDADMVD[23:0] bits are used to display the 24-bit average value.

SDADMVS bit (Status of an A/D Conversion Result)

The SDADMVS bit is used to display whether an overflow occurred as a result of A/D conversion. The maximum or minimum value of the register becomes the conversion result.

SDADMVC[2:0] bits (Channel number for an A/D Conversion Result)

The SDADMVC[2:0] bits are used to display the channel number of the input multiplexer for an A/D conversion result.

33.2.8 Calibration Control Register (CLBC)

Address(es): SDADC24.CLBC 4009 C030h

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

Bit	Symbol	Bit name	Description	R/W
b1, b0	CLBMD[1:0]	Calibration mode	b1 b0 0 0: Internal calibration mode 0 1: External offset calibration mode 1 0: External gain calibration mode. Other settings are prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Only set the CLBC register when the CLBSSR.CLBSS bit is 0.

CLBMD[1:0] bits (Calibration mode)

Internal calibration mode is a mode in which internal offset calibration is first performed, and then internal gain calibration is performed. When internal gain calibration completes, a calibration completion interrupt is output.

To calculate the calibration correction factors for multiple channels using the PGACn.PGACVE bit (n = 0 to 4), after internal calibration is performed for all specified channels, a calibration completion interrupt is output.

[Calibration completion condition]

After 1 is written to the CLBSTR.CLBST bit, calibration completes when both internal offset calibration and internal gain calibration complete for all channels (except single-ended input channels) specified by the PGACn.PGACVE bit (n = 0 to 4).

External offset calibration mode is a mode in which external offset calibration is performed. When external offset calibration completes, a calibration completion interrupt is output.

[Calibration completion condition]:

After 1 is written to the CLBSTR.CLBST bit, calibration completes when external offset calibration completes.

External gain calibration mode is a mode in which external gain calibration is performed. A calibration completion interrupt is output when external gain calibration completes.

[Calibration completion condition]:

After 1 is written to the CLBSTR.CLBST bit, calibration completes when external gain calibration completes.

33.2.9 Calibration Start Control Register (CLBSTR)

Address(es): SDADC24.CLBSTR 4009 C034h

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

Bit	Symbol	Bit name	Description	R/W
b0	CLBST	Calibration start control	0: Disable writing 1: Start calibration.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Only set the CLBSTR register when ADC1.SDADTMD, ADC2.SDADST, and CLBSSR.CLBSS bits are 0.

CLBST bit (Calibration start control)

When calibration and A/D conversion are not running, calibration starts when 1 is written to the CLBST bit. Always read 0 from this bit.

Note: Do not set the CLBST bit to 1 when all of the PGACn.PGACVE bits (n = 0 to 4) are set to 0.
In this case, calibration does not start and the CLBSSR.CLBSS bit remains 0. In addition, no calibration completion interrupt is output. For the gain error correction value and offset error correction value, the previous values are retained.

If an A/D conversion request is made by a software trigger (ADC2.SDADST = 1) or a hardware trigger while calibration is running (CLBSSR.CLBSS = 1), calibration takes priority and A/D conversion does not start. A/D conversion requests are not held during calibration and calibration does not start during A/D conversion.

33.2.10 Calibration Status Register (CLBSSR)

Address(es): SDADC24.CLBSSR 4009 C03Ch

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

Bit	Symbol	Bit name	Description	R/W
b0	CLBSS	Calibration status	0: Calibration is not running 1: Calibration is running.	R
b7 to b1	—	Reserved	These bits are read as 0	R

CLBSS bit (Calibration status)

The CLBSS bit is used to indicate the execution status of calibration. This bit is set to 1 when calibration is running and is set to 0 when calibration finishes.

33.2.11 Calibration Control Protection Release Register (CLBPR)

Address(es): SDADC24.CLBPR 4009 C040h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CLBB0WI	CLBPRO
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CLBPRO	Calibration control protection	0: Read and write to the GCVLRn and OCVLRn (n = 0 to 4) register is disabled 1: Read and write to the GCVLRn and OCVLRn (n = 0 to 4) register is enabled.	R/W
b1	CLBB0WI	Calibration bit write disable	0: Write to the CLBPRO bit is disabled 1: Write to the CLBPRO bit is enabled. This bit is read as 0.	R/W
b7 to b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

CLBPRO bit (Calibration control protection)

The CLBPRO bit enables or disables read and write to the GCVLRn and OCVLRn (n = 0 to 4) register. You must first write 1 to the CLBB0WI bit before setting CLBPRO to 1.

CLBB0WI bit (Calibration bit write disable)

The CLBB0WI bit enables or disables write to the CLBPRO bit.

33.2.12 Gain Error Correction Factor Register n (GCVLRn) (n = 0 to 4)

Address(es): SDADC24.GCVLR0 4009 C048h, SDADC24.GCVLR1 4009 C04Ch, SDADC24.GCVLR2 4009 C050h, SDADC24.GCVLR3 4009 C054h, SDADC24.GCVLR4 4009 C058h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GCVL[15:0]															
Value after reset:	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	GCVL[15:0]	Gain error correction factor	These bits store the gain error correction factor	R/W

Note: Channel number (n = 0 to 4).

GCVL[15:0] bits (Gain error correction factor)

The GCVL[15:0] bits are used to store the gain error correction factor. The read and write operations for this register are only permitted when the CLBPRO bit is set to 1.

When calibration of each channel is complete, the register associated with that channel is automatically updated.

33.2.13 Offset Error Correction Factor Register n (OCVLRn) (n = 0 to 4)

Address(es): SDADC24.OCVLR0 4009 C05Ch, SDADC24.OCVLR1 4009 C060h, SDADC24.OCVLR2 4009 C064h, SDADC24.OCVLR3 4009 C068h, SDADC24.OCVLR4 4009 C06Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	OCVL[23:16]							
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
	OCVL[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
b23 to b0	OCVL[23:0]	Offset error correction factor	These bits store the offset error correction factor	R/W
b31 to b24	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: Channel number (n = 0 to 4)

OCVL[23:0] bits (Offset error correction factor)

The OCVL[23:0] bits are used to store the offset error correction factor. The read and write operations for this register are only permitted when the CLBPRO bit is set to 1.

When calibration of each channel is complete, the register associated with that channel is automatically updated.

33.3 Operation

A sigma-delta A/D converter with a programmable gain instrumentation amplifier is built into the SDADC24. Signals from the input multiplexers (5 channels) pass through the programmable gain instrumentation amplifier (PGA) and enter the sigma-delta A/D converter. The A/D conversion results are filtered through the SINC3 digital filter and stored in an output register. A/D conversion is performed by the SDADC24 reference clock generated by the SDADCCLK.

Set the STC1.CLKDIV[3:0] bits so that the SDADC24 reference clock is output at 4 MHz. In normal A/D conversion mode, the oversampling frequency is 1 MHz. In low-power A/D conversion mode, the oversampling frequency is 0.125 MHz.

A/D conversion is performed based on the AUTOSCAN built-in sequencer. The data rate (output frequency of A/D conversion results) can also be set for each channel. For details on the processing flow, see section 33.4, Control Flows. Figure 33.3 shows a block diagram of the SDADC24.

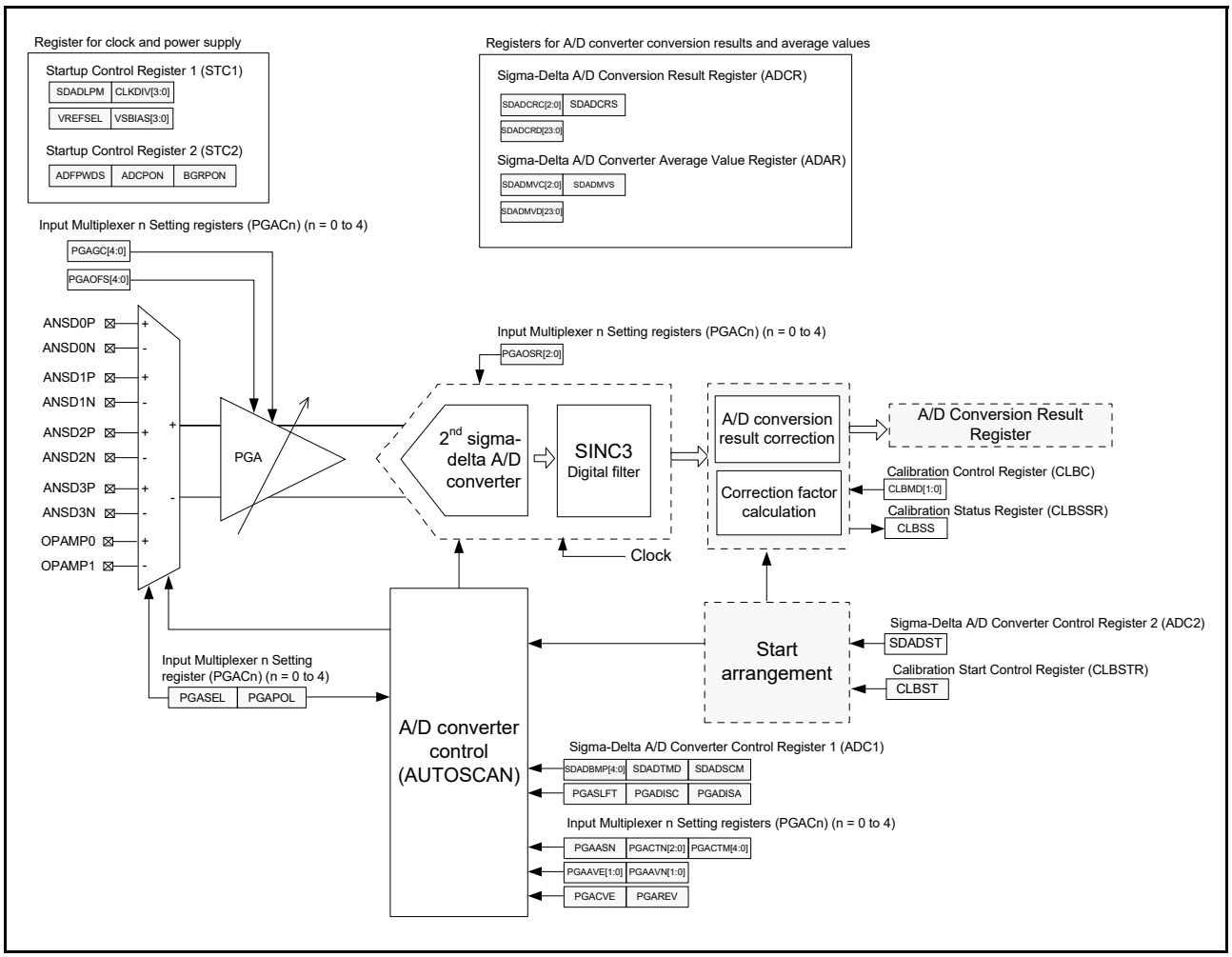


Figure 33.3 SDADC24 block diagram

33.3.1 ADBGR

ADBGR supplies the VREF reference voltage to the ADREG and SBIAS circuits. The supplied VREF reference voltage is then used as the reference voltage of the sigma-delta A/D converter, the reference voltage of the offset adjustment D/A converter inside PGA, and the internal bias voltage connected to the input multiplexer (VBIAS).

33.3.2 ADREG

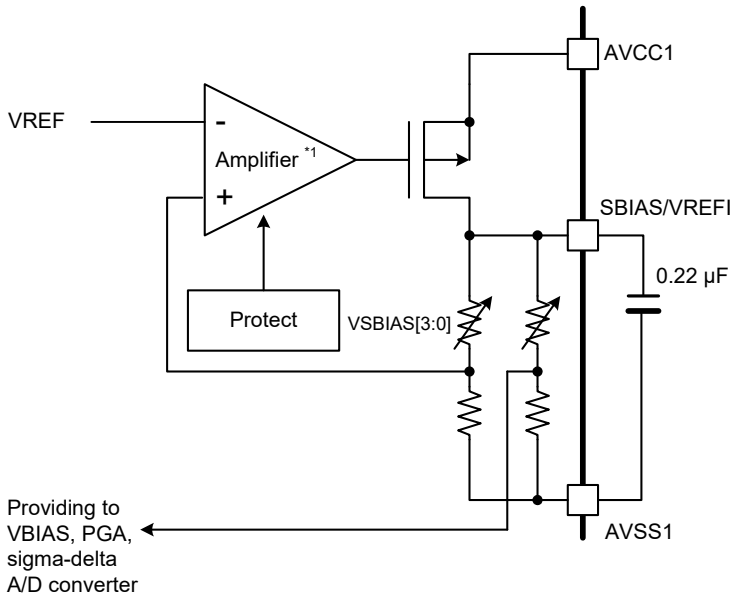
ADREG supplies power to PGA and sigma-delta A/D converter by using the output voltage of ADBGR as the reference. The output voltage is 2.1 V (typical). The output pin of ADREG requires an external capacitor of 0.47 µF (recommended value).

33.3.3 SBIAS and VREFI

33.3.3.1 Description of the SBIAS and VREFI function

SBIAS is a power supply for sensors. When STC1.VREFSEL bit is set to 0, the VREF reference voltage output from ADBGR is input into SBIAS. The output voltage range varies from 0.8 V to 2.2 V, and can be set in units of 0.2 V. The output current is 10 mA (max). SBIAS has an over current (the current exceeding the maximum) protection circuit. If an overcurrent state occurs, the protection circuit protects the internal circuit.

VREFI is an external reference voltage input. When STC1.VREFSEL is set to 1, the reference voltage is externally input. The input voltage range varies from 0.8 V to 2.4 V, and can be set in units of 0.2 V. The SBIAS/VREFI pin requires an external capacitor of 0.22 µF (recommended value).



Note 1. The amplifier is turned off when the external VREF mode (STC1.VREFSEL = 1) is selected.

Figure 33.4 Power circuit for sensors (SBIAS) block diagram

33.3.3.2 SBIAS independent operation

SBIAS can be used independently without using a sigma-delta A/D converter. To use SBIAS independently, set the ADFPWDS bit in the Startup Control Register 2 (STC2) to 1 to turn off ADREG that supplies power to the PGA and the sigma-delta A/D converter.

When SBIAS is used independently, the reference voltage for sensors (VSBIAS) setting can be changed in units of 0.2 V (0.2 V = 1 step) when SBIAS is turned on. For details, see Figure 33.33.

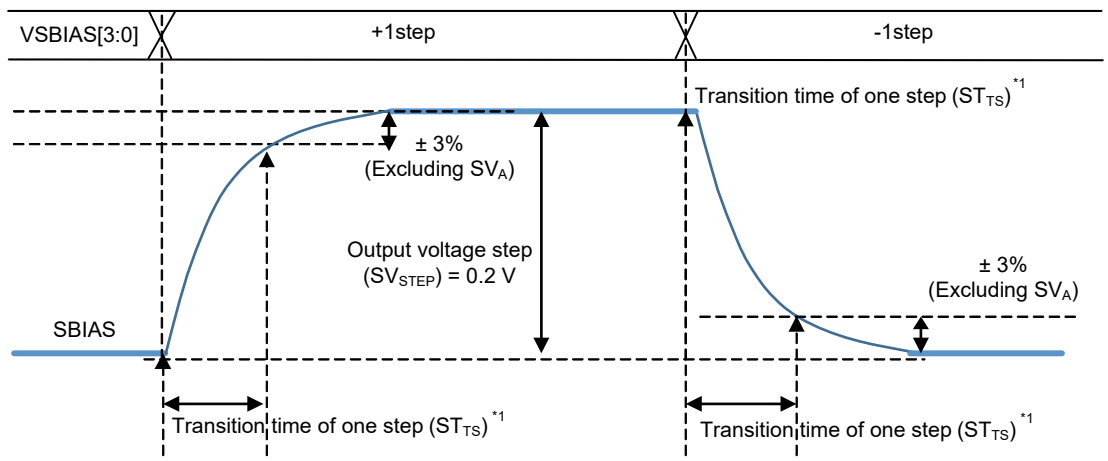
Note: Except when SBIAS is used independently, the setting of the reference voltage for sensors (VSBIAS) must not be changed when SBIAS is turned on.

Note: The reference voltage for sensors (VSBIAS) cannot be changed more than 1 step at one time.

Note: When the reference voltage for sensors (VSBIAS) is changed more than one step at one time, wait for ST_{TS} before the reference voltage can be changed with each step.

Note: Turn SBIAS on during normal mode of MCU. Supply SDADCCLK while waiting for stabilization after turning on SBIAS. It is possible to change to Software standby mode and Snooze mode after stabilization.

Note: When A/D conversion is activated after SBIAS is used independently, the SBIAS operation must be stopped and the SDADC24 power supply must be re-activated from the start.



Note 1. For details of the ST_{TS} time, see section 47, Electrical Characteristics.

Figure 33.5 Changing of the voltage setting for sensors (VSBIAS) in units of 0.2 V (1 step) when turning SBIAS on

Table 33.5 STC2 register settings and power states

BGRPON	ADFPWDS	Status
0	0	ADBGR, SBIAS, ADREG: Power off
0	1	
1	0	ADBGR, SBIAS, ADREG: Power on
1	1	ADBGR, SBIAS: Power on ADREG: Power off

33.3.4 VBIAS

VBIAS supplies the internal bias voltage (VBIAS = 1.0 V (typical)) to the input multiplexer. The internal bias voltage (VBIAS) is used as the reference voltage in single-ended input mode. For details, see section 33.3.6.3, Range of input voltage in single-ended input mode.

33.3.5 Input Multiplexers

The input multiplexers provide five analog input channels. Four of them (input multiplexers 0 to 3) receive external signal input, and one of them (input multiplexer 4) receives the input from an internal OPAMP. The input mode (differential input or single-ended input) can be selected for each channel. If positive-side single-ended input mode is set, the internal bias voltage (VBIAS) is connected on the minus (-) side. If negative-side single-ended input mode is set, the internal bias voltage (VBIAS) is connected on the plus (+) side.

Figure 33.6 shows a block diagram of the input multiplexers.

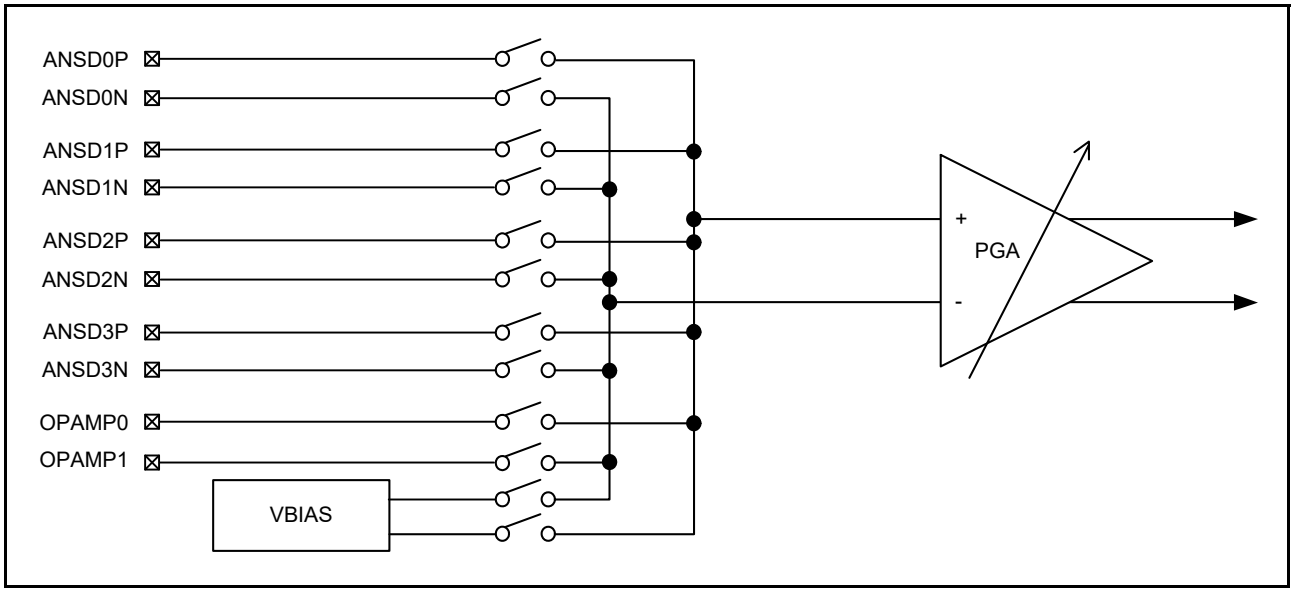


Figure 33.6 Block diagram of the input multiplexers

33.3.5.1 Input Multiplexer Control Registers

Input multiplexer is set using PGACn (n = 0 to 4). Table 33.6 shows the input multiplexer settings. The setting combinations not listed in Table 33.6 are prohibited.

Table 33.6 Input multiplexer settings using PGACn (n = 0 to 4)

Input mode	PGASEL	PGAPOL	PGAREV	Input multiplexer setting
Differential	0	0	0	Differential
Single-ended	1	0	0	Positive-side single A/D conversion
	1	1	0	Negative-side single A/D conversion
	1	1	1	Negative-side single output inverting A/D conversion

33.3.6 Programmable Gain Instrumentation Amplifier (PGA)

The programmable gain instrumentation amplifier (PGA) is an amplifier that features a low offset voltage, low noise, and high input impedance. The PGA can be placed in either differential input mode or single-ended input mode by setting the input multiplexer.

In differential input mode, the multiplication factor of gain (G_{TOTAL}) can be changed from ×1 to ×32 by a combination of the gain of previous-stage amplifier (G_{SET1}) and the gain of next-stage amplifier (G_{SET2}). In single-ended input mode, the gain can only be set for ×1.

Note: A D/A converter for offset voltage adjustment is connected to the next-stage amplifier. In differential input mode, this D/A converter can be used to adjust the offset voltage (from -164 to +164 mV, in 31 levels: 5 bits) for each channel. In single-ended input mode, set the offset voltage to 0 mV.

The single-ended input can be converted to single-ended input with positive-side or negative-side channels.

As a disconnection detection assist function for PGA input, the current source load can be internally connected to the PGA input. A disconnection detection assist function can be used for positive-side and negative-side input pins in single-ended input mode.

Figure 33.7 shows a block diagram of the programmable gain instrumentation amplifier (PGA).

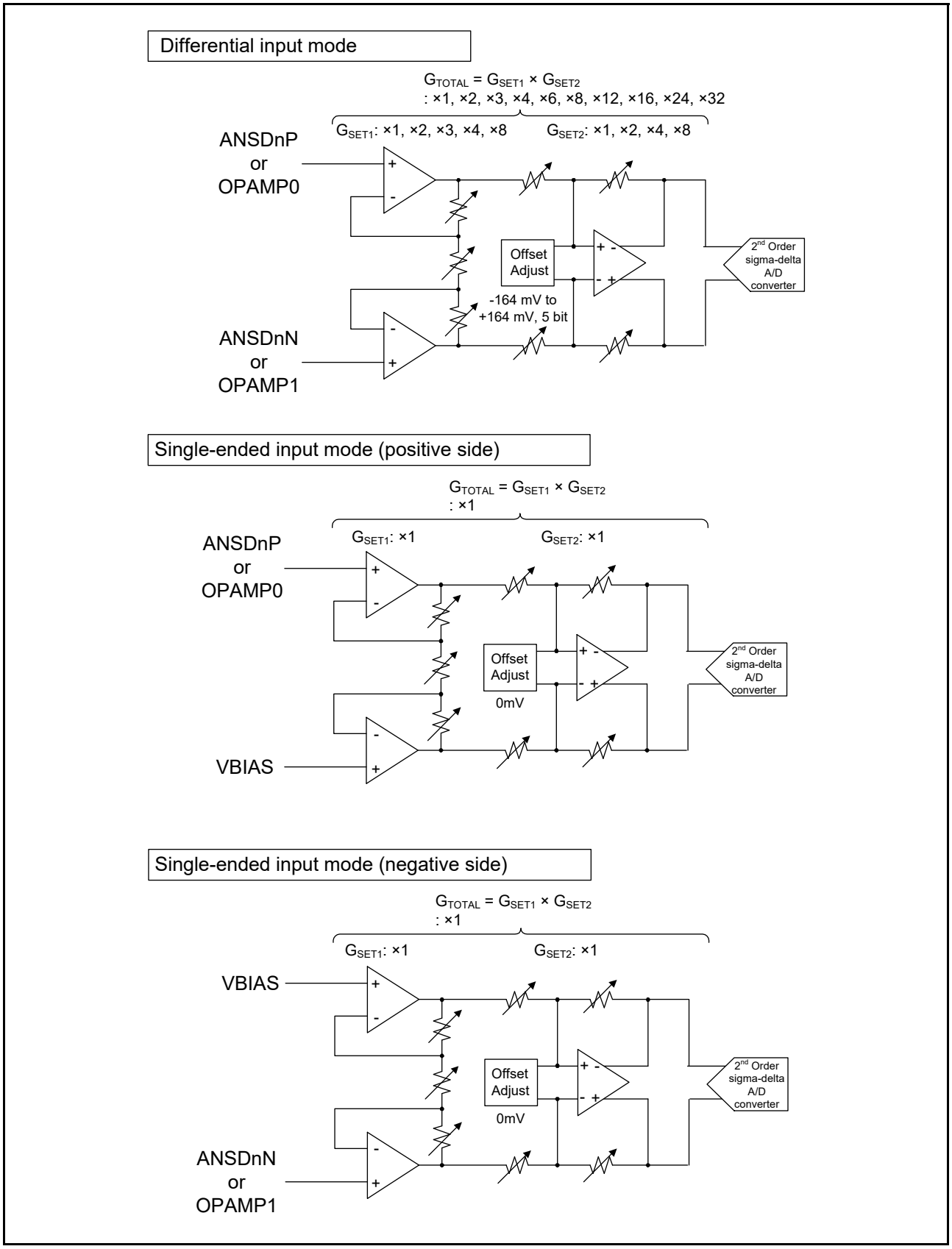


Figure 33.7 Programmable gain instrumentation amplifier (PGA)

33.3.6.1 Range of input voltage

This section describes the range of input voltage for the programmable gain instrumentation amplifier (PGA).

Figure 33.8 and Figure 33.10 show the ranges of input voltage in differential input mode and single-ended input mode.

33.3.6.2 Range of input voltage in differential input mode

In the expressions that follow, VSIG is the differential voltage amplitude, VCOM is the in-phase input voltage, and d_{OFR} is the value calculated by converting the output voltage of the D/A converter for offset voltage adjustment into input voltage. The range of input voltage for one amplifier stage is from 0.2 V to 1.8 V. Therefore, signals that pass through the previous-stage amplifier of the instrumentation amplifier and enter to the next-stage amplifier must satisfy the condition indicated by Expression 1.

In addition, signals that pass through the previous-stage amplifier of the instrumentation amplifier and exit from the next-stage amplifier must satisfy the condition indicated by Expression 2.

Expression 1:

$$0.2\ \text{V} + \frac{|V_{SIG}| \times G_{SET1}}{2} \leq V_{COM} \leq 1.8\ \text{V} - \frac{|V_{SIG}| \times G_{SET1}}{2}$$

Expression 2:

$$-0.8\ \text{V} \leq (V_{SIG} + d_{OFR}) \times G_{TOTAL} \leq 0.8\ \text{V}$$

When d_{OFR} = 0 mV, the input signal can take the differential input voltage at full scale. When $V_{SIG} = V_{ID}$ (full-scale differential input voltage), V_{COM} can be represented using Expression 3.

Expression 3:

$$0.2\ \text{V} + \frac{|V_{ID}| \times G_{SET1}}{2} \leq V_{COM} \leq 1.8\ \text{V} - \frac{|V_{ID}| \times G_{SET1}}{2}$$

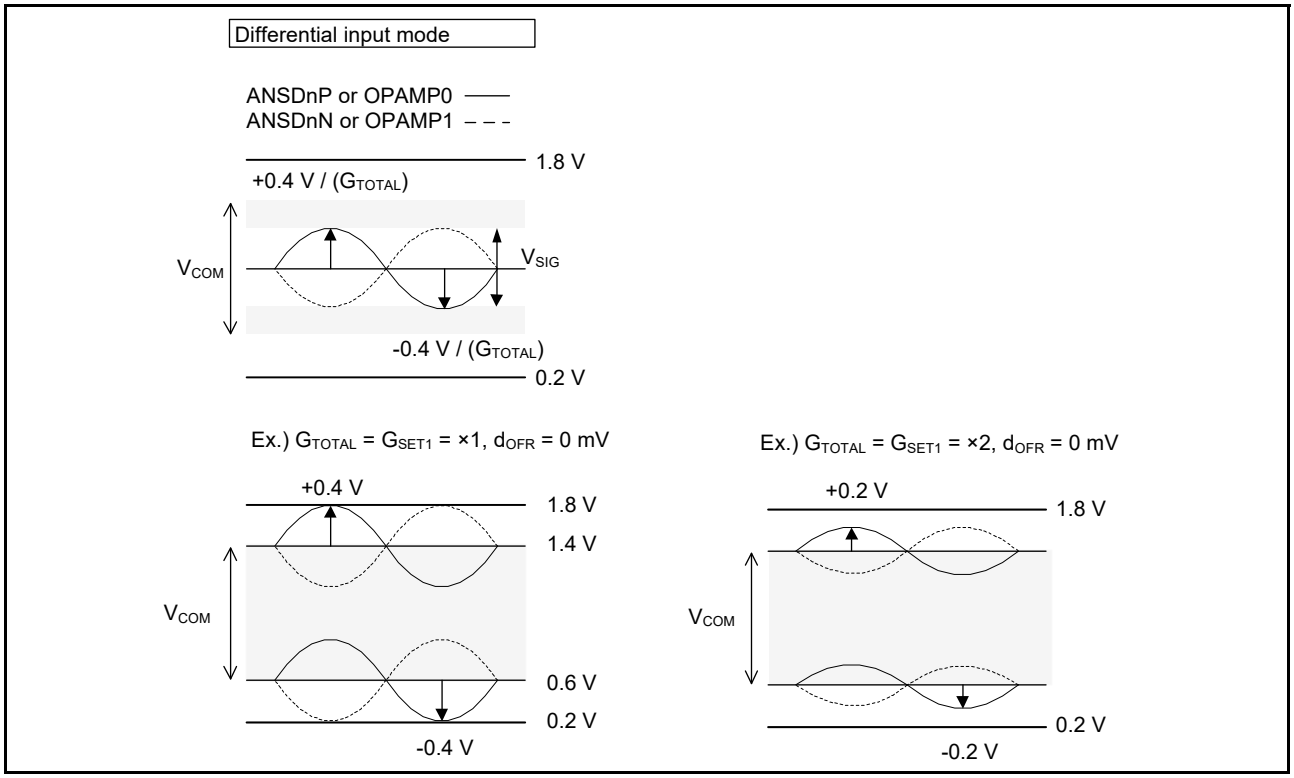


Figure 33.8 Range of input voltage in differential input mode

Figure 33.9 shows the transition of the amplitude of differential input voltage for each channel of the programmable gain instrumentation amplifier (PGA).

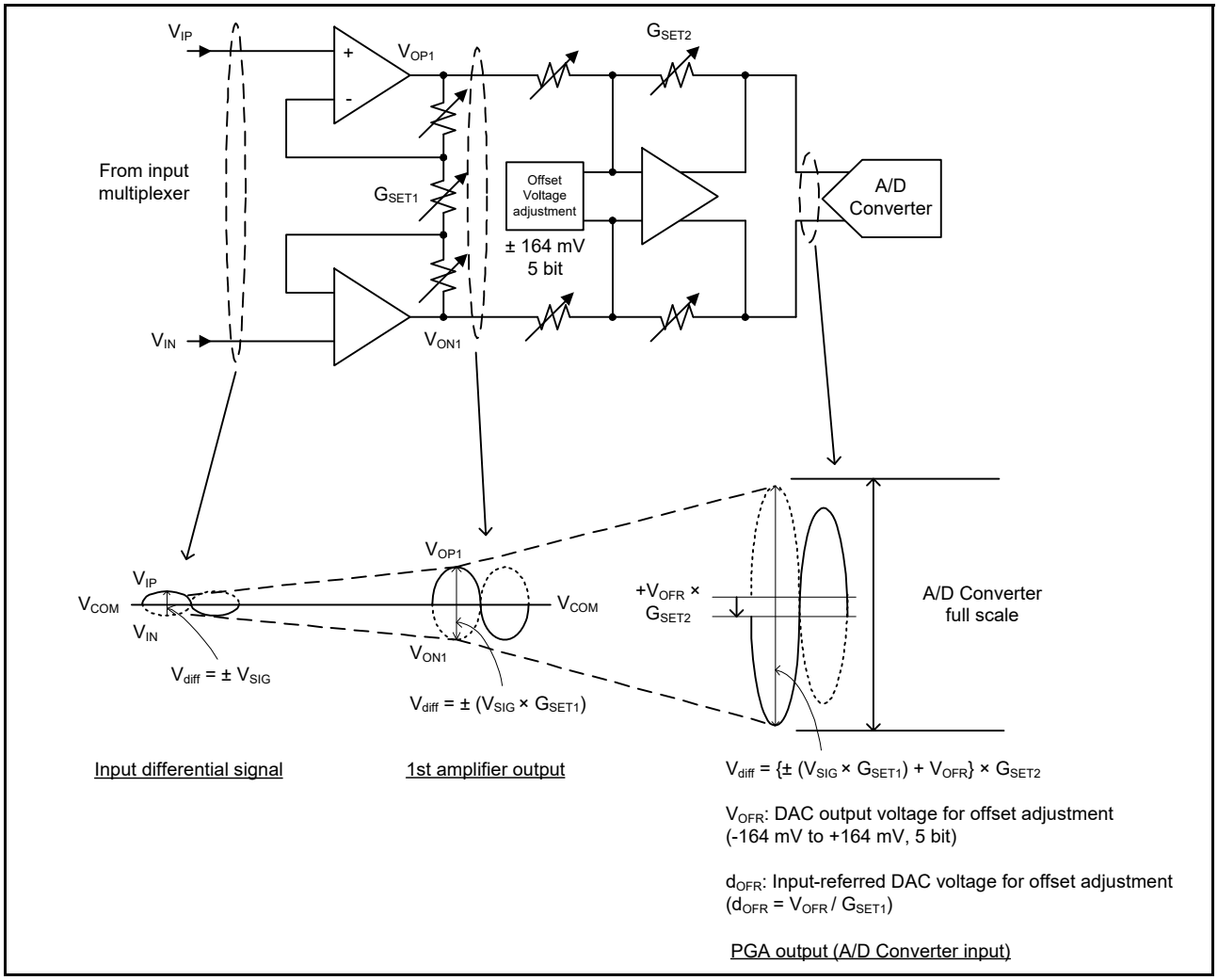


Figure 33.9 Transition of differential input voltage for each channel of the PGA

33.3.6.3 Range of input voltage in single-ended input mode

In positive-side single-ended input mode, the signal from the input multiplexer n (n = 0 to 4) is connected to the non-inverting input of the PGA. The signal is also connected to the inverting input of the PGA using the internal bias voltage (V_{BIAS} = 1.0 V (typical)) as the reference voltage. In negative-side single-ended input mode, the signal from the input multiplexer n (n = 0 to 4) is connected to the inverting input of the PGA, and the internal bias voltage is connected to the non-inverting input of the PGA. The differential signal is output in the range from 0.2 V to 1.8 V for the reference voltage.

The range of input voltage (VI) must satisfy the following expression:

Expression 1: $0.2\ V \leq VI \leq 1.8\ V$

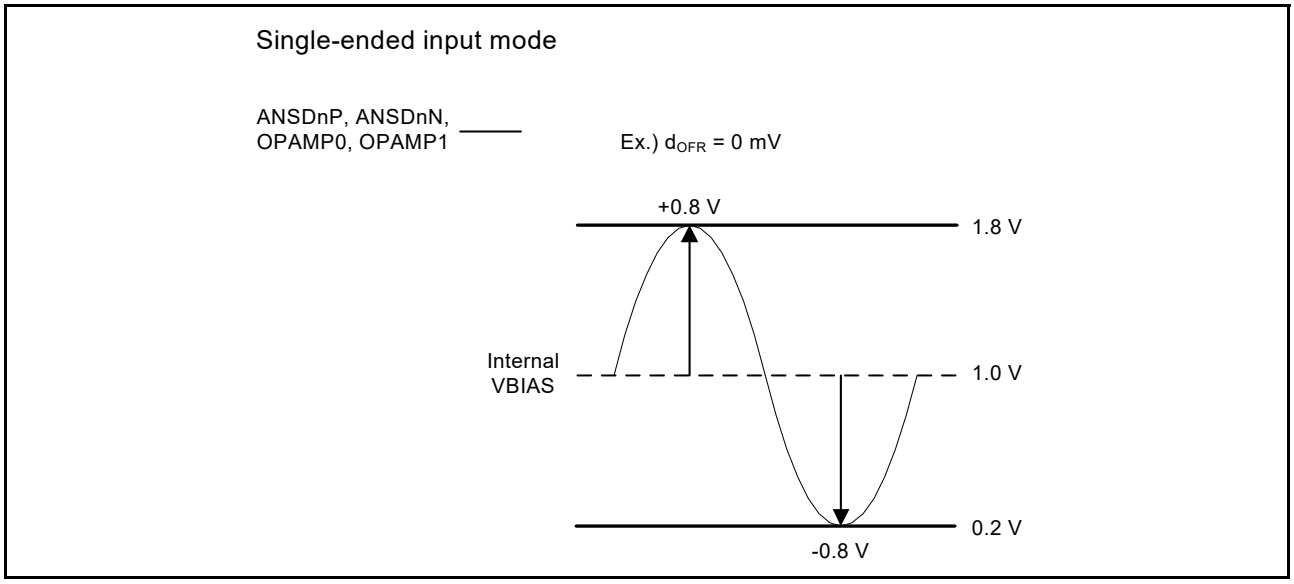


Figure 33.10 Range of input voltage in single-ended input mode

33.3.6.4 Registers for controlling the PGA

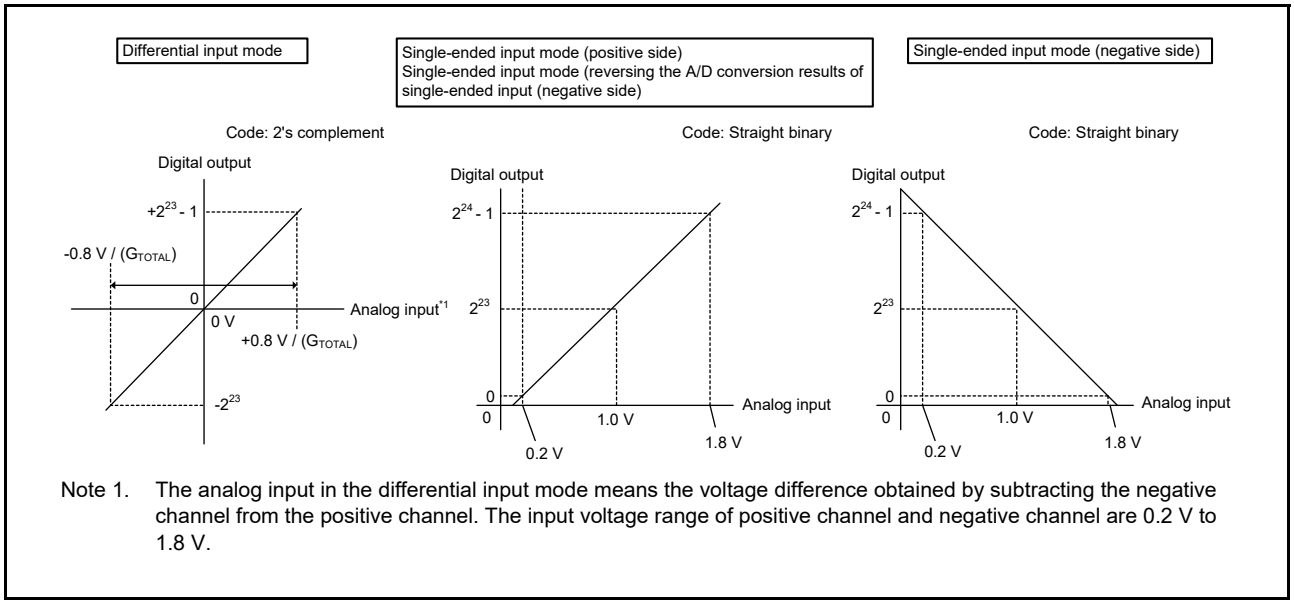
The PGA uses the following registers:

- Input Multiplexer n Setting Register (PGACn) (n = 0 to 4)
- Sigma-Delta A/D Converter Control Register 1 (ADC1).

For details on the settings, see Figure 33.30.

33.3.7 Input Voltage for the SDADC24 and Results of A/D Conversion

This section describes the input voltage for the SDADC24 and the results of A/D conversion. Figure 33.11 shows a result of A/D conversion when the range of input voltage for the A/D converter is full scale.



Note 1. The analog input in the differential input mode means the voltage difference obtained by subtracting the negative channel from the positive channel. The input voltage range of positive channel and negative channel are 0.2 V to 1.8 V.

Figure 33.11 Input voltage for the SDADC24 and results of A/D conversion

Table 33.7 Input voltage for the SDADC24 and results of A/D conversion

Differential input mode		Single-ended input mode (positive side) Single-ended input mode (reversing the A/D conversion results of single-ended input (negative side))		Single-ended input mode (negative side)	
Input voltage for the SDADC24*1	**A/D conversion result (2's complement)**	**Input voltage for the SDADC24**	**A/D conversion result (straight binary)**	**Input voltage for the SDADC24**	**A/D conversion result (straight binary)**
+0.8 V / (G_{TOTAL})	2^{23} - 1	+0.8 V + 1.0 V	2^{24} - 1	+0.8 V + 1.0 V	0
0 V	0	1.0 V	2^{23}	1.0 V	2^{23}
-0.8 V / (G_{TOTAL})	-2^{23}	-0.8 V + 1.0 V	0	-0.8 V + 1.0 V	2^{24} - 1

Note: The results shown in Table 33.7 can be calculated using the following expressions:
- Differential input mode
  - Input voltage for the SDADC24 = (1.6 V / G_{TOTAL}) × (ADCDATA1 / 2^{24})
  - ADCDATA1: 2's complement for the result of 24-bit A/D conversion (ADCR.SDADCRD[23:0])
- Single-ended input mode
  - Input voltage for the SDADC24 (positive side or reversing the A/D conversion results of single-ended input (negative side)) = 1.6 V × (ADCDATA2 / 2^{24}) + 0.2 V
  - Input voltage for the SDADC24 (negative side) = 1.6 V × (1 - ADCDATA2 / 2^{24}) + 0.2 V
  - ADCDATA2: Straight binary value for the result of 24-bit A/D conversion (ADCR.SDADCRD[23:0])

Note 1. The input voltage for the SDADC24 in the differential input mode means the voltage difference obtained by subtracting the negative channel from the positive channel. The input voltage range of positive channel and negative channel are 0.2 V to 1.8 V.

33.3.8 Control of the Sigma-Delta A/D Converter (AUTOSCAN)

A/D conversion is controlled by the built-in sequencer called AUTOSCAN. When 1 is written to the ADC2.SDADST bit to enable AUTOSCAN operation, A/D conversion of each input channel starts on a round-robin basis. A/D conversion of a specific channel can be skipped by setting the ADC1.SDADBMP[4:0] bits. Table 33.8 shows the relationship between the control register settings and the AUTOSCAN operation modes.

The PGACn.PGACTN[2:0] bits (coefficient(n) = 0 to 7) and PGACn.PGACTM[4:0] bits (coefficient(m) = 0 to 31) indicate the number of times A/D conversion is to be performed for each conversion-target channel. If both of the PGACn.PGACTN[2:0] and PGACn.PGACTM[4:0] bits are 0, one-shot operation (A/D conversion is performed only once) is set. Other A/D conversion settings such as PGA gain and oversampling ratio, can also be specified for each channel. If averaging is to be performed, the total number of times A/D conversion is performed is calculated as follows:

The A/D conversion count determined by (count set for PGACn.PGACTN[2:0] and PGACn.PGACTM[4:0] bits) × (count set for the PGACn.PGAAVN[1:0] bits).

The A/D conversion result is stored in ADCR register.

An interrupt request (SDADC_ADI) is generated each time A/D conversion completes. If averaging of A/D conversion results is enabled by setting the PGACn register, you can select the timing of the generated interrupt request (SDADC_ADI), for example whether to generate the request each time A/D conversion is performed or each time the average value is updated. The interrupt request (SDADC_SCANEND) is generated on completion of each cycle of AUTOSCAN from channels 0 to 4.

Table 33.8 Relationship between control register settings and AUTOSCAN operation modes (1 of 2)

ADC1.SDADTMD	**ADC1.SDADSCM**	**PGACn.PGACTN[2:0]**	**PGACn.PGACTM[4:0]**		
Selection of A/D conversion trigger signal	**Selection of autoscan mode**	**A/D conversion count setting bit n**	**A/D conversion count setting bit m**	**Trigger**	**Operation mode**
0	0	≠ 0	≠ 0	Software	Continuous scan
0	0	≠ 0	0	Software	Continuous scan
0	0	0	≠ 0	Software	Continuous scan

Table 33.8 Relationship between control register settings and AUTOSCAN operation modes (2 of 2)

ADC1.SDADTMD	ADC1.SDADSCM	PGACn.PGACTN[2:0]	PGACn.PGACTM[4:0]		
Selection of A/D conversion trigger signal	**Selection of autoscan mode**	**A/D conversion count setting bit n**	**A/D conversion count setting bit m**	**Trigger**	**Operation mode**
0	0	0	0	Software	Continuous scan (scan stops when one shot ends) *1
0	1	≠ 0	≠ 0	Software	Single scan
0	1	≠ 0	0	Software	Single scan
0	1	0	≠ 0	Software	Single scan
0	1	0	0	Software	Single scan (scan stops when one shot ends) *1
1	0	≠ 0	≠ 0	Hardware	Single scan *2
1	0	≠ 0	0	Hardware	Single scan *2
1	0	0	≠ 0	Hardware	Single scan *2
1	0	0	0	Hardware	Single scan (scan stops when one shot ends) *1
1	1	≠ 0	≠ 0	Hardware	Single scan
1	1	≠ 0	0	Hardware	Single scan
1	1	0	≠ 0	Hardware	Single scan
1	1	0	0	Hardware	Single scan (scan stops when one shot ends) *1

Note 1. If the PGACn.PGACTN[2:0] bits are set to 000b and the PGACn.PGACTM[4:0] bits are set to 00000b, one shot operation takes precedence.

Note 2. If a hardware trigger is selected, only single-scan mode is used.

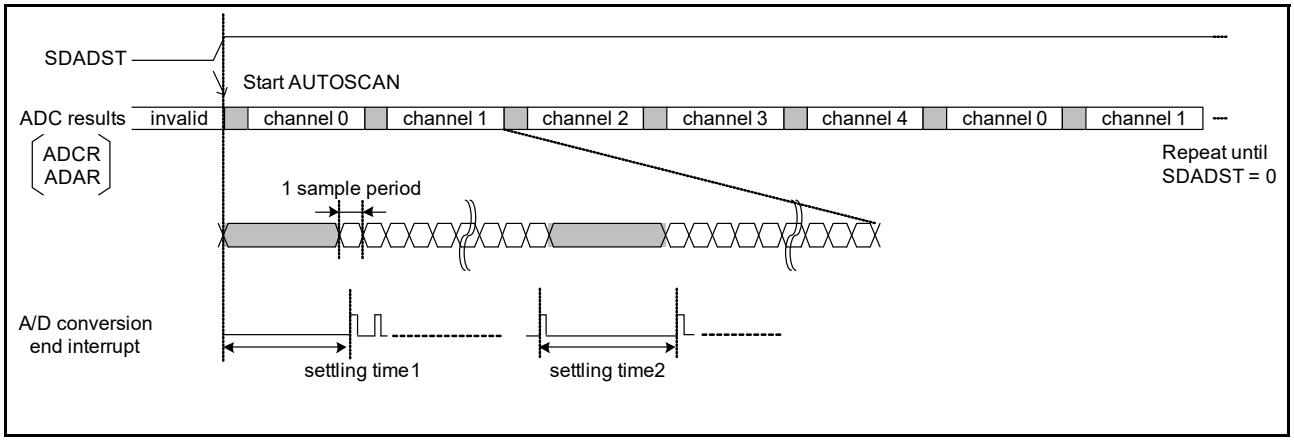


Figure 33.12 AUTOSCAN sequence

See 33.3.9.2 for details of the settling time.

(1) Example for skipping the conversion channels

- ADC1.SDADBMP[4:0] = 01010b
- The PGACTN[2:0] of the PGAC0, PGAC2, and PGAC4 registers is > 000b, and the PGACTM[4:0] of the PGAC0, PGAC2, and PGAC4 registers is > 00000b
- ADC1.SDADSCM = 0.

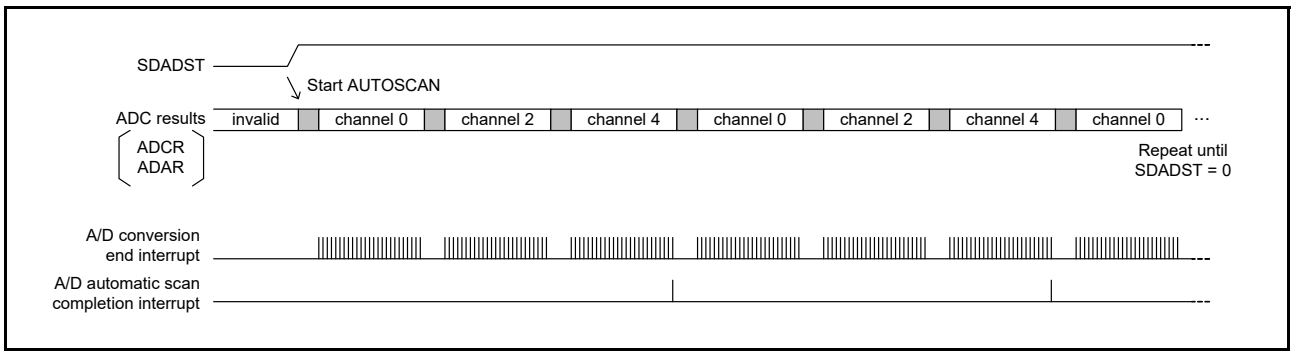


Figure 33.13 Example for skipping the conversion channels

(2) Example for one-shot operation

- ADC1.SDADBMP[4:0] = 11000b
- The PGACTN[2:0] of the PGAC0 and PGAC1 registers is > 000b, and the PGACTM[4:0] of the PGAC0 and PGAC1 registers is > 00000b
- PGAC2.PGACTN[2:0] = 000b, and PGAC2.PGACTM[4:0] = 00000b
- ADC1.SDADSCM = 0.

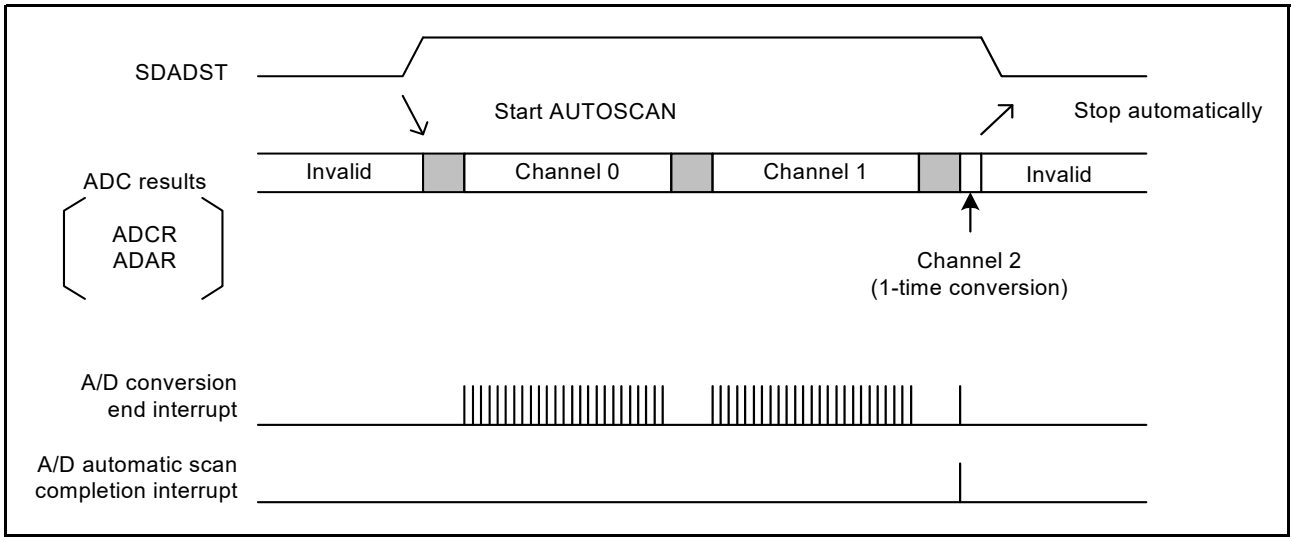


Figure 33.14 Example for one-shot operation

(3) Example for continuous conversion using the same channel

- ADC1.SDADBMP[4:0] = 11101b
- PGAC1.PGACTN[2:0] > 000b, and PGAC1.PGACTM[4:0] > 00000b
- ADC1.SDADSCM = 0.

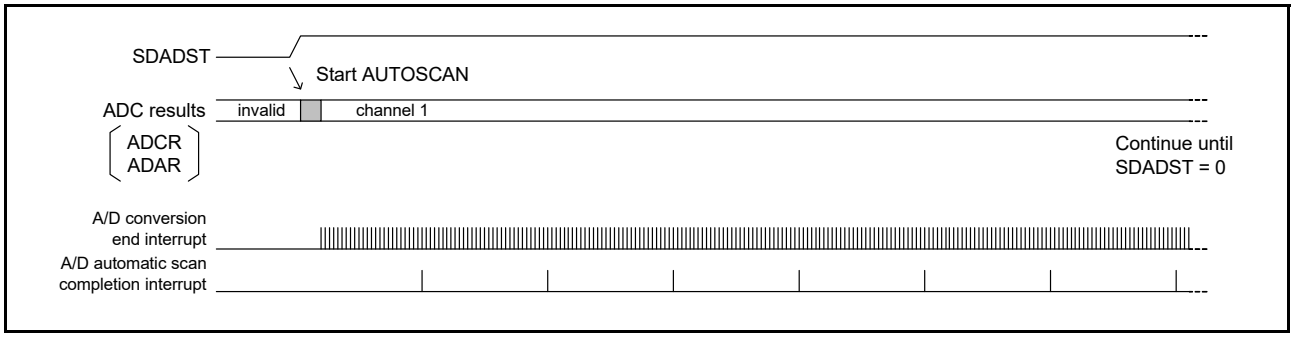


Figure 33.15 Example for continuous conversion using the same channel

(4) Example for single scan operation

- ADC1.SDADBMP[4:0] = 00000b
- ADC1.SDADSCM = 1.

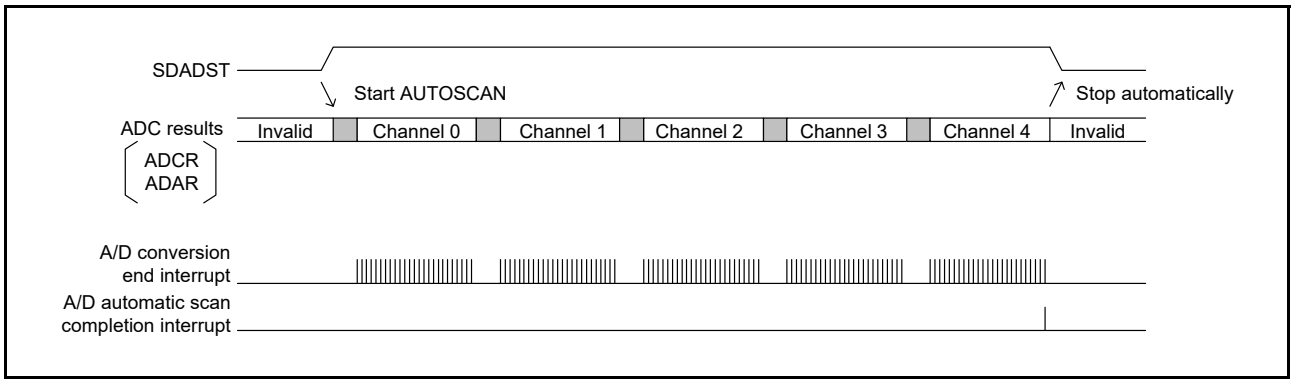


Figure 33.16 Example for single scan operation

Note: Even in continuous conversion, an interrupt request (SDADC_ADI) is generated each time the A/D conversion count set for the PGACn register is reached.

33.3.9 Digital Filter

33.3.9.1 Operation of the digital filter

Down sampling of the A/D conversion result is performed by the SINC3 digital filter. The transfer function of the digital filter is represented by the following expression. The value of M included in the transfer function is the decimation rate of the digital filter, and is determined by the OSR (oversampling ratio) set by the PGACn.PGAOSR[2:0] bits (n = 0 to 4).

$$H(z) = \left(\frac{1}{M} \times \frac{1 - Z^{-M}}{1 - Z^{-1}} \right)^{3}$$

33.3.9.2 Configuration of the digital filter

Figure 33.17 shows a block diagram of the digital filter. Three accumulators and three differentiators are connected in a cascade format. For the A/D converter to become stable, the required settling time must be satisfied. Table 33.9 shows settling time 1 and settling time 2 and are defined as follows:

- Settling time 1 — The time from ADC2.SDADST bit rising to A/D conversion end interrupt. See settling time 1 in Figure 33.12.
- Settling time 2 — The time from the last A/D conversion end interrupt before channel switching to the first A/D conversion end interrupt after channel switching. See settling time 2 in Figure 33.12.

Table 33.9 Settling time for each operation mode

		Normal A/D conversion mode	Low-power A/D conversion mode
Settling time 1	min	3T + 129 μs + 2PCLKB + 9 SDADC24 reference clock*1	3T + 1032 μs + 2PCLKB + 9 SDADC24 reference clock*1
	max	3T + 129 μs + 3PCLKB + 10 SDADC24 reference clock*1	3T + 1032 μs + 3PCLKB + 10 SDADC24 reference clock*1
Settling time 2	min	3T + 129 μs - 1PCLKB	3T + 1032 μs - 1PCLKB
	max	3T + 129 μs + 1PCLKB	3T + 1032 μs + 1PCLKB

Note: The settling time is automatically generated by the AUTOSCAN built-in sequencer.
Note: 3T is the time that is 3 times as long as the sampling time (3 × 1 / fout).
Note 1. Normal A/D conversion mode: 4 MHz, low-power A/D conversion mode: 500 kHz.

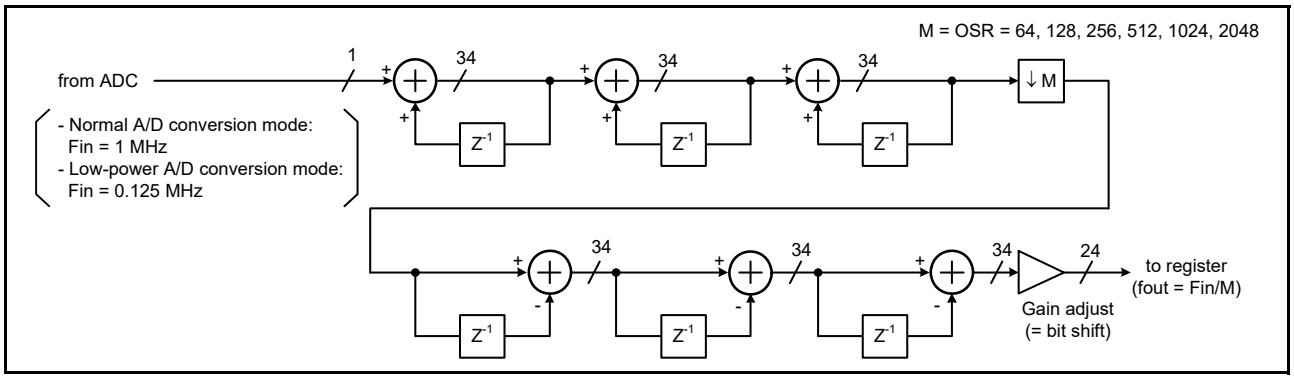


Figure 33.17 Digital filter block diagram

Figure 33.18 shows the frequency response of the SINC3 filter.

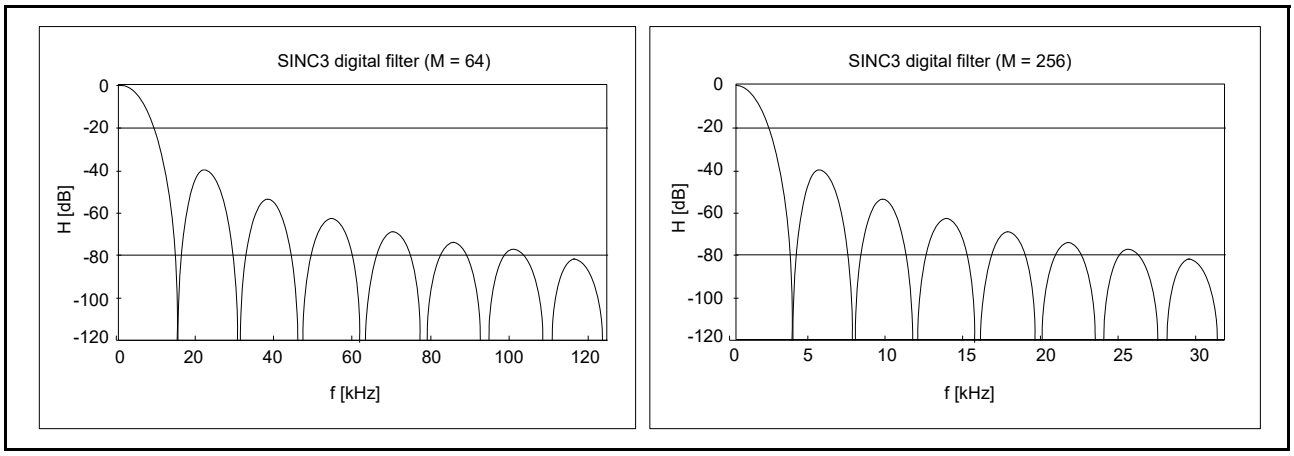


Figure 33.18 Frequency response of the SINC3 filter

33.3.10 Calibration function

Calibration allows high-precision A/D conversion by calculating the offset error correction value and gain error correction value under the conditions of use. The calibration function performs A/D conversion of the internal or user-specified reference voltage, and then determines the most appropriate correction value from the error included in the conversion result. Calibration is started when 1 is written to the CLBSTR.CLBST bit. A/D conversion is performed several times to calculate the correction factor. However, the interrupt request for each A/D conversion is masked. Table 33.10 shows the settings and operations of calibration.

Table 33.10 Settings and operations of calibration

Bits of the control registers			
PGACn.PGASEL	**CLBC.CLBMD[1:0]**	**PGACn.PGACVE**	**Calculating correction factors for calibration**
0 (Differential input mode)	Don't care	0	Disabled
	00	1	Internal calibration operation
	01	1	External offset calibration operation
	10	1	External gain calibration operation

Note: The correction factors are not calculated for the channels set in single-ended input mode.

When calibration must be performed, it must be done in the following cases:

- When differential input mode is used for the first time after reset*1
- When the VREF mode is switched between external VREF mode and internal VREF mode using the STC1.VREFSEL bit in differential input mode
- When the voltage is switched between the SBIAS output voltage and VREFI input voltage using the STC1.VSBIAS[3:0] bits in differential input mode
- When the mode is switched from normal A/D conversion mode to low-power A/D conversion mode, or vice versa in differential input mode
- When the gain is changed for the same channel in differential input mode.

Note: Bits for the channel number of an A/D conversion result (ADCR.SDADCRC[2:0]), bit for displaying the status of an A/D conversion result (ADCR.SDADCRS) and bits for the A/D Converter conversion result (ADCR.SDADCRD[23:0]) are not updated during internal or external calibration.

Note: When performing internal or external calibration, set the Automatic Scan Mode Selection bit of the Sigma-Delta A/D Converter Control Register 1 (ADC1) to 1 for single scan. For details, see section 33.4.5, Internal Calibration Flow and section 33.4.6, External Calibration Flow.

Note: The d_{OFR} voltage cannot be set to a value other than 0 mV (PGACn.PGAOFS[4:0] = 00000b (n = 0 to 4)) during calibration operation.

Note: For external calibration operation, multiple channel settings cannot be set to PGACn.PGACVE = 1 (n = 0 to 4) at the same time.

Note 1. When using the differential input mode with the same STC1.SDADLPM, STC1.VSBIAS[3:0], STC1.VREFSEL and PGACn.PGAGC[4:0] bits setting as in the calibration execution, store the values of GCVLRn and OCVLRn registers after calibration to the data flash. Recalibration can be omitted by copying the stored value to GCVLRn and OCVLRn registers on the next operation. For details, see section 33.4.9, Recalibration Omitted Flow. When a large temperature change occurs or a change in the user environment or use conditions occurs, it is necessary to perform calibration again.

33.3.10.1 Internal calibration operation mode

In internal calibration operation mode, calibration is performed based on the internal reference voltage. The correction values for multiple input channels can be calculated by only one calibration operation. After calibration is started by the CLBSTR.CLBST bit, the offset and gain error correction values are calculated for all input channels set as the calibration targets. Calibration is complete after a calibration completion interrupt (SDADC_CALIEND) is generated. For details, see Figure 33.28.

33.3.10.2 External calibration operation mode

In external calibration operation mode, calibration is performed based on the user-specified reference voltage. The offset calibration calculates the correction value for the A/D conversion result corresponding to the differential analog input at the time of the offset calibration (VIDOCAL) to be corrected to 0. Gain calibration calculates the correction value for the A/D conversion result corresponding to the value (VIDGCAL - VIDOCAL) calculated by subtracting the differential analog input at the time of the offset calibration (VIDOCAL) from the differential analog input at the time of the gain calibration (VIDGCAL) to be corrected to 2^{23} - 1. The correction value for one channel is calculated by one calibration operation. To calculate the offset and gain error correction values, two calibration operations must be performed. When each calibration completes, a calibration completion interrupt (SDADC_CALIEND) is generated. Set the reference voltage for the input channel before each calibration operation is performed (before setting 1 to the CLBSTR.CLBST bit). Table 33.11 shows the user-specified reference voltage in external calibration operation mode. For detail on the setting, see Figure 33.29.

Table 33.11 User-specified reference voltage in external calibration operation mode

User-specified reference voltage		Min	Typ	Max	Unit
The differential analog input at the time of the external offset calibration (VIDOCAL)	ANSDnP - ANSDnN (n = 0 to 3) or OPAMP0 - OPAMP1	*1	0	*1	V
The differential analog input at the time of the external gain calibration (VIDGCAL)		0.4 / G_{TOTAL}*1		0.8 / G_{TOTAL}*1	V
VIDGCAL - VIDOCAL*1		0.4 / G_{TOTAL}		0.8 / G_{TOTAL}	V

Note 1. VIDOCAL and VIDGCAL must be used in a range that satisfies the min and max of VIDGCAL - VIDOCAL.

33.3.10.3 Calibration Time

Each calibration time (t_{CAL}) is expressed in the following expression with t_{SDCAL}, t_{OFCAL}, t_{GCAL}, and t_{EDCAL} defined in Table 33.12:

- Internal Calibration time: $t_{CAL} = t_{SDCAL} + (t_{OFCAL} + t_{GCAL}) \times CH^{*1} + t_{EDCAL}$
- External offset calibration time: $t_{CAL} = t_{SDCAL} + t_{OFCAL} + t_{EDCAL}$
- External gain calibration time: $t_{CAL} = t_{SDCAL} + t_{GCAL} + t_{EDCAL}$

Note 1. CH: Number of calibration channel.

Table 33.12 Required calibration time shown as the number of SDADC24 reference clock and PCLKB cycles

Parameter	Symbol	Internal calibration time [cycle]	External offset calibration time [cycle]	External gain calibration time [cycle]
Calibration start delay time	t_{SDCAL}	1 PCLKB + 4 $t_{SDADC24}$ *1	1 PCLKB + 4 $t_{SDADC24}$ *1	1 PCLKB + 4 $t_{SDADC24}$ *1
Offset error correction factor calculation time	t_{OFCAL}	6658 $t_{SDADC24}$	6658 $t_{SDADC24}$	-
Gain error correction factor calculation time	t_{GCAL}	6662 $t_{SDADC24}$	-	6676 $t_{SDADC24}$
Calibration end delay time	t_{EDCAL}	6 PCLKB + 21 $t_{SDADC24}$ *2	6 PCLKB + 5 $t_{SDADC24}$ *2	6 PCLKB + 5 $t_{SDADC24}$ *2
Calibration time	t_{CAL}	7 PCLKB + 25 $t_{SDADC24}$ + (13320 $t_{SDADC24}$) × CH *3	7 PCLKB + 6667 $t_{SDADC24}$ *3	7 PCLKB + 6685 $t_{SDADC24}$ *3

Note: $t_{SDADC24}$: SDADC24 reference clock (normal A/D conversion mode: 4 MHz, low-power A/D conversion mode: 500 kHz).
Note: CH: Number of calibration channel.
Note 1. Maximum time from software write to A/D conversion start.
Note 2. Maximum time from calibration end to interrupt output.
Note 3. Maximum calibration time.

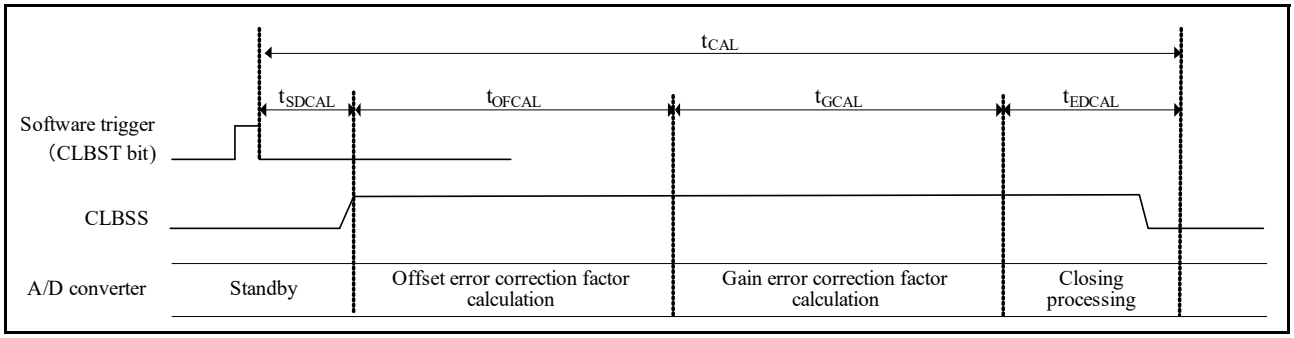


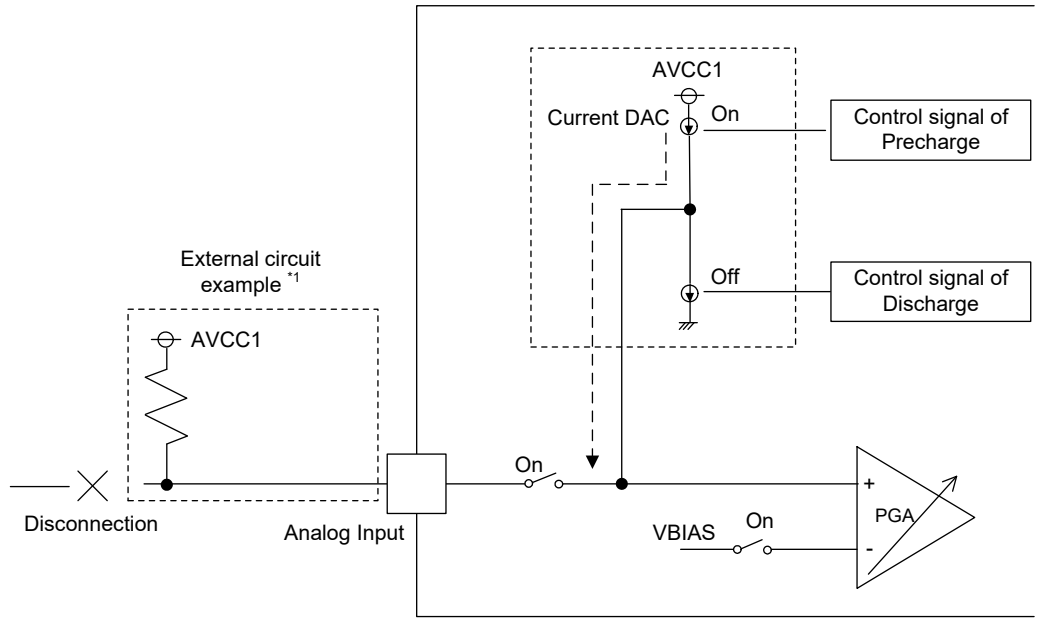
Figure 33.19 Timing of calibration

33.3.11 Disconnection Detection Assist Function

The disconnection detection assist function detects disconnection of single-ended input. By setting the ADC1.PGADISA bit to 1, the current DAC (0.8 µA typical) is connected to the input of the PGA and the disconnection detection assist function is enabled. If the ADC1.PGADISC bit is set to 0, discharge is selected. If the ADC1.PGADISC bit is set to 1, precharge is selected. If there is a disconnection or the current supply of input is less than 0.8 µA (typical), the A/D conversion results converge to approximately 0 or 2^{24} - 1 by repeating the A/D conversion (see Table 33.13). The disconnection detection assist function of the SDADC24 cannot be used at the same time with normal A/D conversion. Use the function while normal A/D conversion is not being performed, for example, before the start of a normal A/D conversion. Figure 33.20 shows an example of detecting a disconnection when precharge is selected. Figure 33.21 shows an example of detecting a disconnection when discharge is selected. For details on the settings, see Figure 33.27.

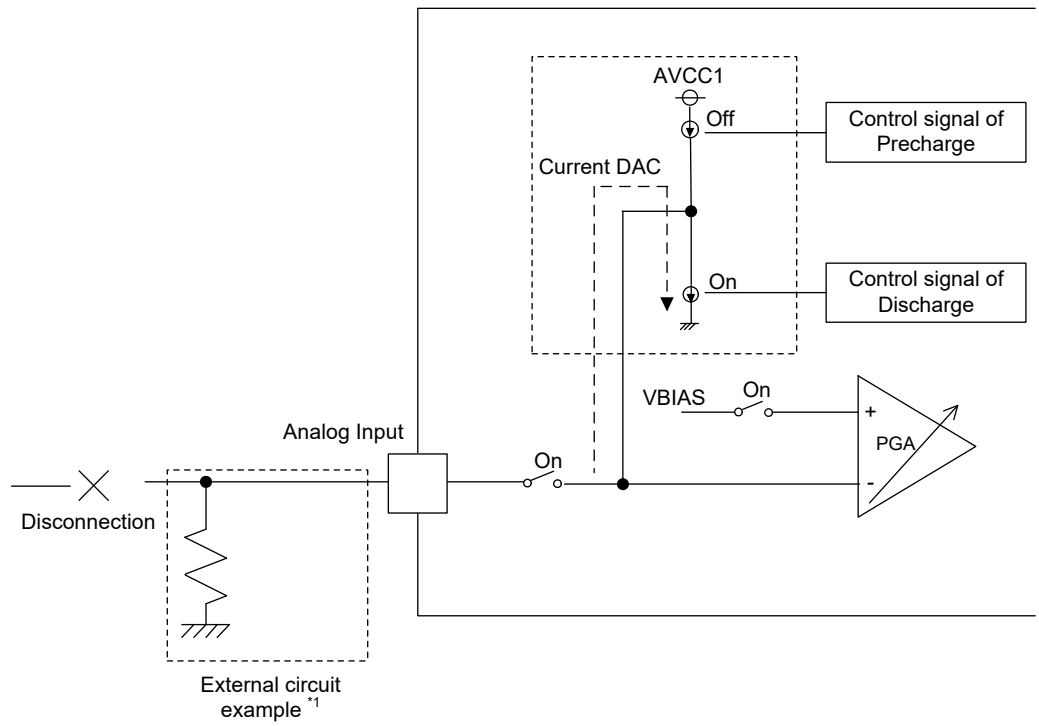
Table 33.13 A/D conversion results when there is a disconnection or the current supply of input is less than 0.8 µA (typical)

	A/D conversion results	
	Precharge	**Discharge**
Single-ended input mode (positive side) Single-ended input mode (reversing the A/D conversion results of single-ended input (negative side))	Approximately 2^{24} - 1	Approximately 0
Single-ended input mode (negative side)	Approximately 0	Approximately 2^{24} - 1



Note 1. Use this function after proper evaluation because the conversion result obtained when a disconnection exists differs depending on the external circuit.

Figure 33.20 Example of detecting disconnection when precharge is selected in single-ended input mode (positive side)



Note 1. Use this function after adequate evaluation because the conversion result obtained when a disconnection exists differs depending on the external circuit.

Figure 33.21 Example of detecting disconnection when discharge is selected in single-ended input mode (negative side)

Note: Use the disconnection detection assist function in single-ended input mode (PGACn.PGASEL = 1). For details, see section 33.4.4, Disconnection Detection Assist Flow. In the disconnection detection assist, do not set the value of d_{OFR} to a value other than 0 mV (PGACn.PGAOFS[4:0] = 00000b (n = 0 to 4)).

33.3.12 Self-Diagnosis Function of PGA Offset

Self-diagnosis of the PGA offset is possible if A/D conversion is performed by setting 1 to the ADC1.PGASLFT bit. The result of self-diagnosis (offset measurement result) is stored in the A/D Conversion Result register. For details on the setting, see Figure 33.26.

Figure 33.22 shows the configuration of self-diagnosis function of the PGA offset.

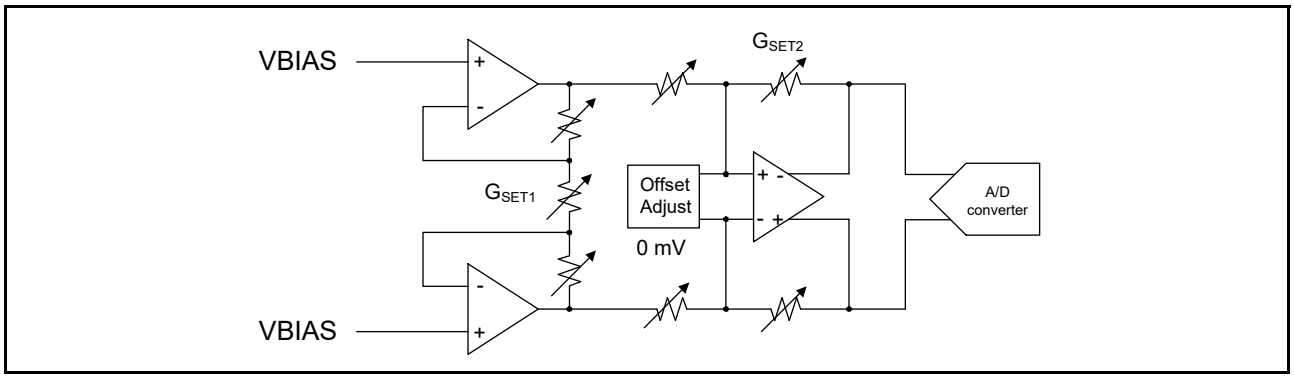


Figure 33.22 Configuration of self-diagnosis function of the PGA offset

Note: In self-diagnosis of PGA offset, do not set the value of d_{OFR} to a value other than 0 mV (PGACn.PGAOFS[4:0] = 00000b (n = 0 to 4)).

33.4 Control Flows

Figure 33.24 to Figure 33.33 show the startup, A/D conversion, and stop flows of the SDADC24. The flow for SBIAS independent operation is shown in Figure 33.33.

The A/D conversion results can be verified using the Sigma-Delta A/D Converter Conversion Result Register (ADCR) and Sigma-Delta A/D Converter Average Value Register (ADAR).

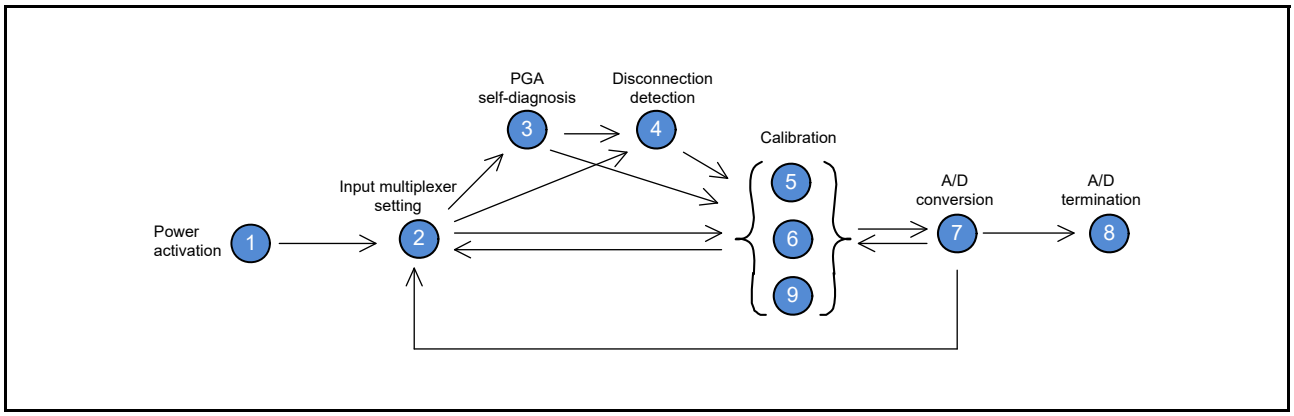


Figure 33.23 Overview of flows

33.4.1 Analog Power Supply Activation Flow

Figure 33.24 shows the flow for analog power supply activation.

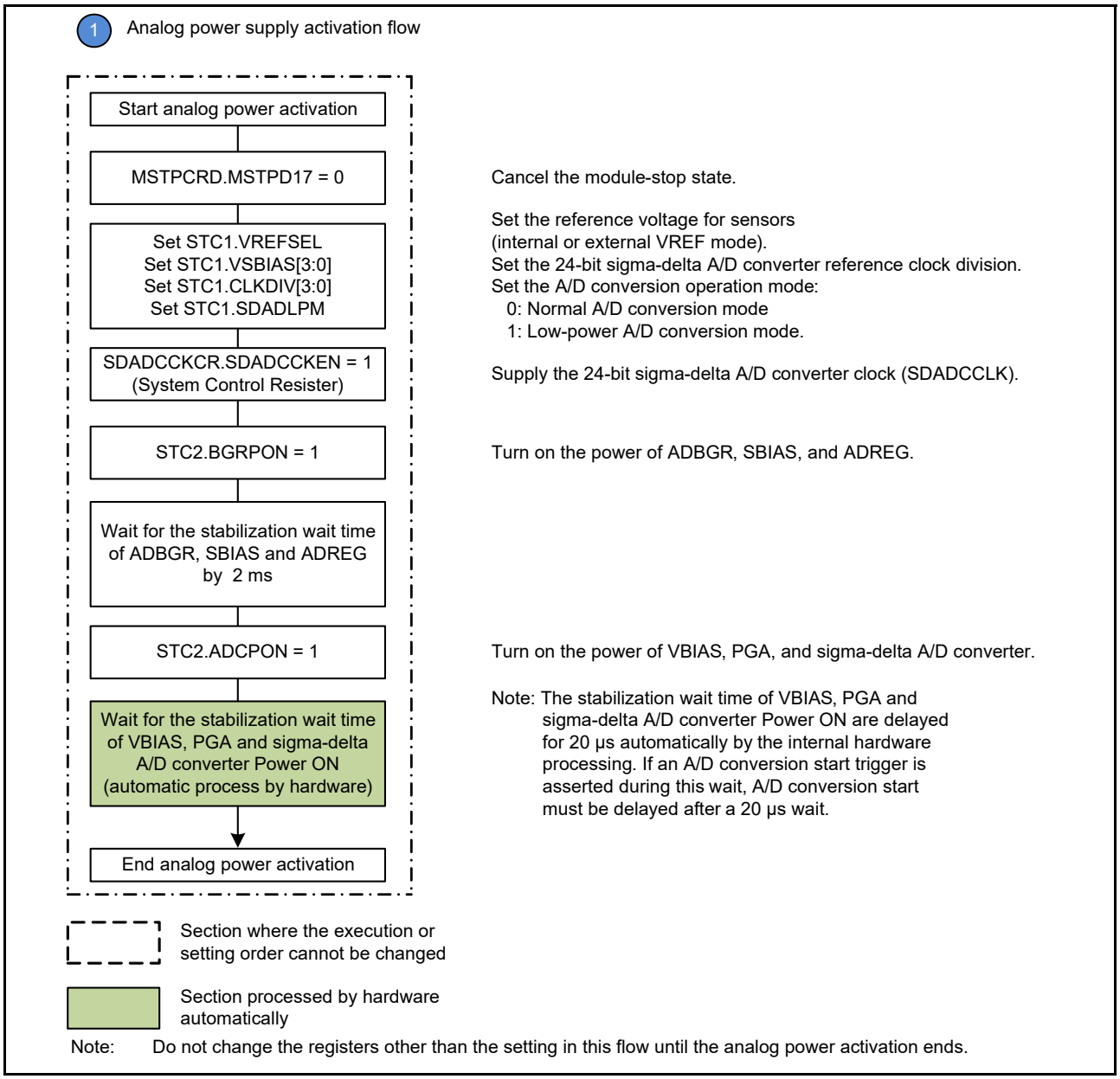


Figure 33.24 Analog power supply activation flow

33.4.2 Input Multiplexer Setting Flow

Figure 33.25 shows the flow for input multiplexer setting.

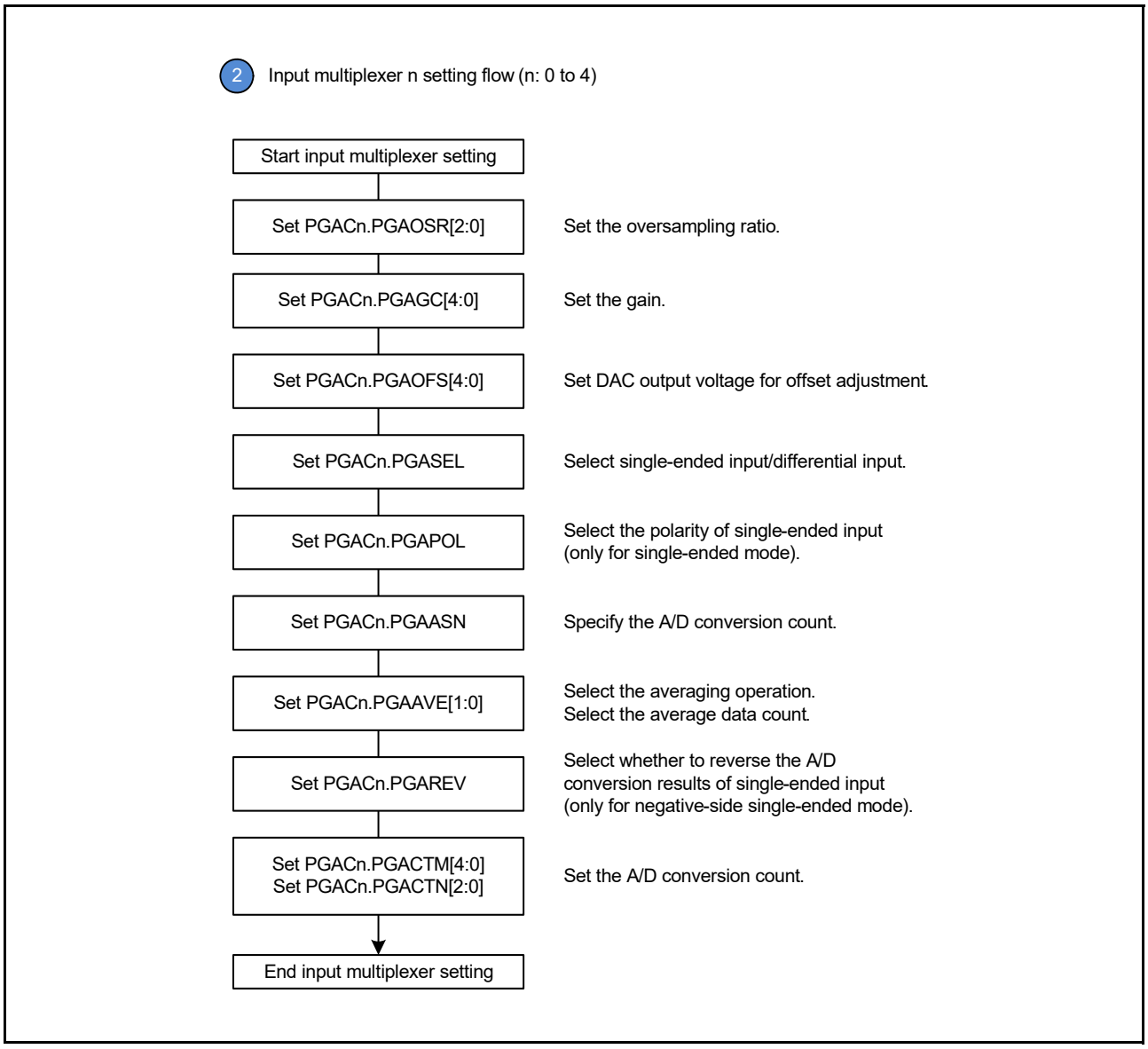
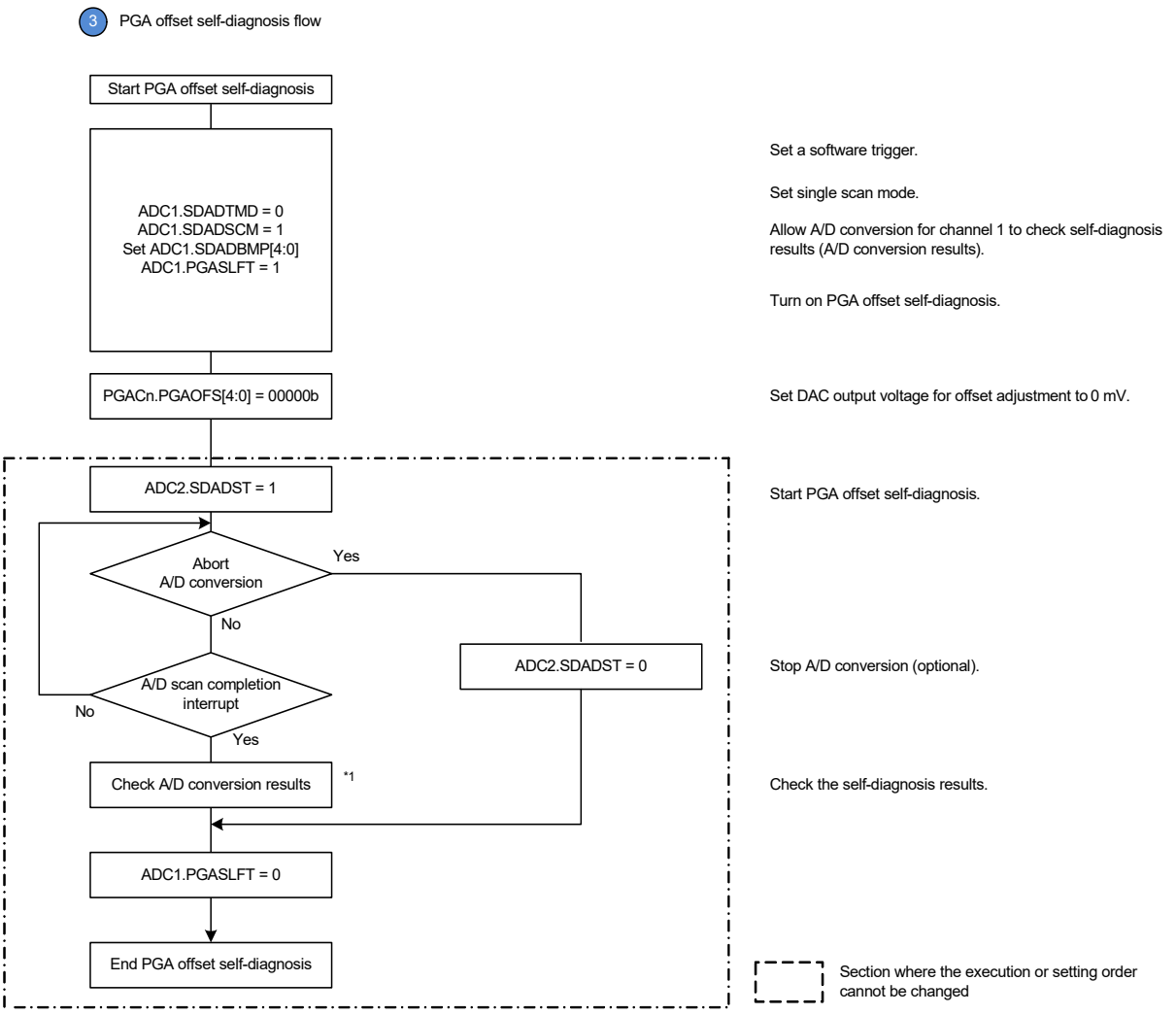


Figure 33.25 Input multiplexer setting flow

33.4.3 Self-Diagnosis Flow of PGA Offset

Figure 33.26 shows the flow for self-diagnosis of PGA offset.

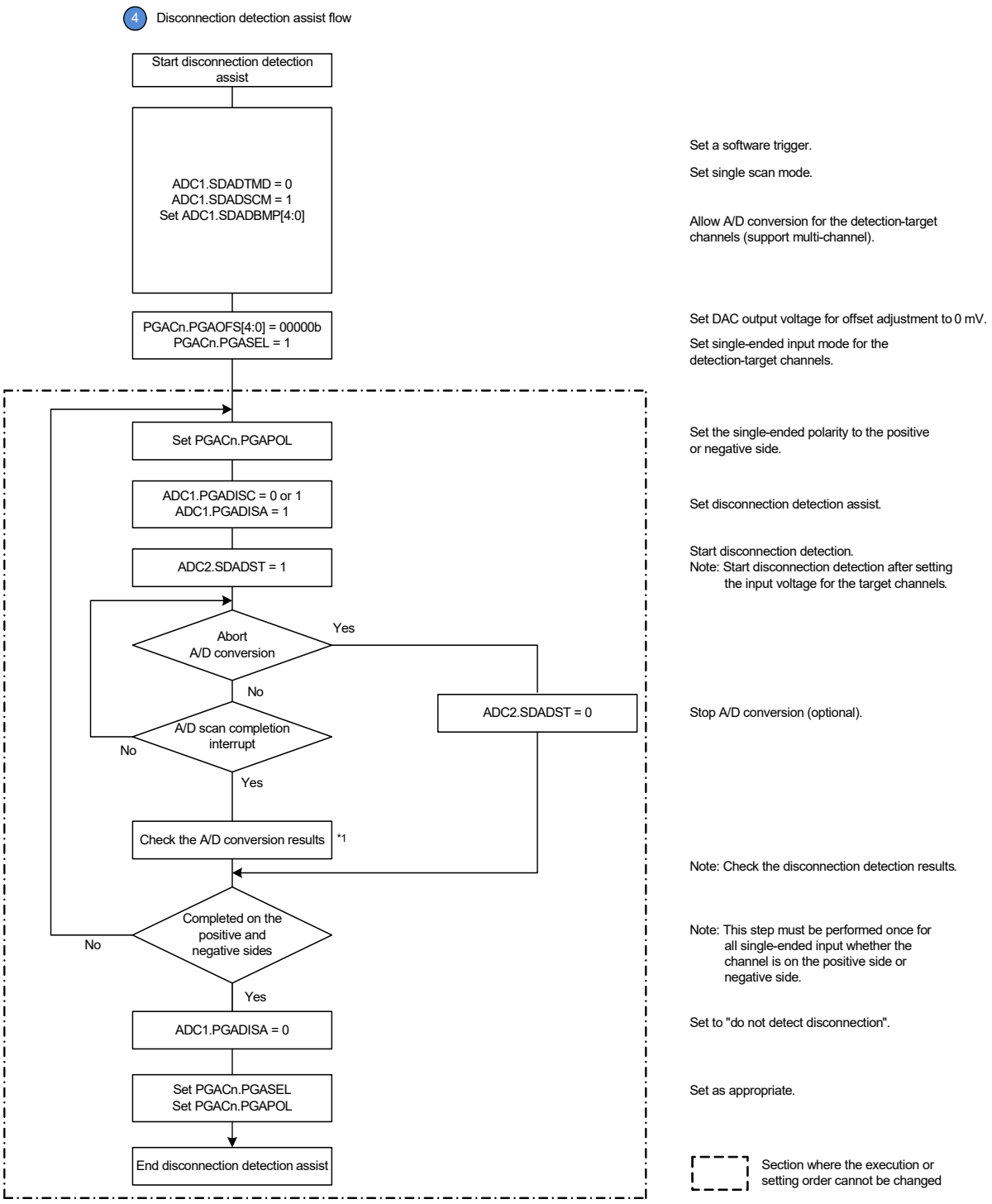


Note 1. The results of PGA offset self-diagnosis can be checked from A/D conversion results.

Figure 33.26 Self-diagnosis flow for PGA offset

33.4.4 Disconnection Detection Assist Flow

Figure 33.27 shows the flow for disconnection detection assist.



Note 1. The disconnection detection status can be checked from the A/D conversion results.

Figure 33.27 Disconnection detection assist flow

33.4.5 Internal Calibration Flow

Figure 33.28 shows the flow for internal calibration.

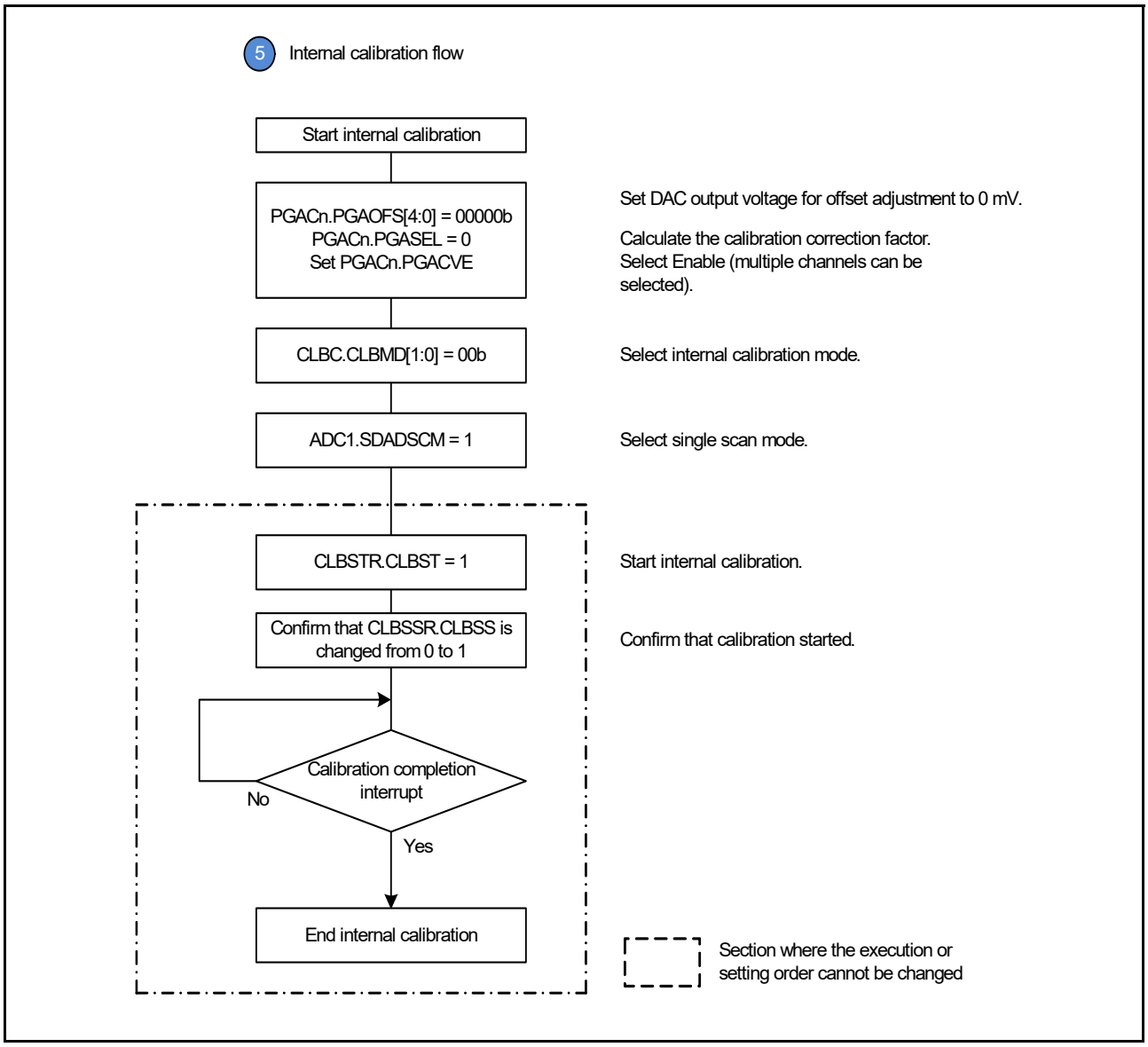


Figure 33.28 Internal calibration flow

33.4.6 External Calibration Flow

Figure 33.29 shows the flow for external calibration.

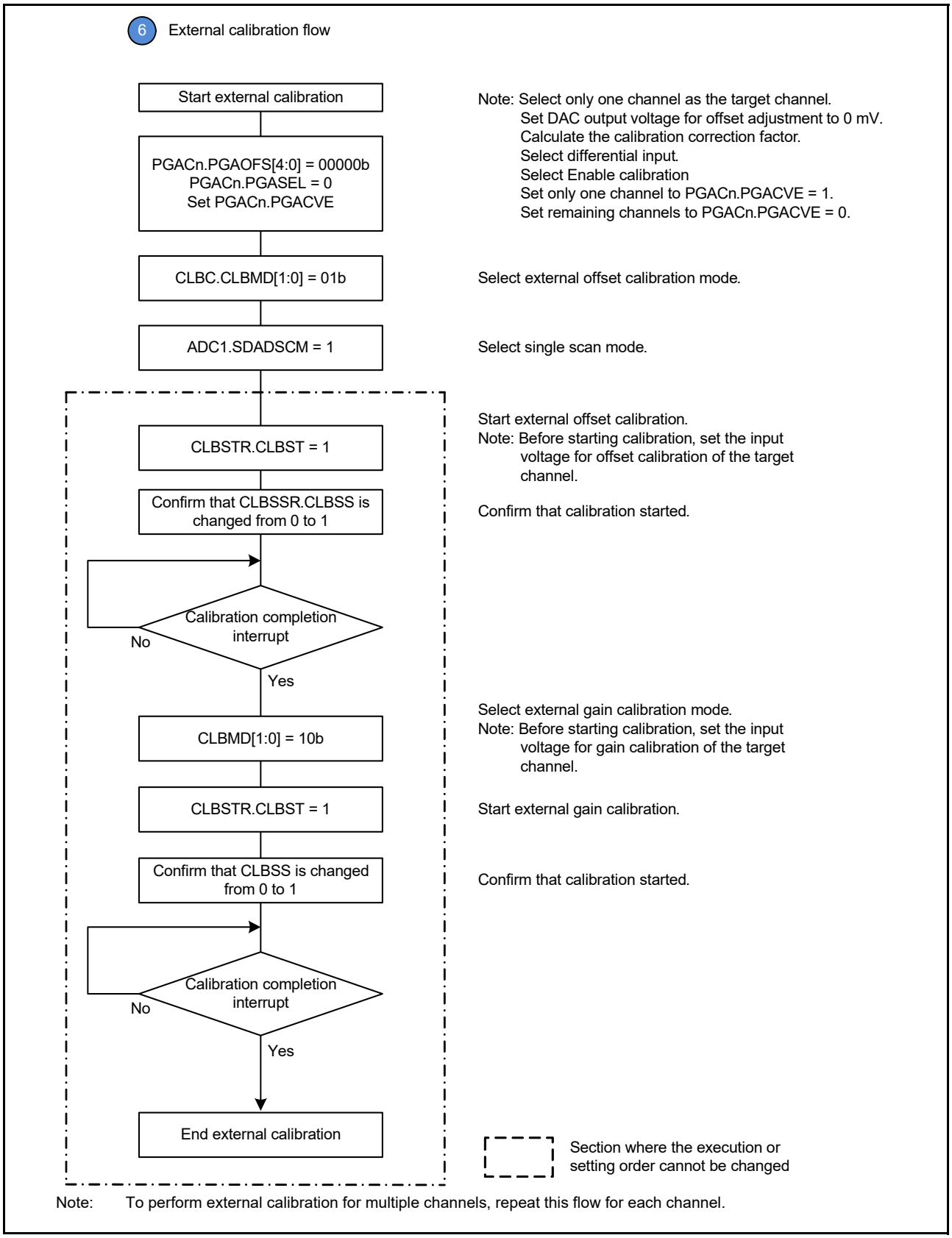


Figure 33.29 External calibration flow

33.4.7 SDADC24 Conversion Flow

Figure 33.30 shows the flow for SDADC24 conversion.

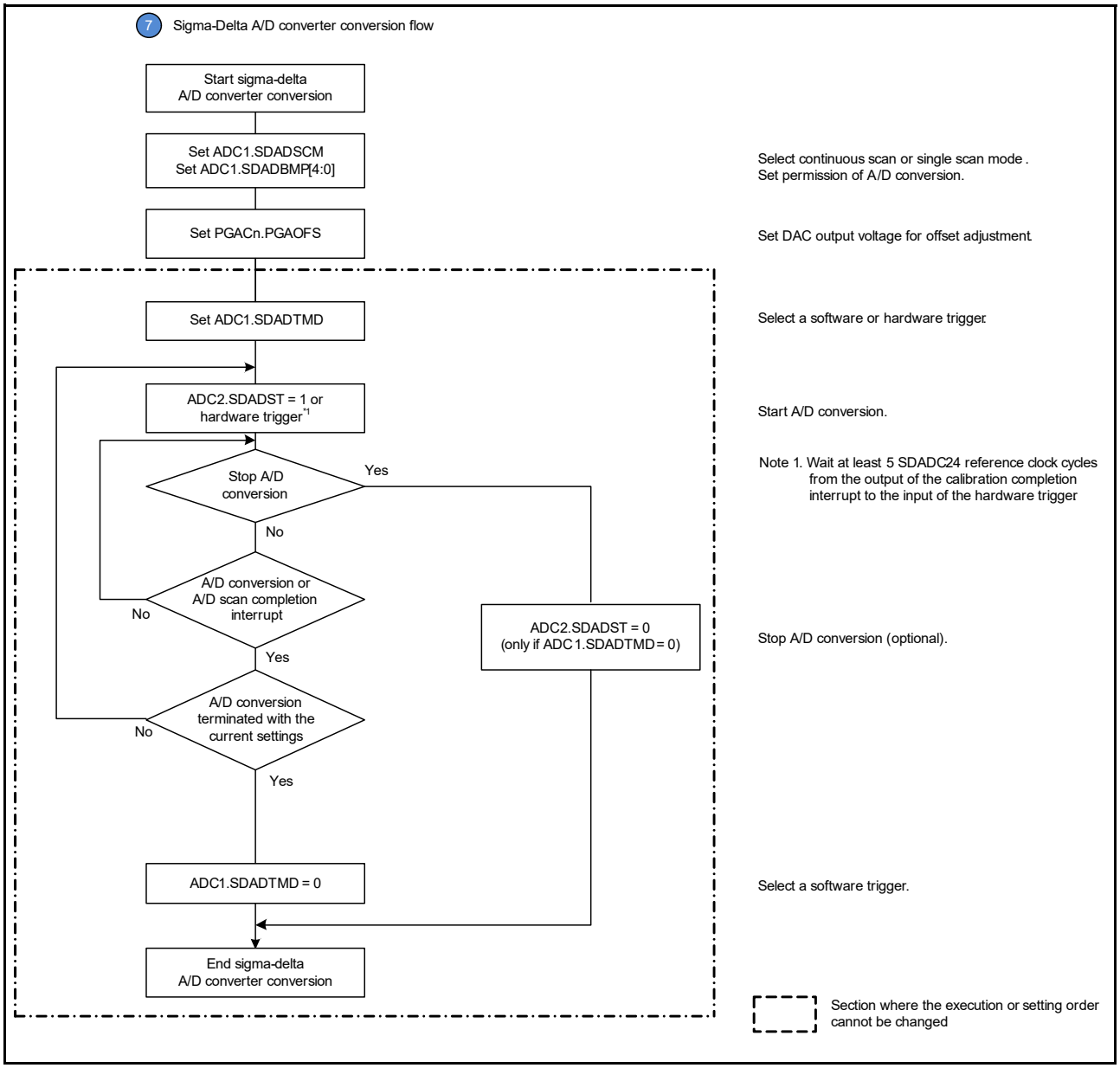


Figure 33.30 SDADC24 conversion flow

33.4.8 SDADC24 Stop Setting Flow

Figure 33.31 shows the flow for SDADC24 stop setting.

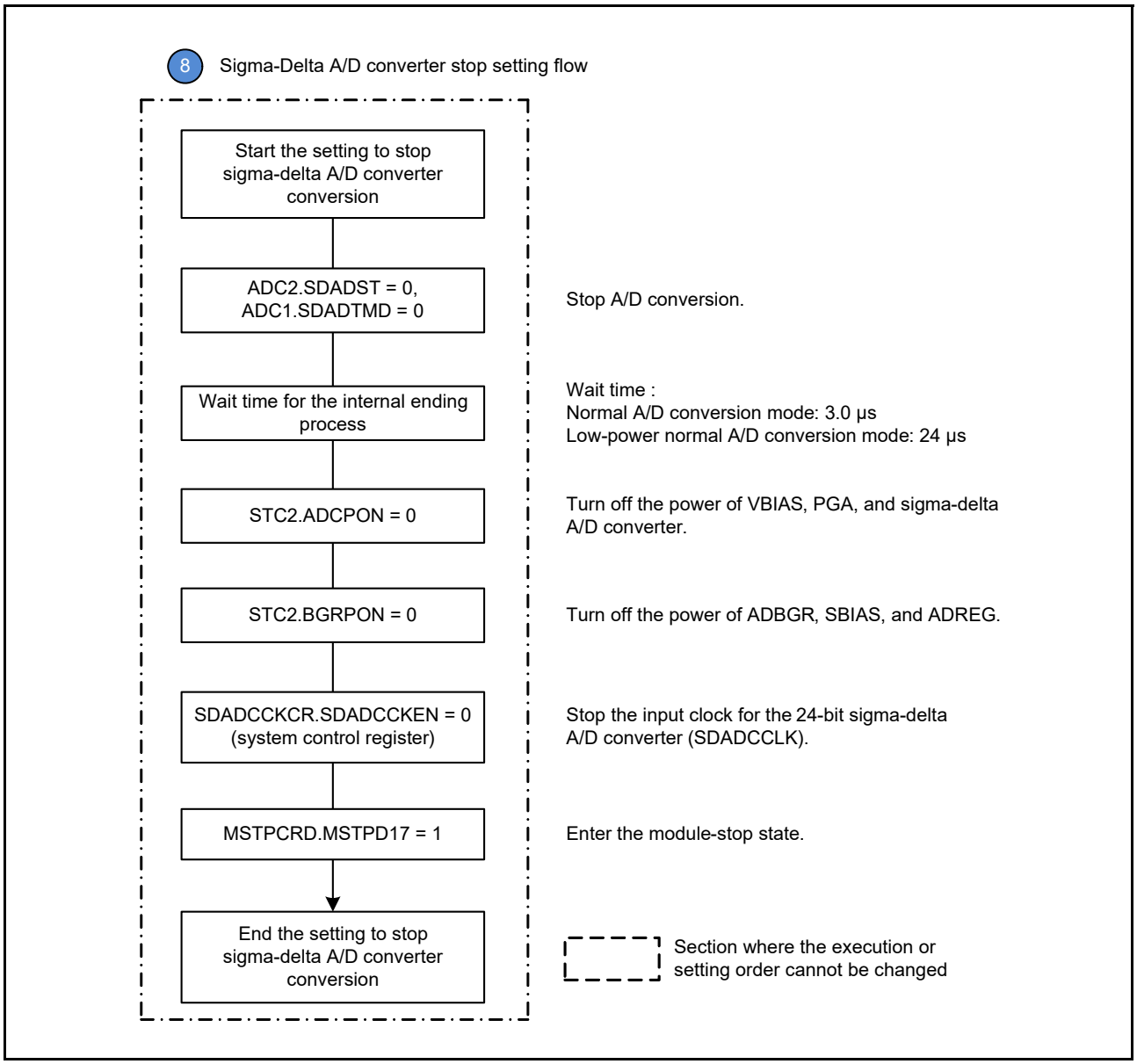


Figure 33.31 SDADC24 stop setting flow

33.4.9 Recalibration Omitted Flow

Figure 33.32 shows the flow for omitting recalibration. However, when there is a large temperature change or there is a change in the user environment or use conditions, it is necessary to perform recalibration.

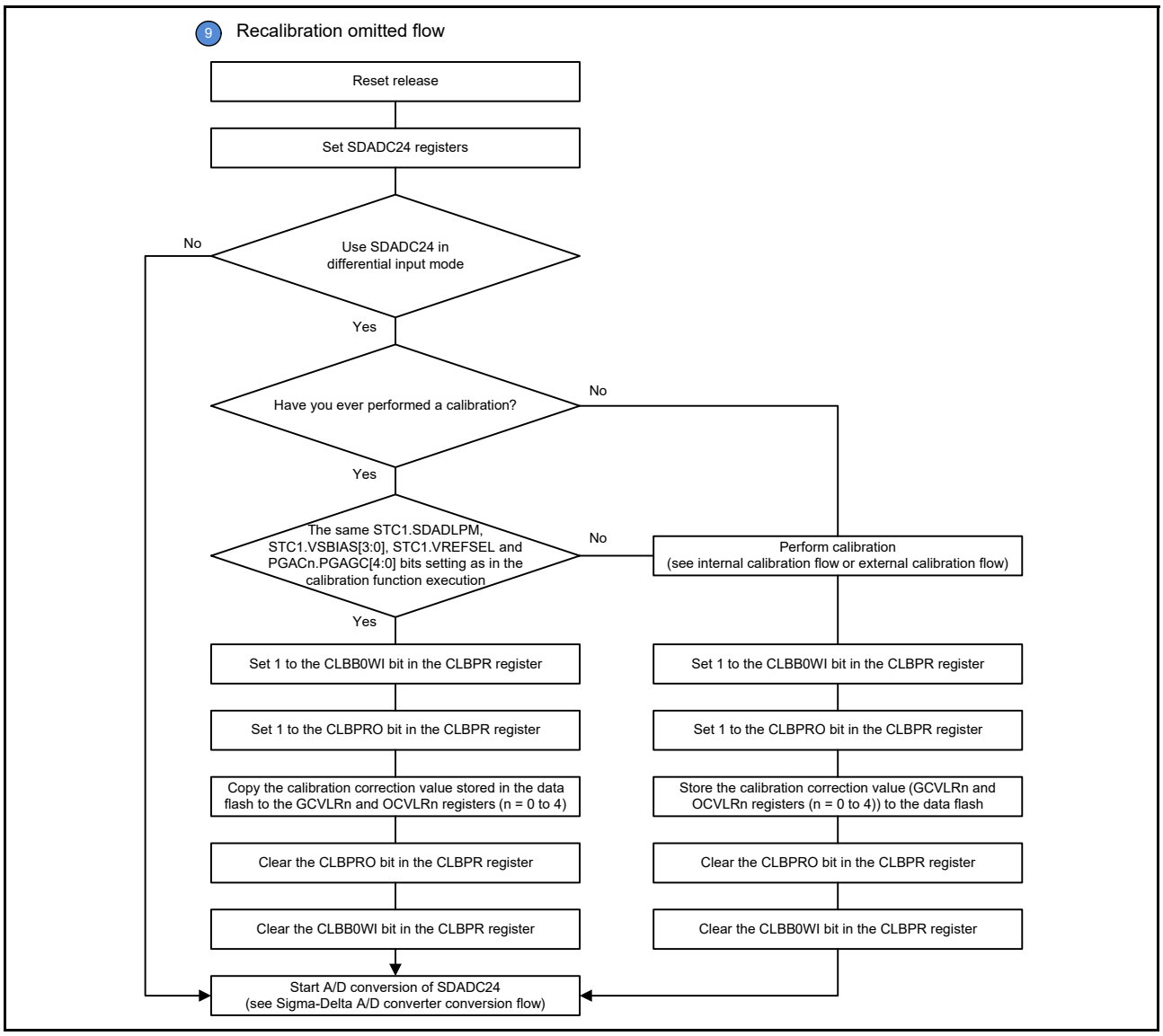
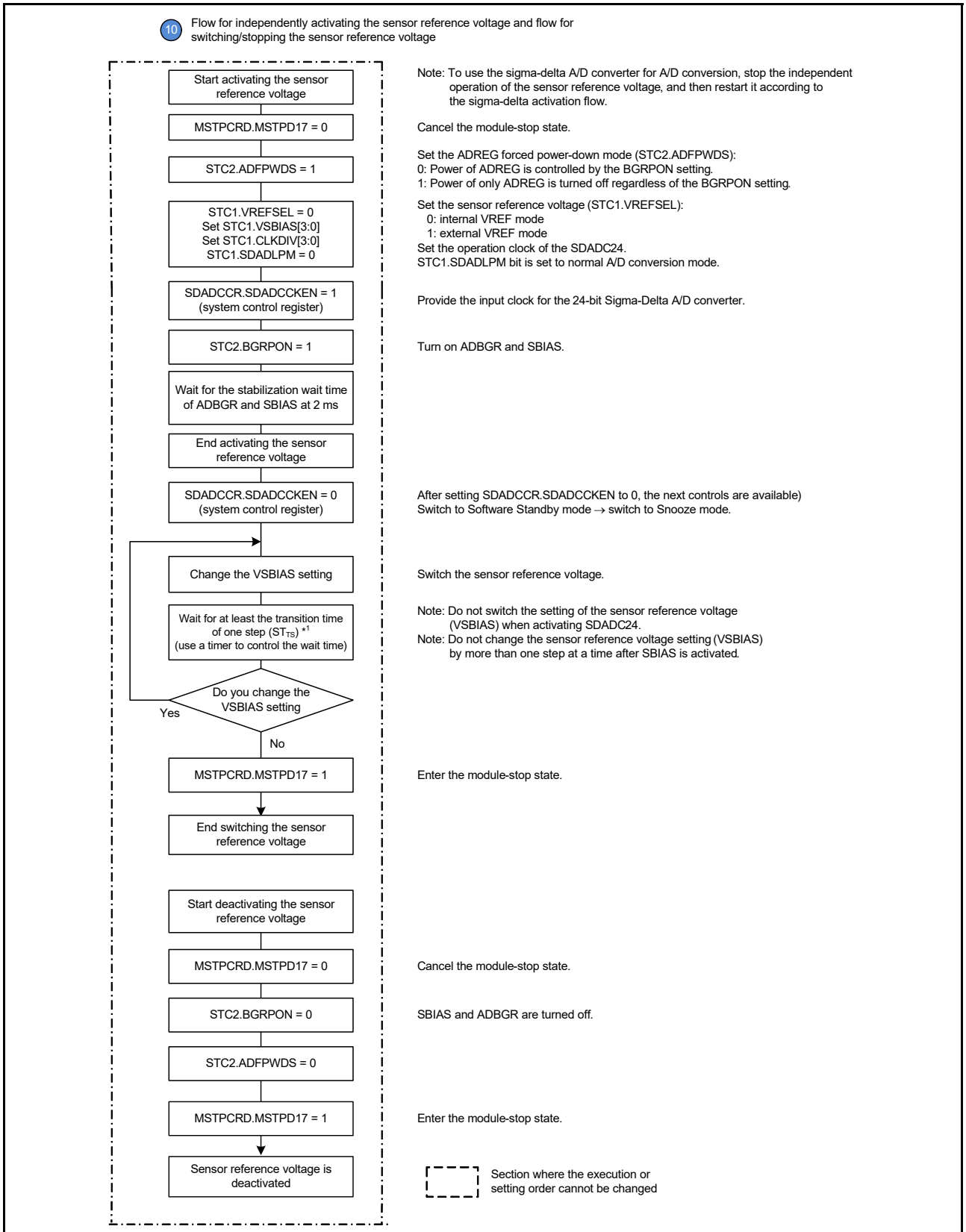


Figure 33.32 Recalibration omitted flow

33.4.10 Flows for Independently Activating and for Switching/Stopping the Sensor Reference Voltage

Figure 33.33 shows the flows for independently activating, and for switching/stopping the sensor reference voltage.



Note 1. For details of the ST_{TS} time, see section 47, Electrical Characteristics.

Figure 33.33 Flows for independently activating and for switching/stopping the sensor reference voltage

33.5 Usage Notes

33.5.1 Notes on Reading Data Registers

The following registers must be read in word units:

- Sigma-Delta A/D Converter Conversion Result Register (ADCR)
- Sigma-Delta A/D Converter Average Value Register (ADAR).

If a register is read multiple times in byte units or halfword units, the A/D converted value initially read might not match with the subsequent A/D converted value that is read. To prevent this, do not read the data registers in byte units or halfword units.

33.5.2 Settings for the Module-Stop State

The Module Stop Control Register D (MSTPCRD) can enable or disable SDADC24 operation. The SDADC24 is initially stopped after a reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

33.5.3 Restrictions on Entering the Low Power States

Before entering the module-stop state or Software Standby mode, be sure to stop A/D conversion and the SDADC24 clock (SDADCCLK).

33.5.4 A/D Conversion Operation Mode and SDADC24 Reference Clock Division Setting

The A/D conversion operation mode setting (STC1.SDADLPM bit setting) and the SDADC24 reference clock division setting (STC1.CLKDIV[3:0] bit settings) must be changed before the ADBGR power is turned on (STC2.BGRPON = 0).

In addition, set the STC1.CLKDIV[3:0] bits so that the SDADC24 reference clock is output at 4 MHz. If the clock is not output at 4 MHz, A/D conversion cannot be performed normally.

33.5.5 Restrictions on SBIAS Operation

The reference voltage for sensors (VSBIAS) cannot be changed more than one step at one time. When the reference voltage for sensors (STC1.VSBIAS) is changed more than one step at a time, wait for ST_{TS} *1 before the reference voltage is changed with each step.

During the MCU normal mode, turn SBIAS on then supply SDADCCLK while waiting for stabilization time. It is possible to change to Software Standby mode and Snooze mode after stabilization. When A/D conversion is activated following the SBIAS independent operation, stop SBIAS then reactivate the SDADC24 power supply from the initial.

Note 1. For details of the ST_{TS} time, see section 47, Electrical Characteristics.

33.5.6 Oversampling Ratio, PGA Gain, and Offset Voltage (d_{OFR}) Setting

In single-ended input mode, use the following settings:

d_{OFR} = 0 mV, G_{TOTAL} = 1, oversampling ratio = 256

In addition, set d_{OFR} = 0 mV for the following operations:

- Disconnection detection assist
- Self-diagnosis of PGA offset
- Calibration.

33.5.7 Restrictions on the Multiplexer 4 (Internal OPAMP)

- OPAMP0 and OPAMP1 must be operated when A/D conversion from the input multiplexer 4 is allowed
- Analog signals externally input to the AMP0O and AMP1O pins cannot be A/D converted by the SDADC24
- P015 and P502 pins cannot be used as general I/O pins, IRQ1 input, or CTS0 function when the SDADC24 is in use.

33.5.8 Relationship between the SDADC24, ADC16 and ACMPHS

Table 33.14 lists the SDADC24 conversion targets that should not be selected as ADC16 and ACMPHS inputs during SDADC24 conversion.

Table 33.14 ADC16 and ACMPHS pins that should not be selected during SDADC24 conversion

Target of SDADC24 conversion	ADC16	ACMPHS
ANSD0P	AN016	IVCMP2
ANSD0N	AN017	IVREF2
ANSD1P	AN018	-
ANSD1N	AN019	-
ANSD2P	AN020	-
ANSD2N	AN021	-
ANSD3P	AN022	-
ANSD3N	AN023	-

34. 12-Bit D/A Converter (DAC12)

34.1 Overview

The MCU provides a 12-bit D/A Converter (DAC12).

Table 34.1 lists the DAC12 specifications, Figure 34.1 shows a block diagram, and Table 34.2 lists the I/O pins.

Table 34.1 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 ADC16, 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)	D/A conversion can be started on input of an event signal

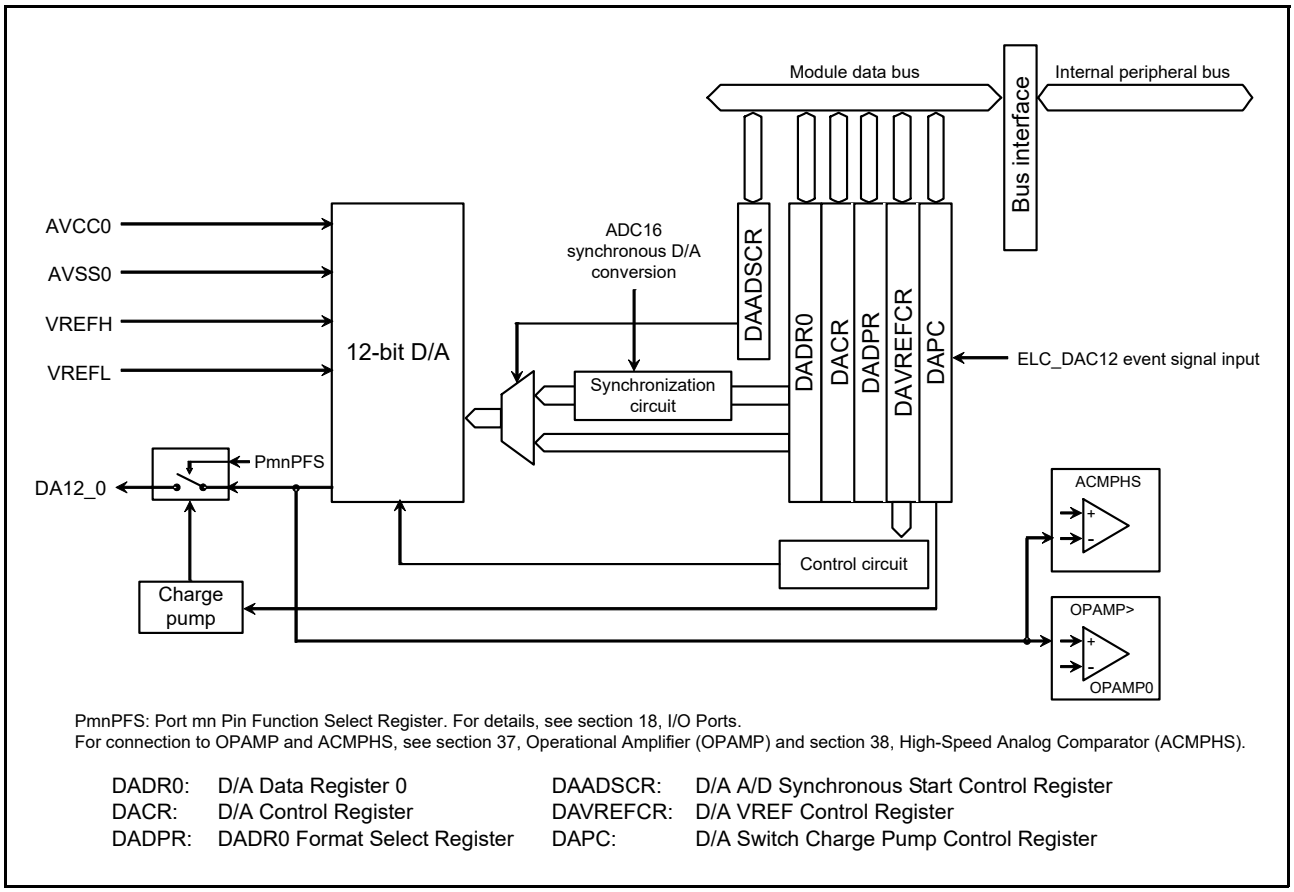


Figure 34.1 DAC12 block diagram

Table 34.2 DAC12 pin configuration (1 of 2)

Pin name	I/O	Function
AVCC0	Input	Analog power supply pin for ADC16, DAC12, DAC8, ACMPHS, and OPAMP. Connect to VCC when these modules are not used.
AVSS0	Input	Analog ground pin for ADC16, DAC12, DAC8, ACMPHS, and OPAMP. Connect to VSS when these modules are not used.

Table 34.2 DAC12 pin configuration (2 of 2)

Pin name	I/O	Function
VREFH	Input	Analog reference top voltage supply pin for DAC12
VREFL	Input	Analog reference ground pin for DAC12
DA12_0	Output	Analog output pin

34.2 Register Descriptions

34.2.1 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 register 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.

34.2.2 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 D/A conversion and analog output (DA12_0) 1: Enable D/A conversion and analog output (DA12_0).	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Only set this register while the ADC16 is halted when DAADSCR.DAADST = 1 (interference reduction between D/A and A/D conversions is enabled). Only set DACR while ADCSR.ADST = 0 and after selecting the software trigger as the ADC16 trigger.

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 ADC16 is halted (ADCSR.ADST = 0) and the software trigger is selected as the ADC16 trigger.

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_DAC12 event occurs, and output of the D/A conversion results starts.

34.2.3 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

34.2.4 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 ADC16 operation (disable interference reduction between D/A and A/D conversion) 1: Synchronize DAC12 operation with ADC16 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 start with the synchronous D/A conversion enable input signal from the ADC16 trigger.

Only set this register while the ADC16 is halted (ADCSR.ADST = 0) and the software trigger is selected as the ADC16 trigger.

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 the ADC16. When the DADR0 register value is modified, D/A conversion does not start until the ADC16 completes A/D conversion.

Only set this bit while the ADC16 is halted (ADCSR.ADST = 0) and the software trigger is selected as the ADC16 trigger. The event link function cannot be used when the DAADST bit is set to 1. Stop the event link function by setting the ELSR12 register of the ELC.

34.2.5 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 1 1 0: VREFH/VREFL selected. Other settings are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The D/A VREF Control Register (DAVREFCR) selects the reference voltage of the DAC12.

REF[2:0] bits (D/A Reference Voltage Select)

The REF[2:0] bits select the reference voltage of the DAC12. 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. Do not rewrite this register during A/D conversion using the ADC16. If this register is rewritten, the accuracy of A/D conversion is not guaranteed.

34.2.6 D/A Switch Charge Pump Control Register (DAPC)

Address(es): DAC12.DAPC 4005 E009h

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

Bit	Symbol	Bit name	Description	R/W
b0	PUMPEN	Charge Pump Enable	0: Charge pump disabled 1: Charge pump enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PUMPEN bit (Charge Pump Enable)

The PUMPEN bit enables or disables charge pump. When AVCC0 < 2.7 V and DAC12 output level is output to a pin, set the MOCOCR.MCSTP bit to 0 and then set the DAPC.PUMPEN bit to 1. When using the DAC12 output as ACMPHS or OPAMP input, set the DAPC.PUMPEN bit to 0. Setting the MOCOCR.MCSTP bit is optional.

Note: Set the DAPC register while no D/A output is selected by the Peripheral Select bit (PSEL[4:0] bits) in the Port mn Pin Function Select register. For details on the PSEL[4:0] bits, see section 18, I/O Ports.

34.3 Operation

The DAC12 includes D/A conversion circuit. 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. Figure 34.2 shows the timing of this operation.

1. Set the MOCOCR.MCSTP bit to 0 (MOCO operating).
2. Set the DAPC.PUMPEN bit to 1 (charge pump enabled). When operating in subosc-speed mode, the following additional steps are required:
   a. Set the MOCOCR.MCSTP bit to 1.
   b. Power control mode change to subosc-speed mode.
3. Wait for the charge pump stabilization time*1.
4. Select the D/A output terminal to analog mode (controlled by the PmnPFS.ASEL bit settings).
5. Select the D/A output terminal (controlled by the PmnPFS.PMR and PmnPFS.PSEL[4:0] bits settings).
6. When the charge pump is enabled, wait for the switching stabilization time*1.
7. Set the data for D/A conversion in the DADR0 register and the data format in the DADPR.DPSEL bit.
8. Set the DACR.DAOE0 bit to 1 to start D/A conversion. The conversion result is output from the analog output pin DA12_0 after the conversion time t_{DCONV} elapses. The conversion result continues to be output until DADR0 is written to again or the DAOE0 bit is set to 0. The output value is expressed by the following formula:

$$\frac{\text{Setting value of DADR0}}{4096} \times \text{Reference voltage}$$

9. To start another conversion, write another value to DADR0. The conversion result is output after the conversion time t_{DCONV} elapses.
   When the DAADSCR.DAADST bit is 1 (interference reduction between D/A and A/D conversion is enabled), a maximum of one A/D conversion time is required for D/A conversion to start. When ADCLK is faster than the peripheral clock, a longer time might be required.
10. To disable analog output, set the DAOE0 bit to 0.

Note: Steps 1. to 6. are for when AVCC0 < 2.7 V and the DAC12 output level is output to a pin. When AVCC0 ≥ 2.7 V and the DAC12 output level is to a pin, steps 1., 2., 3., and 6. are not necessary. Steps 1. to 6. must not be performed when using DAC12 output as ACMPHS or OPAMP input. The MOCOCR.MCSTP bit setting is optional.

Note 1. See section 47, Electrical Characteristics for details of the charge pump stabilization time and the switching stabilization time.

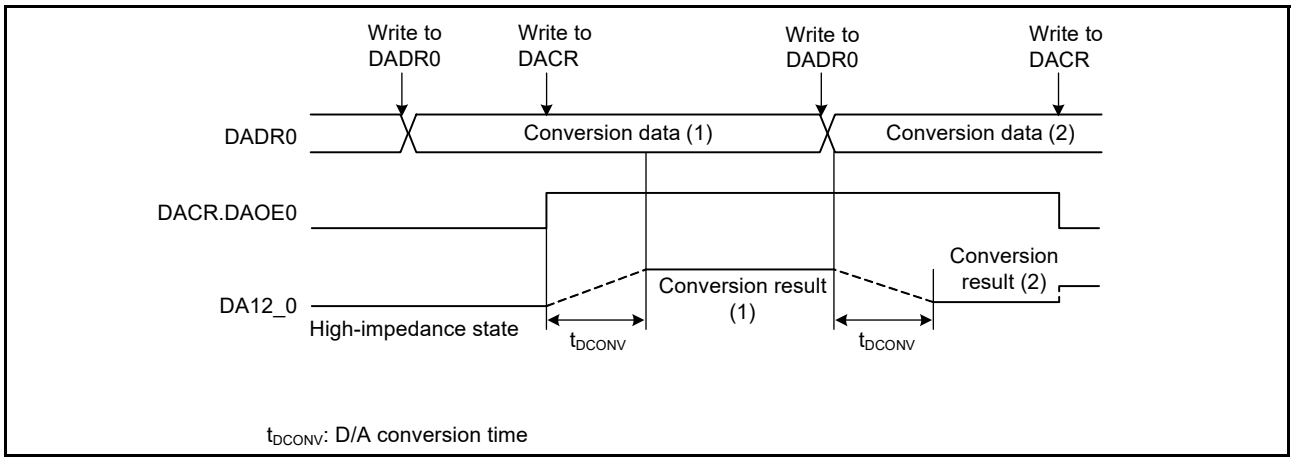


Figure 34.2 Example DAC12 operation

34.3.1 MOCO Stop Procedure after D/A Conversion Disabled

The following procedure describes how to stop MOCO after D/A conversion is disabled (DACR.DAOE0 = 0) when AVCC0 < 2.7 V and the DAC12 output level is to a pin.

1. Set the D/A output terminal to port or another function pin setting. When operating in subosc-speed mode, the following additional steps are required:
   a. Change the power control mode from subosc-speed mode to another mode.
   b. Set the MOCOCR.MCSTP bit to 0.
2. Set the DAPC.PUMPEN bit to 0 (charge pump disabled).
3. Set the MOCOCR.MCSTP bit to 1 (MOCO stopped).

Note: The MOCO must not be stopped when DAC8.DACPC.DAPC = 1 or OPAMP.AMPCPC.PUMPnEN = 1 (n = 0, 1, 2).

34.3.2 Reducing Interference between D/A and A/D Conversion

When D/A conversion starts, the DAC12 generates inrush current. Because the DAC12 and ADC16 share the same analog power supply, the generated inrush current can interfere with ADC16 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 ADC16 is halted, D/A conversion starts in 1 PCLKB cycle
- If the DADR0 register is modified while the ADC16 is in progress (ADCSR.ADST = 1), D/A conversion starts on A/D conversion completion. Therefore, it takes up to one A/D conversion time 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 ADC16. Figure 34.3 shows the timing of this operation.

To perform D/A conversion in synchronization with the ADC:

1. Confirm that the ADC16 is halted and set the DAADSCR.DAADST bit to 1.
2. Confirm that the ADC16 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 ADC16 is halted (ADCSR.ADST = 0) when the DADR0 register is modified, D/A conversion starts in 1 PCLKB cycle
- If the ADC16 is in progress (ADCSR.ADST = 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.

Note: The A/D sampling time must be longer than 3 PCLKB cycles. For details on A/D sampling time, see section 32, 16-Bit A/D Converter (ADC16).

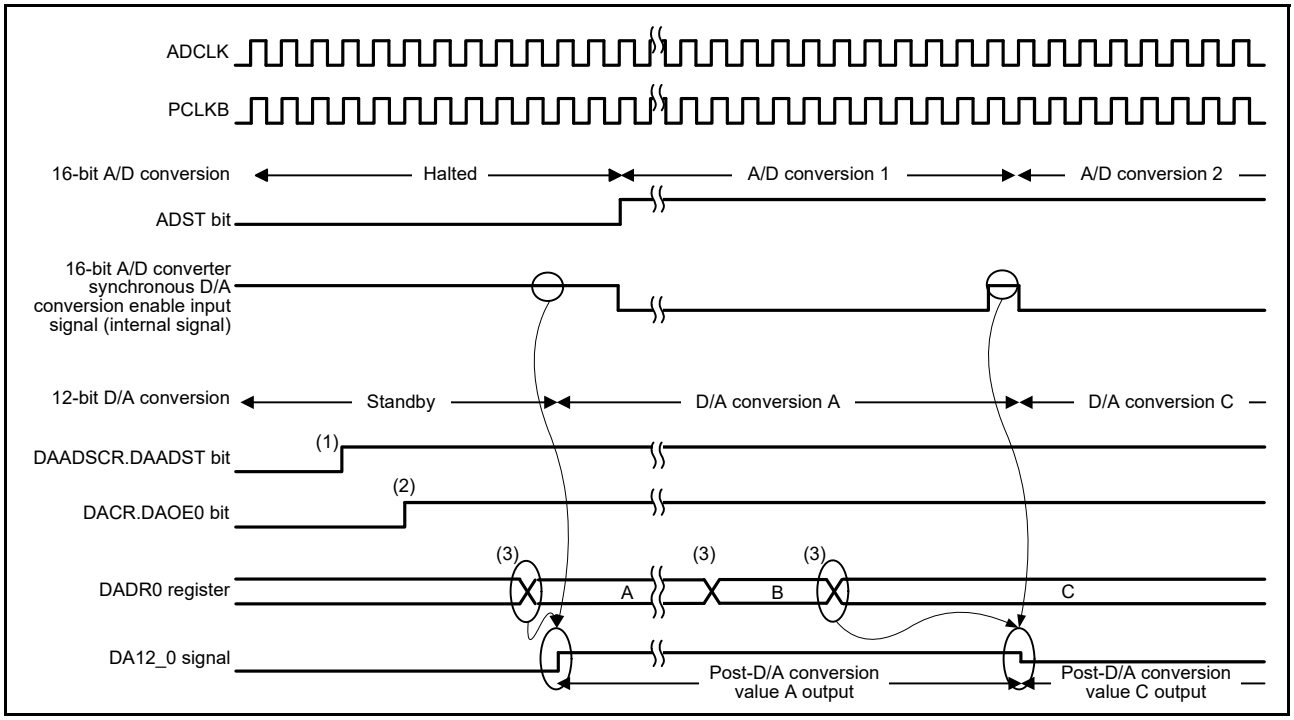


Figure 34.3 Example of conversion when DAC12 is synchronized with ADC16

34.4 Event Link Operation Setting Procedure

To set up an event link operation:

1. Set the DADPR.DPSEL bit and the data for D/A conversion in the DADR0 register.
2. Set the ELC_DAC12 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 is set to 1, and D/A conversion starts.
5. Set the ELSR12.ELS[7:0] bits to 00h to stop event link operation of the DAC12. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

34.5 Usage Notes on Event Link Operation

- When the event specified by the ELC_DAC12 event signal is generated while a write 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.

34.6 Usage Notes

34.6.1 Settings for the Module-Stop Function

The Module Stop Control register D (MSTPCRD) 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.

34.6.2 DAC12 Operation in Module-Stop State

When the MCU enters the module-stop state with D/A conversion enabled, the D/A outputs are saved, and the analog power supply current is the same as during D/A conversion. To reduce the analog power supply current in the module-stop state, disable D/A conversion by setting the DACR.DAOE0 and DAPC.PUMPEN bits to 0.

34.6.3 DAC12 Operation in Software Standby Mode

When the MCU enters Software Standby mode with D/A conversion enabled, the D/A outputs are saved, and the analog power supply current is the same as during D/A conversion. To reduce the analog power supply current in Software Standby mode, disable D/A conversion by setting the DACR.DAOE0 and DAPC.PUMPEN bits to 0.

34.6.4 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 ADC16 in the module-stop state. Doing so can halt D/A conversion in addition to A/D conversion.

34.6.5 D/A Converter Output

Currents on DA12_0 pin cannot be obtained because the output impedance of the D/A converter is high. If the load input impedance is low, insert a follower amplifier between the load and DA12_0 pin or use the on-chip OPAMP to obtain the current. Also make the wiring to the follower amplifier and the load as short as possible (because the output impedance is high). If the wiring becomes long, reconsider the design, for example, shielding the wiring with the ground trace.

34.6.6 DAC12 Output Pin during Charge Pump Enabled

When the charge pump is enabled and the D/A conversion result is output to DA12_0 pin, if any of the following settings is changed, it is necessary to wait for the switching stabilization time to DA12_0 pin output. For details on the switching stabilization time, see section 47, Electrical Characteristics.

- When changing the settings of the AMP2MS register in the OPAMP
- When changing the settings of the AMP2PS register in the OPAMP
- When changing the pin output setting of DAC8.

34.6.7 Connection of D/A Converter Output

DAC12 output can be connected to one of ACMPHS input, OPAMP input, and terminal output, but cannot be connected to two or more at the same time.

35. 8-Bit D/A Converter (DAC8)

35.1 Overview

The MCU provides an 8-bit D/A Converter (DAC8).

Table 35.1 lists the DAC8 specifications, and Figure 35.1 shows a block diagram.

Table 35.1 DAC8 specifications

Parameter	Specifications
Resolution	8 bits
Output channels	2 channels
Interference reduction between analog modules	Reduces interference between D/A and A/D conversion circuits. D/A converted data update timing is controlled by the synchronous D/A conversion enable input signal from the ADC16, which reduces the effect of DAC8 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)	D/A conversion can be started on input of an event signal to reduce power consumption

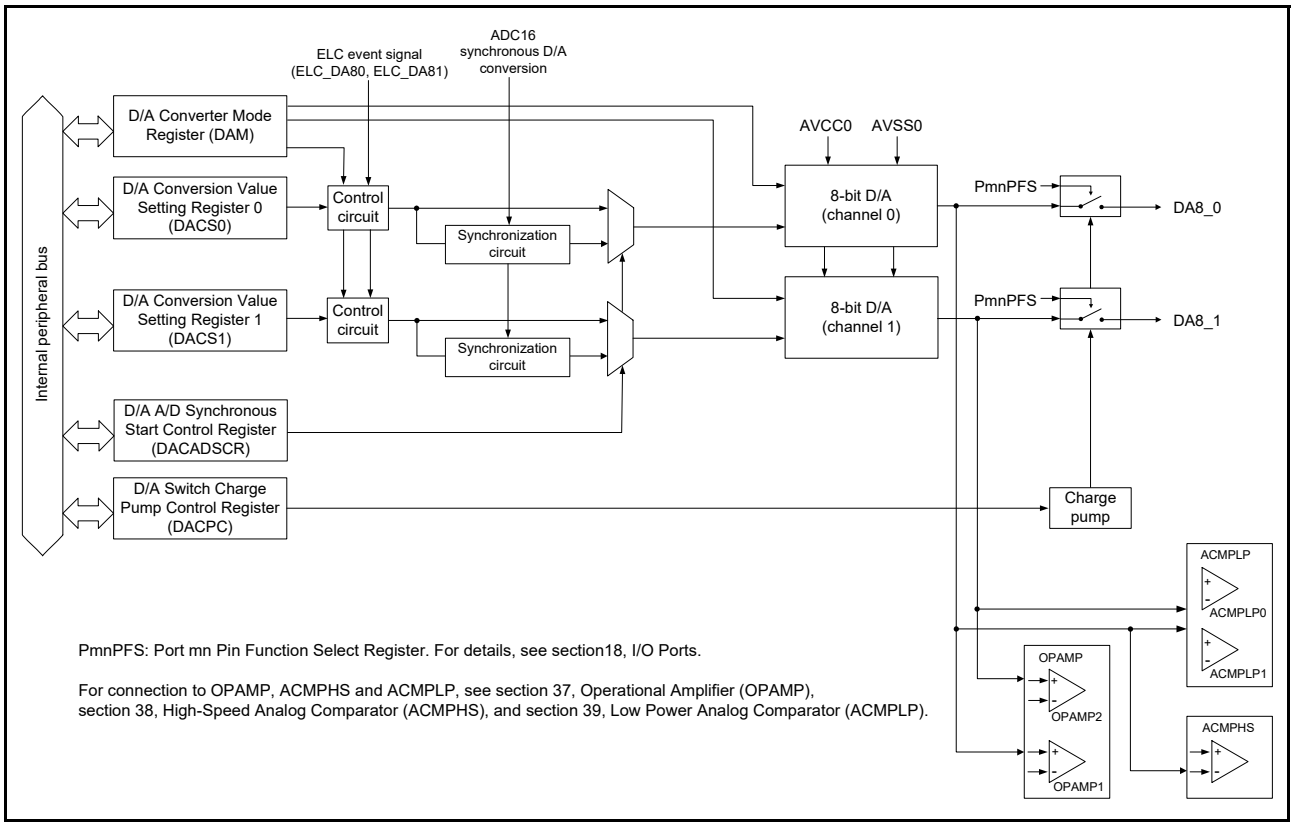


Figure 35.1 DAC8 converter block diagram

Table 35.2 lists the pin configuration of the DAC8.

Table 35.2 DAC8 pin configurations

Pin name	I/O	Function
AVCC0	Input	• Analog power and analog reference top voltage supply pin for ADC16, DAC12, DAC8, and ACMPHS • Connect to VCC when these modules are not used.
AVSS0	Input	• Analog ground and analog reference ground supply pin for ADC16, DAC12, DAC8, and ACMPHS • Connect to VSS when these modules are not used.
DA8_0	Output	Channel 0 analog output pin
DA8_1	Output	Channel 1 analog output pin

35.2 Register Descriptions

35.2.1 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 to store data for D/A conversion. When D/A conversion is enabled, the value in the DACSn register is converted and output to an analog output pin.

35.2.2 D/A Converter Mode Register (DAM)

Address(es): DAC8.DAM 4009 E003h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DACE1	DACE0	—	—	DAMD1	DAMD0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DAMD0	D/A Operation Mode Select 0	0: Channel 0 for normal operation mode 1: Channel 0 for real-time output mode (event link).	R/W
b1	DAMD1	D/A Operation Mode Select 1	0: Channel 1 for normal operation mode 1: Channel 1 for real-time output mode (event link).	R/W
b3, b2	—	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

DAMDn bit (D/A Operation Mode Select n) (n = 0, 1)

The DAMDn bit selects the operation mode of D/A conversion.

DACEn bit (D/A Operation Enable n) (n = 0, 1)

The DACEn bit enables or disables D/A conversion.

When interference reduction between D/A and A/D conversions is enabled (DACADSCR.DACADST = 1), only set the DACEn bits while the ADC16 is halted (ADCSR.ADST = 0) and the software trigger is selected as the ADC16.

35.2.3 D/A A/D Synchronous Start Control Register (DACADSCR)

Address(es): DAC8.DACADSCR 4009 E006h

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

Bit	Symbol	Bit name	Description	R/W
b0	DACADST	D/A A/D Synchronous Conversion	0: Do not synchronize DAC8 operation with ADC16 operation (disable interference reduction between D/A and A/D conversion) 1: Synchronize DAC8 operation with ADC16 operation (enable interference reduction between D/A and A/D conversion).	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

To reduce interference between the D/A and A/D conversion, the DACADSCR register switches on or off the synchronization of the D/A conversion start with the synchronous D/A conversion enable input signal from the ADC16 trigger.

Only set this register while the ADC16 is halted (ADCSR.ADST = 0) and the software trigger is selected as the ADC16 trigger.

DACADST bit (D/A A/D Synchronous Conversion)

Setting the DACADST bit to 0 allows the DACSn register value to be converted into analog data at any time. Setting the DACADST bit to 1 selects synchronous of D/A conversion with the synchronous D/A conversion enable input signal from the ADC16. This means that even if the DACSn register value is modified, D/A conversion does not start until the ADC16 completes A/D conversion.

Only set this bit while the ADC16 is halted (ADCSR.ADST = 0) and the software trigger is selected as the ADC16 trigger. The event link function cannot be used when the DACADST bit is set to 1. Stop the event link function by setting the DAM.DAMDn bit.

The setting of the DACADST bit is common to channels 0 and 1 of the DAC8.

35.2.4 D/A Switch Charge Pump Control Register (DACPC)

Address(es): DAC8.DACPC 4009 E007h

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

Bit	Symbol	Bit name	Description	R/W
b0	PUMPEN	Charge Pump Enable	0: Charge pump disabled 1: Charge pump enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PUMPEN bit (Charge Pump Enable)

The PUMPEN bit enables or disables charge pump. When AVCC0 < 2.7 V and DAC8 output level is output to a pin, set the MOCOCR.MCSTP bit to 0 and then set the DACPC.PUMPEN bit to 1. When using DAC8 output as ACMPHS, ACMPLP, or OPAMP input, set the DACPC.PUMPEN bit to 0. Setting the MOCOCR.MCSTP bit is optional.

Note: Set the DACPC register while no D/A output is selected by the Peripheral Select bit (PSEL[4:0] bits) in the Port mn Pin Function Select register. For details on the PSEL[4:0] bits, see section 18, I/O Ports.

35.3 Operation

35.3.1 Normal Operation Mode

D/A conversion is performed using the write operation to the DACSn (n = 0, 1) register as a trigger.

The following procedure describes an operation example when performing D/A conversion for channel 0.

1. Set the MOCOCR.MCSTP bit to 0 (MOCO operating).
2. Set the DACPC.PUMPEN bit to 1 (charge pump enabled). When operating in subosc-speed mode, the following additional steps are required:
   a. Set the MOCOCR.MCSTP bit to 1.
   b. Change power control mode to subosc-speed mode.
3. Wait for the charge pump stabilization time*1.
4. Select the D/A output terminal to analog mode (controlled by the PmnPFS.ASEL bit settings).
5. Select the D/A output terminal (controlled by the PmnPFS.PMR and PmnPFS.PSEL[4:0] bits settings).
6. When the charge pump is enabled, wait for the switching stabilization time*1.
7. Set the DAM.DAMD0 bit to 0 (normal operation mode).
8. Set the data for D/A conversion to the DACS0 register.
9. Set the DAM.DACE0 bit to 1 to start D/A conversion. The conversion result is output from the analog output pin (DA8_0) after the conversion time elapses. The conversion result continues to be output until DACS0 register is written to again or the DAM.DACE0 bit is set to 0.
   The output value is expressed by the following formula:

$$\frac{\text{Setting value of DACS0}}{256} \times \text{Reference voltage}$$

10. To start another conversion, write another value to DACS0 register. The conversion result is output after the conversion time elapses.
    When the DACADSCR.DACADST bit is 1 (interference reduction between D/A and A/D conversion is enabled), a maximum of one A/D conversion time is required for D/A conversion to start. When ADCLK is faster than the peripheral clock, a longer time might be required.
11. To disable analog output, set the DAM.DACE0 bit to 0.

Note: If the DAM.DACE0 bit is set to 1, 0, and then 1 in order, after the last 1 is set and the conversion time elapses, an analog voltage is output to the DA8_0 pin.

Note: If the DACS0 register is rewritten during the conversion time, the current D/A conversion is canceled and conversion is started again with the rewritten value.

Note: Steps 1. to 6. is performed when AVCC0 < 2.7 V and the DAC8 output level is output to a pin. When AVCC0 ≥ 2.7 V and the DAC8 output level is output to a pin, steps 1., 2., 3. and 6. are not necessary. Steps 1. to 6. must not be performed when using DAC8 output as ACMPHS, ACMPLP, or OPAMP input. The MOCOCR.MCSTP bit setting is optional.

Note 1. See section 47, Electrical Characteristics for details of the charge pump stabilization time and the switching stabilization time.

35.3.2 Real-Time Output Mode (Event Link)

D/A conversion is performed on each channel using an event signal from the ELC as a trigger.

The following procedure describes an operation example when performing D/A conversion for channel 0.

1. Set the MOCOCR.MCSTP bit to 0 (MOCO operating).
2. Set the DACPC.PUMPEN bit to 1 (charge pump enabled). When operating in subosc-speed mode, the following additional steps are required:
   a. Set the MOCOCR.MCSTP bit to 1.
   b. Change power control mode to subosc-speed mode.
3. Wait for the charge pump stabilization time*1.
4. Set the Analog Input Enable bit in PmnPFS.ASEL to 1 (set the DAC8 to analog mode).
5. Select D/A output terminal (controlled by PmnPFS.PMR and PmnPFS.PSEL[4:0] bits settings).
6. When the charge pump is enabled, wait for the switching stabilization time*1.
7. Set the DAM.DAMD0 bit to 0 (normal operation mode).
8. Set the data for D/A conversion to the DACS0 register.
9. Set the DAM.DACE0 bit to 1 to start D/A conversion. The conversion result is output from the analog output pin (DA8_0) after the conversion time elapses.
10. Specify the trigger signal for real-time output mode with the ELSR19 register of the ELC.
11. Set the DAM.DAMD0 bit to 1 (real-time output mode).
12. Start the event source operation. Steps 1. to 12. are performed as the initial settings.
13. On generation of the trigger signal for real-time output mode, D/A conversion starts, and after the conversion time elapses, an analog voltage is output to the DA8_0 pin. The data for D/A conversion must be set to the DACS0 register before the next D/A conversion starts (before generation of the trigger signal for real-time output mode). To disable analog output, set the DAM.DACE0 bit to 0.

Note: Use of the event link function is prohibited when the DACADSCR.DACADST bit is set to 1 to reduce interference between D/A and A/D conversions.

Note: If the DAM.DACE0 bit is set to 1, 0, and then 1 in order, after the last 1 is set and the conversion time elapses, an analog voltage is output to the DA8_0 pin.

Note: When setting the trigger signal generation interval for the real-time output mode on the same channel, set the interval to a value longer than the conversion time. If the trigger signal is generated during the conversion time, the current D/A conversion is canceled, and then D/A conversion is restarted.

Note: Steps 1. to 6. are performed when AVCC0 < 2.7 V and the DAC8 output level is output to a pin. When AVCC0 ≥ 2.7 V and the DAC8 output level is output to a pin, steps 1., 2., 3. and 6. are not necessary. Steps 1. to 6. must not be performed when using DAC8 output as ACMPHS, ACMPLP, or OPAMP input. The MOCOCR.MCSTP bit setting is optional.

Note 1. See section 47, Electrical Characteristics for details of the charge pump stabilization time and the switching stabilization time.

35.3.3 MOCO Stop Procedure after D/A Conversion Disabled

The following procedure describes how to stop MOCO after D/A conversion is disabled (DAM.DACE0 = 0) when AVCC0 < 2.7 V and the DAC8 output level is output to a pin.

1. Set the D/A output terminal to port or another function pin setting. When operating in subosc-speed mode, the following additional steps are required:
   a. Change the power control mode from subosc-speed mode to another mode.
   b. Set the MOCOCR.MCSTP bit to 0.
2. Set the DACPC.PUMPEN bit to 0 (charge pump disabled).
3. Set the MOCOCR.MCSTP bit to 1 (MOCO stopped).

Note: The MOCO must not be stopped when DAC12.DAPC.DAPC = 1 or OPAMP.AMPCPC.PUMPnEN = 1 (n = 0, 1, 2).

35.3.4 Output Start Timing of the D/A Conversion Value

Figure 35.2 shows the output start timing of the D/A conversion value.

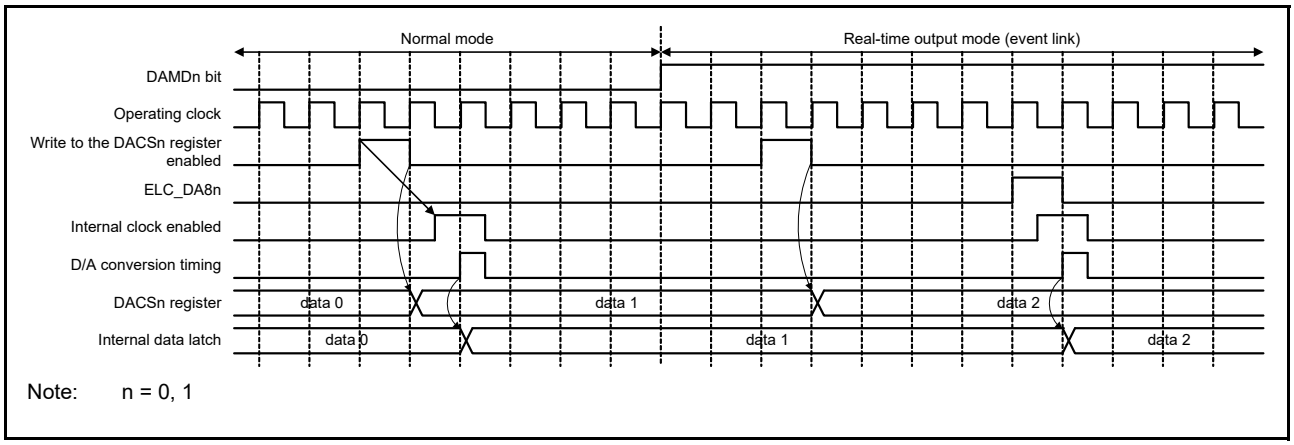


Figure 35.2 Output start timing of the D/A conversion value

Output start timing in normal mode:

- After the DACSn register is written, the internal data latch is set after 1 operating clock cycle*1. After the conversion time elapses, an analog voltage is output to the DA8_n pin.

Note 1. When the DACSn register is written while the A/D conversion stops with the DACADSCR.DACADST bit set to 1, the internal data is set after 2 operating clock cycles.

Output start timing in real-time output mode:

- After an event signal (ELC_DA8n) is received from the ELC, the internal data latch is set after 1 operating clock cycle. After the conversion time elapses, an analog voltage is output to the DA8_n pin.

35.3.5 Minimizing Interference between D/A and A/D Conversion

When D/A conversion starts, the DAC8 generates inrush current. Because the DAC8 and ADC16 share the same analog power supply, the generated inrush current can interfere with 16-bit A/D conversion.

While the DACADSCR.DACADST bit is 1, D/A conversion does not start immediately on updating the DACSn register.

Instead:

- If the DACSn register data is modified while the ADC16 is halted, D/A conversion starts in 2 PCLKB cycles
- If the DACSn register data is modified while the ADC16 is in progress (ADCSR.ADST = 1), D/A conversion starts on A/D conversion completion. Therefore, it takes up to one A/D conversion time for the DACSn register data update to reflect as the D/A conversion circuit output. Until the D/A conversion completes, the DACSn register value does not correspond to the analog output value.

When the DACADSCR.DACADST bit is 1, it is not possible to check through any software means whether the DACSn register value was D/A-converted.

The following sequence provides an example of D/A conversion, in which the DAC8 is synchronized with the ADC16.

1. Confirm that the ADC16 is halted and set the DACADSCR.DACADST bit to 1.
2. Confirm that the ADC16 is halted and set the DAM.DACEn bit to 1.
3. Set the DACSn 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 ADC16 is halted (ADCSR.ADST = 0) when the DACSn register is modified, D/A conversion starts in 2 PCLKB cycles

- If the ADC16 is in progress (ADCSR.ADST = 1) when the DACSn register is modified, D/A conversion starts on A/D conversion completion. If the DACSn register is modified twice during A/D conversion, the first update might not be converted.

Note: The A/D sampling time must be longer than 3 PCLKB cycles. For details on the A/D sampling time, see section 32, 16-Bit A/D Converter (ADC16).

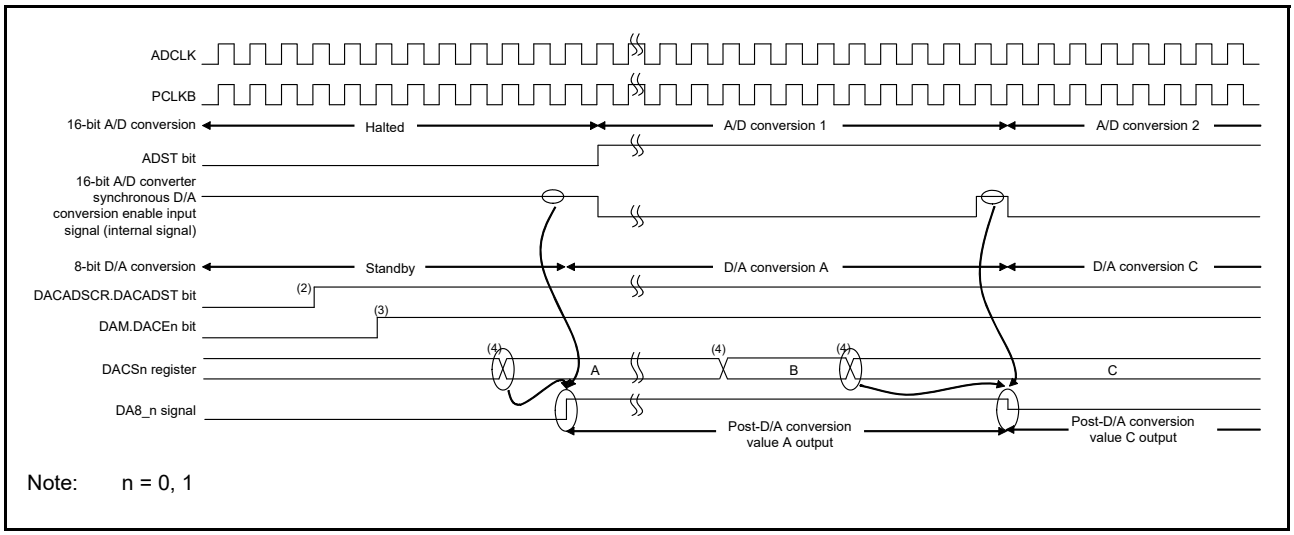


Figure 35.3 Example conversion when DAC8 is synchronized with ADC16

35.4 Usage Notes

35.4.1 Settings for the Module-Stop Function

The Module Stop Control register D (MSTPCRD) can enable or disable DAC8 operation. The DAC8 is stopped after a reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

35.4.2 DAC8 Operation in Module-Stop State

When the MCU enters the module-stop state with D/A conversion enabled, D/A outputs are saved, and the analog power supply current is the same as the one 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 DAM.DACEn bits to 0 and the DACPC.PUMPEN bit to 0.

35.4.3 DAC8 Operation in Software Standby Mode

When the MCU enters Software Standby mode with D/A conversion enabled, D/A outputs are saved, and the analog power supply current is the same as the one during D/A conversion. To reduce the analog power supply current in Software Standby mode, disable D/A conversion by setting the DAM.DACEn and DACPC.PUMPEN bits to 0.

35.4.4 Real-Time Output of the D/A Converter

In real-time output mode:

- Set the value to the DACSn register before the trigger signal for real-time output mode is generated. Do not change the setting value in the DACSn register while the trigger signal is being output.

When the MCU enters Software Standby mode in real-time output mode:

- Disable the ELC_DA8n before entering Software Standby mode. For details, see section 17, Event Link Controller (ELC).

35.4.5 D/A Converter Output

Currents on DA8_0 and DA8_1 pins cannot be obtained because the output impedance of the D/A converter is high. If the load input impedance is low, insert a follower amplifier between the load and DA8_0 and DA8_1 pins or use the on-chip OPAMP to obtain current. Also make the wiring to the follower amplifier and the load as short as possible because the output impedance is high. If the wiring becomes long, consider shielding the wiring with the ground trace.

35.4.6 When Not Using the D/A Converter

When not using the D/A converter, set the DAM.DACEn bit to 0 (output disabled), the DACSn register to 00h, and the DACPC.PUMPEN bit to 0, so that current does not flow and the current consumption can be reduced.

35.4.7 DAC8 Output Pin during Charge Pump Enabled

When the charge pump is enabled and the D/A conversion results are output to DA8_0 and DA8_1 pins, if any of the following settings is changed, it is necessary to wait for the switching stabilization time to DA8_0 and DA8_1 pins output. For details on the switching stabilization time, see section 47, Electrical Characteristics.

- When changing the settings of the AMP2MS register in the OPAMP
- When changing the settings of the AMP2PS register in the OPAMP
- When changing the pin output setting of 12-bit D/A conversion
- When changing the pin output setting of 8-bit D/A conversion of other channel.

35.4.8 Connection of D/A Converter Output

DAC8 channel n (n = 0, 1) output can be connected to one of ACMPHS input, ACMPLP input, OPAMP input, and terminal output, but is prohibited to connect to two or more at the same time.

The connection to the OPAMP input is possible even if the OPAMP external pin is not available.

For details, see section 18.6, Peripheral Select Settings for each Product.

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 ADC16 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	Specifications
Temperature sensor voltage output	Temperature sensor outputs a voltage to the ADC16

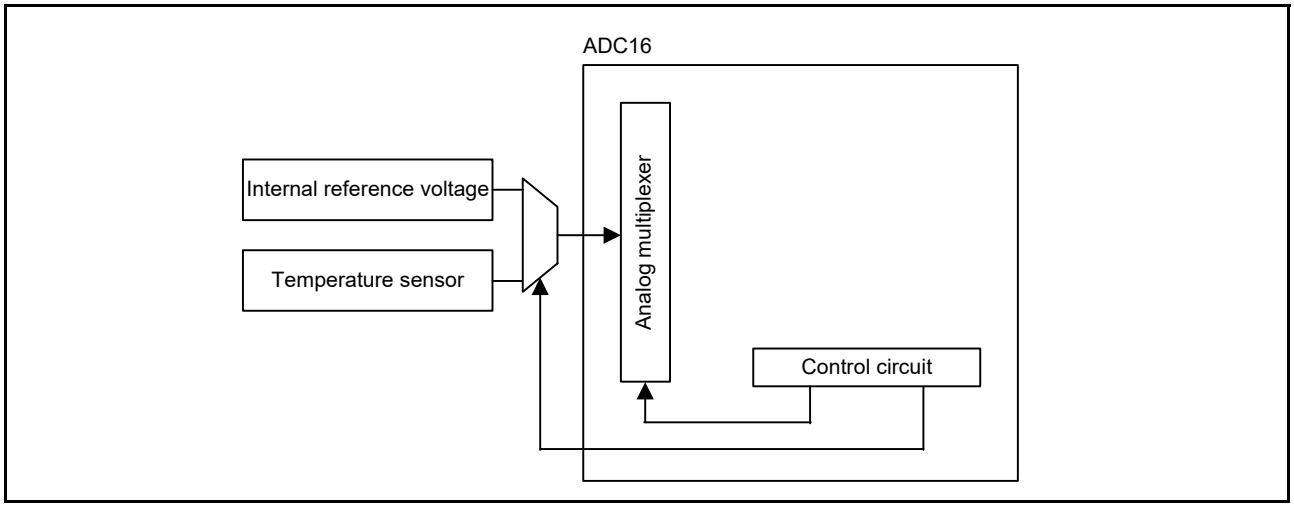


Figure 36.1 Temperature sensor block diagram

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 8 bits of the converted value	R

The TSCDRH register stores temperature sensor calibration data measured for each MCU at factory shipment.

Temperature sensor calibration data is a digital value obtained using the ADC16 to convert the voltage output by the temperature sensor under the condition $T_a = T_j = 125°C$ and AVCC0 = 3.3 V. The TSCDRH register stores the upper 8 bits of the converted value.

36.2.2 Temperature Sensor Calibration Data Register L (TSCDRL)

Address(es): TSN.TSCDRL 407E C228h

b7	b6	b5	b4	b3	b2	b1	b0
TSCDRL[7:0]							

Value after reset: Unique value for each chip

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 TSCDRL register stores temperature sensor calibration data measured for each MCU at factory shipment.

Temperature sensor calibration data is a digital value obtained using the ADC16 to convert the voltage output by the temperature sensor under the condition $T_a = T_j = 125°C$ and AVCC0 = 3.3 V. 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 ADC16. To obtain the die temperature, convert this value into the temperature.

36.3.1 Preparation for Using the Temperature Sensor

The temperature (T) is proportional to the sensor voltage output (Vs), so temperature is calculated with the following formula:

T = (Vs - V1) / Slope + T1

T: Measured temperature (°C)

Vs: Voltage output by the temperature sensor when temperature is measured (V)

T1: Temperature experimentally measured at one point (°C)

V1: Voltage output by the temperature sensor when T1 is measured (V)

T2: Temperature experimentally measured at a second point (°C)

V2: Voltage output by the temperature sensor when T2 is measured (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 ADC16 to measure the voltage V1 output by the temperature sensor at temperature T1.
2. Use the ADC16 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 slope given in Table 47.56 of section 47, Electrical Characteristics, use the ADC16 to measure the voltage V1 output by the temperature sensor at temperature T1, then calculate the temperature characteristic using the following formula:

T = (Vs - V1) / Slope + T1

T: Measured temperature (°C)

Vs: Voltage output by the temperature sensor when the temperature is measured (V)

T1: Sample temperature measurement at first point (°C)

V1: Voltage output by the temperature sensor when T1 is measured (V)

However, this method produces less accurate temperature results than measurement at two points.

In this MCU, the TSCDRH and TSCDRL registers store the temperature value (CAL125) of the temperature sensor measured under the condition $T_a = T_j = 125°C$ and AVCC0 = 3.3 V. By using 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 × CAL125 / 32768 [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 slope given in Table 47.56 ÷ 1000 (V / °C).

Figure 36.2 shows the error in the measured temperature. The variation range is 3σ.

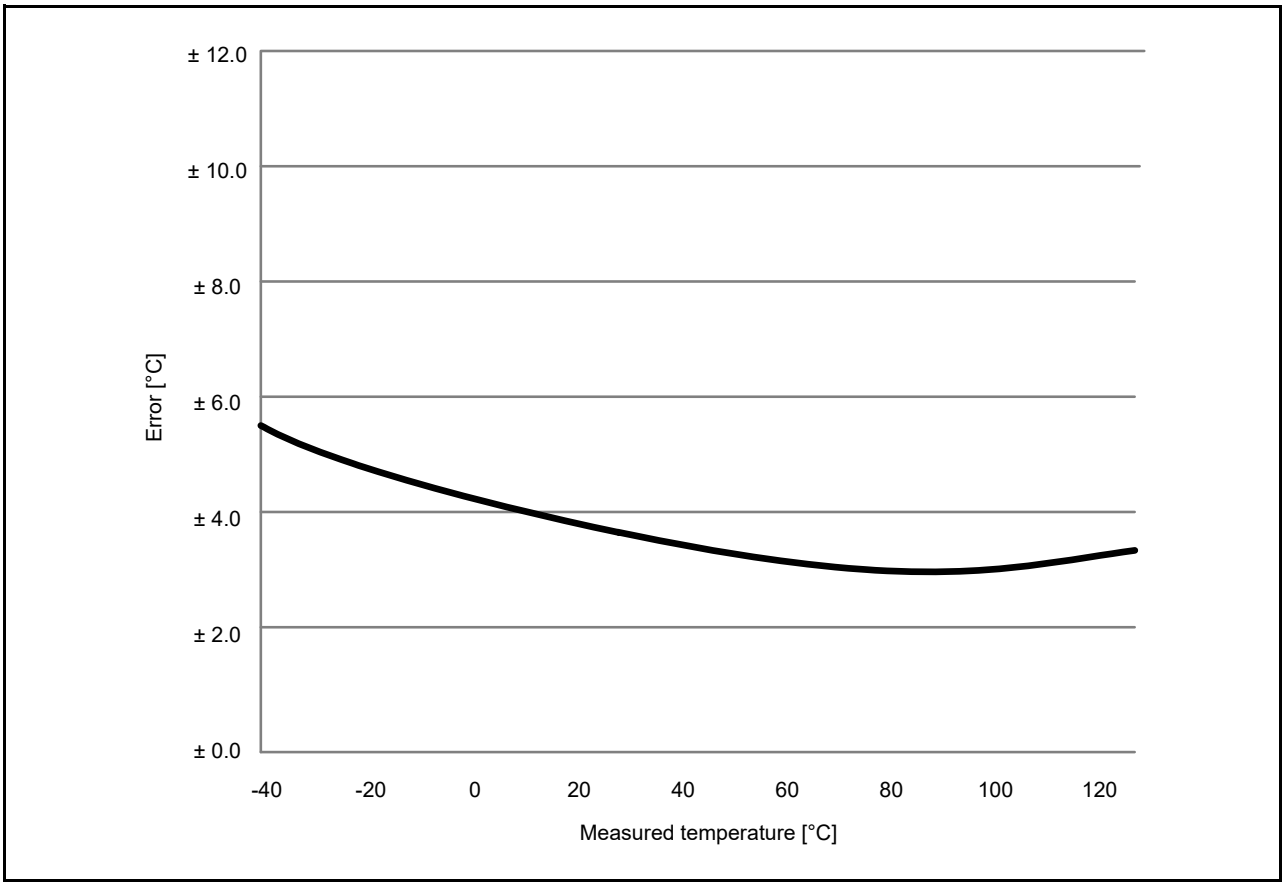


Figure 36.2 Error in the measured temperature (designed values)

36.3.2 Procedures for Using the Temperature Sensor

For details, see section 32, 16-Bit A/D Converter (ADC16).

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 a total of three differential operational amplifier units with two input pins and one output pin.

The operational amplifiers have the following functions:

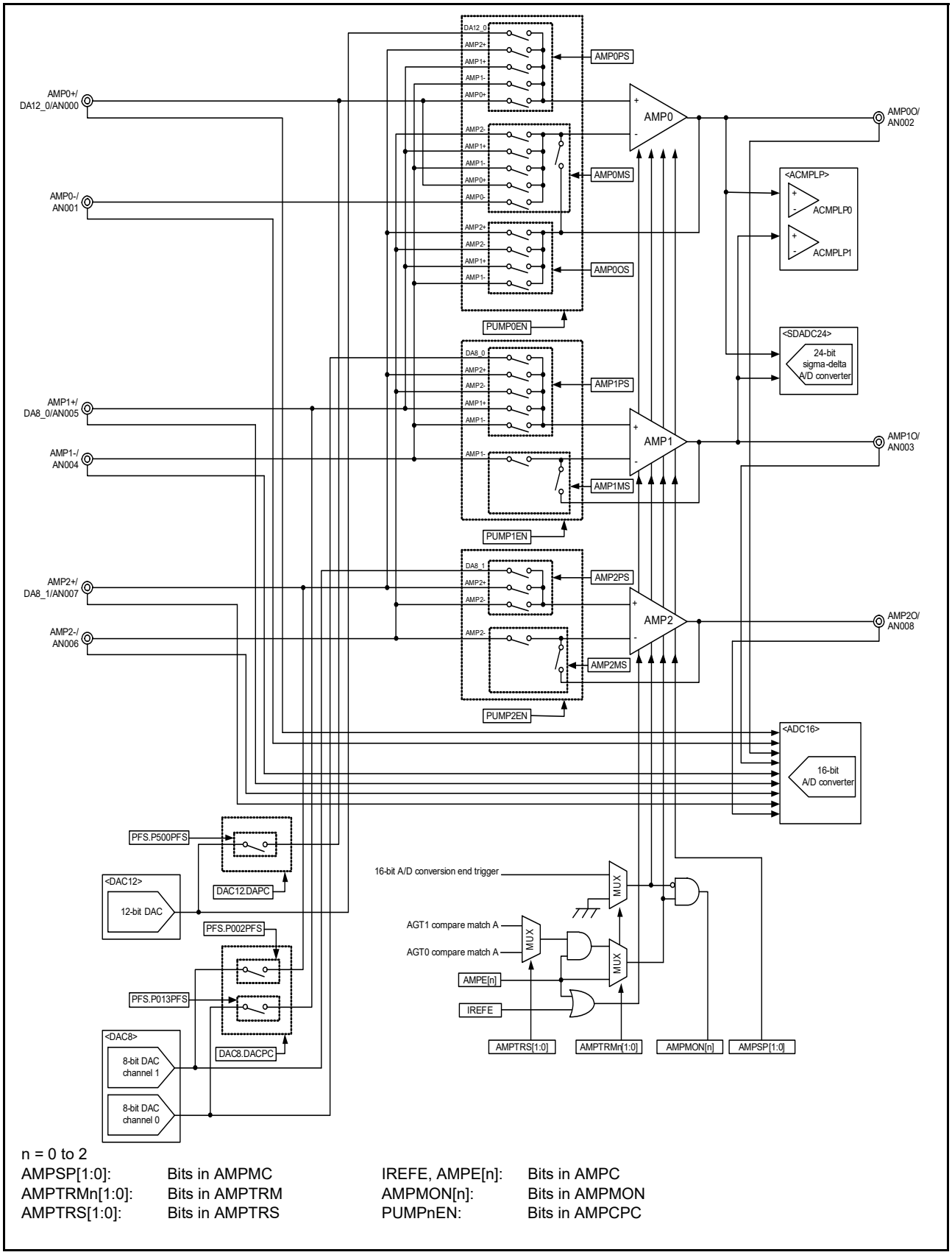
- OPAMP0 and OPAMP1 of the three units can be used to input signals to the Low-Power Analog Comparator (ACMPLP) and the 24-bit Sigma-Delta A/D Converter (SDADC24)
- High-speed mode (high-current consumption), middle-speed mode (medium-current consumption), and low-power mode (slow-speed response) are supported and any mode can be selected based on trade-offs between the response speed and current consumption
- Operation can be started by a trigger from the Low Power Asynchronous General purpose Timer (AGT)
- Operation can be stopped by a 16-bit A/D conversion end trigger
- All units have switches that can select input signals. Additionally, OPAMP0 has a switch that can select the output pin
- The output of the OPAMP can be output from the AMP0O to AMP2O pins without passing through the switch
- The I/O signals of all OPAMP units can be used for the input signals to the ADC16
- The signal output from the DAC8 and DAC12 can be used as the positive input signal for each OPAMP
- A voltage follower circuit can be configured by feeding back its own OPAMP output signal as the negative input signal of OPAMP.

The number of OPAMP input and output pins differs depending on the product. Table 37.1 lists the OPAMP I/O pins and Figure 37.1 shows a block diagram of the OPAMP.

Table 37.1 OPAMP I/O pins

I/O pin	I/O	Function
AVCC0	Input	Analog block power supply pin
AVSS0	Input	Analog block power supply ground pin
AMP0+, AMP0-	Input	Input pin of the OPAMP0 (+, -) *1
AMP0O	Output	Output pin of the OPAMP0
AMP1+, AMP1-	Input	Input pin of the OPAMP1 (+, -) *1
AMP1O	Output	Output pin of the OPAMP1
AMP2+, AMP2-	Input	Input pin of the OPAMP2 (+, -) *1
AMP2O	Output	Output pin of the OPAMP2

Note 1. Connect to the I/O pin by a switch.



n = 0 to 2

AMPSP[1:0]:	Bits in AMPMC	IREFE, AMPE[n]:	Bits in AMPC
AMPTRMn[1:0]:	Bits in AMPTRM	AMPMON[n]:	Bits in AMPMON
AMPTRS[1:0]:	Bits in AMPTRS	PUMPnEN:	Bits in AMPCPC

Figure 37.1 OPAMP block diagram

Figure 37.2 to Figure 37.4 show the block diagrams of OPAMP 0 to 2.

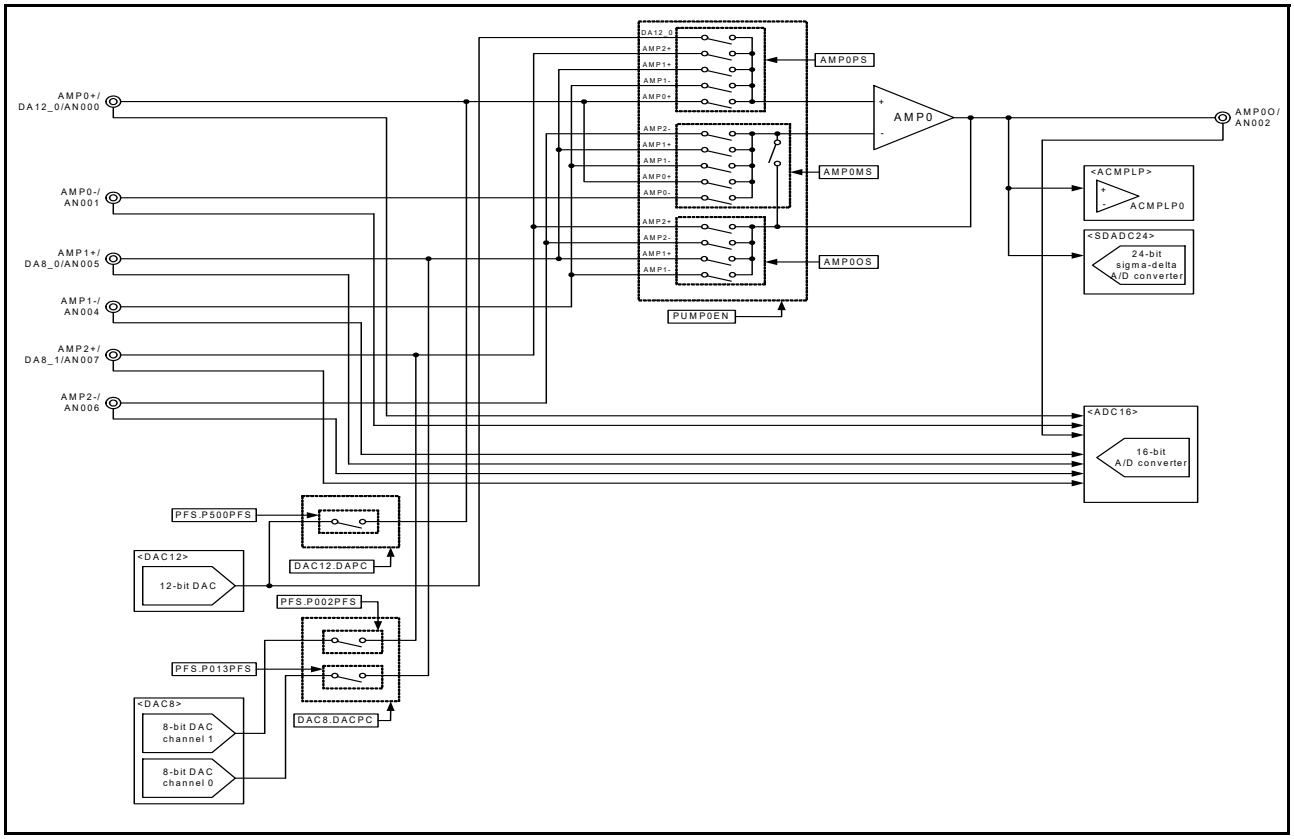


Figure 37.2 OPAMP0 block diagram

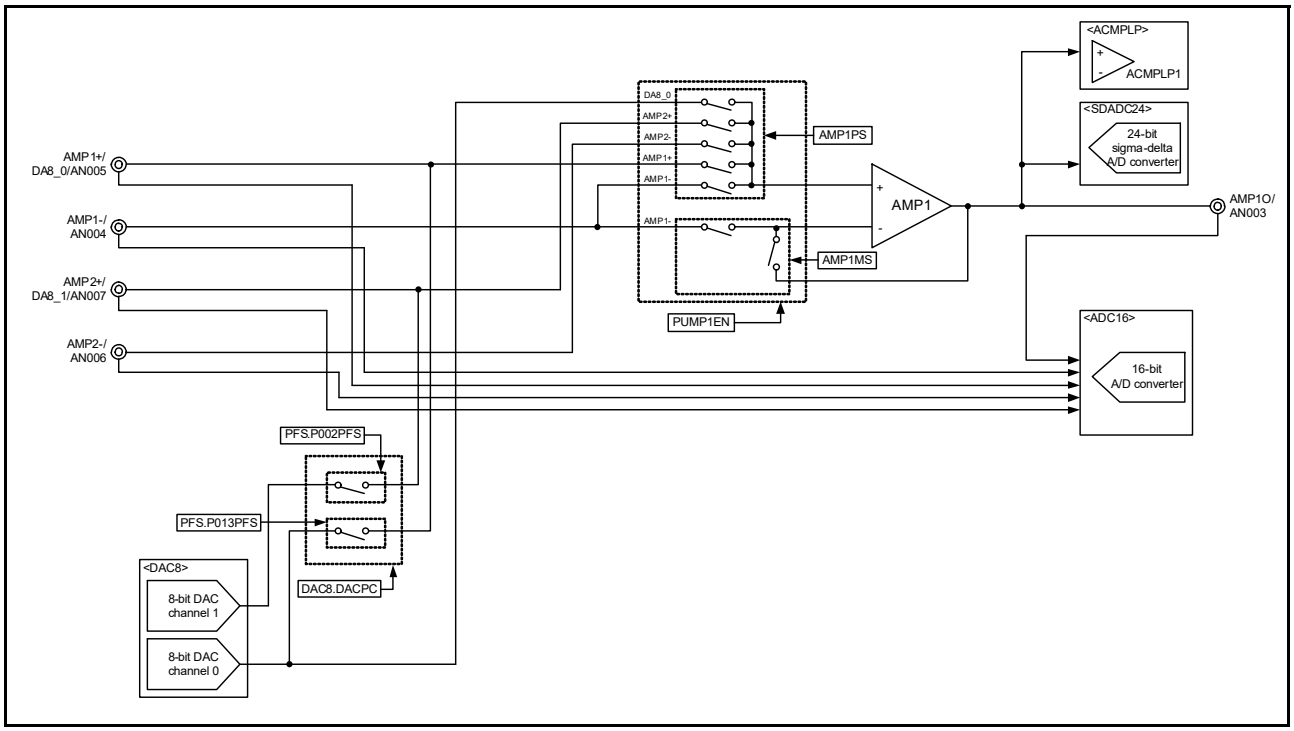


Figure 37.3 OPAMP1 block diagram

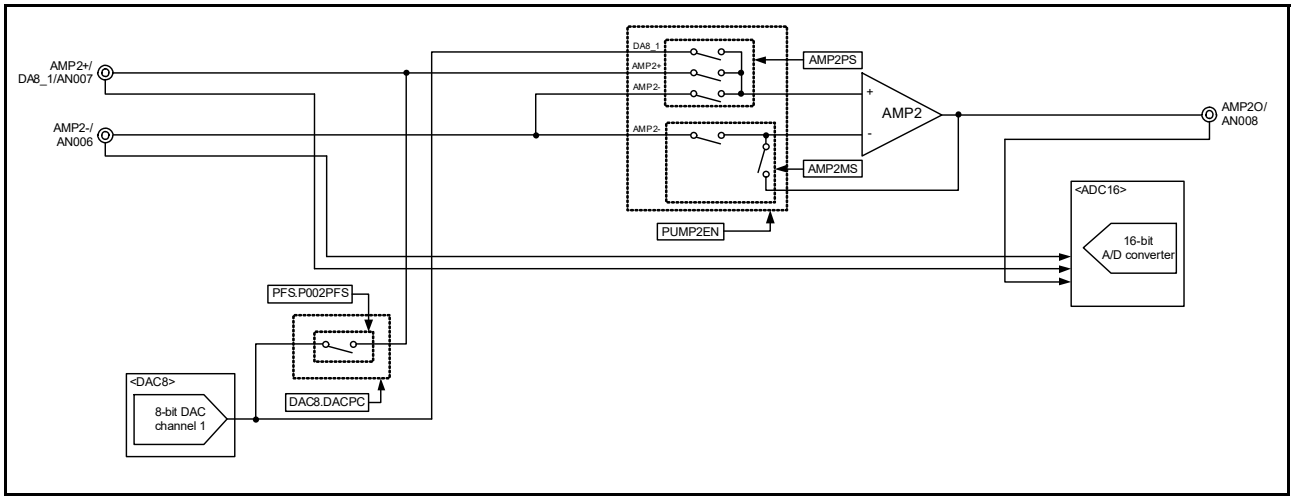


Figure 37.4 OPAMP2 block diagram

37.2 Register Descriptions

37.2.1 Operational Amplifier Mode Control Register (AMPMC)

Address(es): OPAMP.AMPMC 4008 6800h

	b7	b6	b5	b4	b3	b2	b1	b0
	AMPSP[1:0]		—	—	—	—	—	—
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
b7, b6	AMPSP[1:0]	OPAMP Operation Mode Selection	b7 b6 x 0: Low-power mode (low-speed) 0 1: Middle-speed mode 1 1: High-speed mode.	R/W

Note: Set the AMPSP[1:0] bits while the AMPC register is 00h (OPAMP and reference current generator are stopped).

Note: User offset trimming cannot be used in low-power mode. When AMPSP[1:0] = x0b, set the AMPUTOTE register to 00h.

37.2.2 Operational Amplifier Trigger Mode Control Register (AMPTRM)

Address(es): OPAMP.AMPTRM 4008 6801h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AMPTRM2[1:0]		AMPTRM1[1:0]		AMPTRM0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0 b3, b2 b5, b4	AMPTRM0[1:0] AMPTRM1[1:0] AMPTRM2[1:0]	OPAMP Function Activation/ Stop Trigger Control n*2	AMPTRMn[1] AMPTRMn[0] (n = 0 to 2) 0 0: Software trigger mode: •The OPAMPn can be activated or stopped by setting the AMPC register •The OPAMPn cannot be activated by an activation trigger •The OPAMPn cannot be controlled by a 16-bit A/D conversion end trigger. 0 1: Activation trigger mode: •The OPAMPn can be set to wait for an activation trigger or stopped by setting the AMPC register •The OPAMPn can be activated by an activation trigger *1 •The OPAMPn cannot be controlled by a 16-bit A/D conversion end trigger. 1 0: Setting prohibited 1 1: Activation and A/D trigger mode: •The OPAMPn can be set to wait for an activation trigger or stopped by setting the AMPC register •The OPAMPn can be activated by an activation trigger*1 •The OPAMPn can be stopped by a 16-bit A/D conversion end trigger. A 16-bit 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: A 16-bit 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 OPAMP, first specify settings related to the AGT, set the AMPTRS register, 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 AMPTRMn[1:0] bits, make sure that the AMPE[n] bit in the AMPC register is 0 (OPAMPn is stopped).

37.2.3 Operational Amplifier Activation Trigger Select Register (AMPTRS)

Address(es): OPAMP.AMPTRS 4008 6802h

	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: OPAMPn: OPAMP activation trigger n (n = 0 to 2) 0 1: OPAMPn: OPAMP activation trigger 0 (n = 0, 1) OPAMP2: OPAMP activation trigger 1 1 0: Setting prohibited 1 1: OPAMPn: OPAMP activation trigger 0 (n = 0 to 2).	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not change the value of the AMPTRS register after setting the AMPTRM register.

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

37.2.4 Operational Amplifier Control Register (AMPC)

Address(es): OPAMP.AMPC 4008 6803h

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

Bit	Symbol	Bit name	Description	R/W
b2 to b0	AMPE[2:0]	OPAMP Operation Control	0: OPAMPn is stopped 1: Operation of OPAMPn is enabled*1. Activation trigger mode or activation and A/D trigger mode: Wait until AGT is enabled (n = 0 to 2).	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	IREFE	Reference Current Circuit Operation Control	0: Reference current circuit is stopped 1: Operation of reference current circuit is enabled.	R/W

Note 1. Operation of the reference current circuit is also enabled regardless of the IREFE bit setting. Set the bits to 0 for a unit that is not to be used.
Before setting the bits to 1, set the switches in the AMPnMS, AMPnPS, and AMP0OS registers. When AVCC0 < 2.7 V, after setting the switches, wait for the charge pump stabilization time, then set these bits to 1. For details on the stabilization wait time, see section 47, Electrical Characteristics.

37.2.5 Operational Amplifier Monitor Register (AMPMON)

Address(es): OPAMP.AMPMON 4008 6804h

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

Bit	Symbol	Bit name	Description	R/W
b2 to b0	AMPMON[2:0]	OPAMPn Status (n = 0 to 2)	0: OPAMPn is stopped 1: OPAMPn is operating.	R
b7 to b3	—	Reserved	These bits are read as 0	R

Note: This register is used to asynchronously reflect whether each OPAMPn is operating or stopped. To determine the OPAMP state, read this register continuously to determine when the bit state changes. After that, read this register again to confirm whether the state of OPAMP has changed. When an activation trigger or 16-bit A/D conversion end trigger synchronized with the clock or a software trigger in the other interrupt routine is used to control the OPAMP, the timing to operate or stop the OPAMP 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 OPAMP state occurs.

37.2.6 Operational Amplifier 0 Output Select Register (AMP0OS)

Address(es): OPAMP.AMP0OS 4008 6806h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	AMPOS3	AMPOS2	AMPOS1	AMPOS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	AMPOS0	AMP1- pin select	0: AMP1- pin is not connected to the OPAMP0 output 1: AMP1- pin is connected to the OPAMP0 output.	R/W
b1	AMPOS1	AMP1+ pin select	0: AMP1+ pin is not connected to the OPAMP0 output 1: AMP1+ pin is connected to the OPAMP0 output.*1	R/W
b2	AMPOS2	AMP2- pin select	0: AMP2- pin is not connected to the OPAMP0 output 1: AMP2- pin is connected to the OPAMP0 output.	R/W
b3	AMPOS3	AMP2+ pin select	0: AMP2+ pin is not connected to the OPAMP0 output 1: AMP2+ pin is connected to the OPAMP0 output.*2	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When AVCC0 < 2.7 V, before setting the value of these bits, set the PUMP0EN bit in AMPCPC register to 1 (charge pump for AMP0 is enabled) and wait for the stabilization time. For details on the stabilization time, see section 47, Electrical Characteristics.

Note: When AMPCPC.PUMP0EN = 1, do not set a total of five or more bits to 1 between the AMP0OS, AMP0PS, and AMP0MS registers.

Note 1. When connecting to AMP1+ pin, do not output DAC8 channel 0 output level to a pin.

Note 2. When connecting to AMP2+ pin, do not output DAC8 channel 1 output level to a pin.

37.2.7 Operational Amplifier 0 Minus Input Select Register (AMP0MS)

Address(es): OPAMP.AMP0MS 4008 6807h

	b7	b6	b5	b4	b3	b2	b1	b0
	AMPMS7	—	—	AMPMS4	AMPMS3	AMPMS2	AMPMS1	AMPMS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	AMPMS0	AMP0- pin select	0: AMP0- pin is not connected to the AMP0 minus input 1: AMP0- pin is connected to the AMP0 minus input.	R/W
b1	AMPMS1	AMP0+ pin select	0: AMP0+ pin is not connected to the AMP0 minus input 1: AMP0+ pin is connected to the AMP0 minus input.	R/W
b2	AMPMS2	AMP1- pin select	0: AMP1- pin is not connected to the AMP0 minus input 1: AMP1- pin is connected to the AMP0 minus input.	R/W
b3	AMPMS3	AMP1+ pin select	0: AMP1+ pin is not connected to the AMP0 minus input 1: AMP1+ pin is connected to the AMP0 minus input.	R/W
b4	AMPMS4	AMP2- pin select	0: AMP2- pin is not connected to the AMP0 minus input 1: AMP2- pin is connected to the AMP0 minus input.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	AMPMS7	OPAMP0 output select	0: OPAMP0 output is not connected to the AMP0 minus input 1: OPAMP0 output is connected to the AMP0 minus input.*1	R/W

Note: When AVCC0 < 2.7 V, before setting the value of these bits, set the PUMP0EN bit in AMPCPC register to 1 (charge pump for AMP0 is enabled) and wait for the stabilization time. For details on the stabilization time, see section 47, Electrical Characteristics.

Note: When AMPCPC.PUMP0EN = 1, do not set a total of five or more bits to 1 between the AMP0OS, AMP0PS, and AMP0MS registers.

Note 1. The AMPMS7 and AMPMS0 to AMPMS4 bits must not be set to 1 at the same time. Write 80h to this register when configuring a voltage follower. Only set the AMPMS7 bit to 1.
When an operational amplifier input pin is connected to the AMP0 minus input, the OPAMP0 output is not connected to the AMP0 minus input if AMPMS7 = 1.

37.2.8 Operational Amplifier 0 Plus Input Select Register (AMP0PS)

Address(es): OPAMP.AMP0PS 4008 6808h

	b7	b6	b5	b4	b3	b2	b1	b0
	AMPPS7	—	—	—	AMPPS3	AMPPS2	AMPPS1	AMPPS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	AMPPS0	AMP0+ pin select	0: AMP0+ pin is not connected to the AMP0 plus input 1: AMP0+ pin is connected to the AMP0 plus input.	R/W
b1	AMPPS1	AMP1- pin select	0: AMP1- pin is not connected to the AMP0 plus input 1: AMP1- pin is connected to the AMP0 plus input.	R/W
b2	AMPPS2	AMP1+pin select	0: AMP1+ pin is not connected to the AMP0 plus input 1: AMP1+ pin is connected to the AMP0 plus input.	R/W
b3	AMPPS3	AMP2+ pin select	0: AMP2+ pin is not connected to the AMP0 plus input 1: AMP2+ pin is connected to the AMP0 plus input.	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	AMPPS7	DAC12 output select	0: DAC12 output is not connected to the AMP0 plus input 1: DAC12 output is connected to the AMP0 plus input.*1	R/W

Note: When AVCC0 < 2.7 V, before setting the value of these bits, set the PUMP0EN bit in the AMPCPC register to 1 (charge pump for AMP0 is enabled) and wait for the stabilization time. For details on the stabilization time, see section 47, Electrical Characteristics.

Note: When AMPCPC.PUMP0EN = 1, do not set a total of five or more bits to 1 between the AMP0OS, AMP0PS, and AMP0MS registers.

Note 1. To connect the DAC12 to the input pin of the OPAMP, set the DAOE0 bit of the DACR register to 1.
When an operational amplifier input pin is connected to the AMP0 plus input, the DAC12 output is not connected to the AMP0 plus input if AMPPS7 = 1.

37.2.9 Operational Amplifier 1 Minus Input Select Register (AMP1MS)

Address(es): OPAMP.AMP1MS 4008 680Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	AMPMS7	—	—	—	—	—	—	AMPMS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	AMPMS0	AMP1- pin select	0: AMP1- pin is not connected to the AMP1 minus input 1: AMP1- pin is connected to the AMP1 minus input.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	AMPMS7	OPAMP1 output select	0: OPAMP1 output is not connected to the AMP1 minus input 1: OPAMP1 output is connected to the AMP1 minus input.*1	R/W

Note: When AVCC0 < 2.7 V, before setting the value of these bits, set the PUMP1EN bit in the AMPCPC register to 1 (charge pump for AMP1 is enabled) and wait for the stabilization time. For details on the stabilization time, see section 47, Electrical Characteristics.

Note: When AMPCPC.PUMP1EN = 1, do not set a total of five or more bits to 1 between the AMP1PS and AMP1MS registers.

Note 1. The AMPMS7 and AMPMS0 bits must not be set to 1 at the same time. Write 80h to this register when configuring a voltage follower. Only set the AMPMS7 bit to 1.
When an operational amplifier input pin is connected to the AMP1 minus input, the OPAMP1 output is not

connected to the AMP1 minus input if AMPMS7 = 1.

37.2.10 Operational Amplifier 1 Plus Input Select Register (AMP1PS)

Address(es): OPAMP.AMP1PS 4008 680Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	AMPPS7	—	—	—	AMPPS3	AMPPS2	AMPPS1	AMPPS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	AMPPS0	AMP1- pin select	0: AMP1- pin is not connected to the AMP1 plus input 1: AMP1- pin is connected to the AMP1 plus input.	R/W
b1	AMPPS1	AMP1+ pin select	0: AMP1+ pin is not connected to the AMP1 plus input 1: AMP1+ pin is connected to the AMP1 plus input.	R/W
b2	AMPPS2	AMP2- pin select	0: AMP2- pin is not connected to the AMP1 plus input 1: AMP2- pin is connected to the AMP1 plus input.	R/W
b3	AMPPS3	AMP2+ pin select	0: AMP2+ pin is not connected to the AMP1 plus input 1: AMP2+ pin is connected to the AMP1.	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	AMPPS7	DAC8 channel 0 output select	0: DAC8 channel 0 output is not connected to the AMP1 plus input 1: DAC8 channel 0 output is connected to the AMP1 plus input.*1	R/W

Note: When AVCC0 < 2.7 V, before setting the value of these bits, set the PUMP1EN bit in the AMPCPC register to 1 (charge pump for AMP1 is enabled) and wait for the stabilization time. For details on the stabilization time, see section 47, Electrical Characteristics.

Note: When AMPCPC.PUMP1EN = 1, do not set a total of five or more bits to 1 between the AMP1PS and AMP1MS registers.

Note 1. To connect the DAC8 channel 0 to the input pin of AMP1, set the DACE0 bit of the DAM register to 1. When an operational amplifier input pin is connected to the AMP1 plus input, the DAC8 channel 0 output is not connected to the AMP1 plus input if AMPPS7 = 1.

37.2.11 Operational Amplifier 2 Minus Input Select Register (AMP2MS)

Address(es): OPAMP.AMP2MS 4008 680Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	AMPMS7	—	—	—	—	—	—	AMPMS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	AMPMS0	AMP2- pin select	0: AMP2- pin is not connected to the AMP2 minus input 1: AMP2- pin is connected to the AMP2 minus input.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	AMPMS7	OPAMP2 output select	0: OPAMP2 output is not connected to the AMP2 minus input 1: OPAMP2 output is connected to the AMP2 minus input.*1	R/W

Note: When AVCC0 < 2.7 V, before setting the value of these bits, set the PUMP2EN bit in the AMPCPC register to 1 (charge pump for AMP2 is enabled), wait for the stabilization time. For details on the stabilization time, see section 47, Electrical Characteristics.

Note: When AMPCPC.PUMP2EN = 1, do not set a total of two or more bits to 1 between the AMP2PS and AMP2MS registers.

Note 1. The AMPMS7 and AMPMS0 bits must not be set to 1 at the same time. Write 80h to this register when

configuring a voltage follower. Only set the AMPMS7 bit to 1.
When an operational amplifier input pin is connected to the AMP2 minus input, the OPAMP2 output is not connected to the AMP2 minus input if AMPMS7 = 1.

37.2.12 Operational Amplifier 2 Plus Input Select Register (AMP2PS)

Address(es): OPAMP.AMP2PS 4008 680Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	AMPPS7	—	—	—	—	—	AMPPS1	AMPPS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	AMPPS0	AMP2- pin select	0: AMP2- pin is not connected to the AMP2 plus input 1: AMP2- pin is connected to the AMP2 plus input.	R/W
b1	AMPPS1	AMP2+ pin select	0: AMP2+ pin is not connected to the AMP2 plus input 1: AMP2+ pin is connected to the AMP2 plus input.	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	AMPPS7	DAC8 channel 1output select	0: DAC8 channel 1 output is not connected to the AMP2 plus input 1: DAC8 channel 1 output is connected to the AMP2 plus input.*1	R/W

Note: When AVCC0 < 2.7 V, before setting the value of these bits, set the PUMP2EN bit in the AMPCPC register to 1 (charge pump for AMP2 is enabled) and wait for the stabilization time. For details on the stabilization time, see section 47, Electrical Characteristics.

Note: When AMPCPC.PUMP2EN = 1, do not set a total of two or more bits to 1 between AMP2PS and AMP2MS registers.

Note 1. To connect the DAC8 channel 1 to the input pin of the AMP2, set the DACE1 bit of the DAM register to 1.
When an operational amplifier input pin is connected to the AMP2 plus input, the DAC8 channel 1 output is not connected to the AMP2 plus input if AMPPS7 = 1.

37.2.13 Operational Amplifier Switch Charge Pump Control Register (AMPCPC)

Address(es): OPAMP.AMPCPC 4008 6812h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PUMP2EN	PUMP1EN	PUMP0EN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PUMP0EN	Charge Pump for AMP0 Enable	0: Charge Pump for the AMP0 disabled 1: Charge Pump for the AMP0 enabled.	R/W
b1	PUMP1EN	Charge Pump for AMP1 Enable	0: Charge Pump for the AMP1 disabled 1: Charge Pump for the AMP1 enabled.	R/W
b2	PUMP2EN	Charge Pump for AMP2 Enable	0: Charge Pump for the AMP2 disabled 1: Charge Pump for the AMP2 enabled.*1	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When AVCC0 < 2.7 V, perform the following instructions:
- Set the MOCOCR.MCSTP bit to 0, then set the PUMPnEN bit to 1
- Set PUMPnEN bit while the AMPnMS and AMPnPS registers are 00h (no connection).
  Additionally, set the PUMP0EN bit when the AMP0OS register is 00h (no connection).

Note 1. It is prohibited to output the D/A converter output level to DA8_0, DA8_1 and DA12_0 pins, during 16-bit A/D conversion of AMP2O.

37.2.14 Operational Amplifier User Offset Trimming Enable Register (AMPUOTE)

Address(es): OPAMP.AMPUOTE 4008 6817h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	AMP2TE	AMP1TE	AMP0TE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	AMP0TE	AMP0OT write enable	0: Write to the AMP0OTP and AMP0OTN registers is not possible 1: Write to the AMP0OTP and AMP0OTN registers is possible.	R/W
b1	AMP1TE	AMP1OT write enable	0: Write to the AMP1OTP and AMP1OTN registers is not possible 1: Write to the AMP1OTP and AMP1OTN registers is possible.	R/W
b2	AMP2TE	AMP2OT write enable	0: Write to the AMP2OTP and AMP2OTN registers is not possible 1: Write to the AMP2OTP and AMP2OTN registers is possible.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: For procedure on user offset trimming, see section 37.9, User Offset Trimming.

Note: User offset trimming cannot be used in low-power mode. When AMPMC.AMPSP[1:0] = x0b, set the AMPUTOTE register to 00h.

37.2.15 Operational Amplifier n Offset Trimming Pch Register (AMPnOTP) (n = 0 to 2)

Address(es): OPAMP.AMP0OTP 4008 6818h, OPAMP.AMP1OTP 4008 681Ah, OPAMP.AMP2OTP 4008 681Ch

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

Bit	Symbol	Bit name	Description	R/W
b4 to b0	TRMP[4:0]	AMPn input offset trimming Pch side	AMPn input offset trimming Pch side value: b4 b3 b2 b1 b0 0 0 0 0 0: -16 0 0 0 0 1: -15 0 0 0 1 0: -14 . . 0 1 1 1 1: -1 1 0 0 0 0: Center Code 1 0 0 0 1: +1 . . 1 1 1 0 1: +13 1 1 1 1 0: +14 1 1 1 1 1: +15	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: The rewrite timing of TRMP[4:0] bits where n = 0 to 2, is as follows:

(A) An initial setting value in factory shipment is written:

- When the first MSTP (module-stop state) is released after a reset release
- When writing to the AMPSP[1:0] bits of the AMPMC register where AMPnTE = 0
- When the AMPnTPE bit is changed from 1 to 0.

(B) The user setting value is written:

- When writing to TRMP[4:0] where AMPnTE = 1.

Note: There is an initial value set at factory shipment for each operation mode of each OPAMP. Therefore, when changing the mode (changing the values of AMPSP[1:0] bits in the AMPMC register) after user offset trimming, it is necessary to execute user offset trimming again in that mode.

Note: User offset trimming cannot be used in low-power mode (AMPMC.AMPSP[1:0] = x0b).

37.2.16 Operational Amplifier n Offset Trimming Nch Register (AMPnOTN) (n = 0 to 2)

Address(es): OPAMP.AMP0OTN 4008 6819h, OPAMP.AMP1OTN 4008 681Bh, OPAMP.AMP2OTN 4008 681Dh

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

Bit	Symbol	Bit name	Description	R/W
b4 to b0	TRMN[4:0]	AMPn input offset trimming Nch side	AMPn input offset trimming Nch side value: b4 b3 b2 b1 b0 0 0 0 0 0: -16 0 0 0 0 1: -15 0 0 0 1 0: -14 . . 0 1 1 1 1: -1 1 0 0 0 0: Center Code . . 1 0 0 0 1: +1 . . 1 1 1 0 1: +13 1 1 1 1 0: +14 1 1 1 1 1: +15	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: The rewrite timing of TRMN[4:0] bits where n = 0 to 2, is as follows:

(A) An initial setting value in factory shipment is written:

- When the first MSTP (module-stop state) is released after a reset release
- When writing to the AMPSP[1:0] bits of the AMPMC register where AMPnTE = 0
- When the AMPnTE bit is changed from 1 to 0.

(B) The user setting value is written:

- When writing to TRMN[4:0] where AMPnTE = 1.

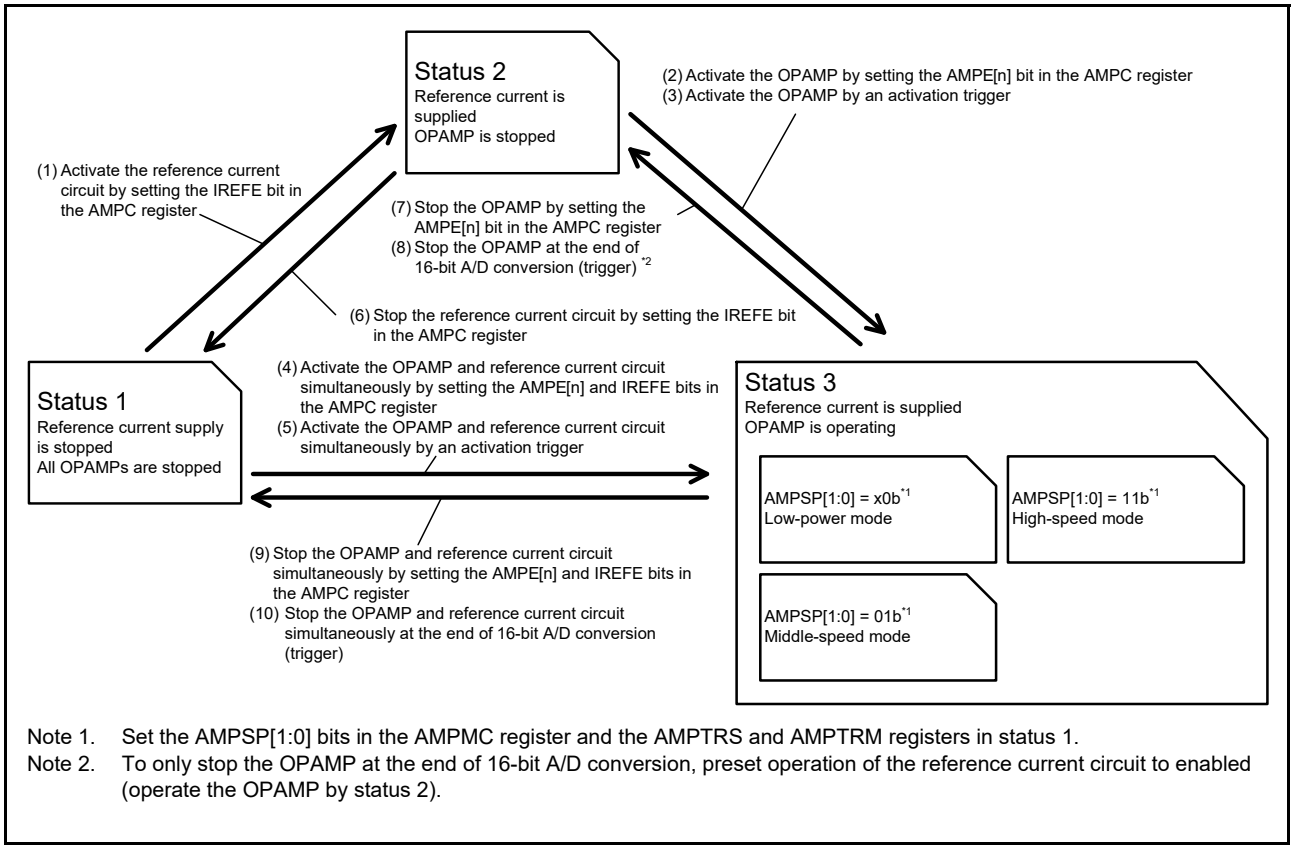
Note: There is an initial value set at factory shipment for each operation mode of each OPAMP. Therefore, when changing the mode (changing the values of AMPSP[1:0] bits in the AMPMC register) after user offset trimming, it is necessary to execute user offset trimming again in that mode.

Note: User offset trimming cannot be used in low-power mode (AMPMC.AMPSP[1:0] = x0b).

37.3 Operation

37.3.1 State Transitions

Figure 37.5 shows state transitions when the OPAMP and reference current circuit are activated or stopped using the OPAMP control circuit.



Note 1. Set the AMPSP[1:0] bits in the AMPMC register and the AMPTRS and AMPTRM registers in status 1.
Note 2. To only stop the OPAMP at the end of 16-bit A/D conversion, preset operation of the reference current circuit to enabled (operate the OPAMP by status 2).

Figure 37.5 OPAMP state transitions

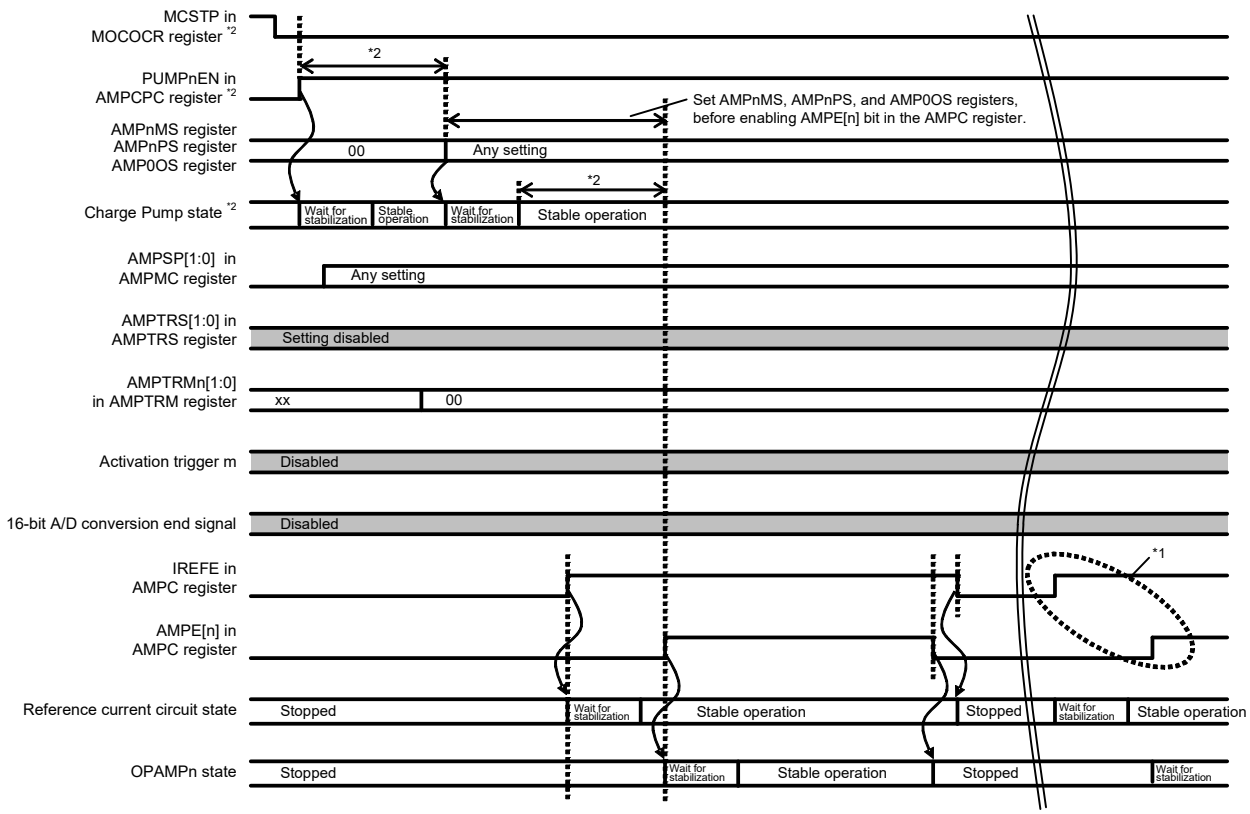
A stabilization wait time is required after supply of the reference current and operation of the OPAMP are set before each operation actually starts. For details on the stabilization wait time, see section 47, Electrical Characteristics.

The OPAMP cannot be activated or stopped continuously in steps (2) → (8), (2) → (10), (3) → (10), and (4) → (10).

An activation trigger and end of 16-bit A/D conversion are used to activate or stop only the OPAMP that is preset for use by setting the AMPTRM register.

37.3.2 OPAMP Control Operation

Figure 37.6 to Figure 37.9 show OPAMP control operation.

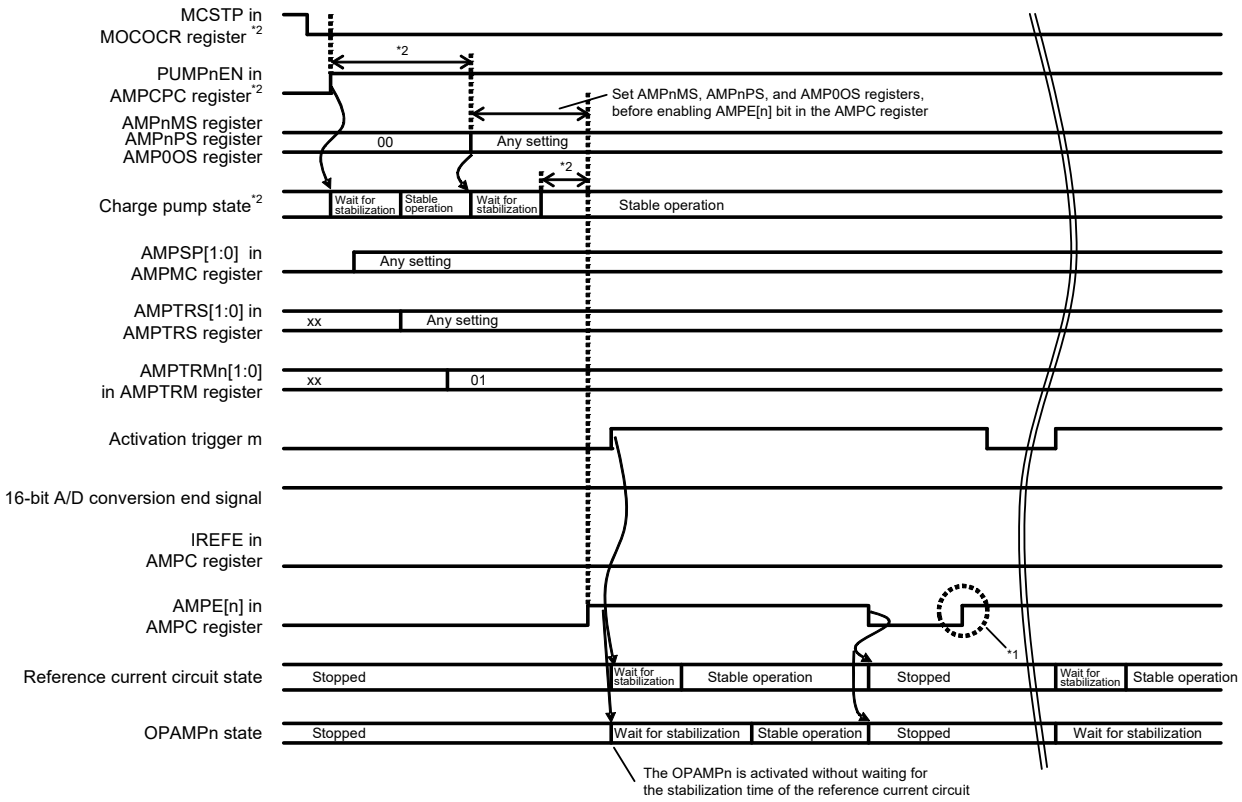


Note: n: Unit number (n = 0 to 2)
m: An activation trigger used to control OPAMPn selected by the AMPTRS register.

Note 1. When operating or stopping the OPAMP continuously, set the IREFE and AMPE[n] bits again as in the first setting after the OPAMP is stopped.

Note 2. When AVCC0 < 2.7 V, enable the charge pump. After enabling the charge pump, wait for the stabilization time and then set switches in the AMPnMS, AMPnPS, and AMP0OS registers. After setting the switches, set AMPE[n] bit to 1 after waiting for the charge pump stabilization time. For details on the charge pump stabilization time, see section 47, Electrical Characteristics.

Figure 37.6 OPAMP control operation when software trigger mode is used for control, and when the reference current circuit and OPAMP are activated or stopped by software trigger mode

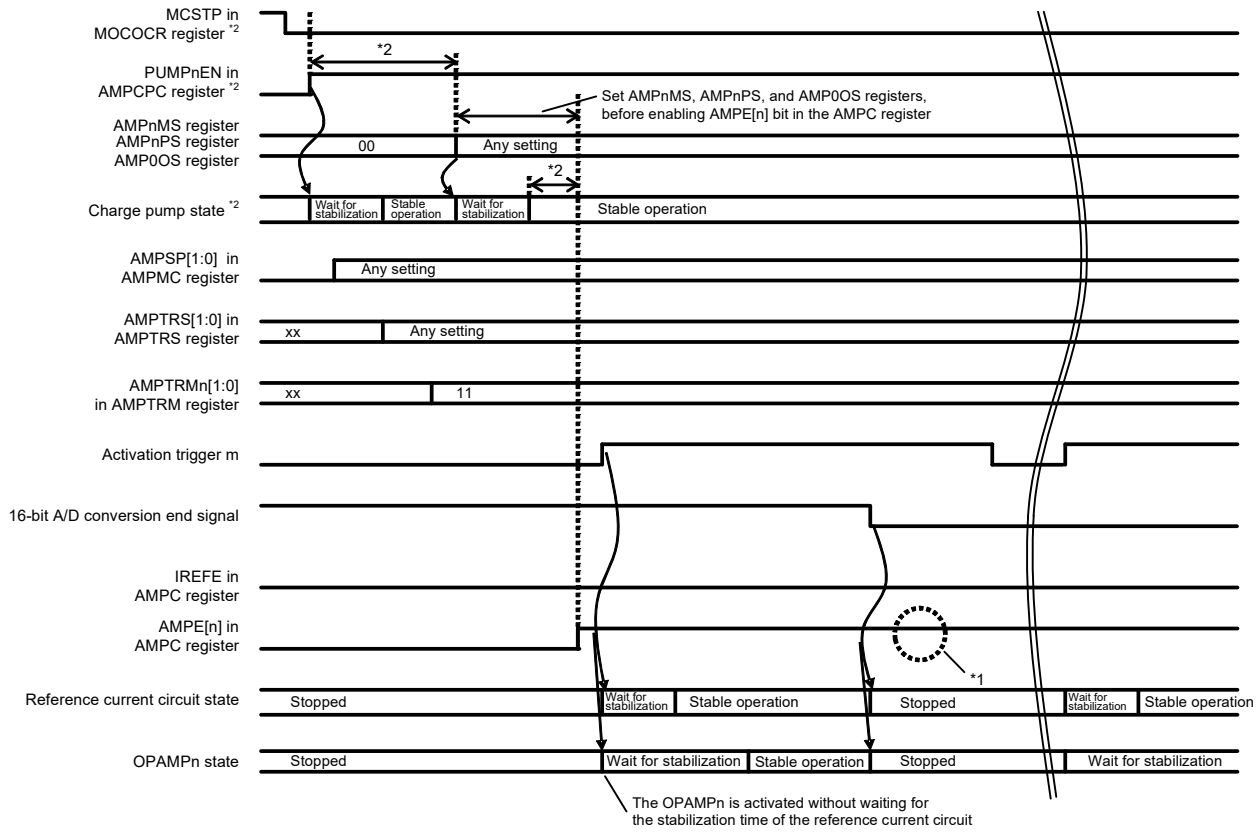


Note: n: Unit number (n = 0 to 2)
m: An activation trigger used to control OPAMPn selected by the AMPTRS register.
Set the AGT function.

Note 1. When operating or stopping the OPAMP continuously, use the AMPE[n] bit again as in the first setting, and set the OPAMP to wait for an activation trigger after it is stopped.

Note 2. When AVCC0 < 2.7 V, enable the charge pump. After enabling the charge pump, wait for the stabilization time and then set switches in the AMPnMS, AMPnPS, and AMP0OS registers. After setting the switches, set AMPE[n] bit to 1 after waiting for the charge pump stabilization time. For details on the charge pump stabilization time, see section 47, Electrical Characteristics.

Figure 37.7 OPAMP control operation when activation trigger mode is used for activation, and when the reference current circuit and OPAMP are activated by an activation trigger and stopped by setting the AMPC register

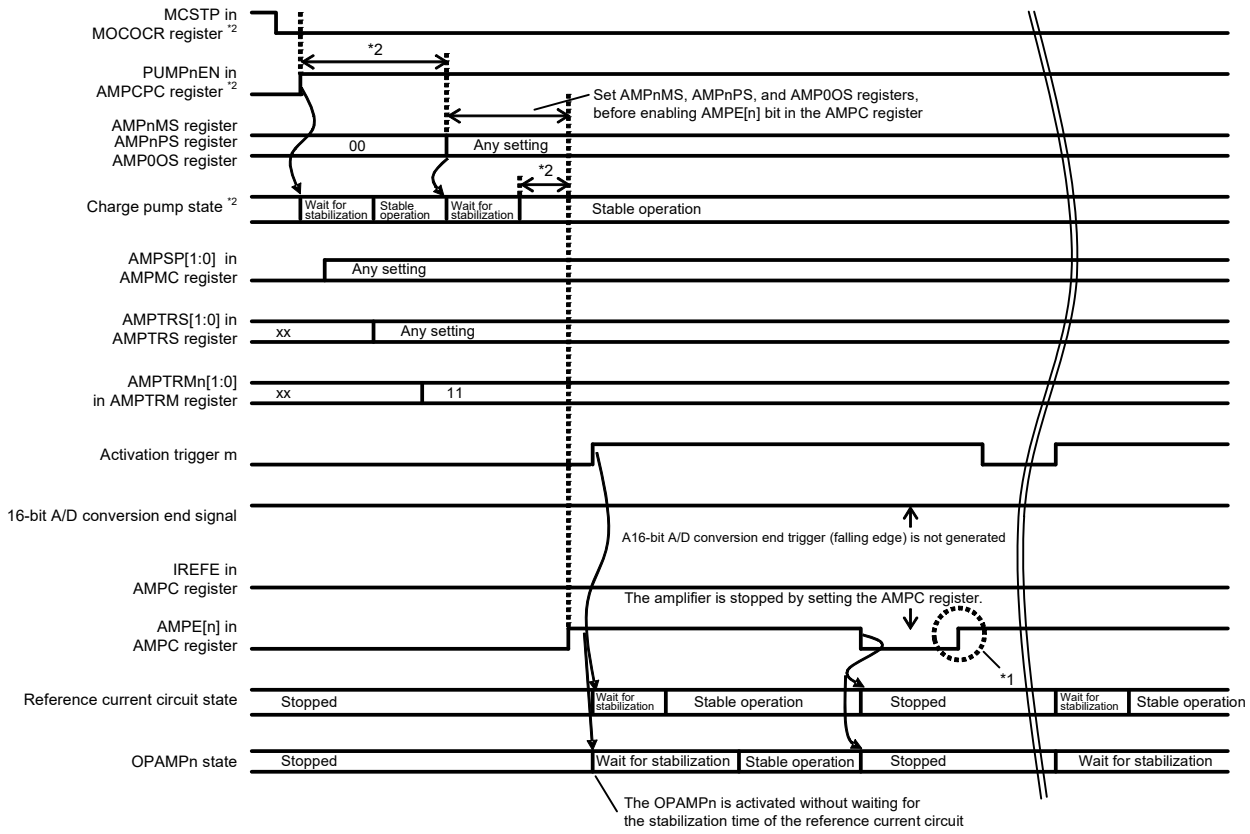


Note: n: Unit number (n = 0 to 2)
m: An activation trigger used to control OPAMPn selected by the AMPTRS register.
Set the AGT function.

Note 1. When operating/stopping the OPAMP continuously, it is not necessary to set the registers again because the OPAMP waits for an activation trigger after it is stopped.

Note 2. When AVCC0 < 2.7 V, the charge pump must be enabled. After enabling the charge pump, wait for the stabilization time and then set switches (in the AMPnMS, AMPnPS, and AMP0OS registers). After setting the switches, set AMPE[n] bit to 1 after waiting for the charge pump stabilization time. For details on the charge pump stabilization time, see section 47, Electrical Characteristics.

Figure 37.8 OPAMP control operation with activation and A/D trigger mode (1), and with the reference current circuit and OPAMP activated by an activation trigger and stopped by a 16-bit A/D conversion end (trigger)



Note: n: Unit number (n = 0 to 2)
m: An activation trigger used to control OPAMPn selected by the AMPTRS register.
Set the AGT function. See section 37.4, Software Trigger Mode for the procedure to activate the OPAMP with an activation trigger.

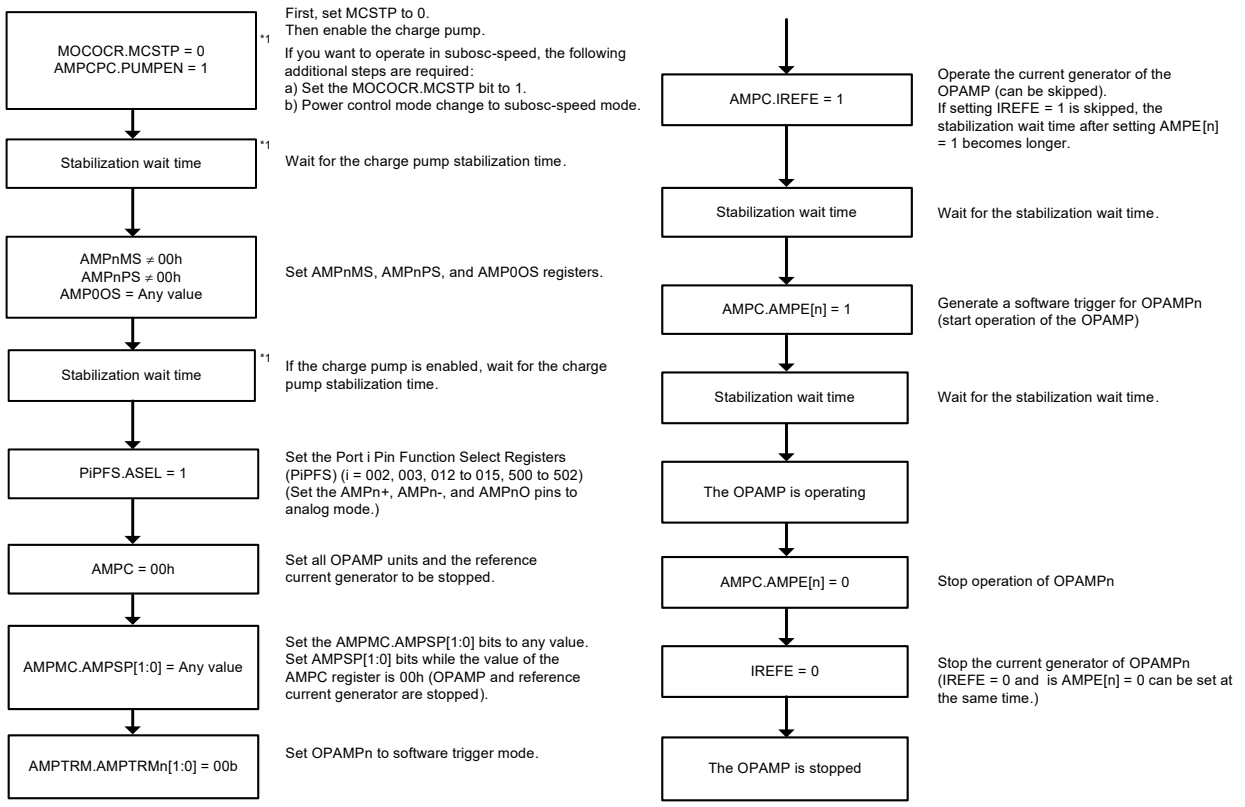
Note 1. When operating or stopping the OPAMP continuously, use the AMPE[n] bit again as in the first setting, and set the OPAMP to wait for an activation trigger after it is stopped.

Note 2. When AVCC0 < 2.7 V, enable the charge pump. After enabling the charge pump, wait for the stabilization time and then set switches in the AMPnMS, AMPnPS, and AMP0OS registers. After setting the switches, set AMPE[n] bit to 1 after waiting for the charge pump stabilization time. For details on the charge pump stabilization time, see section 47, Electrical Characteristics.

Figure 37.9 OPAMP control operation with activation and A/D trigger mode (2), and with the reference current circuit and OPAMP stopped by setting the AMPC register to be activated by an activation trigger and stopped by a 16-bit A/D conversion end (trigger)

37.4 Software Trigger Mode

This section describes the procedure to activate and stop the OPAMP using a software trigger. Figure 37.10 shows an example of each register setting.



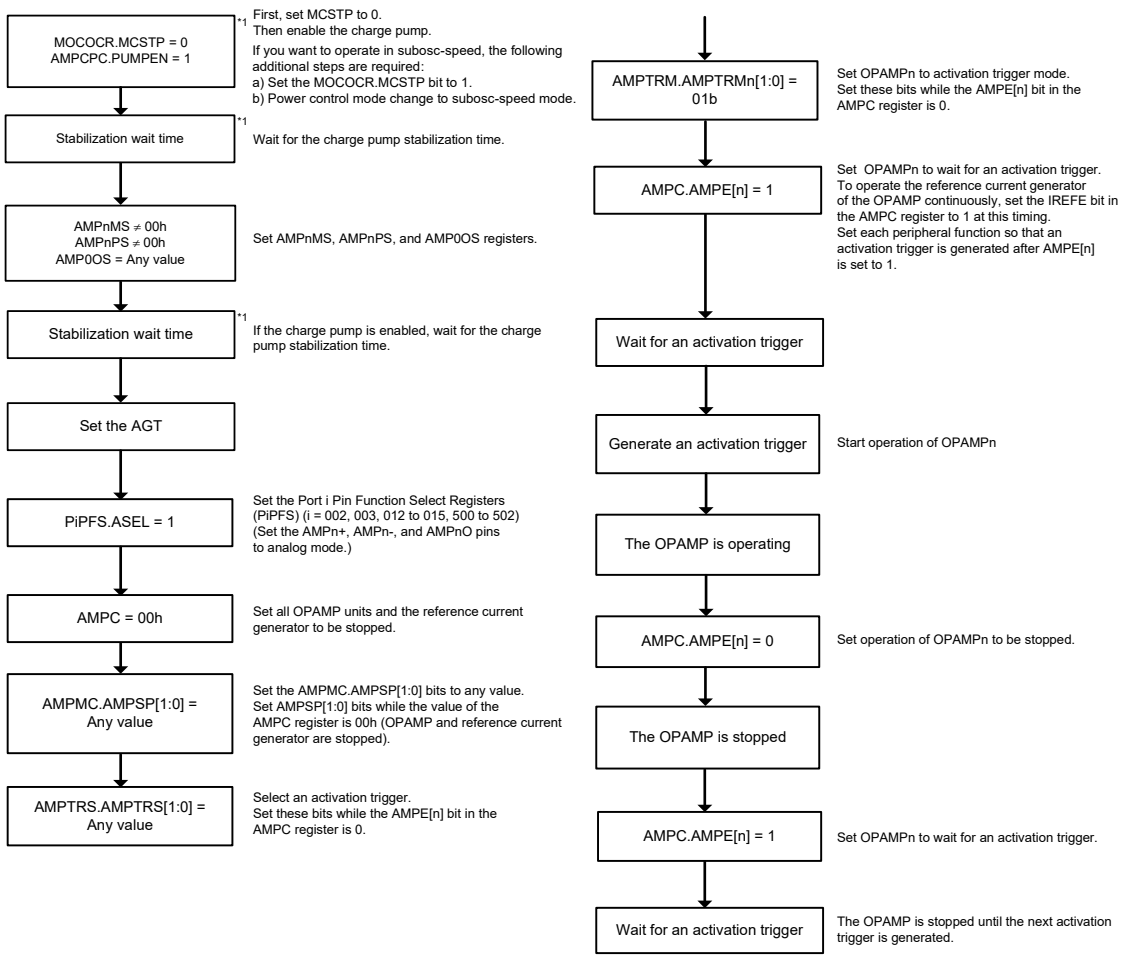
Note: For details on the stabilization wait time, see section 47, Electrical Characteristics.

Note 1. When AVCC0 < 2.7 V, these steps are required. For details on the charge pump stabilization wait time, see section 47, Electrical Characteristics.

Figure 37.10 Procedure to start and stop OPAMP in software trigger mode

37.5 Activation Trigger Mode

This section describes the procedure to activate the OPAMP using an activation trigger and to stop the OPAMP with software. Figure 37.11 shows an example of each register setting.

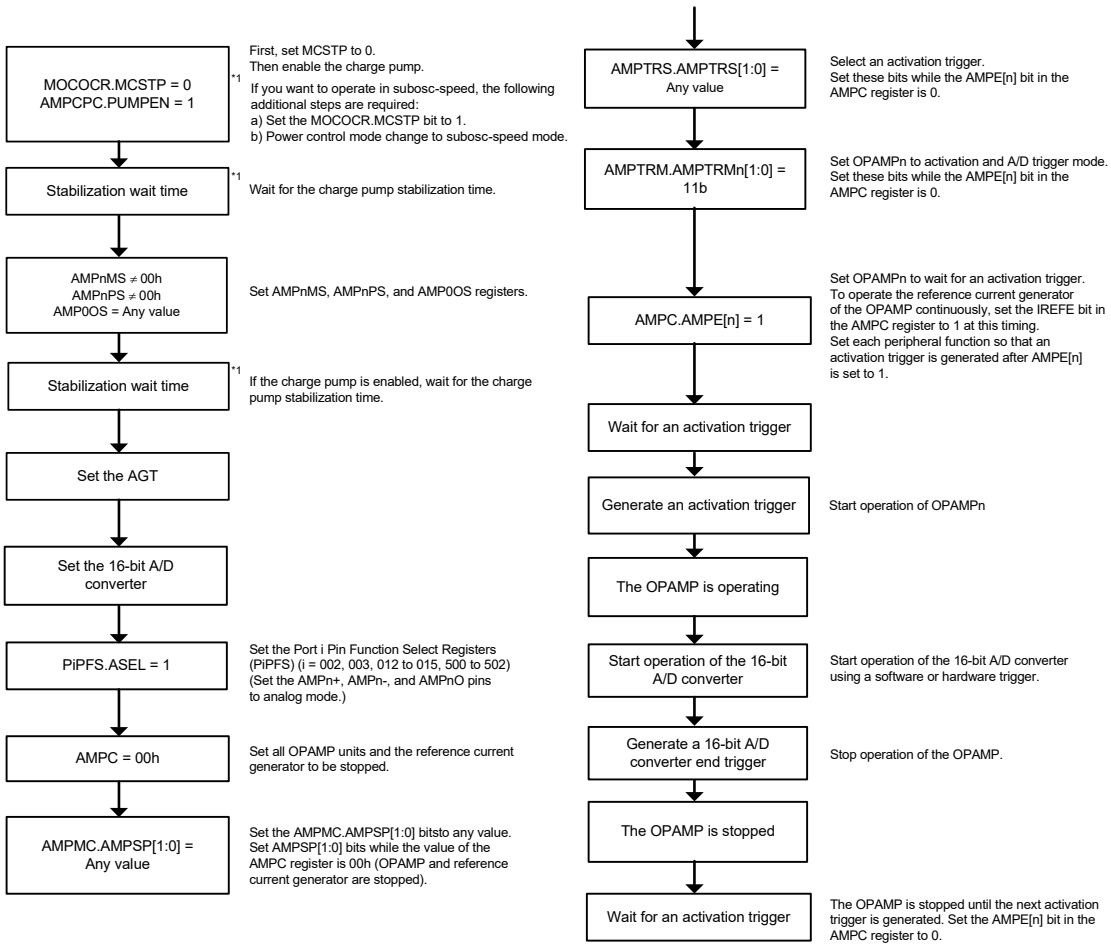


Note 1. When AVCC0 < 2.7 V, these steps are required. For details on the charge pump stabilization wait time, see section 47, Electrical Characteristics.

Figure 37.11 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 OPAMP using an activation trigger and to stop the OPAMP with a 16-bit A/D conversion end trigger. Figure 37.12 shows an example of each register setting.



Note 1. When AVCC0 < 2.7 V, these steps are required. For details on the charge pump stabilization wait time, see section 47, Electrical Characteristics.

Figure 37.12 Procedure to activate OPAMP using an activation trigger and to stop OPAMP with a 16-bit A/D conversion end trigger

37.7 MOCO Stop Procedure after OPAMP Stopped

The following procedure describes how to stop MOCO after OPAMP is stopped (AMPC.AMPE[2:0] = 000b) when AVCC0 < 2.7 V.

1. Stop all OPAMP units including the reference current generator. If operating in subosc-speed mode, the following additional steps are required:
   a. Change the power control mode from subosc-speed mode to another mode.
   b. Set the MOCOCR.MCSTP bit to 0.
2. Set the AMPCPC register to 0 (charge pump disabled).
3. Set the MOCOCR.MCSTP bit to 1 (MOCO is stopped).

Note: The MOCO must not be stopped when DAC12.DAPC.DAPC = 1 or DAC8.DACPC.DAPC = 1.

37.8 Changing OPAMP Switches

This section describes the procedure to change OPAMP switches in the AMPnMS, AMPnPS, and AMP0OS registers without stopping the operational amplifiers. Figure 37.13 shows an example. Although the switch settings can be changed while OPAMP operates (AMPE[n] = 1), carefully consider the effect so that changing the switch settings during operation does not impact the user-prepared peripheral components.

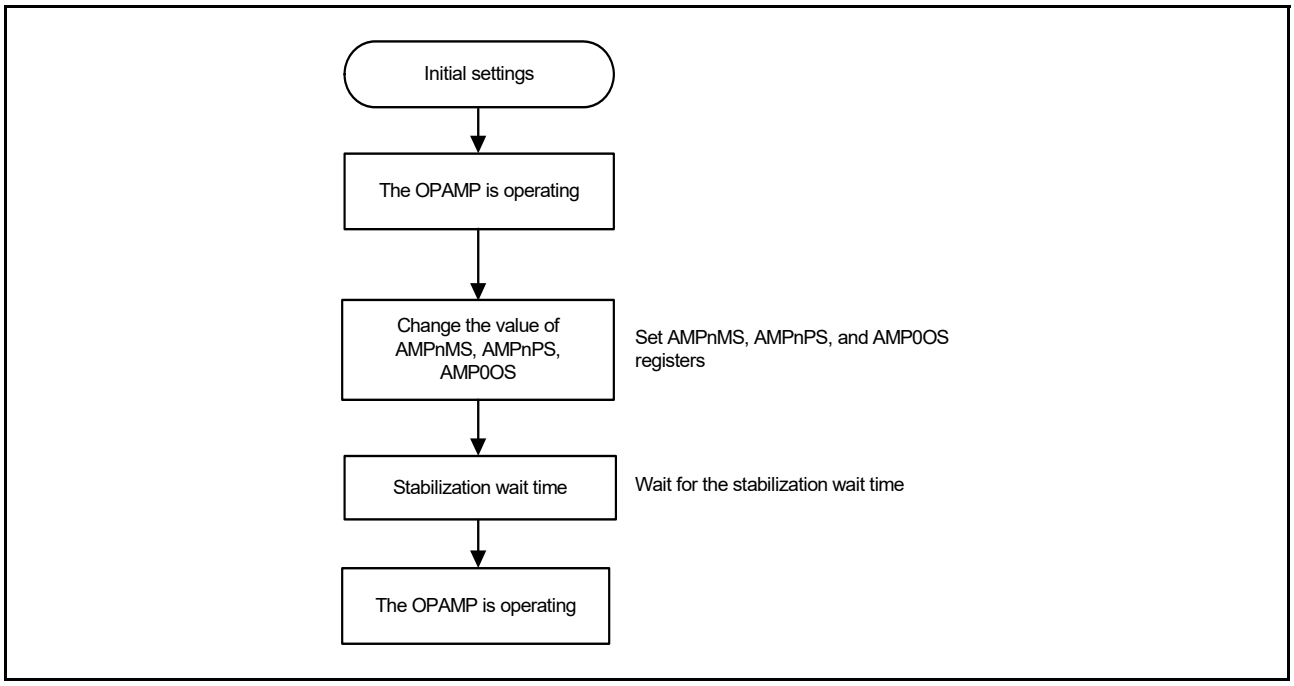


Figure 37.13 Procedure to change OPAMP switches in the AMPnMS, AMPnPS and AMP0OS registers during OPAMP operation

37.9 User Offset Trimming

This section describes the procedure to execute user offset trimming. Figure 37.14 and Figure 37.15 show the connection diagram and an example. The offset trimming value for each operation mode is configured at factory shipment. However, you can set the offset trimming value to suit your usage environment of the product.

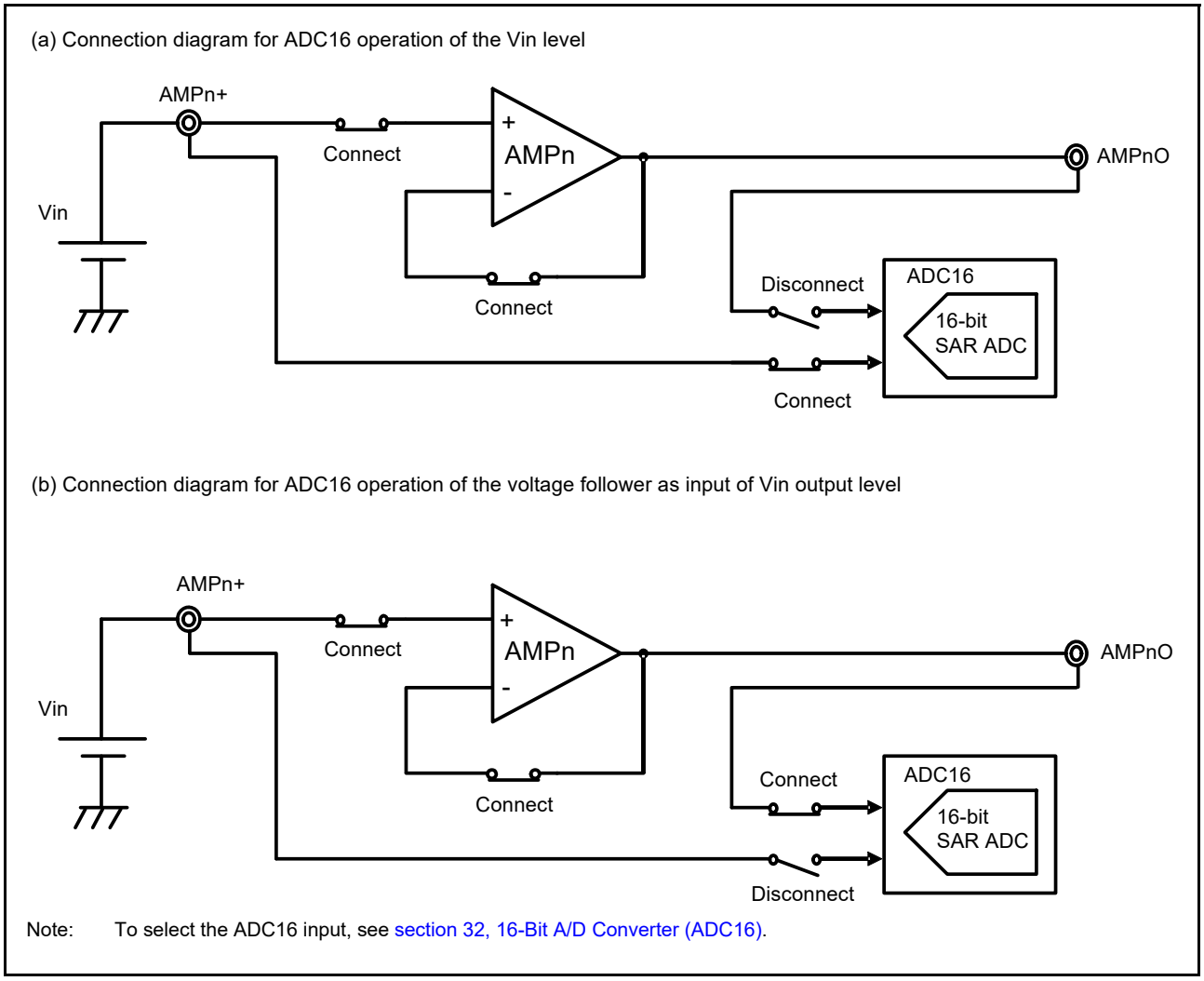
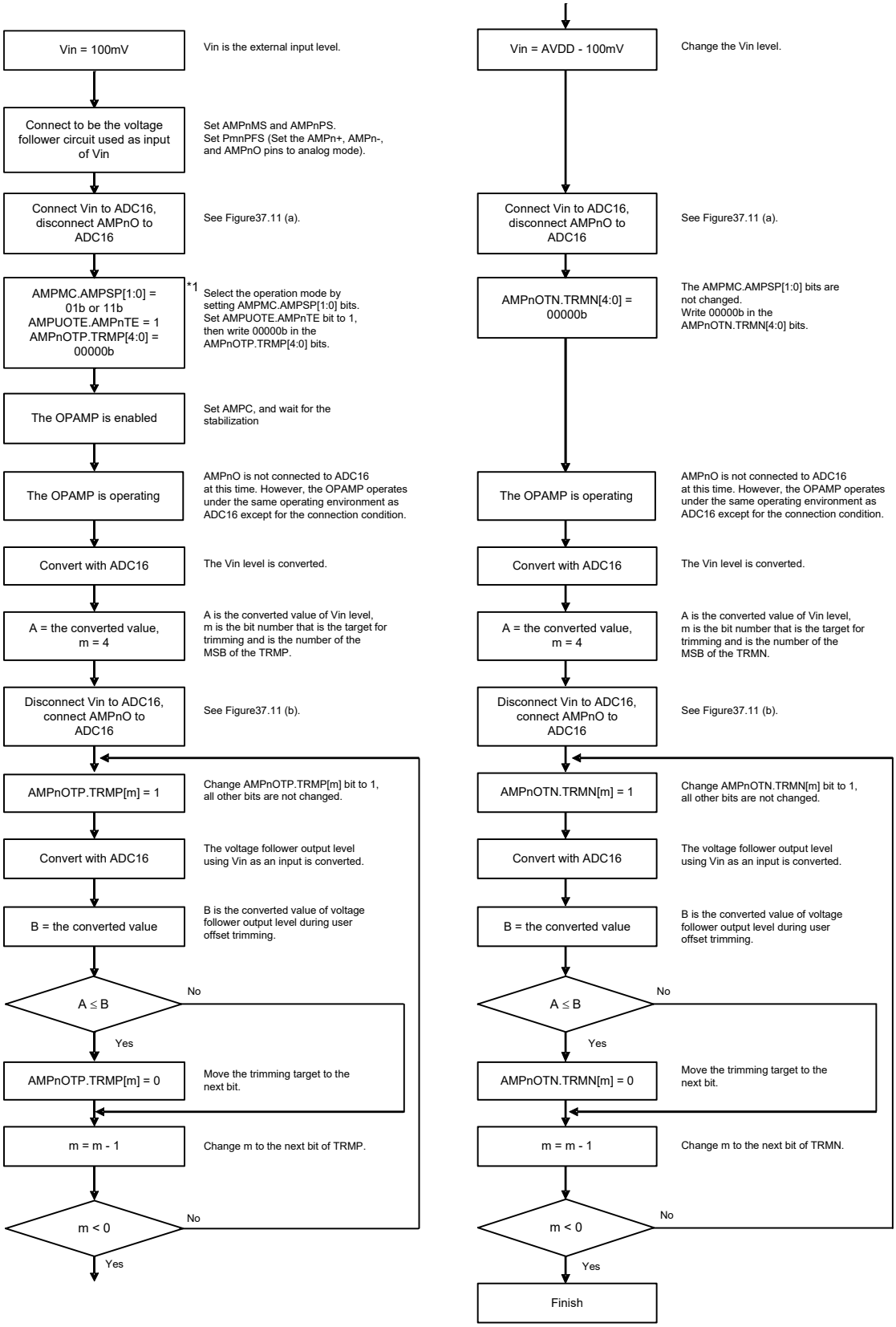


Figure 37.14 Connection diagram to execute user offset trimming



Note 1. User offset trimming cannot be used in low-power mode. When AMPMC.AMPSP[1:0] = x0b, set the AMPUTOTE register to 00h.

Figure 37.15 Procedure to execute user offset trimming

37.10 Using the OPAMP as a Configurable Amplifier

A configurable amplifier can be used as various types of operational amplifiers by controlling the configurable switches in combination with external resistors and capacitors. Figure 37.16 to Figure 37.19 provide typical examples.

37.10.1 Voltage Follower

A general operational amplifier can configure a voltage follower by feeding back its own output signal as its own negative input signal. To configure a feedback circuit, set bit [7] (AMPMS7) of the Operational Amplifier n Minus Input Select Register (AMPnMS) to 1.

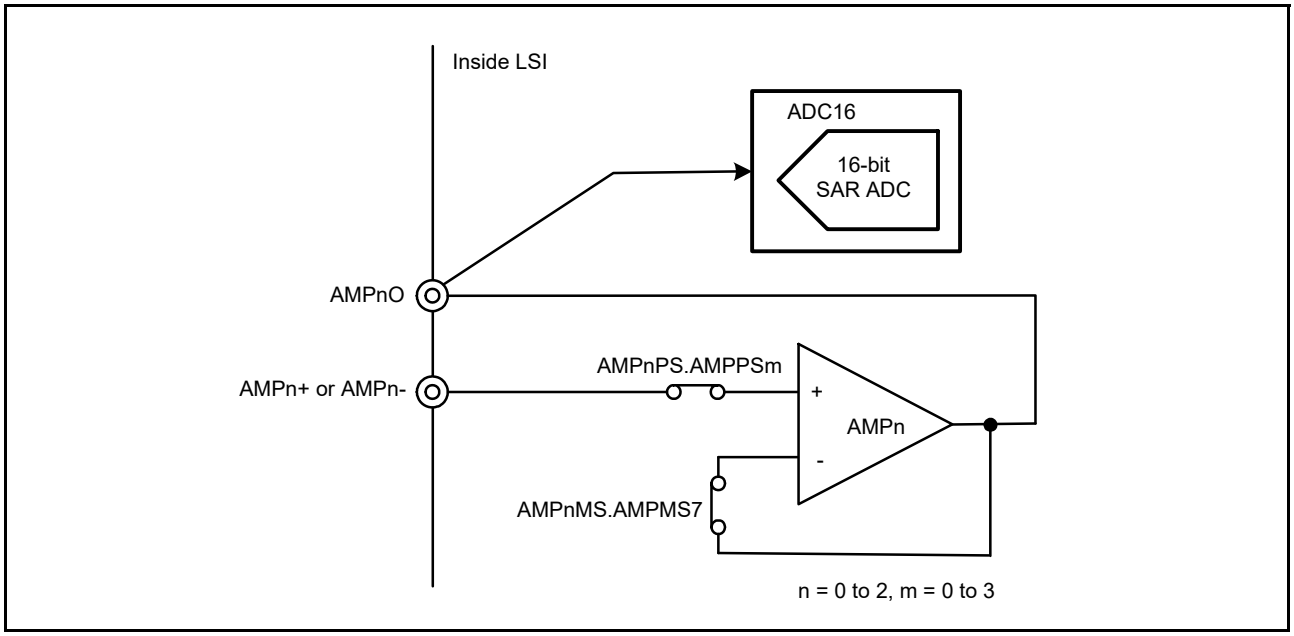


Figure 37.16 Example of using a configurable amplifier as a voltage follower

Use general-purpose analog port n (AMP1+, AMP1-, AMP2+ or AMP2-) to input the pre-amplifier output signal to the post amplifier. To connect the signal output from the voltage follower of operational amplifier 0 to the positive input of operational amplifier 1, for example, set bit [1] (AMP0OS1) of the Operational Amplifier 0 Output Select Register (AMP0OS) to 1. The operational amplifier 0 output is then connected to the general-purpose analog I/O port AMP1+. Then, connect AMP1+ to the positive input signal of post amplifier 1 by setting bit [1] (AMP1PPS) of the Operational Amplifier 1 Plus Input Select Register (AMP1PS) to 1.

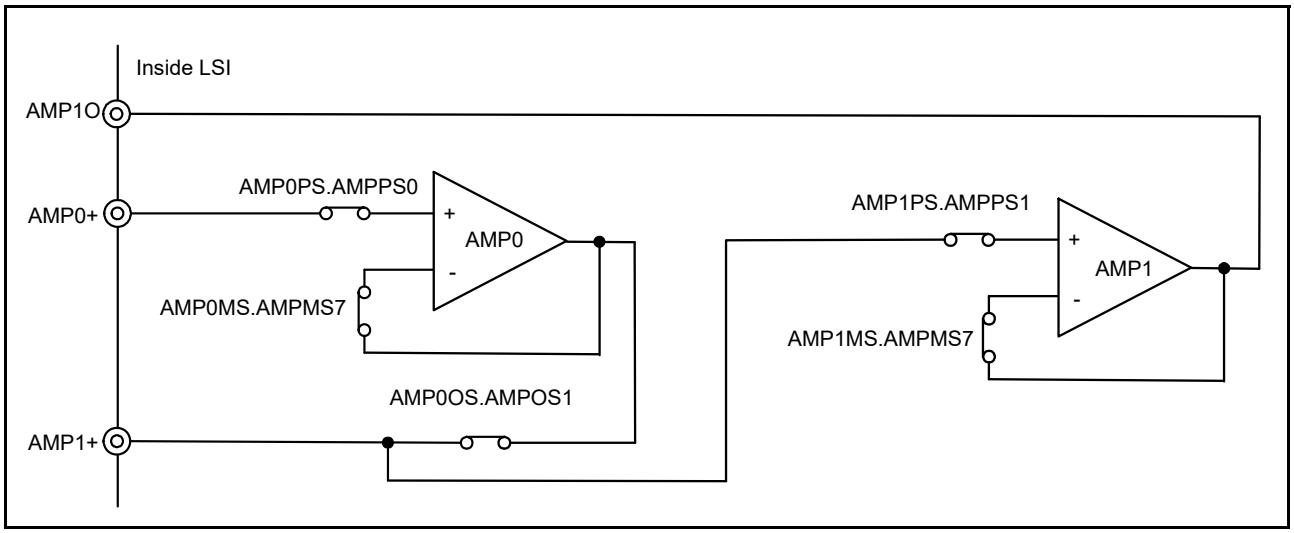


Figure 37.17 Example of using a configurable amplifier as a cascaded voltage follower

37.10.2 Programmable Non-Inverting Amplifier

A programmable non-inverting amplifier can be configured using a combination of configurable switches and external resistors connected to general-purpose analog ports. Figure 37.18 shows an example of a non-inverting amplifier.

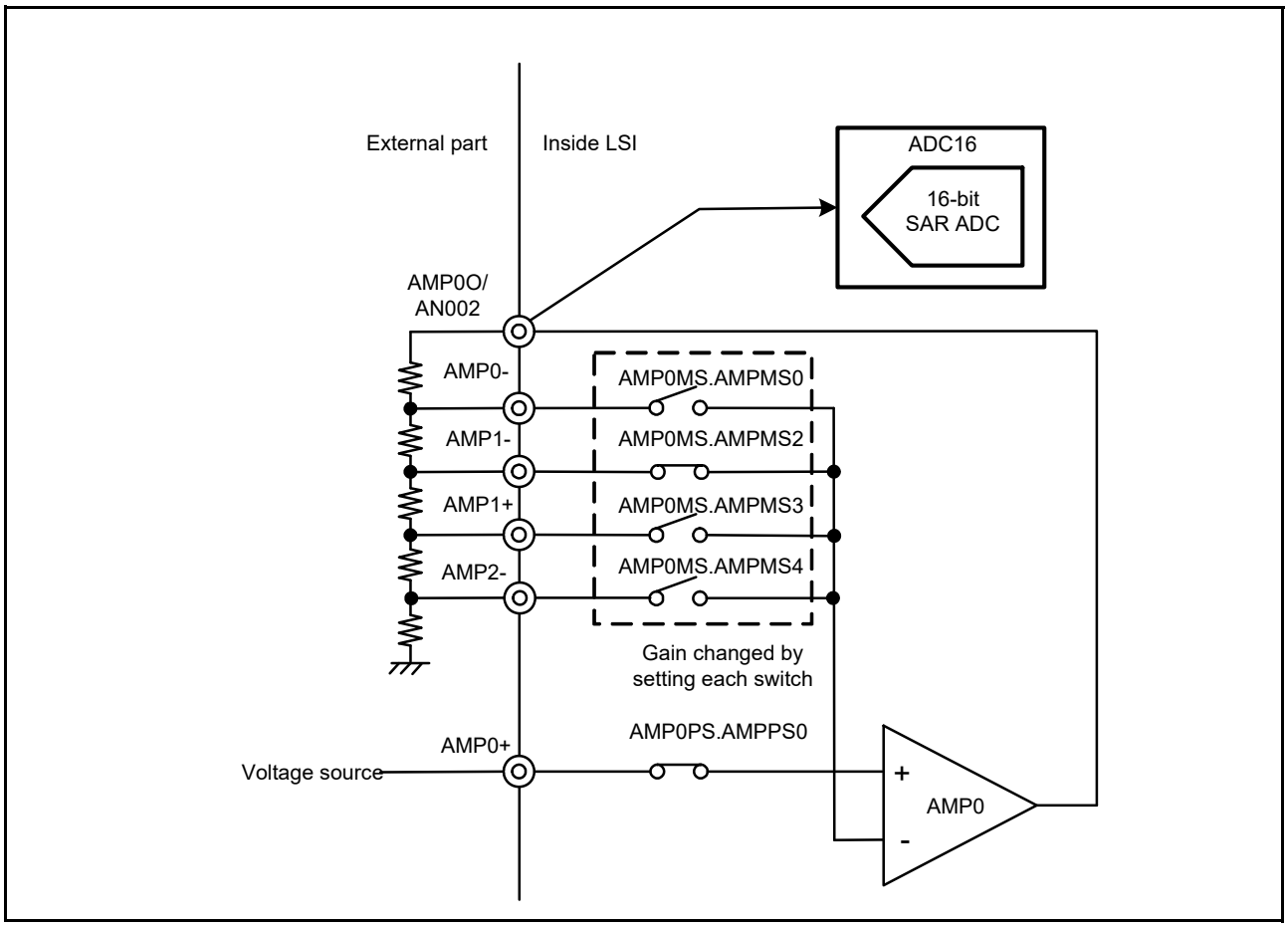


Figure 37.18 Example of using a configurable amplifier as a programmable non-inverting amplifier

37.10.3 Programmable Trans-Impedance Amplifier

Figure 37.19 shows an example of a trans-impedance amplifier in which the gain can be switched using software.

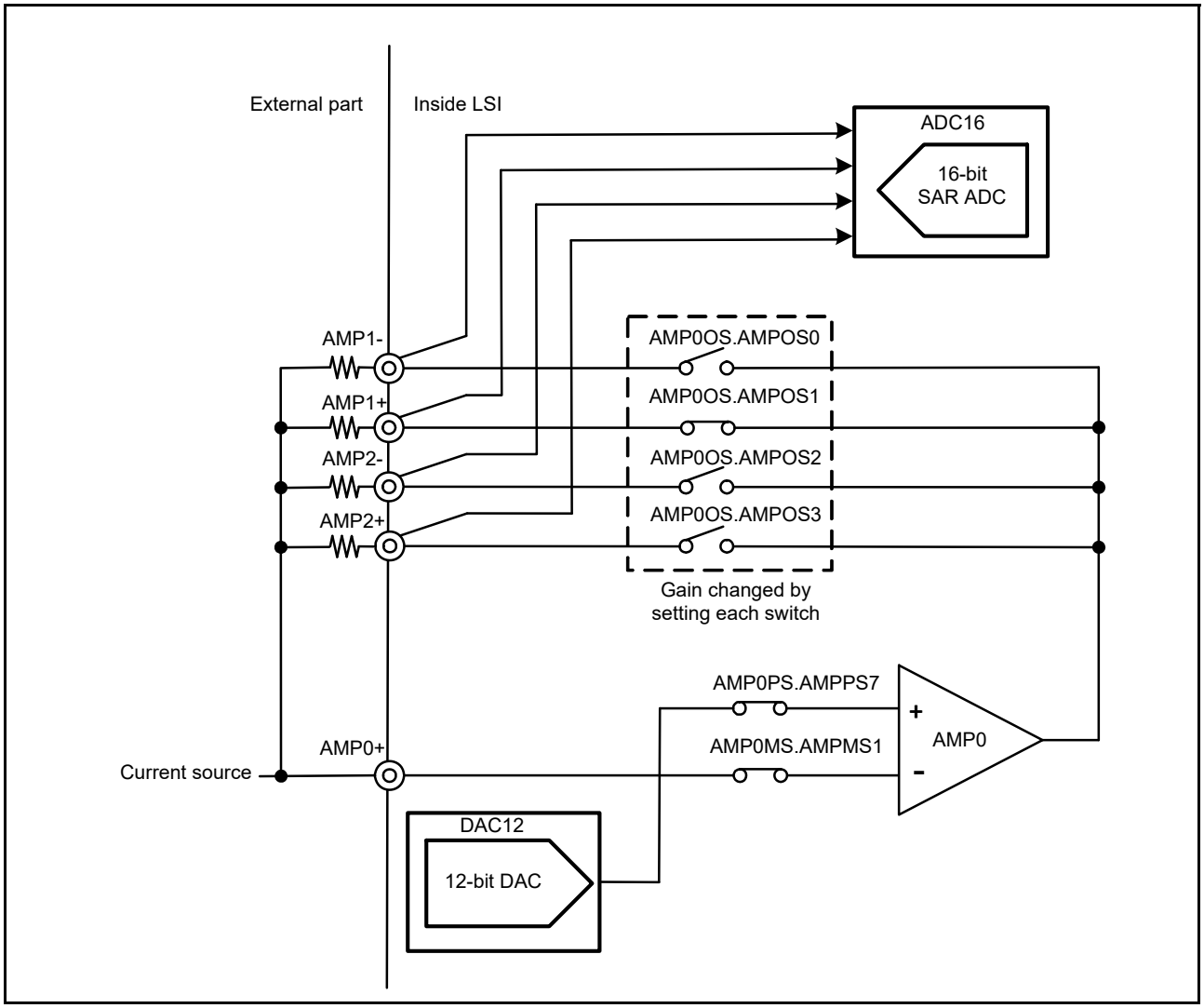


Figure 37.19 Example of using a configurable amplifier as a programmable trans-impedance amplifier

37.10.4 Using the Configurable Amplifier as a D/A Converter Output Amplifier

You can use the configurable amplifier to output the D/A converter output external to the MCU. To input the output from the D/A converter to the positive input pin of operational amplifier n, set bit [7] (AMPPS7) of the Operational Amplifier n Plus Input Select register (AMPnPS) to 1, or output the D/A conversion to the terminal with the Port mn Pin Function Select register (PmnPFS) and set the bit of the operational amplifier n plus input selection register shared by the D/A converter output terminal and the AMP+ input terminal to 1. To output the D/A converter output from the AMPnO pin, configure a voltage follower by setting bit [7] (AMPMS7) of the Operational Amplifier n Minus Input Select Register (AMPnMS) to 1.

The connection to the D/A converter is possible even if the OPAMP external pin is not available.

For details, see section 18.6, Peripheral Select Settings for each Product.

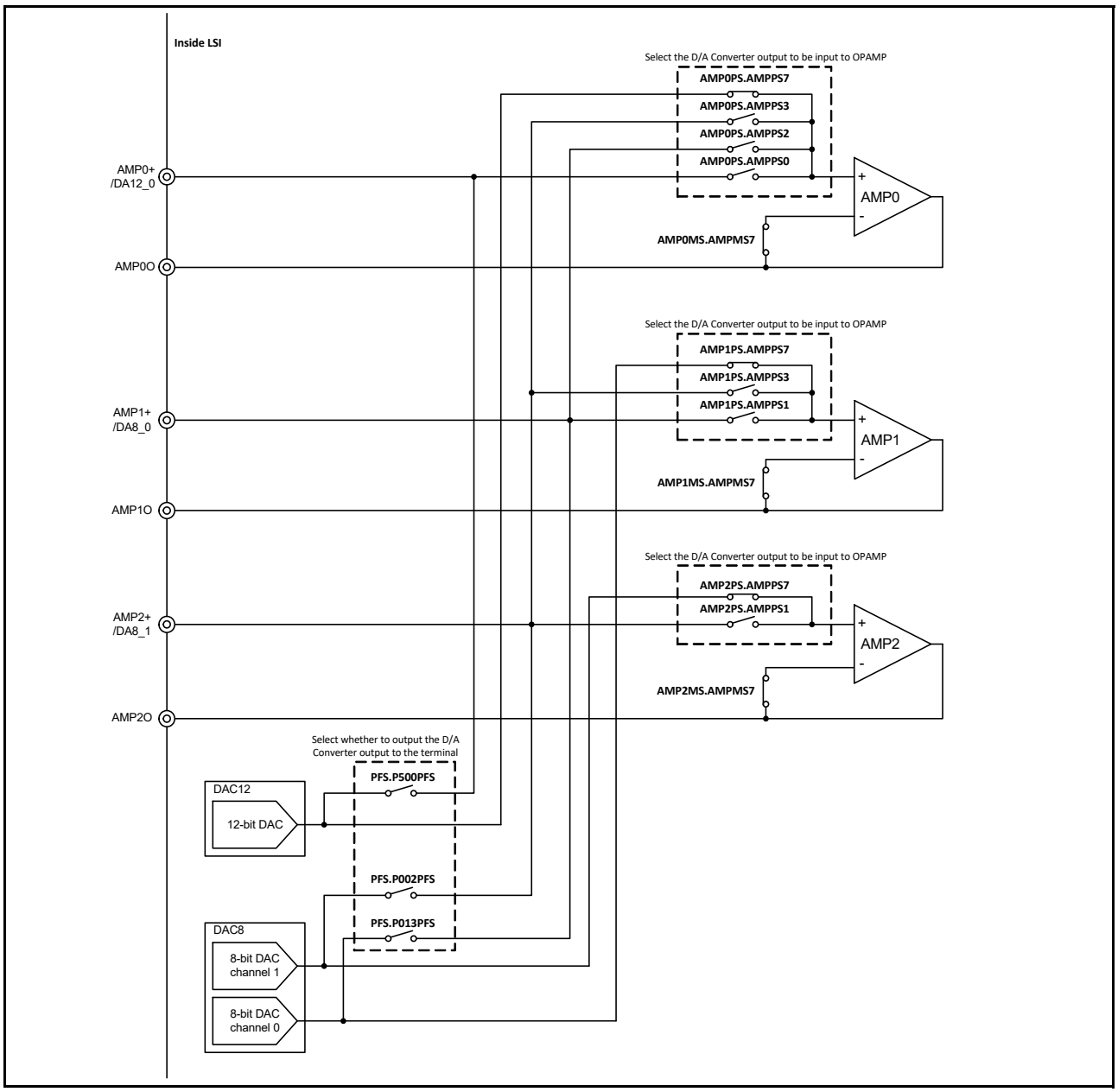


Figure 37.20 Using the configurable amplifier as a D/A converter output amplifier

37.11 Usage Notes

- In addition to the AMPC register settings, the OPAMP function can be activated by an activation trigger and stopped at the end of 16-bit A/D conversion. The reference current circuit can be stopped at the end of 16-bit A/D conversion. Design applications such as circuits and programs should conform to the operation flows to prevent these asynchronous triggers from causing conflicts between the activation and stop control.
- When connecting a bypass capacitor to AVCC0/AVSS0, which are the power supply pins of the OPAMP, place the capacitor as close to the chip as possible, that is, make the wiring as short as possible, and minimize the transfer of noise between the capacitor and the device, board, or peripheral components.
- To prevent the OPAMP inputs being left open, the OPAMP operation is forcibly stopped when all input switches are off.
- When using the pins that also function as digital I/O port pins as the OPAMP ports, do not switch the function to a digital I/O port.
- When the charge pump is enabled and OPAMP2 is operating, if either of the following settings is changed, it is necessary to wait for the settling time to the OPAMP2:
  - When changing the pin output setting of DAC12
  - When changing the pin output setting of DAC8.

  For details on the settling time, see section 47, Electrical Characteristics.
- 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. High-Speed Analog Comparator (ACMPHS)

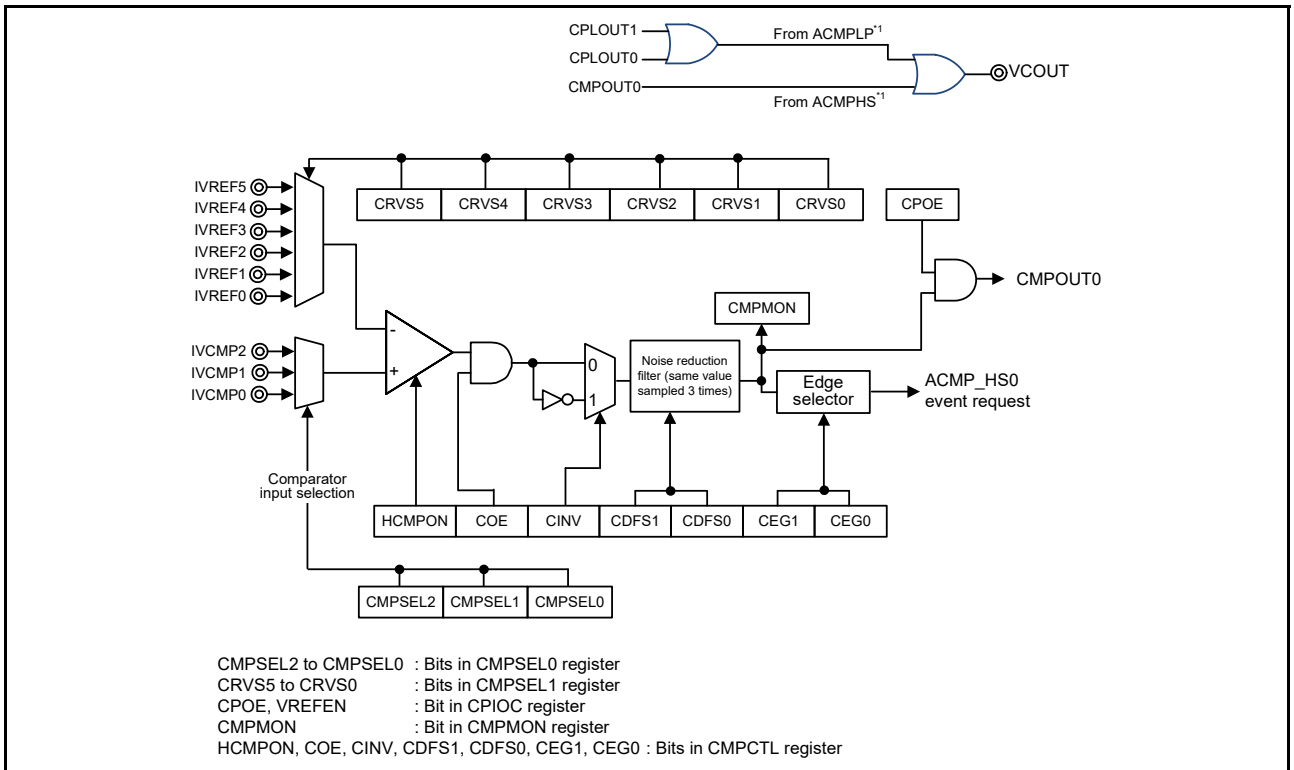
38.1 Overview

The High-Speed Analog Comparator (ACMPHS) compares a reference voltage with an analog input voltage. The comparison result can be read by software and also be output externally. The reference voltage can be selected from either an input to the IVREFi (i = 0 to 2) pin, an output from internal D/A converter, or from the internal reference voltage (Vref) generated internally in the MCU. Such flexibility is useful in applications that require go/no-go comparisons to be performed between analog signals without requiring A/D conversion.

Table 38.1 lists the ACMPHS specifications, Figure 38.1 shows a block diagram, and Table 38.2 shows the input source configuration of the ACMPHS.

Table 38.1 ACMPHS specifications

Parameter	Specifications
Number of channels	1 channel (ACMPHS0)
Analog input voltage	Input from internal ADC16 input pin
Reference voltage	• Output from internal DAC12 • Output from internal DAC8 (channel 0) • Internal reference voltage (Vref) • Input from internal ADC16 input pin.
ACMPHS output	• Comparison result • Generation of ELC event output • Monitor output from register.
Interrupt request signal	• Interrupt request generated on valid edge detection from comparison result • Rising edge, falling edge, or both edges can be selected.
Digital filter function	• One of three sampling frequencies can be selected • Not using the filter function can be selected.



Note 1. ACMPHS0 and ACMPLP results are output in VCOUT.

Figure 38.1 ACMPHS block diagram

Table 38.2 Input source configuration of the ACMPHS

Comparator	Reference voltage input source						Analog voltage input source			Output pin
	IVREF5	IVREF4	IVREF3	IVREF2	IVREF1	IVREF0	IVCMP2	IVCMP1	IVCMP0	
ACMPHS0	Vref*3	DAC12 output	DAC8 channel 0 output	AN017*4	AN004*4,*5	AN001*4	AN016*4	AN005 *1,*4,*5	AN000 *2,*4	VCOUT

Note 1. When the output level of DAC8 channel 0 is not output to pin (DA8_0), it can be used as AN005 analog input.
Note 2. When the output level of DAC12 is not output to pin (DA12_0), it can be used as AN000 analog input.
Note 3. Internal reference voltage.
Note 4. When using for ACMPHS input, it cannot be selected as A/D conversion target.
Note 5. When using for ACMPHS input, it cannot be selected as analog reference voltage supply pin for DAC12.

38.2 Register Descriptions

38.2.1 Comparator Control Register (CMPCTL)

Address(es): ACMPHS0.CMPCTL 4008 5000h

	b7	b6	b5	b4	b3	b2	b1	b0
	HCMPON	CDFS[1:0]		CEG[1:0]		—	COE	CINV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CINV	Comparator Output Polarity Selection*1, *2	0: Comparator output not inverted 1: Comparator output inverted.	R/W
b1	COE	Comparator Output Enable	0: Comparator output disabled (the output signal is low level) 1: Comparator output enabled.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4, b3	CEG[1:0]	Selection of Valid Edge (Edge Selector)	b4 b3 0 0: No edge selection 0 1: Rising edge selection 1 0: Falling edge selection 1 1: Both edge selection.	R/W
b6, b5	CDFS[1:0]	Noise Filter Selection*1, *2, *3	b6 b5 0 0: Noise filter not used 0 1: Noise filter sampling frequency is PCLKB divided by 8 1 0: Noise filter sampling frequency is PCLKB divided by 16 1 1: Noise filter sampling frequency is PCLKB divided by 32.	R/W
b7	HCMPON	Comparator Operation Control*4	0: Operation stopped (the comparator outputs a low-level signal) 1: Operation enabled (enable input to the comparator pins).	R/W

Note 1. Change CDFS[1:0] and CINV bits only after disabling the ACMPHS output (COE = 0).
Note 2. If the CDFS[1:0] and CINV bits are changed, an ACMPHS interrupt request and an ELC event can be generated. Before changing these bits, set the ELC.ELSRn register to 0000h (the ACMPHS output is not linked). After changing these bits, set the IR flag in the ICU.IELSRn register to 0 to clear the interrupt status.
Note 3. If the CDFS[1:0] bits are changed from 00b (noise filter not used) to a value other than 00b (noise filter used), perform sampling four times, update the filter output, then use the ACMPHS interrupt request or the ELC event.
Note 4. A stabilization wait time is required to permit ACMPHS operation after enabling it (HCMPON = 1). The operation stabilization wait time for ACMPHS0 is 1 μs.

The CMPCTL register does the following:

- Controls the ACMPHS operation
- Enables or disables the ACMPHS output
- Selects the noise filter
- Selects the valid edge of the interrupt signal
- Selects the interrupt.

A reset clears this register to 00h.

38.2.2 Comparator Input Select Register (CMPSEL0)

Address(es): ACMPHS0.CMPSEL0 4008 5004h

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

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CMPSEL[2:0]	Comparator Input Selection*1	b2 b0 0 0 0: No input 0 0 1: IVCMP0 selected*2 0 1 0: IVCMP1 selected*2 1 0 0: IVCMP2 selected.*2 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. Use the following procedure to modify the CMPSEL[2:0] bits. Writing a value other than 0000 0000b while the value of the CMPSEL0 register is not 0000 0000b is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.
To change the CMPSEL[2:0] bits:
1. Set the CMPCTL.COE bit to 0.
2. Set the CMPSEL0 register to 0000 0000b.
3. Set a new value in the CMPSEL[2:0] bits, with 1 set in only one of the bits.
4. Wait for the input switching stabilization wait time of 200 ns.
5. Set the CMPCTL.COE bit to 1.
6. Clear IR flag in the ICU.IELSRn register to clear the interrupt status.

Note 2. For details, see Table 38.2, Input source configuration of the ACMPHS.

38.2.3 Comparator Reference Voltage Select Register (CMPSEL1)

Address(es): ACMPHS0.CMPSEL1 4008 5008h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CRVS[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	CRVS[5:0]	Reference Voltage Selection*1	b5 b0 0 0 0 0 0 0: No reference voltage 0 0 0 0 0 1: IVREF0 selected*2 0 0 0 0 1 0: IVREF1 selected*2 0 0 0 1 0 0: IVREF2 selected*2 0 0 1 0 0 0: IVREF3 selected*2 0 1 0 0 0 0: IVREF4 selected*2 1 0 0 0 0 0: IVREF5 selected.*2 Other settings are prohibited.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Use the following procedure to modify the CRVS[5:0] bits. Writing a value other than 0000 0000b while the value of the CMPSEL1 register is not 0000 0000b is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.
To change the CRVS[5:0] bits:
1. Set the CMPCTL.COE bit to 0.
2. Set the CMPSEL1 register to 0000 0000b.
3. Set a new value in the CRVS[5:0] bits, with 1 set in only one of the bits.
4. Wait for the input switching stabilization wait time of 200 ns.
5. Set the CMPCTL.COE bit to 1.
6. Set the IR flag in the ICU.IELSRn register to clear the interrupt status.

Note 2. For details, see Table 38.2, Input source configuration of the ACMPHS.

38.2.4 Comparator Output Monitor Register (CMPMON)

Address(es): ACMPHS0.CMPMON 4008 500Ch

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

Bit	Symbol	Bit name	Description	R/W
b0	CMPMON	Comparator Output Monitor*1	0: Comparator output low 1: Comparator output high.	R
b7 to b1	—	Reserved	These bits are read as 0.	R

Note 1. When ACMPHS operation is enabled (HCMPON = 1 and COE = 1) but the noise filter is not in use (CDFS[1:0] = 00b), ensure that the CMPMON bit is read twice and the values are only used after the two consecutive values match.

38.2.5 Comparator Output Control Register (CPIOC)

Address(es): ACMPHS0.CPIOC 4008 5010h

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

Bit	Symbol	Bit name	Description	R/W
b0	CPOE	Comparator Output Selection	0: VCOUT pin output of the comparator disabled (the output signal is low) 1: VCOUT pin output of the comparator enabled.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	VREFEN	Internal Reference Voltage Enable	0: Internal reference voltage (Vref) disabled 1: Internal reference voltage (Vref) enabled.	R/W

38.3 Operation

The ACMPHS compares a reference voltage to an analog input voltage. Operation is not guaranteed when the register values are changed during ACMPHS operation. Table 38.3 shows the procedures for setting the registers associated with the ACMPHS.

Table 38.3 Procedure for setting registers associated with ACMPHS (1 of 2)

Step	Register	Bit	Setting
1	Associated MSTPCRD register	MSTPD28	0: Input clock supply.
2	Associated Port mn Pin Function Select register (PmnPFS)	ASEL	1: Select the function of pins IVREF and IVCMP.
3	ACMPHS0.CPIOC	VREFEN	1: When using the internal reference voltage (Vref).
4	Associated DAC12 and DAC8		When using DAC12 and DAC8, set in the register.
5	ACMPHS0.CMPSEL0 ACMPHS0.CMPSEL1	CMPSEL[2:0] CRVS[5:0]	Select the ACMPHS0 input, with 1 set in only one of the bits.
6	CMPCTL	CDFS[1:0], CEG1, CEG0, and CINV	Set the ACMPHS0 control.
		HCMPON	1: Enable the ACMPHS0 operation.
7	Waiting for the ACMPHS stabilization time (min. 1 μs).		
8	CMPCTL	COE	1: Enable the ACMPHS0 output.
9	CPIOC	CPOE	1: Set the VCOUT output.
	Associated Port mn Pin Function Select register (PmnPFS)	PSEL, PMR	Select the VCOUT port function.
10	IELSRn	IR, IELS[7:0]	When using an interrupt, select the interrupt status flag and the ICU event link.*1
11	ELSRn	ELS[7:0]	When using an ELC, select the event link.*2
12	POEGGn	CDRE0	When using a POEG, enable the request from the ACMPHS.
13	Operation started		
14	CMPCTL	COE	0: When changing IVREF or IVCMP, disable ACMPHS0 output.
15	ACMPHS0.CMPSEL1	CRVS[5:0]	Change the CRVS[5:0] bits as follows: 1. Set the CMPSEL1 register to 0000 0000b. 2. Set a new value to the CRVS[5:0] bits, with 1 set in only one of the bits.

Table 38.3 Procedure for setting registers associated with ACMPHS (2 of 2)

Step	Register	Bit	Setting
16	ACMPHS0.CMPSEL0	CMPSEL[2:0]	Change the CMPSEL[n] bits as follows: 1. Set the CMPSEL0 register to 0000 0000b. 2. Set a new value to CMPSEL[n] bits, with 1 set in only one of the bits.
17	Waiting for the ACMPHS0 switching stabilization time (min. 200 ns).		
18	CMPCTL	COE	1: Enable the ACMPHS0 output.
19	Operation restarted		

Note 1. After ACMPHS0 is set, an unnecessary interrupt might occur until operation becomes stable, so initialize the interrupt flag.

Note 2. After ACMPHS0 is set, an unnecessary interrupt might occur until operation becomes stable, so initialize the event link select.

Figure 38.2 shows an example of ACMPHS operation. The VCOUT output becomes 1 when the analog input voltage is higher than the reference input voltage, and the VCOUT output becomes 0 when the analog input voltage is lower than the reference voltage. When the ACMPHS output changes, an interrupt request and an ELC event are output.

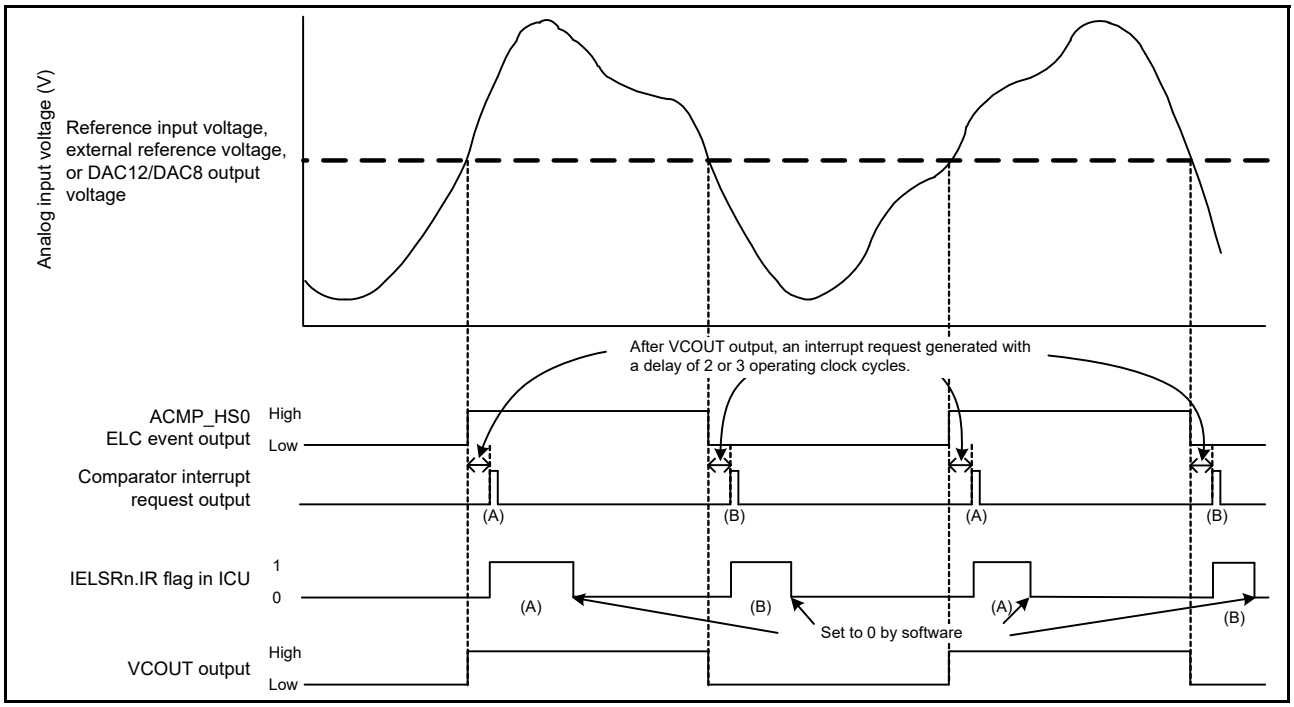


Figure 38.2 ACMPHS operation example

Figure 38.2 applies when CPOE = 1 (pin output enabled), CDFS[1:0] = 00b (filter not used), and CEG1 = CEG0 = 1 (both edge selection). When CINV = 0, CEG0 = 1, and CEG1 = 0 (rising edge selection for non-inversion output signal from the ACMPHS), the ICU.IELSRn.IR flag changes as shown by (A) only. When CINV = 0, CEG0 = 0, and CEG1 = 1 (falling edge selection for non-inversion output signal from the ACMPHS), the IR flag changes as shown by (B) only. When CPOE = 1, VCOUT directly outputs the ELC event output.

38.4 Noise Filter

The ACMPHS contains a noise filter. The sampling clock is selected in the CMPCTL.CDFS[1:0] bits. The ACMPHS signal is sampled on each sampling clock. If the same value is sampled three times, the noise filter output at the next sampling clock cycle is used as the ACMPHS output.

Figure 38.3 shows the configuration of the noise filter and edge detector, and Figure 38.4 shows an example of the noise filter and interrupt operation. When using an interrupt and ELC is in Software Standby mode, set the CMPCTL.CDFS[1:0] bits to 00b (noise filter not used).

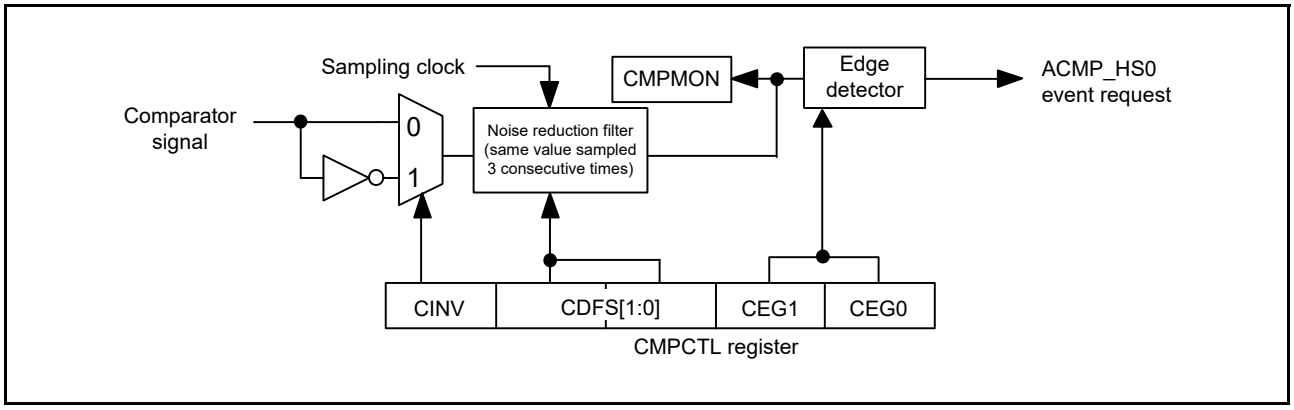


Figure 38.3 Noise filter and edge detection configuration

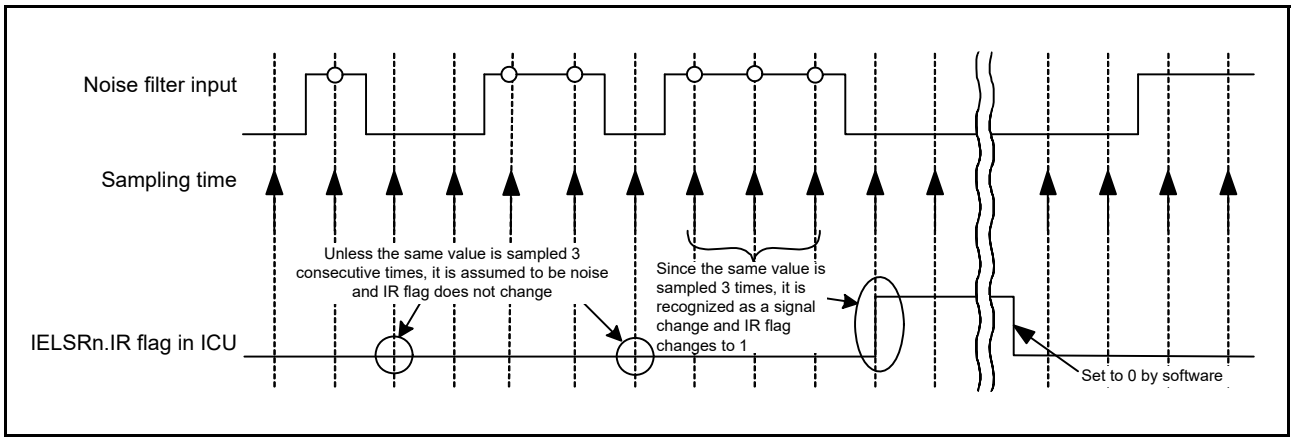


Figure 38.4 Noise filter and interrupt operation example

The operation example in Figure 38.4 applies when the CMPCTL.CDFS[1:0] bits are 01b, 10b, or 11b (noise filter used).

38.5 ACMPHS Interrupts

The ACMPHS generates one interrupt request from ACMPHS0. To use an ACMPHS interrupt, select it in the IELSRn register in the Interrupt Controller Unit (ICU). Set at least one of the CMPCTL.CEG0 or CMPCTL.CEG1 bits to 1, that is, to a value other than 00b for no edge selection.

For details on the register setting related to ACMPHS interrupt request, see section 38.2.1, Comparator Control Register (CMPCTL).

38.6 ACMPHS Output to the Event Link Controller (ELC)

The ELC uses the ACMPHS interrupt request signal as an ELC event signal, enabling link operation for the preset module. To use the ACMPHS ELC event, select it in the ELSRn register in the ELC. When using the ELC event request, set at least one of the CMPCTL.CEG0 or CMPCTL.CEG1 bits to 1, that is, to a value other than 00b for no edge selection.

38.7 ACMPHS Pin Output

The comparison result from the ACMPHS can be output to external pins. Use the CMPCTL.CINV and CPIOC.CPOE bits to set the output polarity (non-inverted or inverted output) and enable or disable output. To output the ACMPHS comparison result to the VCOUT output pin, set the associated Port mn Pin Function Select register (PmnPFS) in the I/O register.

38.8 Usage Notes

38.8.1 Settings for the Module-Stop Function

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

38.8.2 Relationship with A/D Converter

Simultaneous use of ACMPHS analog input and A/D converter analog input have restrictions. For details, see section 32.7.12, Relationship between the ADC16, OPAMP, ACMPHS, and SDADC24, and section 33.5.8, Relationship between the SDADC24, ADC16 and ACMPHS.

39. Low-Power Analog Comparator (ACMPLP)

39.1 Overview

The Low-Power Analog Comparator (ACMPLP) compares a reference voltage with an analog input voltage. Comparator channels ACMPLP0 and ACMPLP1 are independent of each other.

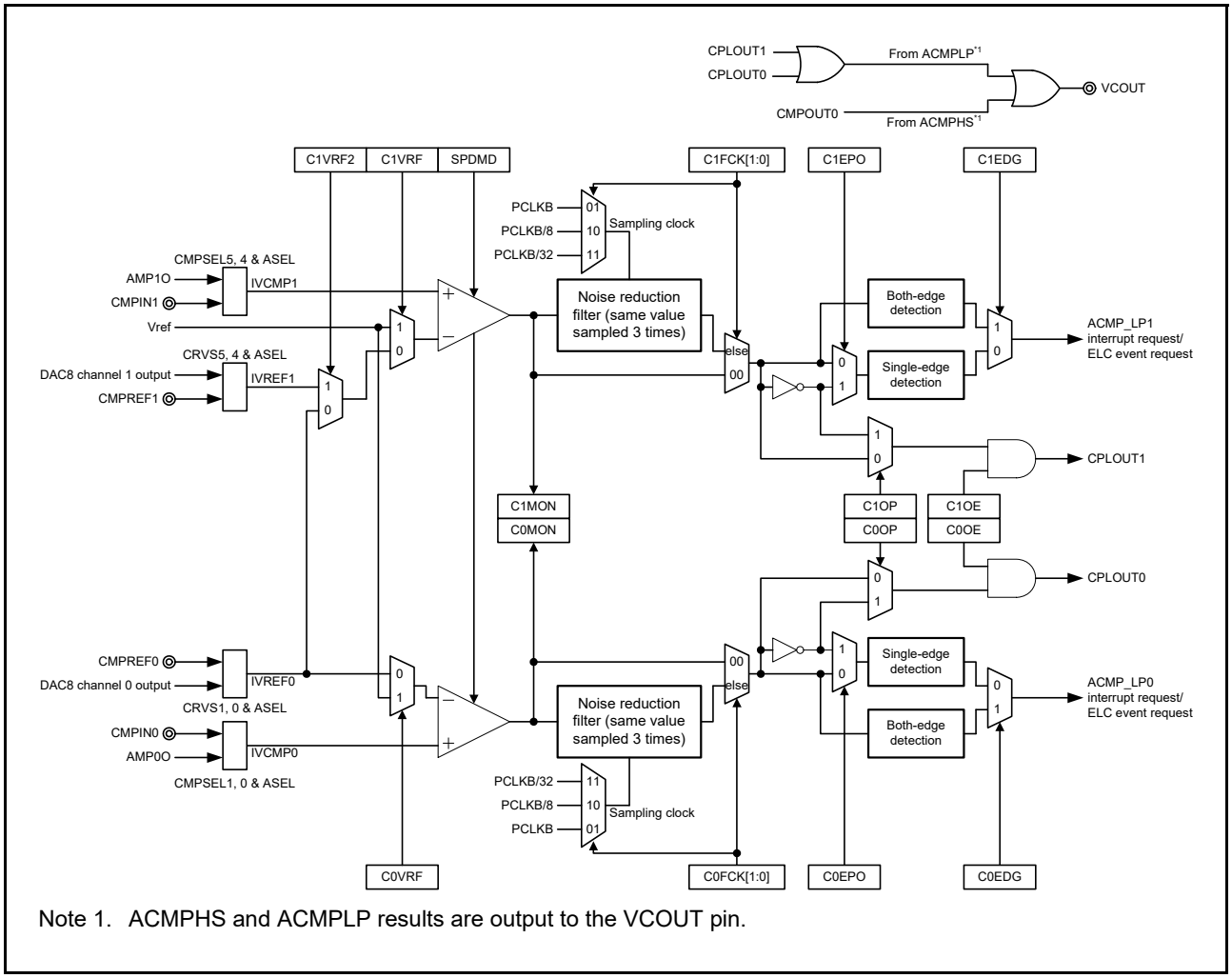
The comparison result of the reference input voltage and analog input voltage can be read by software. The comparison result can also be output externally. The reference voltage can be selected from an input to the CMPREFi (i = 0, 1) pin, an output from the internal DAC8, and the internal reference voltage (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 39.1 lists the ACMPLP specifications, Figure 39.1 shows a block diagram of the ACMPLP when the window function is disabled, and Figure 39.2 shows a block diagram of the ACMPLP when the window function is enabled. Table 39.2 lists the I/O pins of the ACMPLP.

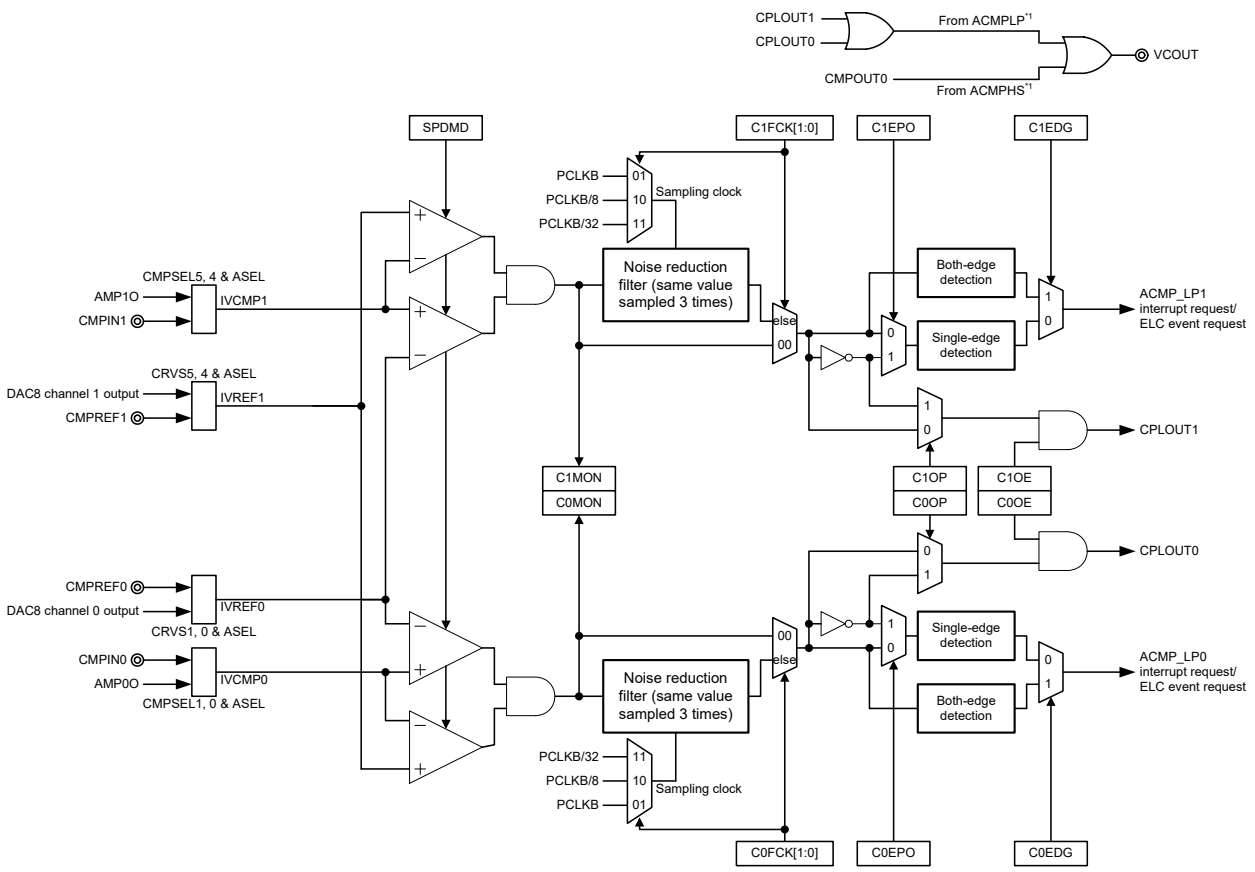
Table 39.1 ACMPLP specifications

Parameter	Specifications
Number of channels	2 channels: ACMPLP0 and ACMPLP1
Analog input voltage	• Input from CMPINi (i = 0, 1) pin • Output from internal operational amplifier (AMP0O, AMP1O).
Reference voltage	• Standard mode One of the following can be selected: - Internal reference voltage (Vref) - Input from CMPREFi (i = 0, 1) pin - Output from the internal DAC8. • Window mode One of the following can be selected: - Input from CMPREFi (i = 0, 1) pin (CMPREF0: low reference, CMPREF1: high reference) - Output from the internal DAC8.
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.



Note 1. ACMPHS and ACMPLP results are output to the VCOUT pin.

Figure 39.1 ACMPLP block diagram when window function is disabled in standard mode



Note 1. ACMPHS and ACMPLP results are output to the VCOUT pin.

Figure 39.2 ACMPLP block diagram when window function is enabled in window function mode

Table 39.2 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/ DAC8 channel 0 output) • Vref (selectable)	Low reference voltage: • IVREF0 (CMPREF0/ DAC8 channel 0 output) High reference voltage: • IVREF1 (CMPREF1/ DAC8 channel 1 output)	• IVCMP0 (CMPIN0/AMP0O)		VCOUT*1
ACMPLP1	• IVREF0 (CMPREF0/ DAC8 channel 0 output) • IVREF1 (CMPREF1/ DAC8 channel 1 output) • Vref (selectable)		• IVCMP1 (CMPIN1/AMP1O)		

Note 1. ACMPHS0 and ACMPLPn (n = 0, 1) results are output to the VCOUT pin.

39.2 Register Descriptions

39.2.1 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: Disabled 1: Enabled.	R/W
b1	C0WDE	ACMPLP0 Window Function Mode Enable*1,*2	0: Disabled 1: Enabled.	R/W
b2	C0VRF	ACMPLP0 Reference Voltage Selection	0: IVREF0 input 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: Disabled 1: Enabled.	R/W
b5	C1WDE	ACMPLP1 Window Function Mode Enable*1,*2	0: Disabled 1: Enabled.	R/W
b6	C1VRF	ACMPLP1 Reference Voltage Selection	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 this bit setting.

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.
When the reference level is equal to the input level, the bit value is undefined.

Note 4. The setting is valid only when in standard mode. When in window function mode, IVREF0 or IVREF1 is selected regardless of this bit setting.

39.2.2 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 modified, an ACMPLP interrupt request and an ELC event request can be generated. Change these bits only after setting event link to deselected. Also, be sure to clear the associated interrupt request flag.

39.2.3 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: Disabled 1: Enabled.	R/W
b2	C0OP	ACMPLP0 VCOUT Output Polarity Selection*1	0: Non-inverted 1: Inverted.	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	C1OE	ACMPLP1 VCOUT Pin Output Enable*1	0: Disabled 1: Enabled.	R/W
b6	C1OP	ACMPLP1 VCOUT Output Polarity Selection*1	0: Non-inverted 1: Inverted.	R/W
b7	SPDMD	ACMPLP0/ACMPLP1 Speed Selection*2	0: Comparator low-speed mode 1: Comparator high-speed mode.	R/W

Note 1. ACMPHS0 and ACMPLPn (n = 0, 1) results are bundled on the VCOUT pin.
Note 2. When rewriting the SPDMD bit, be sure to set the CiENB bit (i = 0, 1) in the COMPMDR register to 0 in advance.

39.2.4 Comparator Input Select Register (COMPSEL0)

Address(es): ACMPLP.COMPSEL0 4008 5E04h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CMPSEL[5:4]		—	—	CMPSEL[1:0]	
Value after reset:	0	0	0	1	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b1, b0	CMPSEL[1:0]	ACMPLP0 Input (IVCMP0) Selection	b1 b0 0 0: No input 0 1: CMPIN0 input selected 1 0: AMP0O output selected. Other settings are prohibited.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	CMPSEL[5:4]	ACMPLP1 Input (IVCMP1) Selection	b5 b4 0 0: No input 0 1: CMPIN1 input selected 1 0: AMP1O output selected. Other settings are prohibited.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

39.2.5 Comparator Reference Voltage Select Register (COMPSEL1)

Address(es): ACMPLP.COMPSEL1 4008 5E05h

	b7	b6	b5	b4	b3	b2	b1	b0
	C1VRF2	—	CRVS[5:4]		—	—	CRVS[1:0]	
Value after reset:	1	0	0	1	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b1, b0	CRVS[1:0]	ACMPLP0 Reference Voltage (IVREF0) Selection	b1 b0 0 0: No reference voltage 0 1: CMPREF0 selected 1 0: DAC8 channel 0 output selected. Other settings are prohibited.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	CRVS[5:4]	ACMPLP1 Reference Voltage (IVREF1) Selection	b5 b4 0 0: No reference voltage 0 1: CMPREF1 selected 1 0: DAC8 channel 1 output selected. Other settings are prohibited.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	C1VRF2	ACMPLP1 Reference Voltage Selection 2	0: IVREF0 selected 1: IVREF1 selected.	R/W

39.3 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 39.3 shows the procedure for setting the ACMPLP associated registers.

Table 39.3 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[1:0], CMPSEL[5:4]		
3	COMPOCR	SPDMD	Select the comparator response speed 0: Low-speed mode 1: High-speed mode.*1	
4	COMPMDR	CiWDE	0: Window function mode disabled	1: Window function mode enabled.*2
		CiVRF	Select the reference voltage.	Window comparator operation (reference = IVREF0 and IVREF1*3).
	COMPSEL1	CRVS[1:0], CRVS[5:4], C1VRF2		
	COMPMDR	CiENB	1: Operation enabled.	
5	Waiting for the comparator stabilization time T_{cmp} (min. 100 μs).			
6	COMPFIR	CiFCK[1:0]	Select whether the digital filter is used or not and the sampling clock.	
		CiEPO, CiEDG	Select the edge detection condition for an interrupt request (rising edge/ falling edge/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[4:0], 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.

Figure 39.3 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 voltage, the COMPMDR.CiMON bit is set to 1.
- If the analog input voltage is lower than the reference voltage, the CiMON bit is set to 0.

ACMPLPi outputs an interrupt to the ICU. For details on the interrupt, see section 39.5, ACMPLP Interrupts. ACMPLPi also outputs an event signal to the ELC to activate other modules. For details on the ELC, see section 39.6, ELC Event Output. Do not change the values of the registers during the comparison.

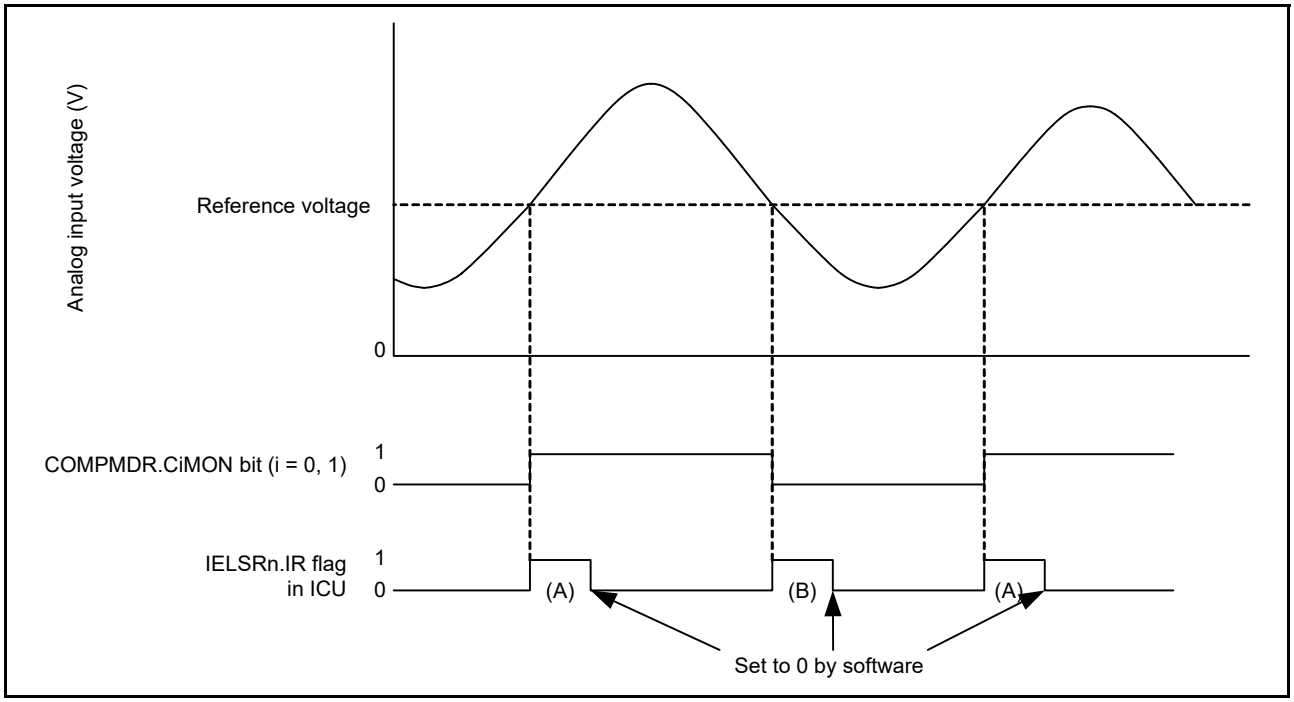


Figure 39.3 Operating example of ACMPLPi (i = 0, 1) when window function is disabled

Figure 39.3 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), ICU.IELSRn.IR changes as shown by (A) only
- When CiEDG = 0 and CiEPO = 1 (falling edge), ICU.IELSRn.IR changes as shown by (B) only.

Figure 39.4 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 ICU. For details on the interrupt, see section 39.5, ACMPLP Interrupts. ACMPLPi also outputs an event signals to the ELC to activate other modules. For details on the ELC, see section 39.6, ELC Event Output. Do not change the values of the registers during comparison.

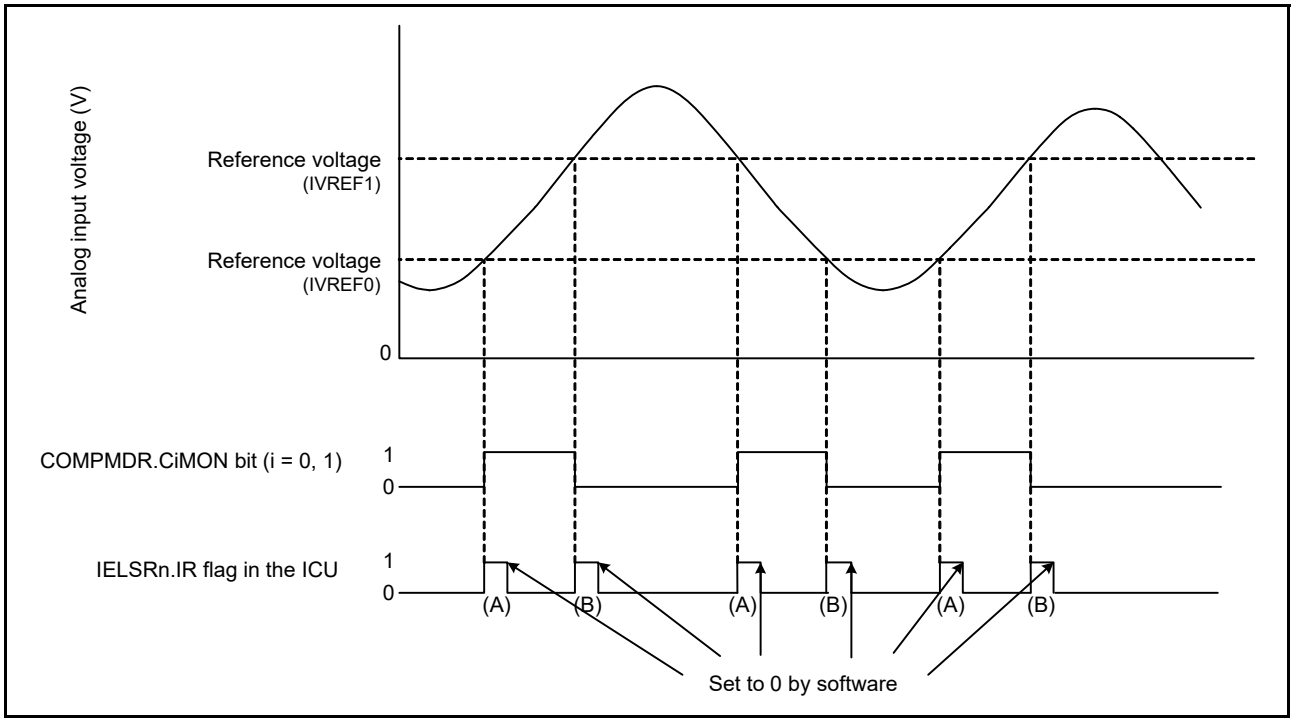


Figure 39.4 Operating example of ACMPLPi (i = 0, 1) when window function is enabled

Figure 39.4 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), ICU.IELSRn.IR changes as shown by (A) only.
- When CiEDG = 0 and CiEPO = 1 (falling edge), ICU.IELSRn.IR changes as shown by (B) only.

39.4 Noise Filter

Figure 39.5 shows the configuration of the ACMPLPi noise filter and edge detection configuration, and Figure 39.6 shows an operating example of the ACMPLPi noise filter.

The sampling clock can be selected in the COMPFIR.CiFCK[1:0] bits. The signal (internal signal) output from ACMPLPi is sampled at every sampling clock cycle. When the level matches three times, the corresponding IELSRn.IR bit is set to 1 (interrupt requested) and an ELC event is output.

When using an interrupt and ELC in Software Standby mode, set the COMPFIR.CiFCK[1:0] bits to 00b (bypass).

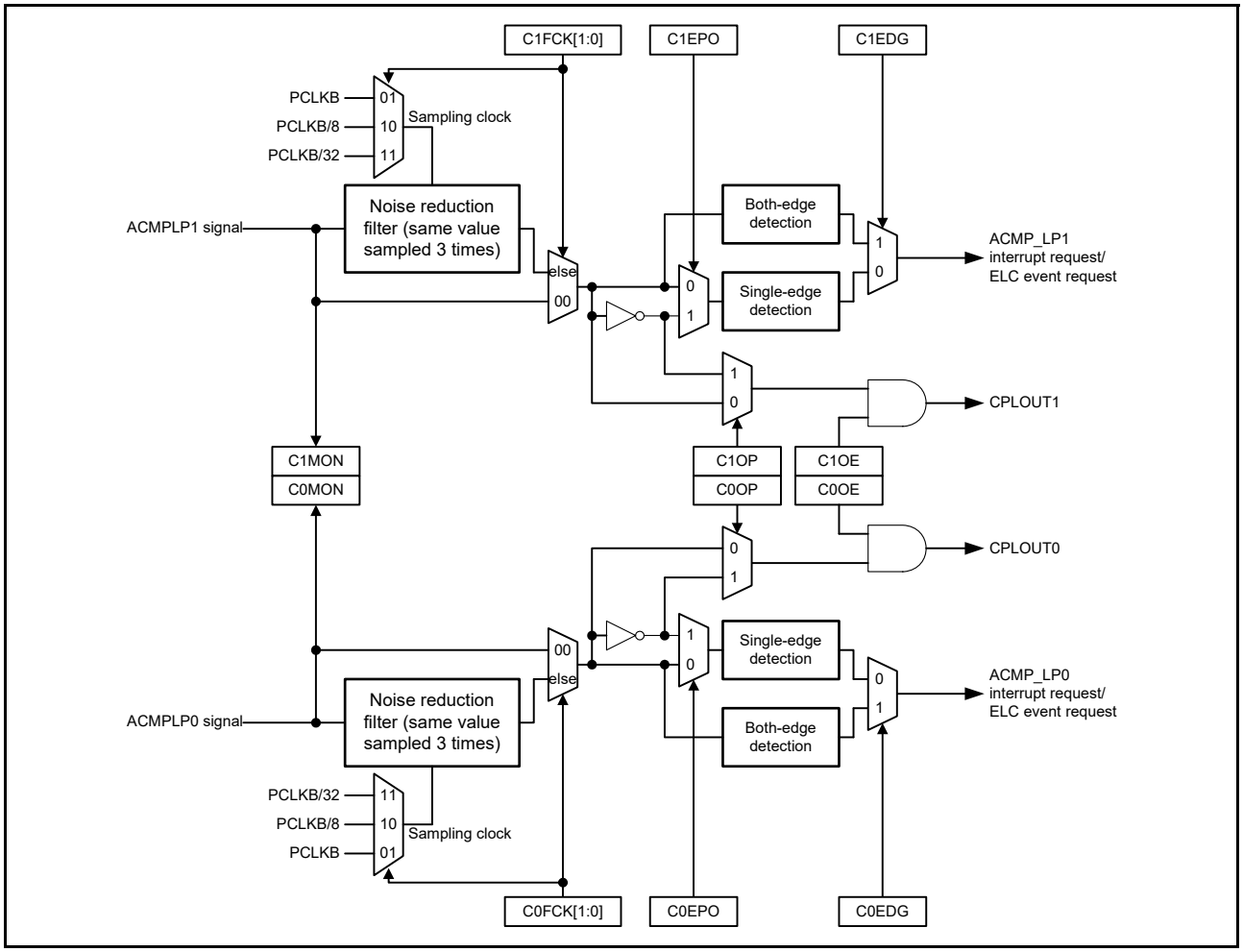


Figure 39.5 Noise filter and edge detection configuration

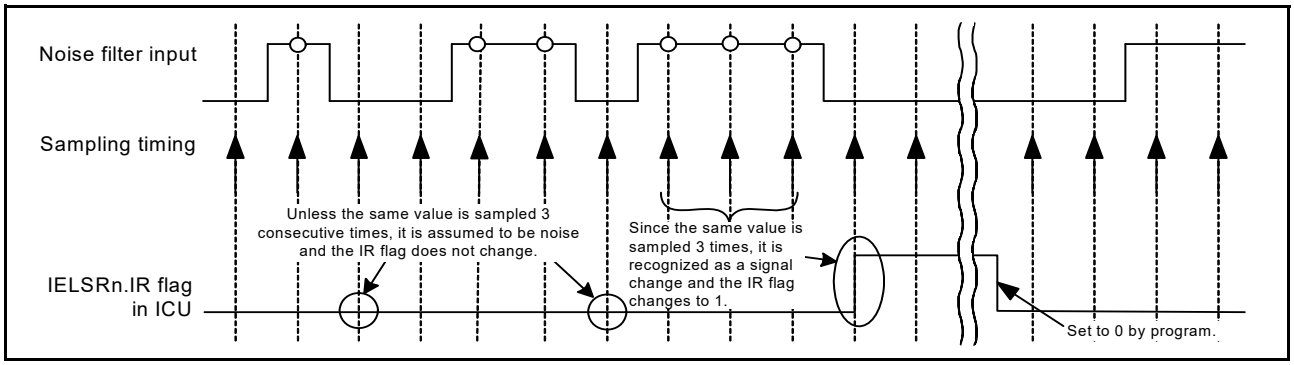


Figure 39.6 Noise filter and interrupt operation example

39.5 ACMPLP Interrupts

The ACMPLP generates two interrupt requests from ACMPLP0 and ACMPLP1 sources. The ACMPLPi (i = 0 and 1) interrupt can be used by selecting it in the IELSRn register in the ICU.

To use the ACMPLPi interrupt, select either single-edge detection or both-edge detection using the COMPFIR.CiEDG bit. When single-edge detection is selected, select the polarity using the CiEPO bit. The interrupt output can also be passed through the noise filter with four different sampling clocks. 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.CiFCK[1:0] to 00b to bypass the ACMPLP0 noise filter. The ACMPLP1 interrupt request cannot be used to release Software Standby mode or Snooze mode.

39.6 ELC Event Output

The ELC uses the ACMPLP interrupt request signal as an ELC event signal, enabling link operation for the preset module. The ELC event of ACMPLP can be used by selecting it in the ELSRn register in the ELC. When using ELC event request, set the COMPFIR.CiFCK[1:0] bits to 01b, 10b or 11b (otherwise no sampling).

39.7 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.

39.8 Comparator Pin Output

The comparison result from ACMPLPi can be output to external pins. The COMPOCR.CiOP and CiOE bits can be used to set the output polarity (non-inverted output or inverted output) and to enable or disable the comparison output. For the register settings and associated comparator output, see section 39.2.3, ACMPLP Output Control Register (COMPOCR).

To output the ACMPLP comparison result to the VCOUT output pin by the CPLOUTi, set the corresponding Port mn Pin Function Select register (PmnPFS) in the I/O register.

39.9 Usage Notes

39.9.1 Settings for the Module-Stop State

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

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 conductors. Because the human body is an electrical conductor, when a finger is placed close to the electrode, the electrostatic capacitance increases.

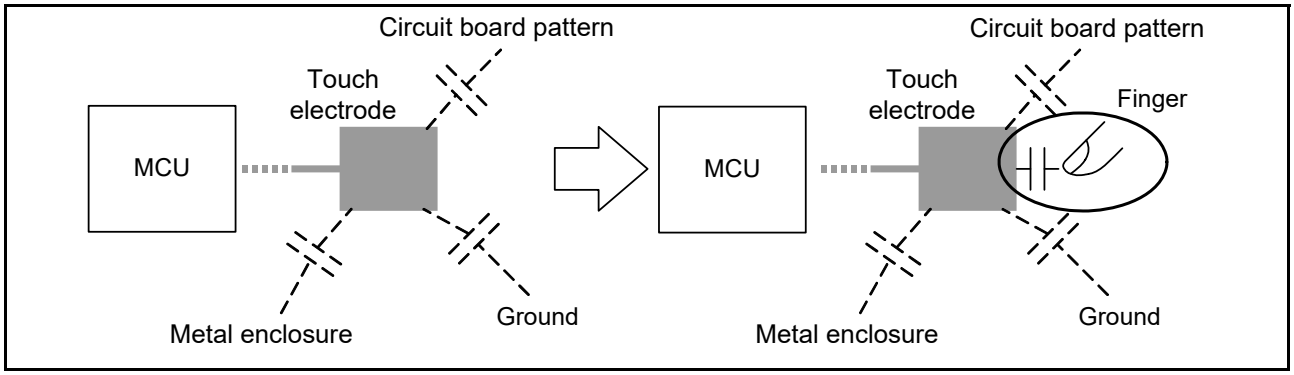


Figure 40.1 Increased electrostatic capacitance because of the presence of a finger

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, one as a transmit electrode, and the other as a receive electrode, and the CTSU detects the change in the electrostatic capacitance generated between the two when a finger is placed close to them.

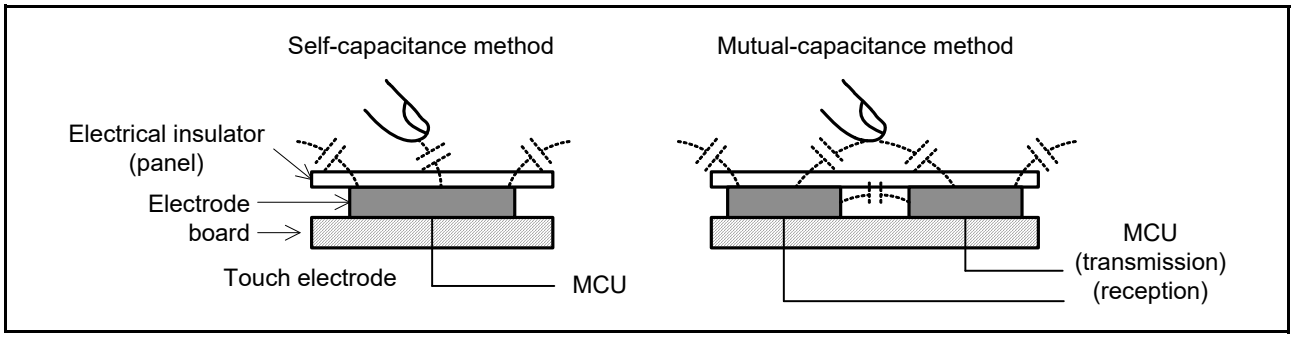


Figure 40.2 Self-capacitance and mutual-capacitance methods

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 (1 of 2)

Parameter		Specifications
Operating clocks		PCLKB, PCLKB/2, or PCLKB/4
Pins	Electrostatic capacitance measurement	26 channels (TS00 to TS25)
	TSCAP	Low Pass Filter (LPF) connection pin

Table 40.1 CTSU specifications (2 of 2)

Parameter		Specifications
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.

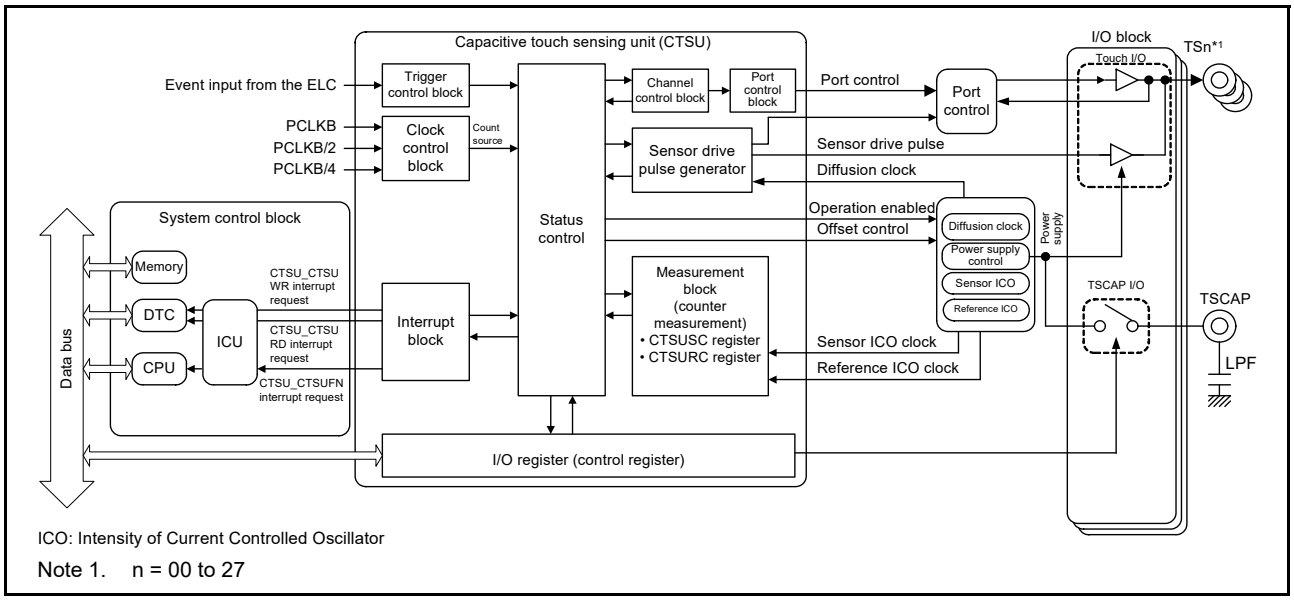


Figure 40.3 CTSU block diagram

Table 40.2 CTSU pin configuration

Pin name	I/O	Function
TS00 to TS25	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	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 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, set 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 that 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, 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 internal 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 the CTSU 1: Power on the CTSU.	R/W
b1	CTSUCSW	CTSU LPF Capacitance Charging Control	This bit controls charging of the LPF capacitance connected to the TSCAP pin: 0:Turn off capacitance switch 1: Turn on capacitance switch.	R/W
b2	CTSUATUNE0	CTSU Power Supply Operating Mode Setting	VCC ≥ 2.4 V 0: Normal operating mode 1: Low-voltage operating mode. VCC < 2.4 V 0: Setting prohibited 1: Low-voltage operating mode.	R/W
b3	CTSUATUNE1	CTSU Power Supply Capacity Adjustment	0: Normal output 1: High-current output.	R/W

Bit	Symbol	Bit name	Description	R/W
b5, b4	CTSUCLK[1:0]	CTSU Operating Clock Select	These bits select the operating clock: b5 b4 0 0: PCLKB 0 1: PCLKB/2 (PCLKB divided by 2) 1 0: PCLKB/4 (PCLKB divided by 4) 1 1: Setting prohibited.	R/W
b7, b6	CTSUMD[1:0]	CTSU Measurement Mode Select	These bits select the measurement mode: b7 b6 0 0: Self-capacitance single scan mode 0 1: Self-capacitance multi-scan mode 1 0: Setting prohibited 1 1: Mutual capacitance full scan mode.	R/W

Only set the CTSUCR1 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUPON bit (CTSU Power Supply Enable)

The CTSUPON bit controls the power supply to the CTSU. Set the CTSUPON and CTSUCSW bits to the same value.

CTSUCSW bit (CTSU LPF Capacitance Charging Control)

The CTSUCSW bit controls charging of the LPF capacitor connected to the TSCAP pin by turning the capacitance switch on or off. After the capacitance switch is turned on, wait until the capacitance connected to the TSCAP pin is charged for the specified time before starting measurement by setting CTSUCR0.CTSUSTRT to 1. Before starting measurement, use an I/O port to output low to the TSCAP pin, and discharge the existing LPF capacitance. Set the CTSUPON and CTSUCSW bits to the same value.

CTSUATUNE0 bit (CTSU Power Supply Operating Mode Setting)

The CTSUATUNE0 bit sets the power supply operating mode. Set this bit according to the lower limit of VCC to operate the CTSU. For example, when using touch measurement in a system where VCC varies depending on the 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. Normally, 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[1:0] 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 are used to determine the measurement time and the measurement pulse count. Recommended setting is 3 (0011b).	R/W

Bit	Symbol	Bit name	Description	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) 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 determine the measurement time and the measurement pulse count using the following formulas, where the base pulse count 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 40.2.19, 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 the 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 the 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 0: TS02 0 0 0 0 1 1: TS03 0 0 0 1 0 0: TS04 0 0 0 1 0 1: TS05 0 0 0 1 1 0: TS06 0 0 0 1 1 1: TS07 0 0 1 0 0 0: TS08 0 0 1 0 0 1: TS09 0 0 1 0 1 0: TS10 0 0 1 0 1 1: TS11 0 0 1 1 0 0: TS12 0 0 1 1 0 1: TS13 0 0 1 1 1 0: TS14 0 0 1 1 1 1: TS15 0 1 0 0 0 0: TS16 0 1 0 0 0 1: TS17 0 1 0 0 1 0: TS18 0 1 0 0 1 1: TS19 0 1 0 1 0 0: TS20 0 1 0 1 0 1: TS21 0 1 0 1 1 0: TS22 0 1 0 1 1 1: TS23 0 1 1 0 0 0: TS24 0 1 1 0 0 1: TS25 Other than the above settings, starting measurement by setting CTSUCR0.CTSUSTRT to 1 is prohibited after these bits are set.	R/W*1

Bit	Symbol	Bit name	Description	R/W
			In other measurement modes, these bits indicate the channel that is currently being measured: b5 b0 0 0 0 0 0 0: TS00 0 0 0 0 0 1: TS01 0 0 0 0 1 0: TS02 0 0 0 0 1 1: TS03 0 0 0 1 0 0: TS04 0 0 0 1 0 1: TS05 0 0 0 1 1 0: TS06 0 0 0 1 1 1: TS07 0 0 1 0 0 0: TS08 0 0 1 0 0 1: TS09 0 0 1 0 1 0: TS10 0 0 1 0 1 1: TS11 0 0 1 1 0 0: TS12 0 0 1 1 0 1: TS13 0 0 1 1 1 0: TS14 0 0 1 1 1 1: TS15 0 1 0 0 0 0: TS16 0 1 0 0 0 1: TS17 0 1 0 0 1 0: TS18 0 1 0 0 1 1: TS19 0 1 0 1 0 0: TS20 0 1 0 1 0 1: TS21 0 1 0 1 1 0: TS22 0 1 0 1 1 1: TS23 0 1 1 0 0 0: TS24 0 1 1 0 0 1: TS25 1 1 1 1 1 1: Measurement 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 only enabled in self-capacitance single scan mode (CTSUCR1.CTSUMD[1:0] = 00b).

Only set the CTSUMCH0 register when 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 the enabled channels (000000b to 011011b). In all other modes, these bits 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 0: TS02 0 0 0 0 1 1: TS03 0 0 0 1 0 0: TS04 0 0 0 1 0 1: TS05 0 0 0 1 1 0: TS06 0 0 0 1 1 1: TS07 0 0 1 0 0 0: TS08 0 0 1 0 0 1: TS09 0 0 1 0 1 0: TS10 0 0 1 0 1 1: TS11 0 0 1 1 0 0: TS12 0 0 1 1 0 1: TS13 0 0 1 1 1 0: TS14 0 0 1 1 1 1: TS15 0 1 0 0 0 0: TS16 0 1 0 0 0 1: TS17 0 1 0 0 1 0: TS18 0 1 0 0 1 1: TS19 0 1 0 1 0 0: TS20 0 1 0 1 0 1: TS21 0 1 0 1 1 0: TS22 0 1 0 1 1 1: TS23 0 1 1 0 0 0: TS24 0 1 1 0 0 1: TS25 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]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHAC0[7:0]	CTSU Channel Enable Control 0	These bits select whether the associated TSn pin is measured: 0: Do not measure 1: Measure. These bits specify the TS00 to TS07 pins.	R/W

Only set the CTSUCHAC0 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC0[7:0] bits (CTSU Channel Enable Control 0)

The CTSUCHAC0[7:0] bits select the receive and transmit pins whose electrostatic capacitance is to be measured. CTSUCHAC0[0] is associated with TS00, and CTSUCHAC0[7] with TS07.

40.2.8 CTSU Channel Enable Control Register 1 (CTSUCHAC1)

Address(es): CTSU.CTSUCHAC1 4008 1007h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHAC1[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHAC1[7:0]	CTSU Channel Enable Control 1	These bits select whether the associated TSn pin is measured: 0: Do not measure 1: Measure. These bits specify the TS08 to TS15 pins.	R/W

Only set the CTSUCHAC1 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC1[7:0] bits (CTSU Channel Enable Control 1)

The CTSUCHAC1[7:0] bits select the receive and transmit pins whose electrostatic capacitance is to be measured. CTSUCHAC1[0] is associated with TS08, and CTSUCHAC1[7] with TS15.

40.2.9 CTSU Channel Enable Control Register 2 (CTSUCHAC2)

Address(es): CTSU.CTSUCHAC2 4008 1008h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHAC2[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHAC2[7:0]	CTSU Channel Enable Control 2	These bits select whether the associated TSn pin is measured: 0: Do not measure 1: Measure. These bits specify the TS16 to TS23 pins.	R/W

Only set the CTSUCHAC2 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC2[7:0] bits (CTSU Channel Enable Control 2)

The CTSUCHAC2[7:0] bits select the receive and transmit pins whose electrostatic capacitance is to be measured. CTSUCHAC2[0] is associated with TS16 and CTSUCHAC2[7] with TS23.

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 TSn pin is measured: 0: Do not measure 1: Measure. These bits specify pins TS24 to TS25.	R/W

Note 1. The MCU does not support TS26 to TS31 pins. Therefore, CTSUCHAC3[2] to CTSUCHAC3[7] 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 whose electrostatic capacitance is to be measured. CTSUCHAC3[0] is associated with TS24, and CTSUCHAC3[1] with TS25.

40.2.11 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]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHTRC0[7:0]	CTSU Channel Transmit/Receive Control 0	0: Reception 1: Transmission. These bits specify the TS00 to TS07 pins.	R/W

Only set the CTSUCHTRC0 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC0[7:0] bits (CTSU Channel Transmit/Receive Control 0)

In full scan mode, the CTSUCHTRC0[7:0] bits allocate reception or transmission to the associated TSn pins. The setting is ignored in self-capacitance single scan and multi-scan modes. CTSUCHTRC0[0] is associated with TS00, and CTSUCHTRC0[7] with TS07.

40.2.12 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]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHTRC1[7:0]	CTSU Channel Transmit/Receive Control 1	0: Reception 1: Transmission. These bits specify the TS08 to TS15 pins.	R/W

Only set the CTSUCHTRC1 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC1[7:0] bits (CTSU Channel Transmit/Receive Control 1)

In full scan mode, the CTSUCHTRC1[7:0] bits allocate reception or transmission to the associated TSn pins. The setting is ignored in self-capacitance single scan and multi-scan modes. CTSUCHTRC1[0] is associated with TS08, and CTSUCHTRC1[7] with TS15.

40.2.13 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]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHTRC2[7:0]	CTSU Channel Transmit/Receive Control 2	0: Reception 1: Transmission. These bits specify the TS16 to TS23 pins.	R/W

Only set the CTSUCHTRC2 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC2[7:0] bits (CTSU Channel Transmit/Receive Control 2)

In full scan mode, the CTSUCHTRC2[7:0] bits allocate reception or transmission to the associated TSn pins. The setting is ignored in self-capacitance single scan and multi-scan modes. CTSUCHTRC2[0] is associated with TS16, and CTSUCHTRC2[7] with TS23.

40.2.14 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 TS24 to TS25 pins.	R/W

Note 1. The MCU does not support the TS26 to TS31 pins. Therefore, CTSUCHTRC3[2] to CTSUCHTRC3[7] 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 TSn pins. The setting is ignored in self-capacitance single scan and multi-scan modes. CTSUCHTRC3[0] is associated with TS24, and CTSUCHTRC3[1] with TS25.

40.2.15 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	These bits should be set 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	These bits should be set 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 set 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 set these bits to 11b. If these bits are not set, touch measurement might be performed incorrectly.

40.2.16 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 counters 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 to 0 when the reference counter is read by software or the DTC. This flag can also be cleared using the CTSUCR0.CTSUINIT bit.

CTSUSOVF flag (CTSU Sensor Counter Overflow Flag)

The CTSUSOVF flag is set to 1 when the sensor counter, CTSUSC, overflows. On overflow, the counter value reads as FFFFh. Measurement processing continues for the specified period.

No interrupt occurs on an overflow. To determine the channel on which the overflow occurred, read the measurement result of each channel after measurement completes, 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 using the CTSUCR0.CTSUINIT bit.

CTSUROVF flag (CTSU Reference Counter Overflow Flag)

The CTSUROVF flag is set to 1 when the reference counter, CTSURC, overflows. On overflow, the counter value reads as FFFFh. Measurement processing continues for the specified period.

No interrupt occurs on an overflow. To determine the channel on which the overflow occurred, read the measurement result of each channel after measurement completes, 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 using the CTSUCR0.CTSUINIT bit.

CTSUPS flag (CTSU Mutual Capacitance Status Flag)

In mutual-capacitance full scan mode, when CTSUCR1.CTSUMD[1:0] bits are 11b, the CTSUPS flag indicates whether the measurement is the first or second of two measurements for each channel. When measurement is stopped, or when in other measurement modes, this flag is always 0.

40.2.17 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 division derived from the base clock frequency division setting. To calculate the correct setting for CTSUSSDIV[3:0], see the relationship between the 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

40.2.18 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 TSn 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 TSn pin that is to be measured next after a CTSU_CTSUWR interrupt is generated.

40.2.19 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 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 = 1) in mutual-capacitance full scan mode (CTSUCR1.CTSUMD[1:0] = 11b).

After a CTSU_CTSUWR interrupt is generated, write first to the CTSUSSC register, next to the CTSUSO0 register, and then to the CTSUSO1 register. The write to the CTSUSO1 register causes a transition to Status 3. Set all of 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 the 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 settings 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. Normally, 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.

40.2.20 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. They read FFFFh when an overflow occurs.	R

After a CTSU_CTSURD interrupt occurs, read first from the CTSUSC counter, then from the CTSURC counter.

CTSUSC[15:0] bits (CTSU Sensor Counter)

The CTSUSC[15:0] bits are configured as an increment counter for the sensor ICO.

Read these bits after a CTSU_CTSURD interrupt is generated. After the CTSURC counter is read, these bits are immediately cleared before the CTSU measurement status counter value changes to Status 4 (the CTSUST.CTSUSTC[2:0] flags change to 100b) in the next measurement. They can also be cleared using the CTSUCR0.CTSUINIT bit.

40.2.21 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. They 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 the 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 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 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 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 using the CTSUCR0.CTSUINIT bit.

40.2.22 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	—	—	—	—	—	—	—	CTSUTSOC	CTSUCLKSEL1	—	—	CTSUDRV	CTSUTSOD	CTSUSPMD[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	CTSUSPMD[1:0]	Calibration Mode	Calibration Mode b1 b0 0 0: Capacitance measurement mode 1 0: Calibration mode Others: Setting prohibited	R/W
b2	CTSUTSOD	TS Pins Fixed Output	TS Pins Fixed Output 0: Capacitance measurement mode 1: Output High or Low from TS terminals	R/W
b3	CTSUDRV	Calibration Setting 1	Calibration Setting 1 0: Capacitance measurement mode 1: Calibration setting 1	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CTSUCLKSEL1	Calibration Setting 3	Calibration Setting 3 0: Capacitance measurement mode 1: Calibration setting 3	R/W
b7	CTSUTSOC	Calibration Setting 2	Calibration Setting 2 0: Capacitance measurement mode 1: Calibration setting 2	R/W
b14 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	CTSUICOMP	TSCAP Voltage Error Monitor	This bit monitors the error status of the TSCAP voltage: 0: Normal TSCAP voltage 1: Abnormal TSCAP voltage.*1	R

Note 1. When CTSUCR1.CTSUPON bit is 0, this bit is set to 1.

CTSUSPMD[1:0] bits (Calibration Mode)

The CTSUSPMD[1:0] bits are used to calibrate the CTSU. When measuring the capacitance, set these bits to 00b.

CTSUTSOD bit (TS Pins Fixed Output)

The CTSUTSOD bit is used to calibrate the CTSU. When setting this bit to 1, the TS pins are forced to the logic level specified by the CTSUCR0.CTSUIOC bit. When measuring the capacitance, set this bit to 0.

CTSUDRV bit (Calibration Setting 1)

The CTSUDRV bit is used to calibrate the CTSU. When measuring capacitance, set this bit to 0.

CTSUCLKSEL1 bit (Calibration Setting 3)

The CTSUCLKSEL1 bit is used to calibrate the CTSU. When measuring capacitance, set this bit to 0.

CTSUTSOC bit (Calibration Setting 2)

The CTSUTSOC bit is used to calibrate the CTSU. When measuring capacitance, set this bit to 0.

CTSUICOMP bit (TSCAP Voltage Error Monitor)

The CTSUICOMP bit monitors the TSCAP voltage and it is set to 1 if the voltage becomes abnormal.

If the offset current specified in the CTSUSO0 register exceeds the sensor ICO input current during touch measurement, the TSCAP voltage becomes abnormal and touch measurement cannot be performed correctly.

If the TSCAP voltage becomes abnormal, the sensor ICO counter value becomes 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 any value other than 0, always 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.

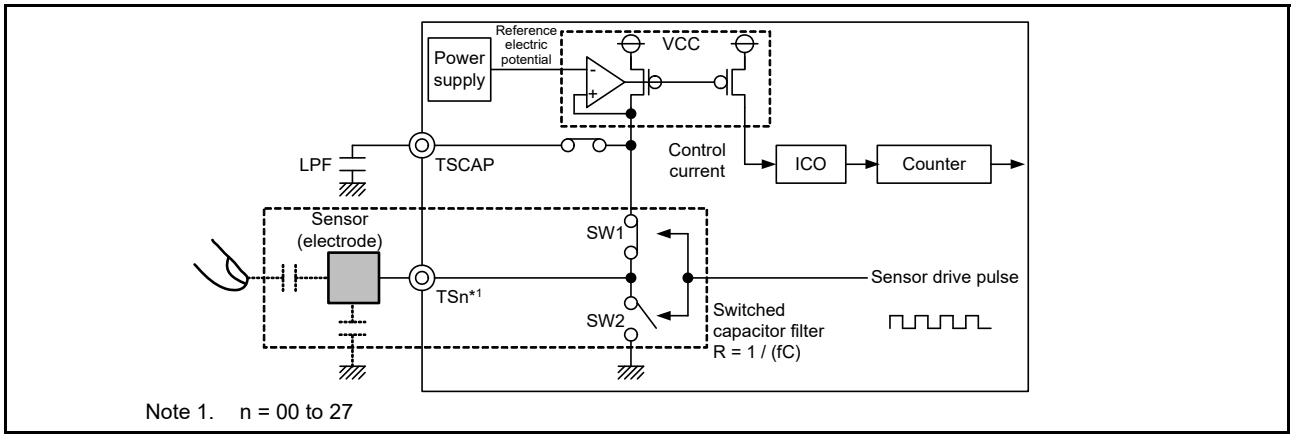


Figure 40.4 Measurement circuit

Figure 40.5 to Figure 40.7 explain the electrostatic capacitance measurement operation principles of the CTSU current frequency conversion. The operation is as follows:

1. The electrostatic capacitance of the electrode is charged by turning SW1 on and SW2 off. See 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 steps 1. and 2. At this point, if a finger is in close proximity, the capacitance and the flowing current change. A clock is generated by supplying a control current, which 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. See Figure 40.7.

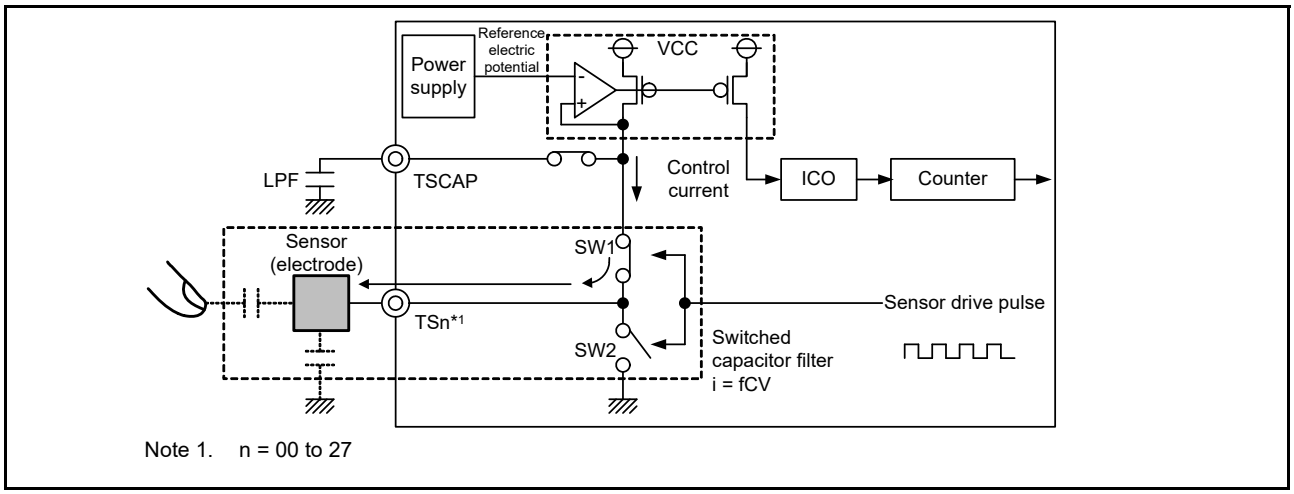


Figure 40.5 Charging operation

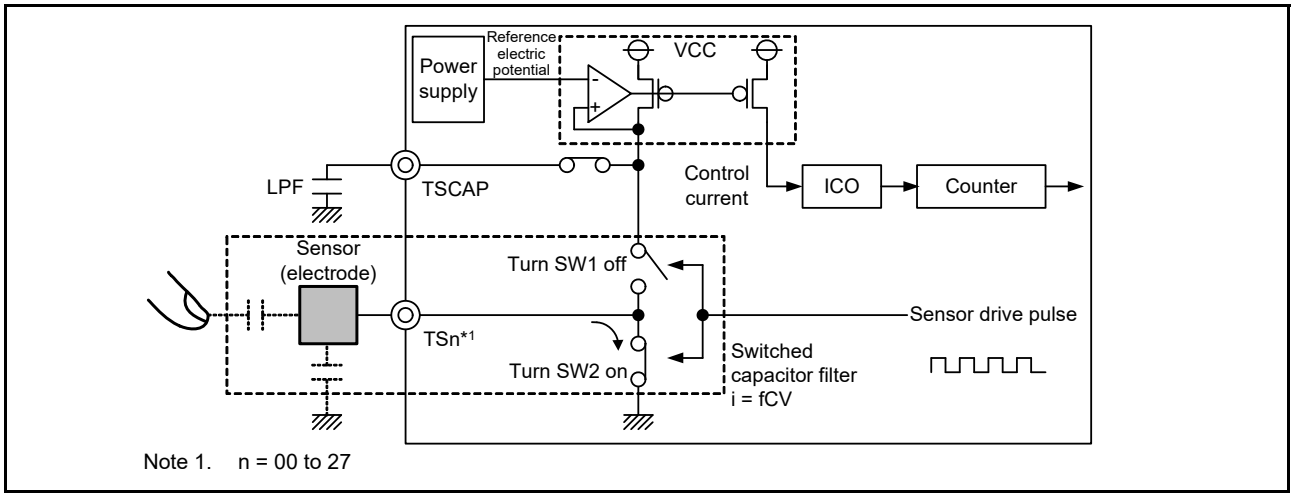


Figure 40.6 Discharging operation

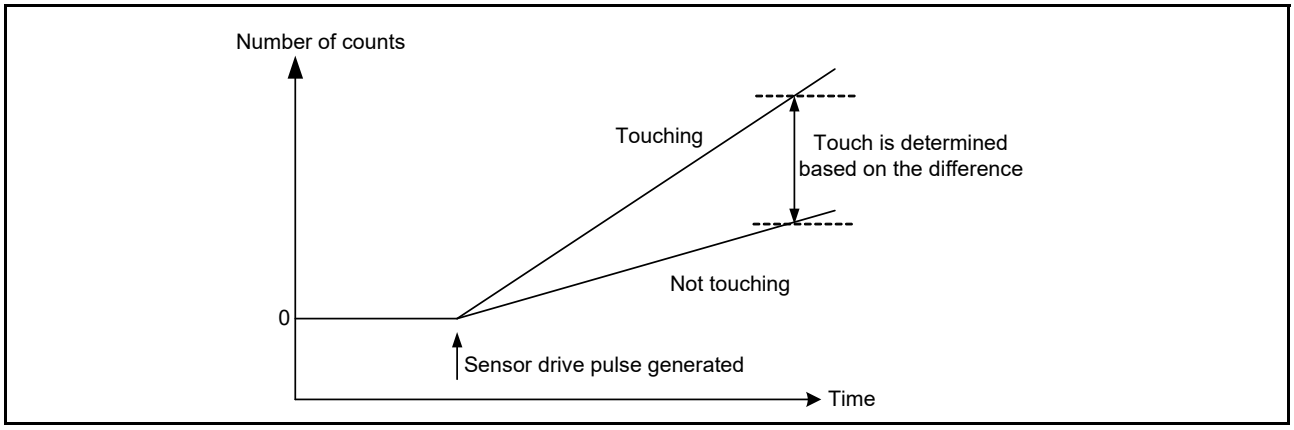


Figure 40.7 Change in measured value when finger is touching and not touching

40.3.2 Measurement Modes

The CTSU supports self-capacitance and mutual-capacitance methods. Figure 40.8 shows these methods.

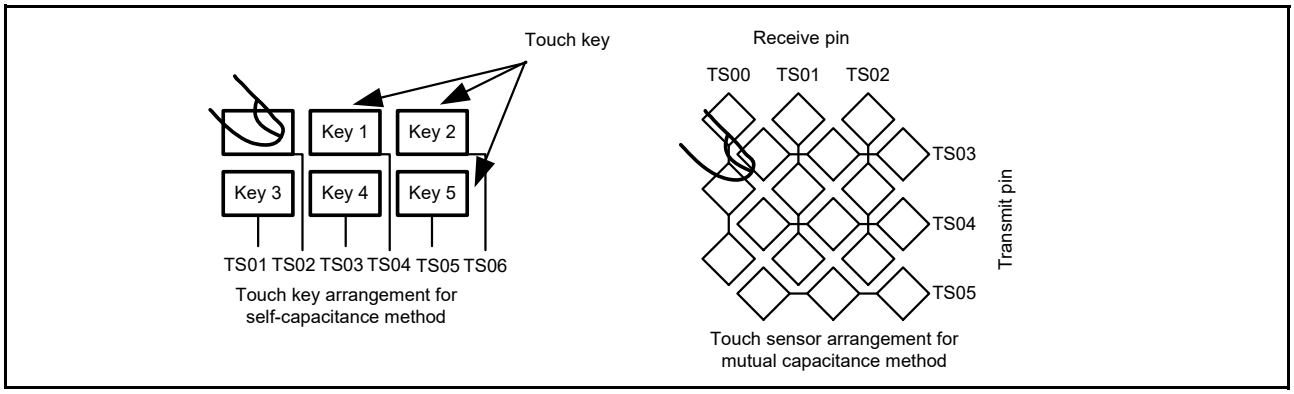


Figure 40.8 Overview of self-capacitance method and mutual-capacitance method

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, capacitance can be measured in both single scan and multi-scan modes. In the mutual-capacitance method, the capacitance between two opposing electrodes (transmit and receive pins) is measured.

40.3.2.1 Initial settings flow

Figure 40.9 shows the flow for the initial CTSU settings.

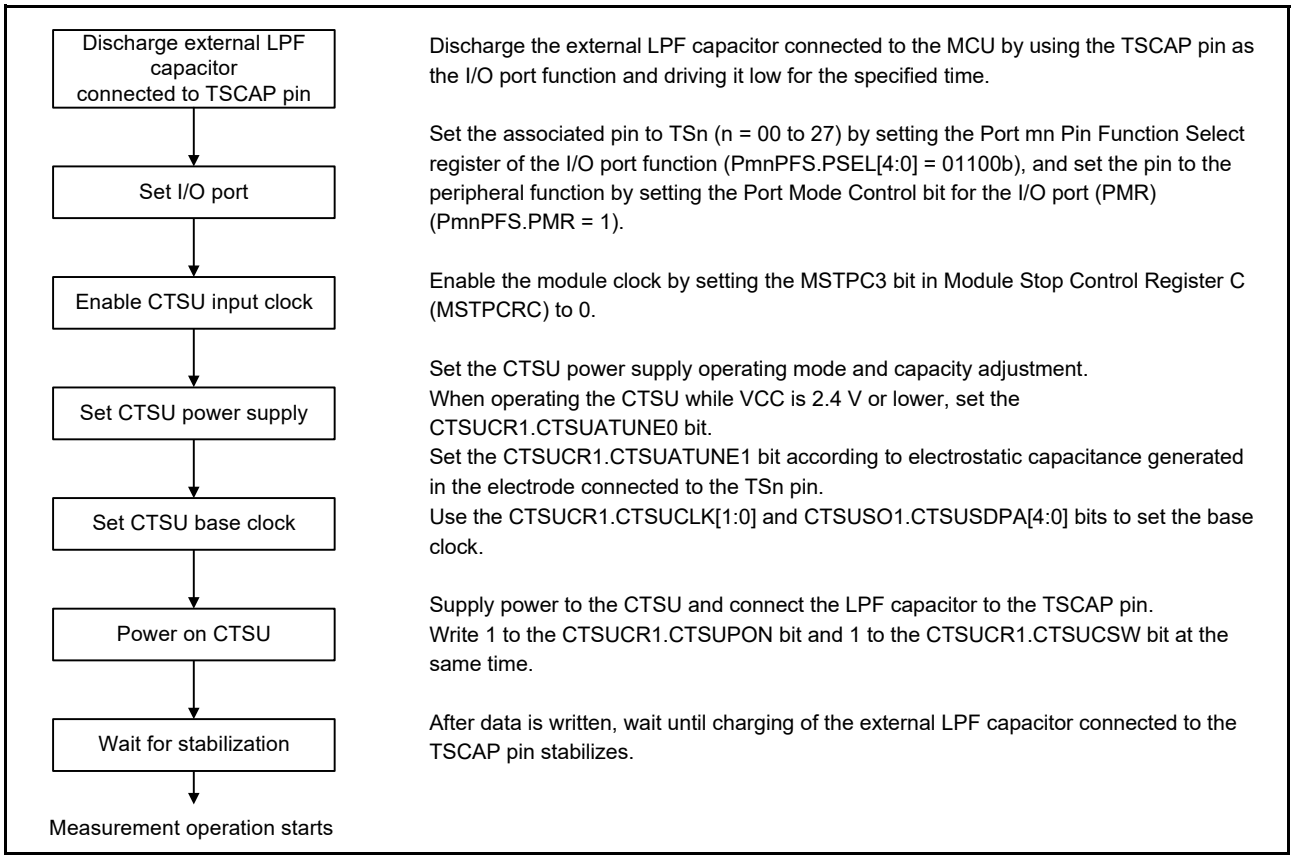


Figure 40.9 CTSU initial setting flow

Figure 40.10 shows the flow for stopping CTSU operation and invoking the standby state.

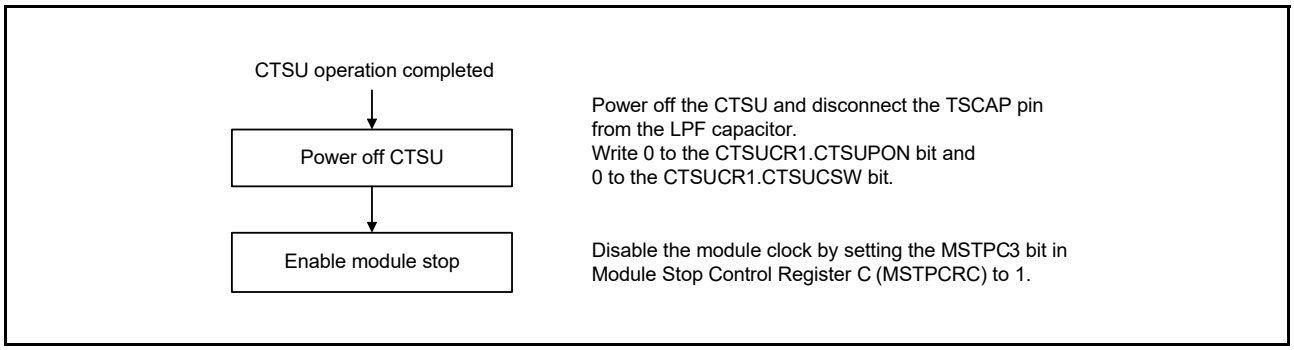
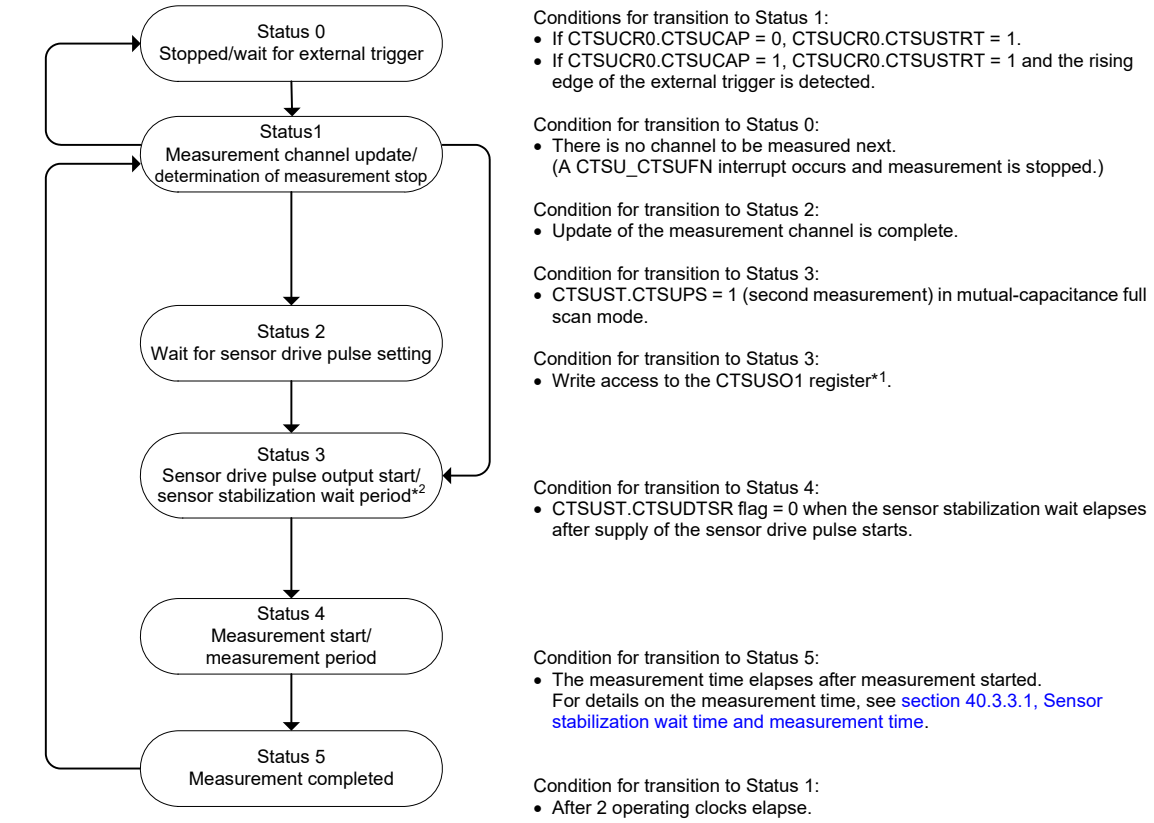


Figure 40.10 CTSU stopping flow

To restart operation, follow the initial setting flow shown in Figure 40.9.

40.3.2.2 Status counter

The CTSU Status Register (CTSUST) indicates the current measurement status, which applies to all three modes. Figure 40.11 shows the status operation transitions.



Note 1. When using the DTC or ICU to set the registers during 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 of 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.

The following are cases when there is no channel to be measured:

- No target measurement channel is specified in the CTSUCHAC0 to CTSUCHAC3 registers
- In self-capacitance single scan mode, the channel specified in the CTSUMCH0 register is not a measurement target in the CTSUCHAC0 to CTSUCHAC3 registers
- In full scan modes, there is no transmit or receive channel to be measured based on the combined settings of the CTSUCHAC0 to CTSUCHAC3, and CTSUCHTRC0 to CTSUCHTRC3 registers.

If there is no channel to be measured based on these settings, a CTSU_CTSUFN interrupt occurs immediately after a transition to Status 1, and the counter status transitions to Status 0.

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.

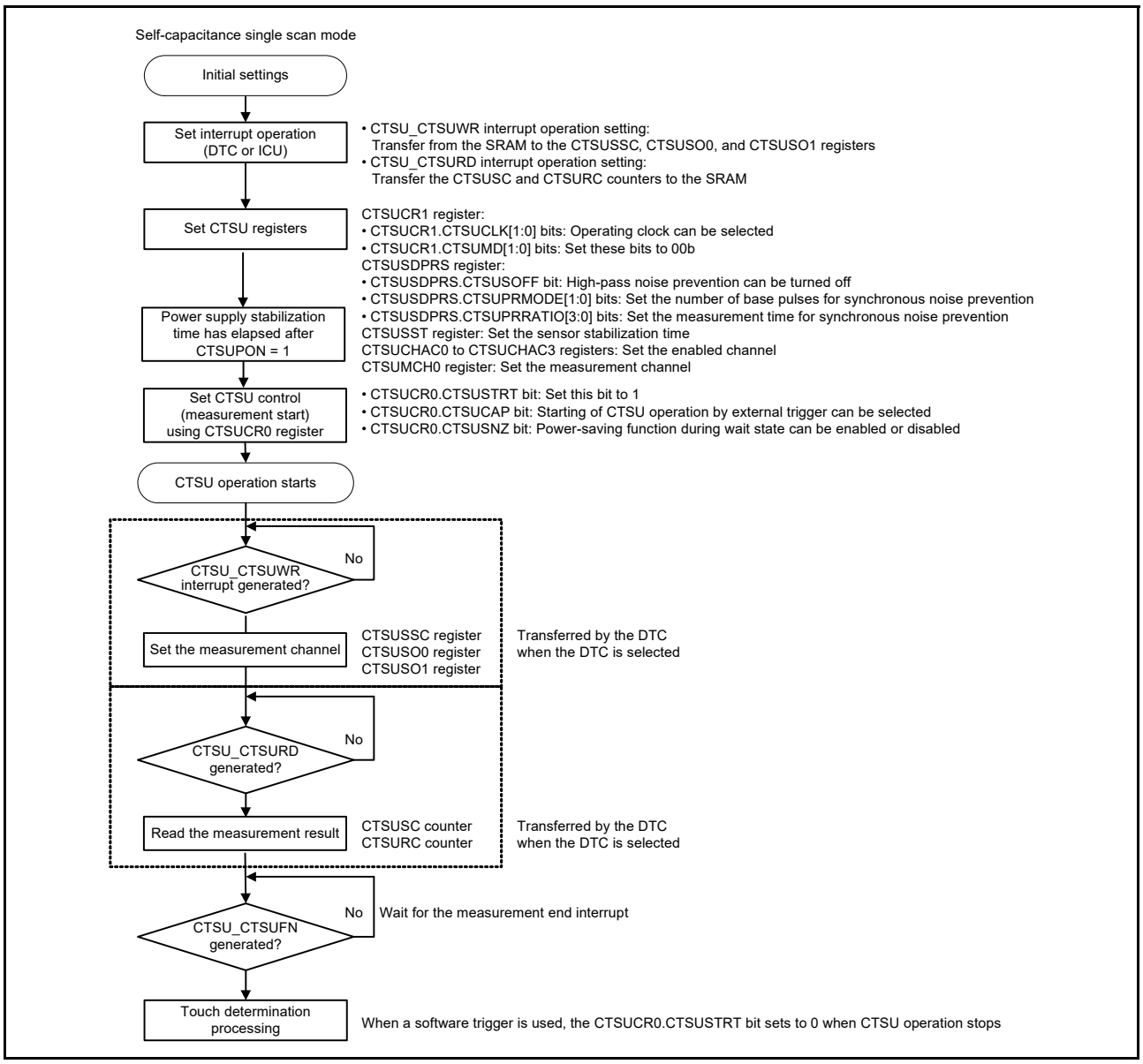


Figure 40.12 Software flow and operation example of self-capacitance single scan mode

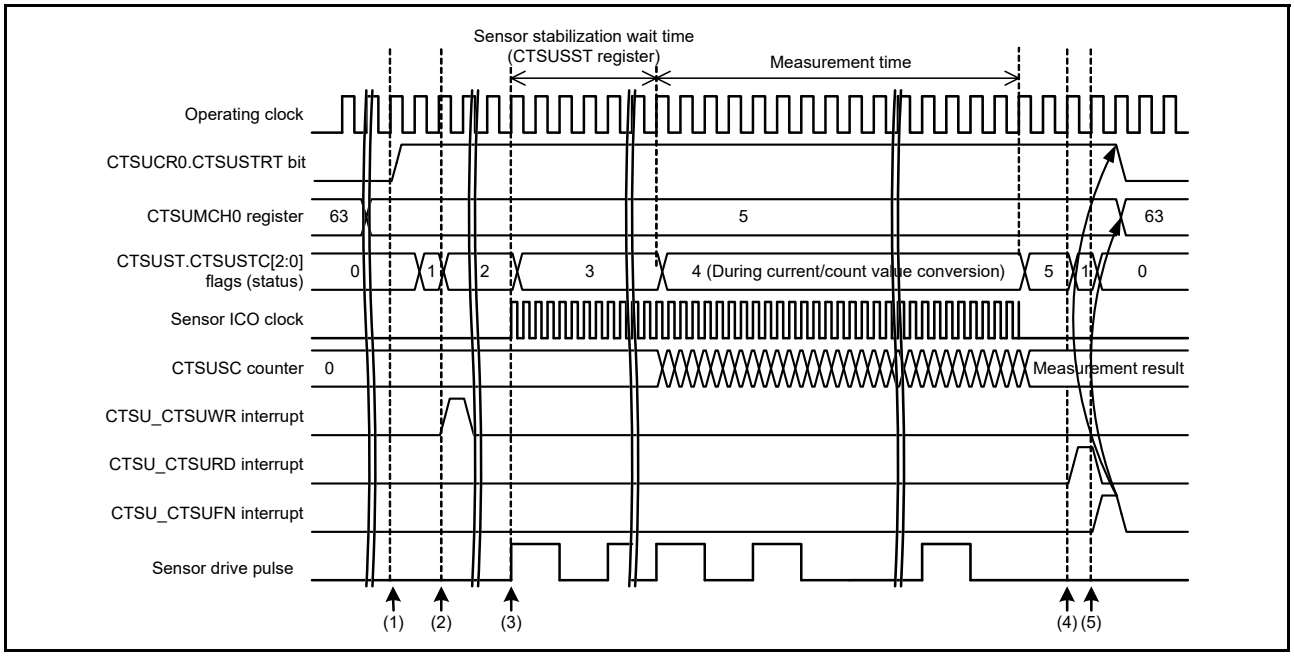


Figure 40.13 Timing of self-capacitance single scan mode when the measurement start condition is a software trigger

The following sequence describes the operation shown in Figure 40.13:

1. After the 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 associated 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 CTSUCHAC3 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.

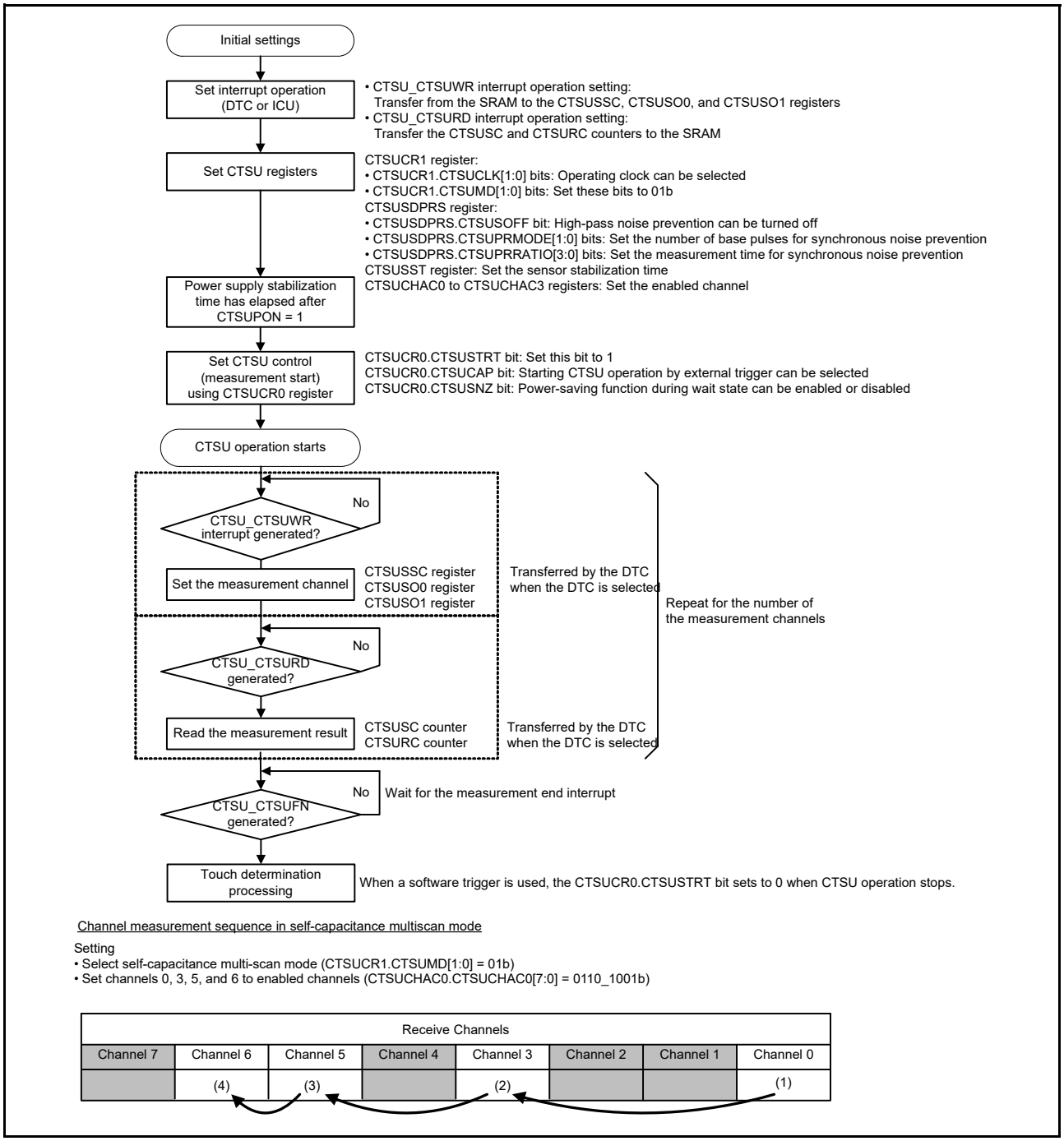


Figure 40.14 Software flow and operation example of self-capacitance multi-scan mode

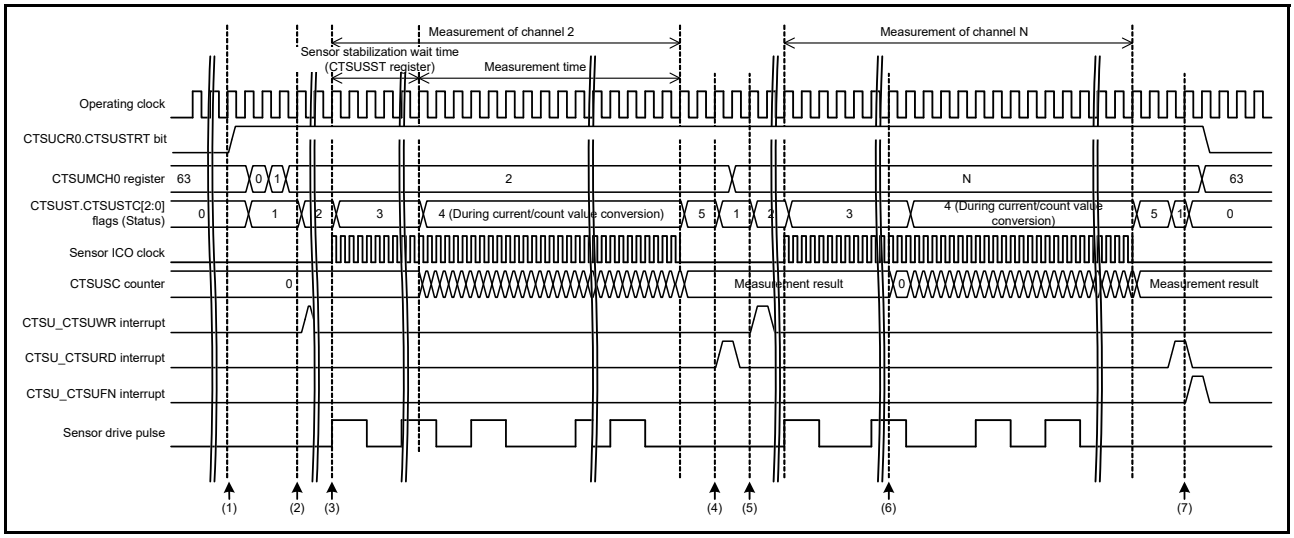


Figure 40.15 Timing of self-capacitance multi-scan mode when the measurement start condition is a software trigger

The following sequence describes the operation shown in Figure 40.15:

1. After the 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 associated 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 associated 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 channel measurements, 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, which results in a higher touch sensitivity.

Electrostatic capacitance is measured sequentially on channels set to transmission or reception in the CTSUCHTRC0 to CTSUCHTRC3 registers, and specified as measurement targets in the CTSUCHAC0 to CTSUCHAC3 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.

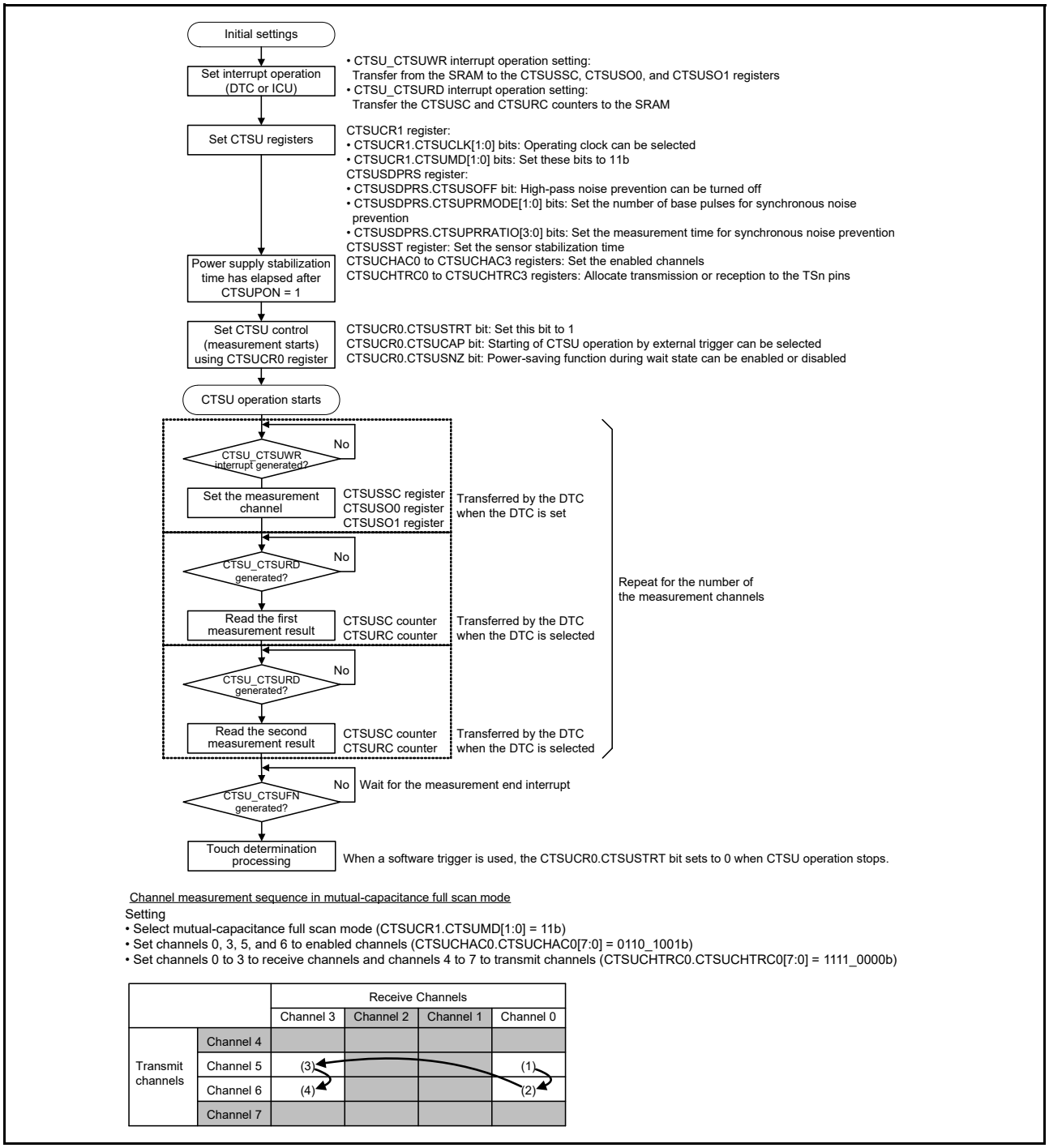


Figure 40.16 Software flow and operation example of mutual-capacitance full scan mode

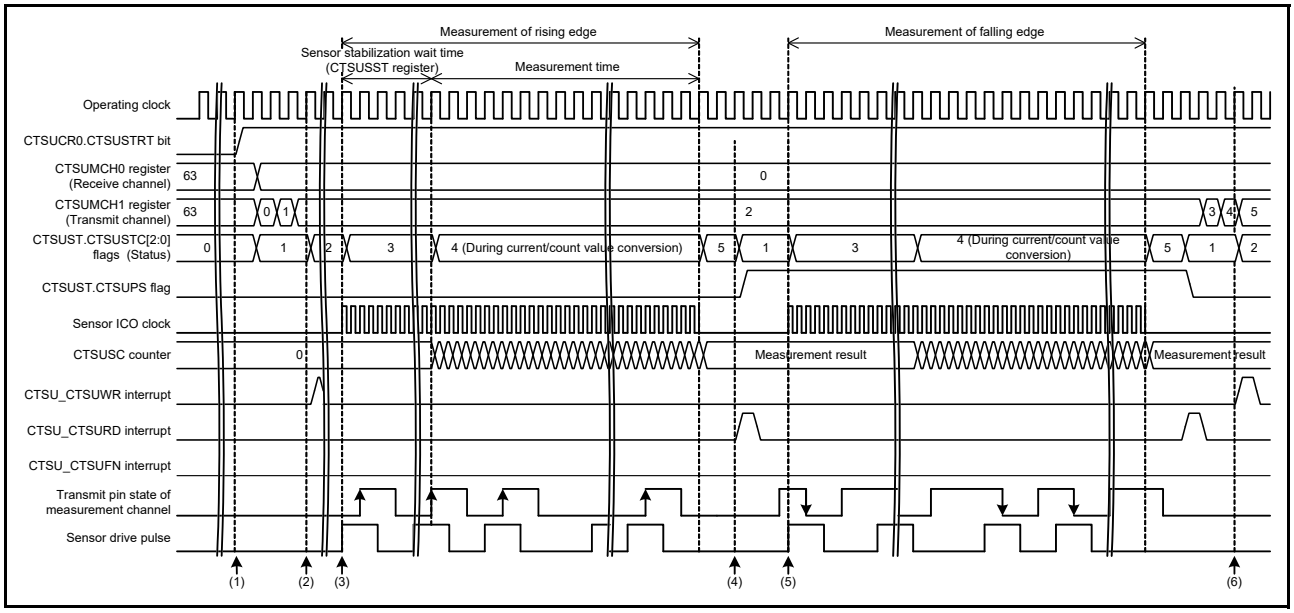


Figure 40.17 Timing of mutual-capacitance full scan mode when the measurement start condition is a software trigger

The following sequence describes the operation shown in Figure 40.17:

1. After the 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 associated 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 rising edge pulse is output to the transmit pin on the measurement channel for 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, measurement is performed by outputting a pulse that becomes a 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 channel measurements, 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)

Status	Touch pins for receive channels		Touch pins for transmit channels		Remarks
	Measured channels	Non-measured channels	Measured channels	Non-measured channel	
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)

Status	Touch pins for receive channels		Touch pins for transmit channels		Remarks
	Measured channels	Non-measured channels	Measured channels	Non-measured channel	
3	Pulse	Low	Pulse	Low	The transmit pulse phase 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 Functions 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.

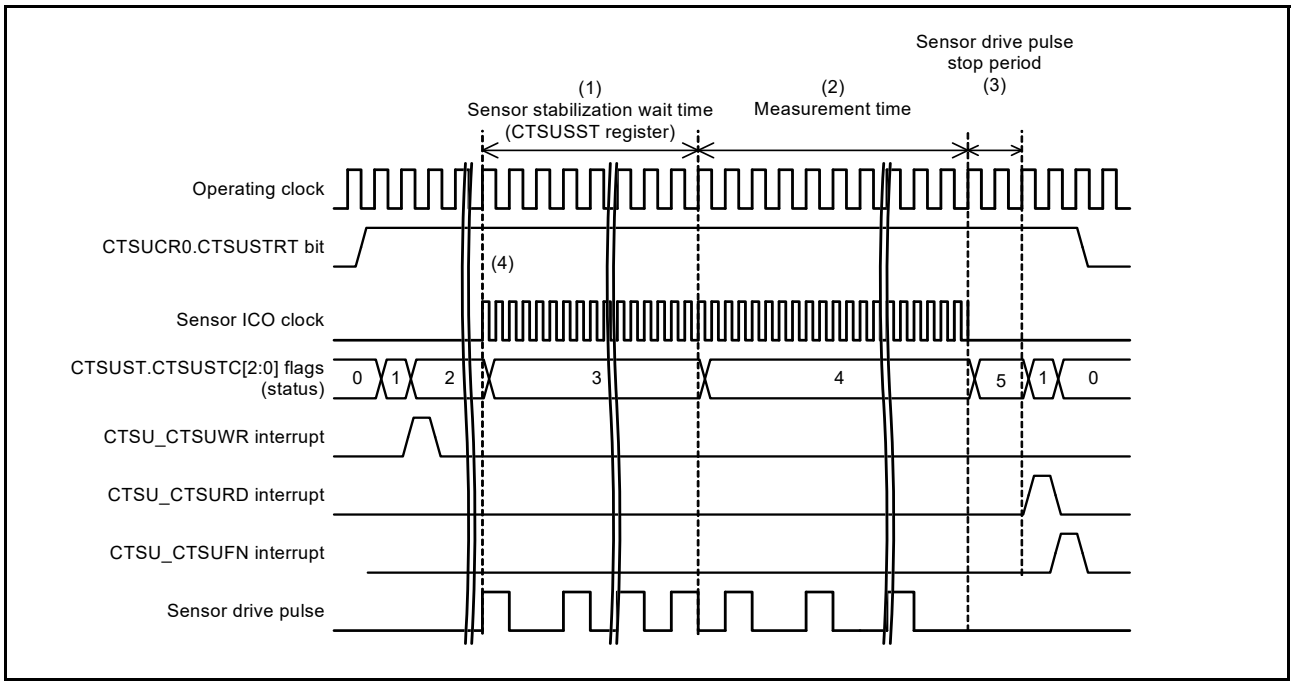


Figure 40.18 Sensor stabilization wait and measurement timing

The following sequence describes the operation shown in Figure 40.18:

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 associated channel stops.
3. After the measurement time elapses, the status transitions to Status 1 after 2 operating clock cycles and a CTSU_CTSURD interrupt occurs. Read the data from the CTSUSC and CTSURC counters. At this time, the sensor drive pulse is output low. When measurement of all specified channels completes, the CTSUCR0.CTSUSTRT bit is set to 0.
4. The sensor ICO clock oscillates while the CTSUST.CTSUSTC[2:0] flags are 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 settings for the selected channel from the SRAM to the 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.

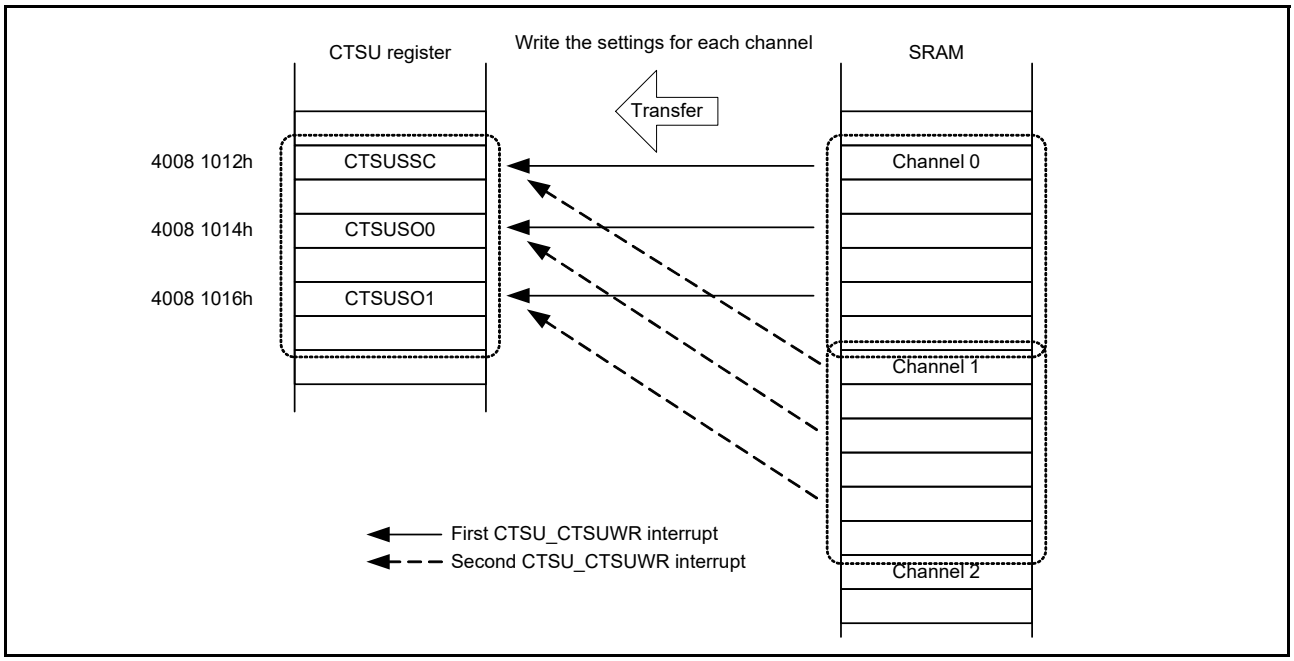


Figure 40.19 Example of DTC transfer operation using the CTSU_CTSUWR interrupt

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 number 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: Transfer three times with one interrupt.

(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 results from the CTSUSC and CTSURC counters (Figure 40.20).

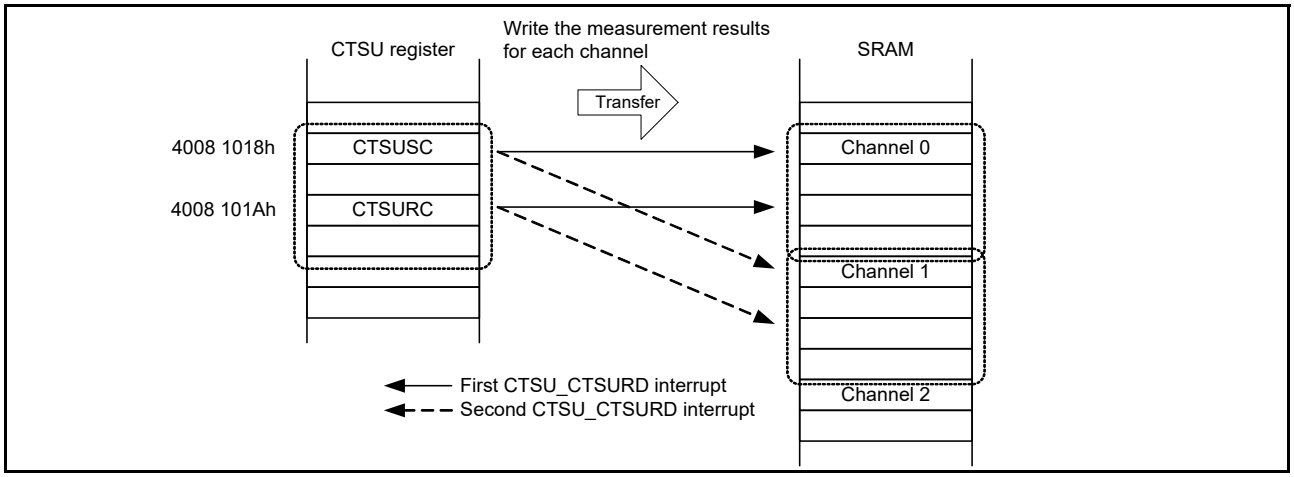


Figure 40.20 Example of DTC transfer operation using the CTSU_CTSURD interrupt

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 continues from the previous interrupt handling.
- Number of transfers: Transfer twice with one interrupt.

(3) Measurement end interrupt (CTSU_CTSUFN)

After all channels are measured, an interrupt occurs when Status 1 transitions to Status 0. In the software, check the overflow flags (CTSUST.CTSUSOVF and CTSUROVF) 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 Constraint on Software Trigger

When 10b (PCLKB/4) is selected in the CTSUCR1.CTSUCLK[1:0] bits, to restart measurement by writing 1 to the CTSUCR0.CTSUSTRT bit after measurement is complete, wait for at least 3 cycles to elapse after an interrupt occurs, then write to the CTSUCR0.CTSUSTRT bit.

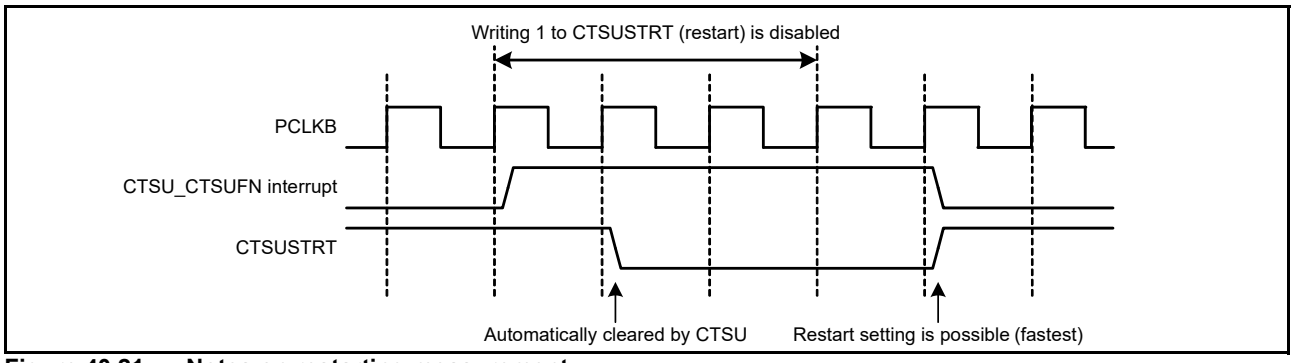


Figure 40.21 Notes on restarting measurement

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 operating clock cycle 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, perform the processing for stopping or 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 the CTSU internal voltage. The traces between the TSCAP pin and the capacitor, and the capacitor and ground should be as short and wide as physically possible.

The capacitor connected to the TSCAP pin should be fully discharged using I/O port control to output low, before turning on the switch (CTSUCR1.CTSUCSW = 1) to establish a connection.

40.4.6 Restrictions on Measurement Operation (CTSUCR0.CTSUSTRT = 1)

During measurement (CTSUCR0.CTSUSTRT = 1), do not use the settings for stopping the peripheral clock or changing the port settings related to the touch pins (TSn and TSCAP) in the higher layers of the system.

If control settings non-compliant with this constraint are made, and operation is forced to stop (CTSUCR0.CTSUSTRT = 0 and CTSUCR0.CTSUINIT = 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	Specifications
Data operation function	16-bit data comparison, addition, and subtraction
Module-stop function	The module-stop state can be set to reduce power consumption
Interrupts and event link function (DOC_DOPCI)	An interrupt occurs on the following conditions: • The compared values either match or mismatch • The result of data addition is greater than FFFFh • The result of data subtraction is less than 0000h.

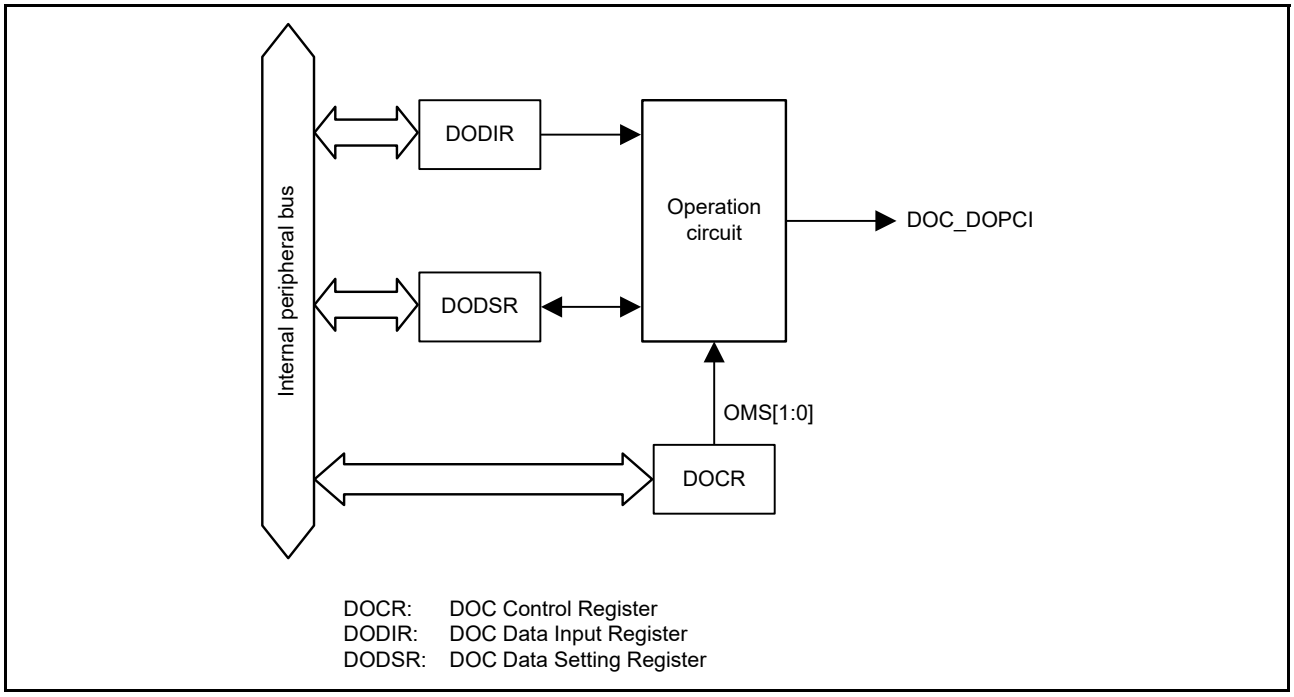


Figure 41.1 DOC block diagram

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	There bits are read as 0. The write value should be 0.	R/W
b5	DOPCF	Data Operation Circuit Flag	Indicates the result of an operation	R
b6	DOPCFCL	DOPCF Clear	0: Save DOPCF flag state 1: Clear DOPCF flag.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Only valid when data comparison mode is selected.

OMS[1:0] bits (Operating Mode Select)

The OMS[1:0] bits select the operating mode of the DOC.

DCSEL bit (Detection Condition Select)

The DCSEL bit selects the detection condition in data comparison mode. This bit is only valid when data comparison mode is selected.

DOPCF flag (Data Operation Circuit Flag)

[Setting conditions]

- When the condition selected in the DCSEL bit is met
- When the result of data addition is greater than FFFFh
- When the result of data subtraction 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 the 16-bit reference data in the DODSR register.
3. Write the 16-bit data for comparison to the DODIR register.
4. Continue writing 16-bit data until all data to be compared is written to the DODIR register.
5. When DOCR.DCSEL = 0, if a value written to DODIR does not match that in DODSR, the DOCR.DOPCF flag is set to 1.

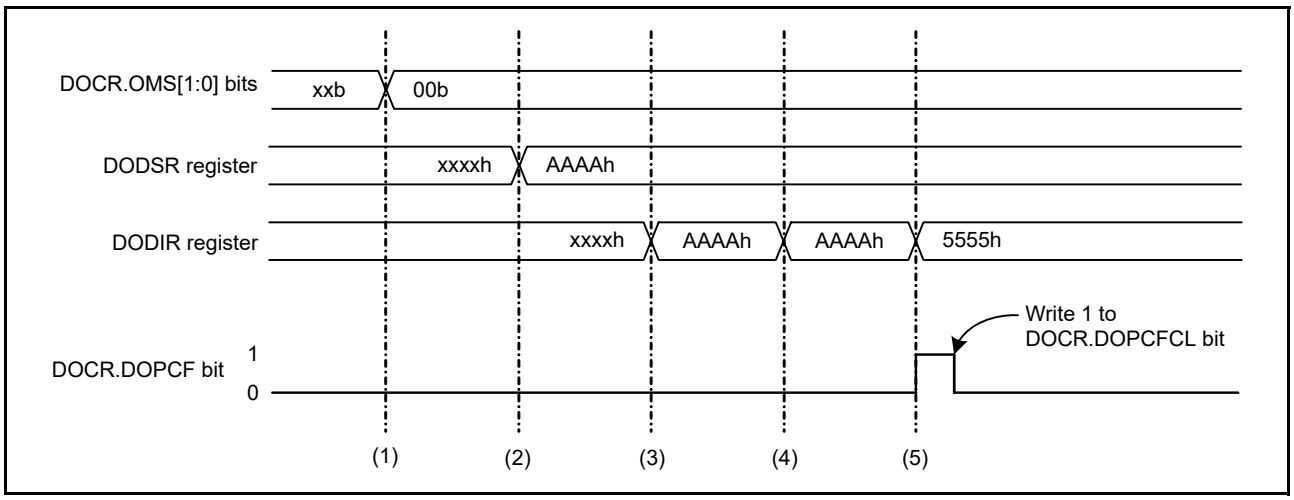


Figure 41.2 Example operation in data comparison mode

41.3.2 Data Addition Mode

Figure 41.3 shows an example DOC operation in data addition mode. The steps are as follows:

1. Write 01b to the DOCR.OMS[1:0] bits to select data addition mode.
2. Set 16-bit data as the initial value in the DODSR register.
3. Write the 16-bit data to be added to the DODIR register. The result of the operation is stored in the DODSR register.
4. Continue writing 16-bit data to the DODIR register until all data to be added is written.
5. If the result of an operation is greater than FFFFh, the DOCR.DOPCF flag is set to 1.

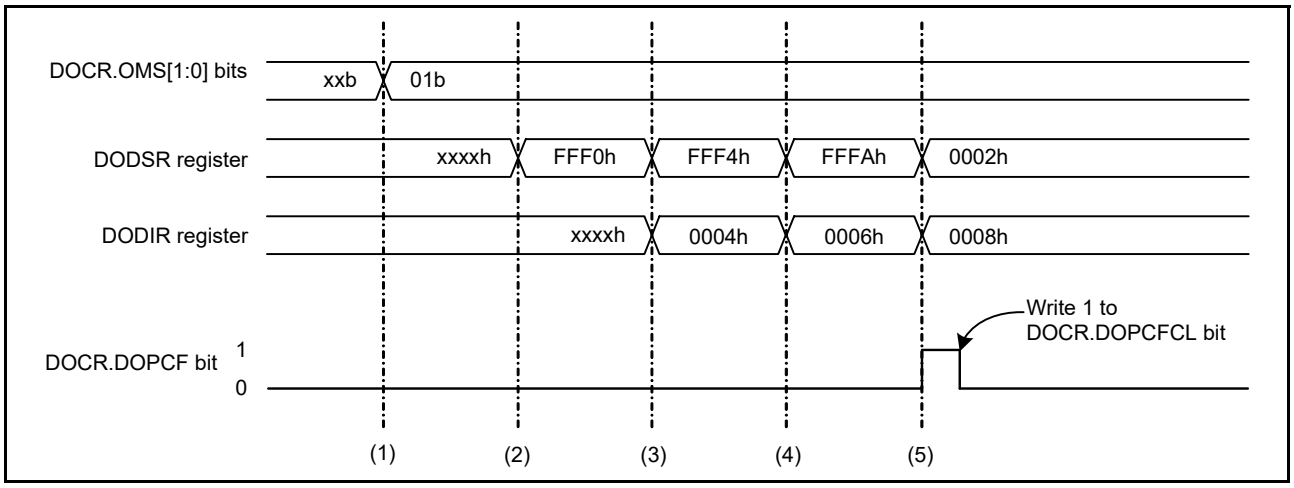


Figure 41.3 Example operation in data addition mode

41.3.3 Data Subtraction Mode

Figure 41.4 shows an example DOC operation in data subtraction mode. The steps are as follows:

1. Write 10b to the DOCR.OMS[1:0] bits to select data subtraction mode.
2. Set 16-bit data as the initial value in the DODSR register.
3. Write the 16-bit data to be subtracted to the DODIR register. The result of the operation is stored in the DODSR register.
4. Continue writing 16-bit data to the DODIR register until all data to be subtracted is written.
5. If the result of an operation is less than 0000h, the DOCR.DOPCF flag is set to 1.

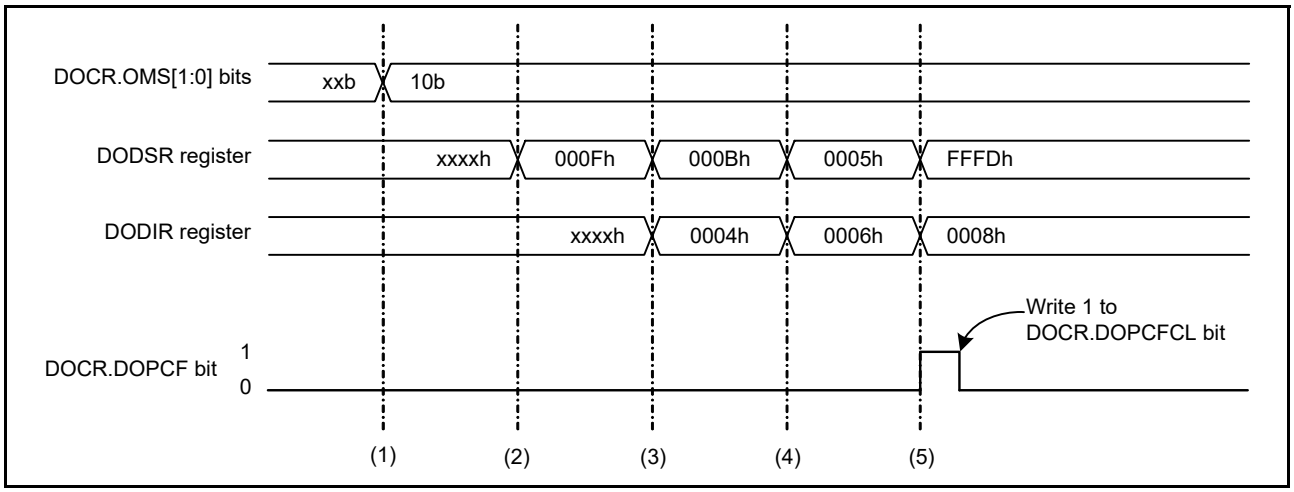


Figure 41.4 Example operation in data subtraction mode

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:

- Compared values either match or mismatch
- Data addition result is greater than FFFFh
- 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 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 subject to ECC. Parity check is performed on the other areas.

Table 42.1 lists the SRAM specifications.

Table 42.1 SRAM specifications

Parameter	Specifications without ECC	Specifications with ECC
SRAM capacity	SRAM0: 16 KB	SRAM0 (ECC area): 16 KB
SRAM address	SRAM0: 2000 4000h to 2000 7FFFh	SRAM0 (ECC area): 2000 0000h to 2000 3FFFh
Access*1	0 wait	
Module-stop function	Not 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: SRAM0 (without ECC) and Trace_RAM are shared. For the Trace_RAM specifications, see the *ARM® CoreSight™ MTB-M23 Technical Reference Manual* (ARM DDI 0564C).

Note 1. For details, see section 42.3.6, 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. Always set the SRAMPRCR bit in SRAMPRCR to enabled 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 the generation of either a reset or a 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 writes to the SRAMPRCR bit	R/W

SRAMPRCR bit (Register Write Control)

The SRAMPRCR bit controls the write mode of the PARIOAD register. Setting the bit to 1 enables writes to the PARIOAD register. When you write to this bit, always write 78h to KW[6:0] 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 SRAMPRCR bit, always write 78h to these 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 (write protection disabled). 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 SRAM0.

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 SRAM0. 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 SRAM0 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 error ECC error in the SRAM0 (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 SRAM0. This register also functions as an interrupt or a 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 SRAM0. 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 SRAM0 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 ECCPRCR bit, write 78h to the KW[6:0] bits simultaneously. When a value other than 78h is written to the KW[6:0] bits, 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 writing of 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 writes to the ECCPRCR2 bit. When writing to ECCPRCR2 bit, write 78h to the KW2[6:0] bits simultaneously. When a value other than 78h is written to the KW2[6:0] bits, 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 of ECC test, see section 42.3.3, 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 SRAM0 (ECC area).

42.2.11 Trace Control (for the MTB)

The Micro Trace Buffer (MTB) has programmable registers to control the behavior of the trace features and the POSITION, MASTER, FLOW, and BASE registers. The following shows the registers in offset order from the base address:

Base address: 4001 9000h
Base address + 000h: MTB_POSITION value on reset: Bits [31:0] = UNKNOWN
Base address + 004h: MTB_MASTER value on reset: Bits [31] = 0, Bits [30:10] = UNKNOWN,
Bits [9:8] = 0, Bits [7]=1, Bits [6:5] = 0,
Bits [4:0] = UNKNOWN
Base address + 008h: MTB_FLOW value after reset: Bits [31:2] = UNKNOWN, Bits [1:0] = 0
Base address + 00Ch: MTB_BASE value after reset: Bits [31:0] = 2000 4000h

For more information on these registers, see the *ARM® CoreSight™ MTB-M23 Technical Reference Manual* (ARM DDI 0564C).

Note: Do not attempt to access reserved or unused address locations. This can result in UNPREDICTABLE behavior.

The MTB for trace is limited from 2000 4000h to 2000 7FFFh.

42.2.12 CoreSight™ (for MTB)

See the *ARM® CoreSight™ Architecture Specification* for more information about the registers and access types. The following shows the registers in offset order from the base address:

Base address: 4001 9000h
Base address + FF0h to FFCh: Component ID
Base address + FE0h to FDCh: Peripheral ID
Base address + FCCh: Device Type Identifier
Base address + FC8h: Device Configuration
Base address + FBCh: Device Architecture
Base address + FB8h: Authentication Status
Base address + FB4h: Lock Status
Base address + FB0h: Lock Access

For more information on these registers, see the *ARM® CoreSight™ MTB-M23 Technical Reference Manual* (ARM DDI 0564C).

Note: Do not attempt to access reserved or unused address locations. This can result in UNPREDICTABLE behavior.

42.3 Operation

42.3.1 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 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 bit 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.

Because the SRAM data is undefined after a power on, accessing the SRAM when the ECC function is enabled and error checking is selected, causes an ECC error. To avoid this, write 32-bit data to the area used in the SRAM before using the ECC function.

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.

42.3.2 ECC Error Generation

When the ECC function is enabled and error checking is applied to SRAM0 (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 the ECC checking revealed a 2-bit error or a 1-bit error, respectively.

To mask ECC 1-bit errors, set the ECC1STSEN.E1STSEN bit to 0 to disable updating of the ECC1ERR bit. An ECC error is not generated when the ECC function is disabled or enabled and error checking is not selected.

An ECC error can generate a non-maskable interrupt or a reset, as specified 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.

42.3.3 ECC Decoder Testing

Figure 42.1 shows the ECC decoder testing.

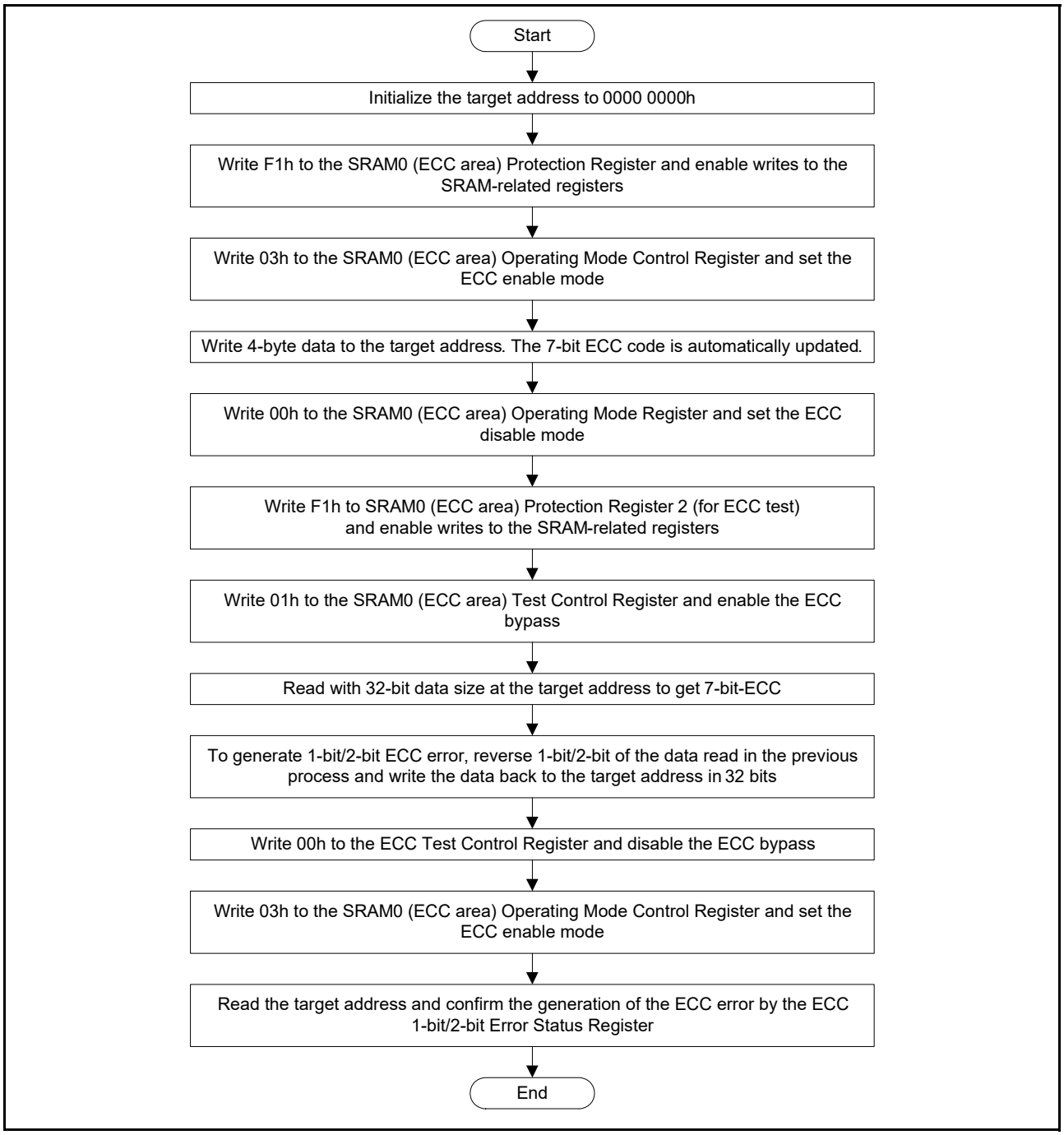


Figure 42.1 ECC decoder testing

42.3.4 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 of the PARIOAD register. When the OAD bit is set to 1, a parity error is output to the reset function. When the OAD bit is set to 0, a parity error is output to the ICU as a non-maskable interrupt.

Parity errors 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.

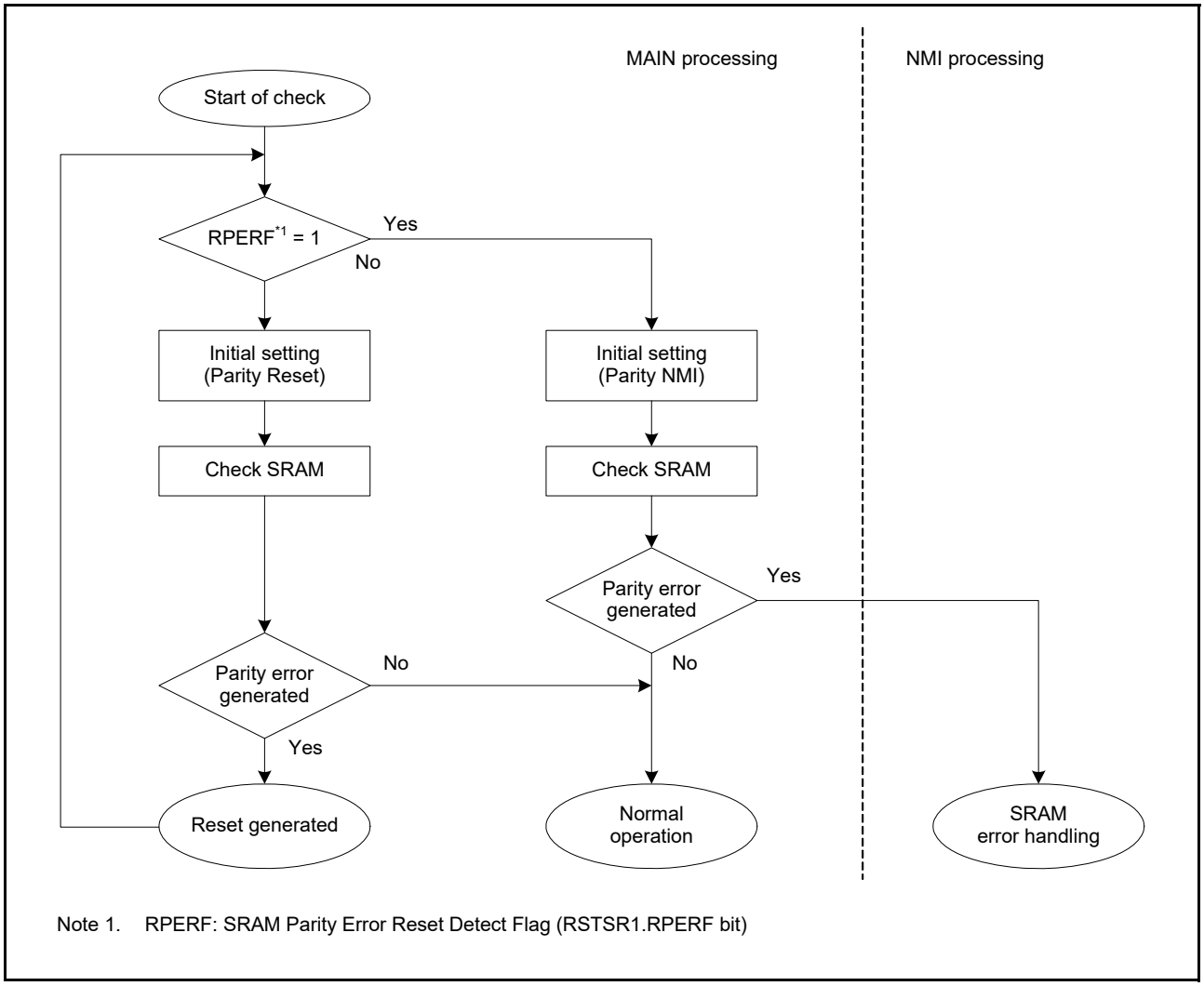
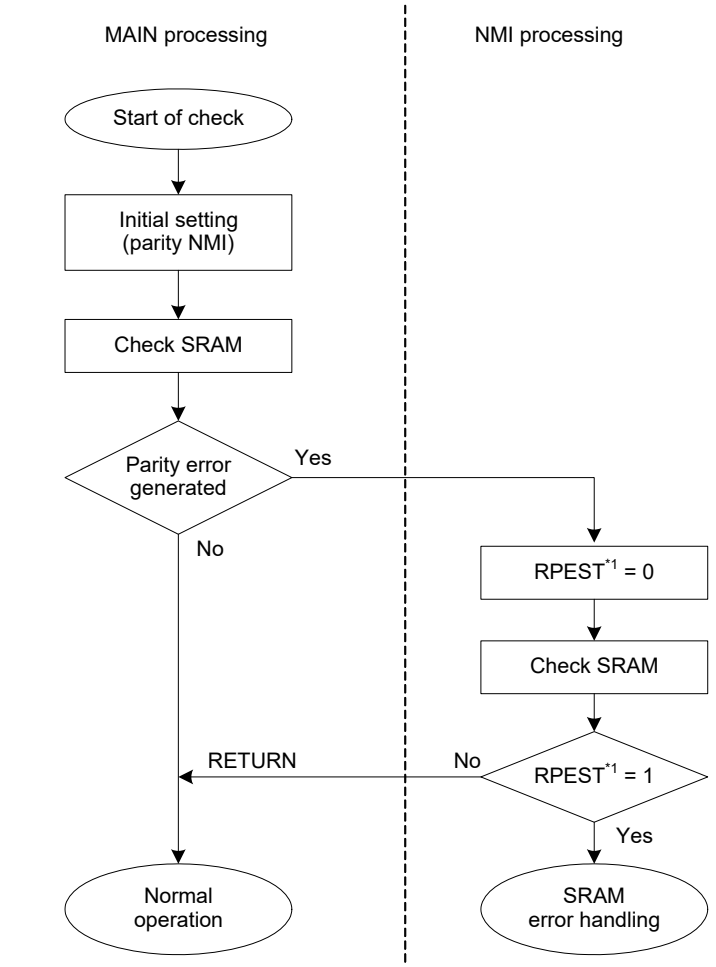


Figure 42.2 Flow of SRAM parity check when SRAM parity reset is enabled



Note 1. RPEST: SRAM Parity Error Interrupt Status Flag (NMISR.RPEST bit)

Figure 42.3 Flow of SRAM parity check when SRAM parity interrupt is enabled

42.3.5 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 PARIOAD register. TDC activation is not supported for SRAM parity errors.

Table 42.2 SRAM error sources

SRAM error source	DTC activation
ECC error (SRAM0 area with ECC)	Not possible
Parity error (SRAM0 area without ECC)	Not possible

42.3.6 Access Cycle

Table 42.3 SRAM0 (ECC area 2000 0000h to 2000 3FFFh)

	Read (cycle)		Write (cycle)	
	Word access	**Halfword/Byte access**	**Word access**	**Halfword/Byte access**
ECC Off ECCMOD[1] = 0	2		2	
ECC On ECCMOD[1] = 1	2		2	4

Table 42.4 SRAM0 (parity area 2000 4000h to 2000 7FFFh)

Read (cycles)		Write (cycles)	
Word access	**Halfword/Byte access**	**Word access**	**Halfword/Byte access**
2		2	

42.4 Usage Notes

42.4.1 Instruction Fetch from the SRAM Area

When using SRAM0 to operate a program, initialize the SRAM area so that the CPU can correctly prefetch the 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 2-byte area from the end address of a program with a 4-byte boundary. Renesas recommends using the NOP instruction for data initialization.

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 256-KB code flash memory and 8-KB data flash memory. The Flash Control Block (FCB) controls the programming commands.

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 Code flash memory and data flash memory specifications

Parameter	Code flash memory	Data flash memory
Memory capacity	• 256 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 with or reading of data in flash memory	
Protection	Protection against erroneous overwriting of flash memory	
Background operations (BGOs)	Code flash memory can be read during data flash memory programming	
Units of programming and erasure	• 64-bit units for programming in user area • 2-KB units for erasure in user area.	• 8-bit units for programming in data area • 1-KB units for erasure in data area.
Other functions	Interrupts accepted during self-programming	
	An expansion area of flash memory (option bytes) can be set in the initial MCU settings	
On-board programming	Programming in serial programming mode (SCI boot mode): • Asynchronous serial interface (SCI9) used • Transfer rate adjusted automatically. Programming in serial programming mode (USB boot mode*1): • USBFS used • Dedicated hardware not required, so direct connection to a PC is possible. Programming in on-chip debug mode: • SWD interface used • Dedicated hardware not required. Programming by a routine for code and data flash memory programming within the user program: • Allows code and data flash memory programming without resetting the system.	

Note 1. USB boot mode does not exist with 32-pin products.

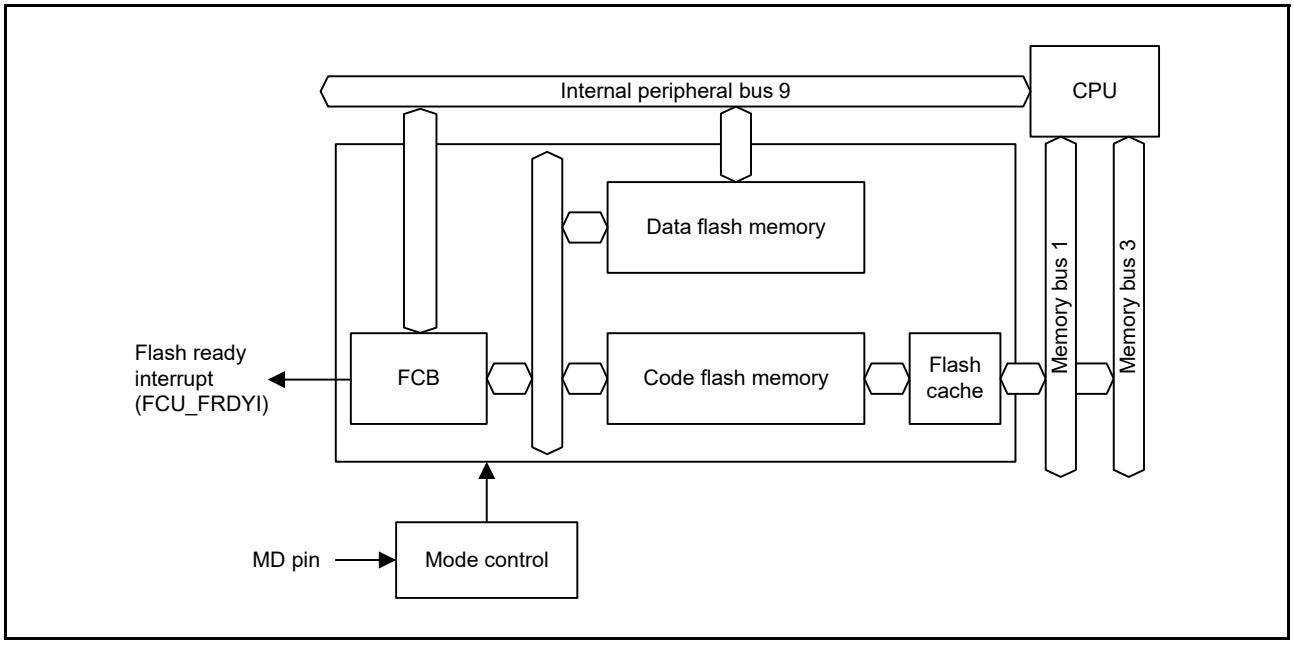


Figure 43.1 Flash memory-related modules block diagram

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 addresses of the code flash memory. The user space of the code flash memory is divided into 2-KB blocks, which serve as the units of erasure. The user area is available for storing the user program.

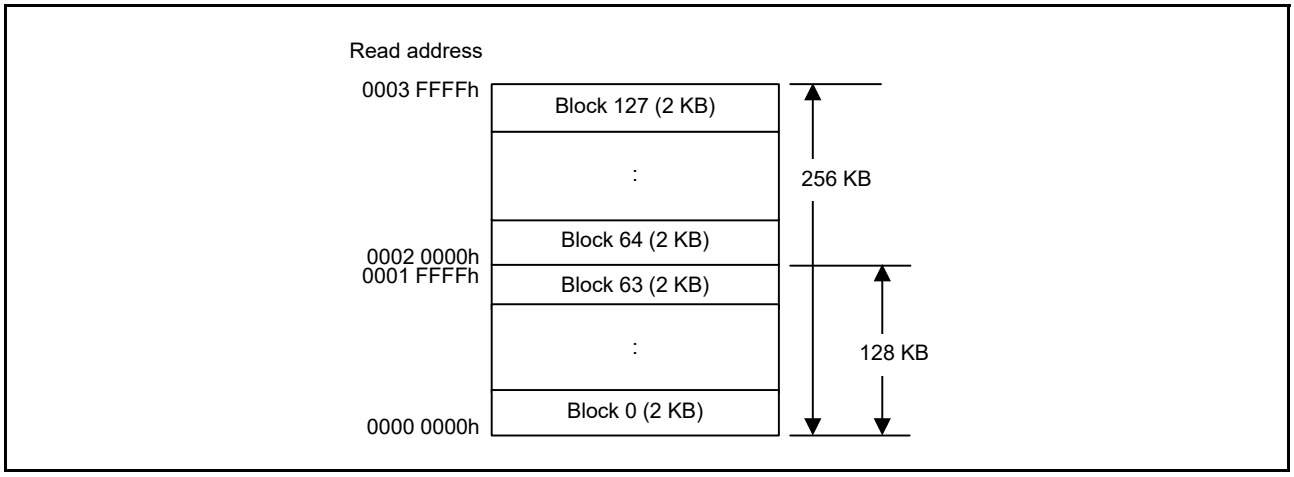


Figure 43.2 Mapping of the code flash memory

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
256 KB	0000 0000h to 0003 FFFFh	0000 0000h to 0003 FFFFh	0 to 127

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.

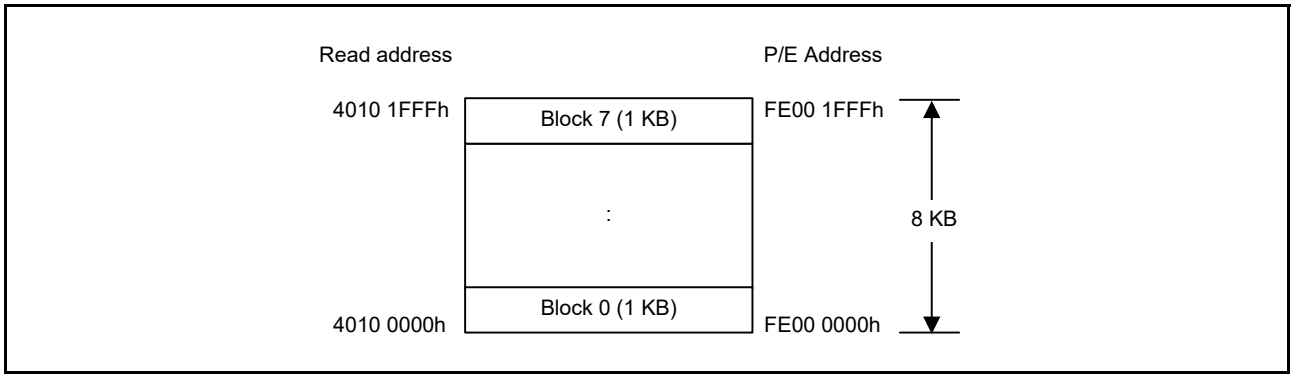


Figure 43.3 Mapping of the data flash memory

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 DTC
- 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 setting: MEMWAIT = 0: 0 wait MEMWAIT = 1: 1 or 2 waits	Cache hit: 0 wait Cache miss: According to SYSTEM.MEMWAIT Register setting: MEMWAIT = 0: 0 wait MEMWAIT = 1: 1 or 2 waits	Cache hit: 0 wait Cache miss: According to SYSTEM.MEMWAIT Register setting: MEMWAIT = 0: 0 wait MEMWAIT = 1: 1 or 2 waits

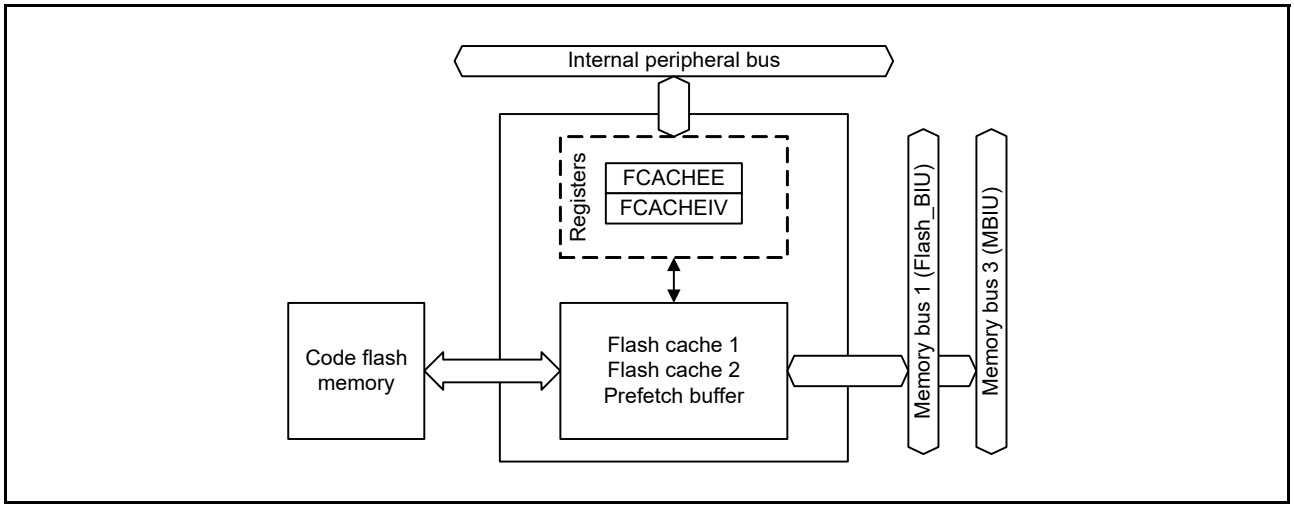


Figure 43.4 FCACHE block diagram

43.4 Register Descriptions

43.4.1 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 14.5, Notes on using Flash Cache for details on HPROT[3].

43.4.2 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
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.

43.4.3 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 used to enable or disable access to the data flash. After setting the DFLCTL.DFLEN bit, data flash stop recovery time (tDSTOP) is required before reading the data flash or entering the data flash P/E mode.

Setup time for each operating mode:

- High-speed operating mode: 5 µs
- Middle-speed operating mode: 720 ns
- Low-speed operating mode: 720 ns
- Low-voltage operating mode: 10 µs

43.4.4 Factory MCU Information Flash Root Table (FMIFRT)

Address(es): 407F B19Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	Value depends on the product															

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

Bit	Description	R/W
b31 to b0	Base address of unique ID	R

The FMIFRT is a read-only register that stores a base address of the Unique ID register, Part Numbering register and MCU Version register. The FMIFRT should be read in 32-bit units. The base address of the RA2A1 MCU is 0x0100_3C00.

43.4.5 Unique ID Register n (UIDRn) (n = 0 to 3)

Address(es): UIDR0 FMIFRT+14h, UIDR1 FMIFRT+18h, UIDR2 FMIFRT+1Ch, UIDR3 FMIFRT+20h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	Unique value for each MCU															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	Unique value for each MCU															

Bit	Description	R/W
b31 to b0	Unique ID	R

The UIDRn is a read-only register that stores a 16-byte ID code (unique ID) for identifying the individual MCU. The UIDRn register should be read in 32-bit units.

43.4.6 Part Numbering Register n (PNRn) (n = 0 to 3)

Address(es): PNR0 FMIFRT+24h, PNR1 FMIFRT+28h, PNR2 FMIFRT+2Ch, PNR3 FMIFRT+30h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	Value depends on the product															

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

Bit	Description	R/W
b31 to b0	Product part number	R

The PNRn is a read-only register that stores a 16-byte part numbering. The PNRn register should be read in 32-bit units. Each byte corresponds to the ASCII code representation of the product part number as described in Product list. The first character ("R", 0x52 in ASCII code) of the part number is stored in the byte with the smallest address (FMIFRT + 24h).

Example of product part number: R7FA2A1AB3CFM

43.4.7 MCU Version Register (MCUVER)

Address(es): FMIFRT+44h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	Value depends on the MCU							

Bit	Description	R/W
b7 to b0	MCU version	R

The MCUVER is a read-only register that stores the MCU version. The MCUVER register should be read in 8-bit units. The higher the value, the newer the MCU version.

43.5 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 set the power control mode 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.

43.5.1 Notice to use Flash Cache

When using flash cache by access from the CPU, Arm® MPU should also be set to cacheable.

See the *ARM®v8-M Architecture Reference Manual* and the *ARM® Cortex®-M23 Processor User Guide*.

43.6 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.

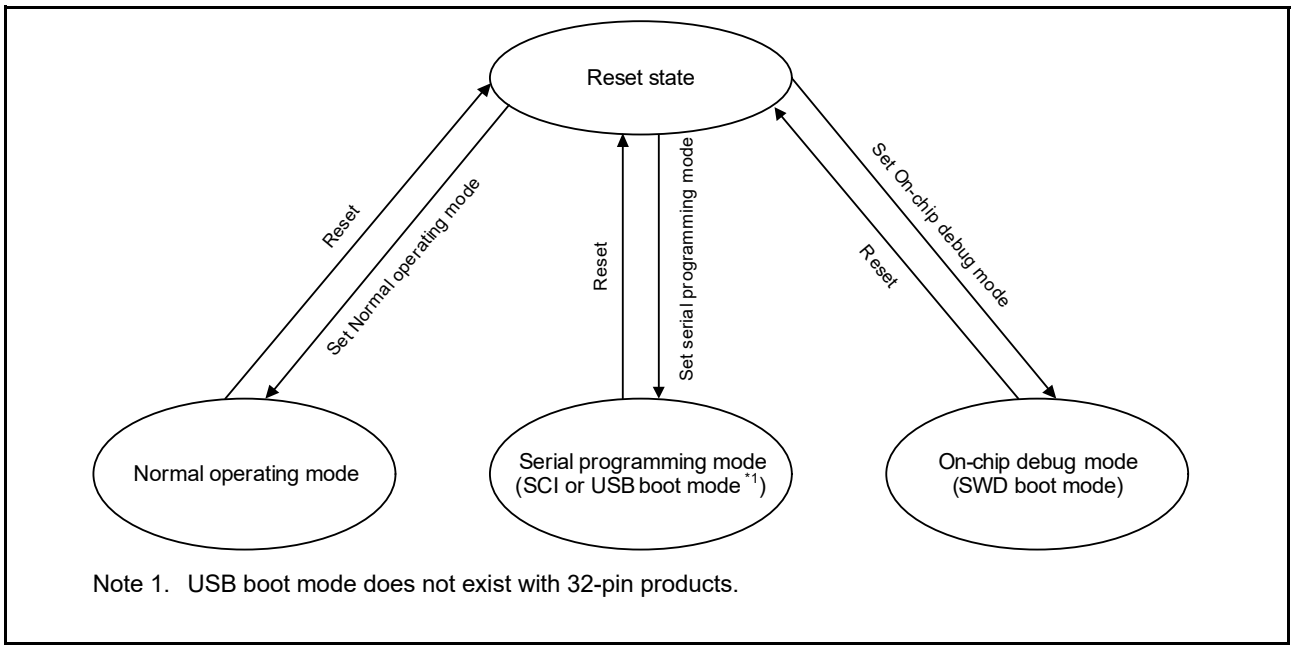


Figure 43.5 Mode transitions associated with flash memory

The flash memory areas where programming and erasure are permitted and where the boot program executes at a reset, differ with the 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*1)	On-chip debug mode (SWD boot mode)
Programmable and erasable areas	• Code flash memory • Data flash memory.	• Code flash memory • Data flash memory.	• Code flash memory • Data flash memory.
Erasure in block units	Possible	Possible	Possible
Boot program at a reset	User area program	Embedded program for serial programming	Depends on debug command

Note 1. USB boot mode does not exist with 32-pin products.

43.6.1 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*1) On-chip debug mode (SWD boot mode)	FFh, …, FFh (all bytes FFh)	Protection disabled	ID code validation is not performed, the ID code always matches, and connection to the programmer or the on-chip debugger is permitted.
	Bit [127] = 1, bit [126] = 1, and at least one of all 16 bytes is not FFh	Protection enabled	Matching ID code: Authentication ends and connection with the programmer or 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 content of the user flash (code and data) area and 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 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 perfomed, the ID code is always mismatching, and connection to the programmer or the on-chip debugger is prohibited, but the ALeRASE command will be accepted. For the prohibition of the ALeRASE command, see section 2.8.3.4 (1) When MSB of OSIS is 0 (bit [127] = 0). The entering of Renesas test mode is protected.

Note 1. USB boot mode does not exist with 32-pin products.

43.7 Overview of Functions

By using a dedicated flash-memory programmer to program the on-chip flash memory through a serial interface (serial programming mode) or through SWD interface (on-chip debug mode), the device can be programmed before or after it is mounted on the target system. Additionally, security functions to prohibit overwriting of the user program prevent tampering by third parties.

Programming by the user program (self-programming) is available for applications that might require updating after system manufacturing or shipment. Protection features for safely overwriting the flash memory area are also provided. Additionally, interrupt processing during self-programming is supported so that programming can 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

Programming method	Functional overview	Operating mode
Serial programming	A dedicated flash-memory programmer connected through the SCI or USBFS interface enables on-board programming of the 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 can also program the flash memory. The background operation capability makes it possible to fetch instructions or otherwise read data from code flash memory while the data flash memory is programmed. As a result, a program resident in code flash memory is able to program the data flash memory.	Normal operating mode
SWD programming	A dedicated flash-memory programmer or an on-chip debugger connected through SWD enables on-board programming of the flash memory after the device is mounted on the target system.	On-chip debug mode
	A dedicated flash-memory programmer or an on-chip debugger connected through SWD and a dedicated programming adapter board allow off-board programming of the flash memory, for example, programming of the device, before it is mounted on the target system.	

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 (1 of 2)

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)

Table 43.8 Basic functions (2 of 2)

Function	Functional overview	Availability	
		Serial programming	Self-programming
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 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 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

43.7.1 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.

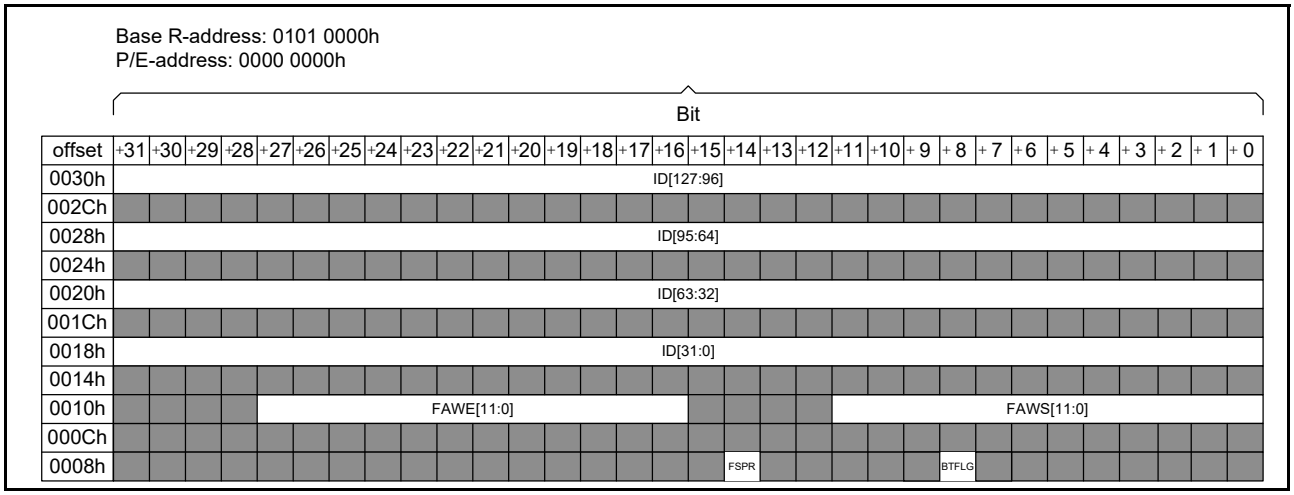


Figure 43.6 Configuration area bit map

43.7.2 Startup Area Select

The startup area select function allows the boot program to be safely updated. The startup area is 8 KB of space located in the user area. The FCB controls the address of the startup area based on the Startup Area Select flag (BTFLG) that is located in the configuration area or the AWSC register. The startup area can be locked by the FSPR bit.

Figure 43.7 shows an overview of the startup program protection.

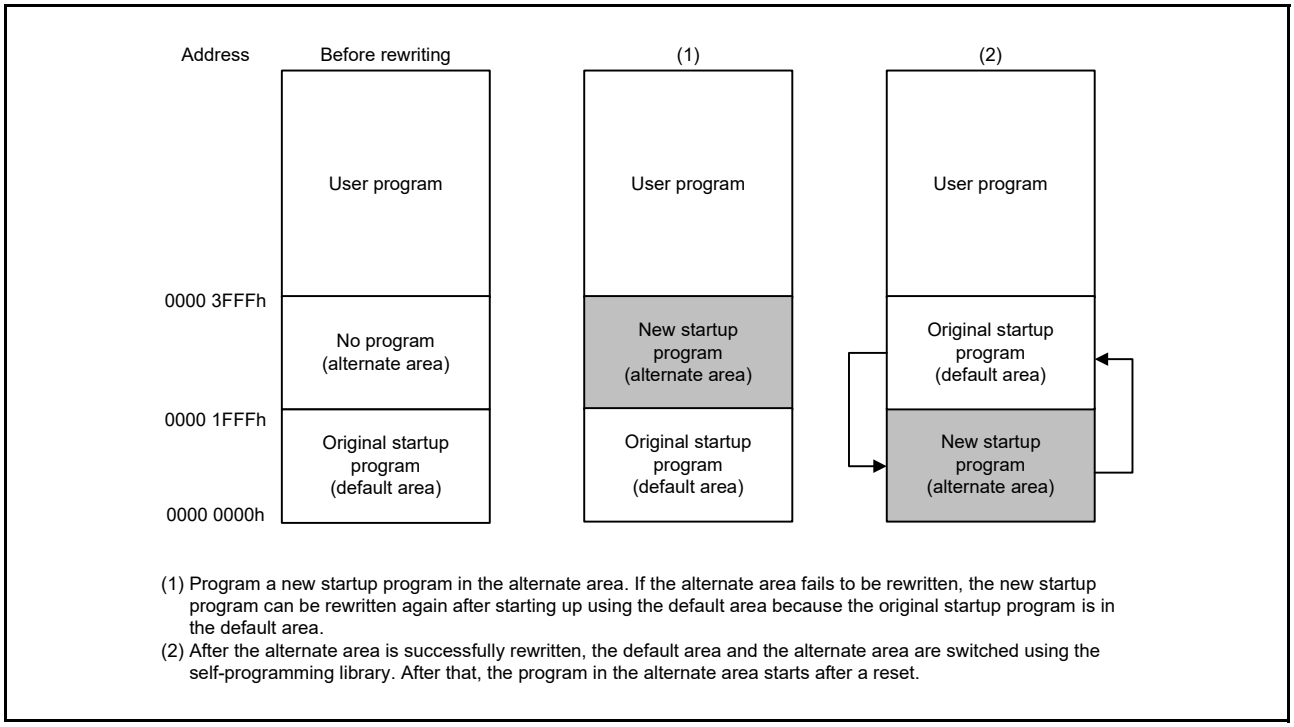


Figure 43.7 Overview of startup program protection

43.7.3 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 an overview of flash area protection.

The access window is specified in both the FAWS [11:0] and FAWE [11:0] bits*1.

Setting of the FAWE[11:0] and the 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 bits to one block lower than the one 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.

Note 1. For information on the AWS.FAWS[11:0] and AWS.FAWE[11:0] bits, see section 7.2.5, Access Window Setting Register (AWS).

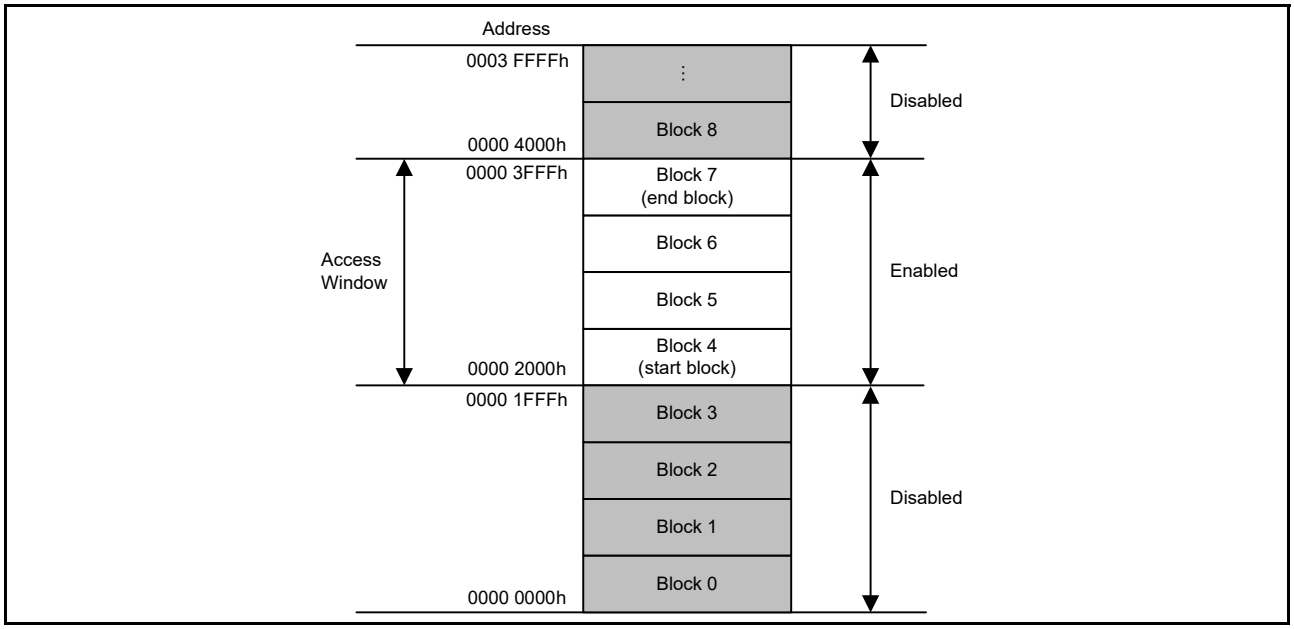


Figure 43.8 Flash area protection overview

43.8 Programming Commands

The FCB controls the programming commands.

43.9 Suspend Operation

The forced stop command forces the blank check command or block erase command to stop. When a forced stop is executed, the stopped address values are stored in the registers. The command can restart from the stopped address after a reset to the registers for command execution by copying the saved addresses.

43.10 Protection

The types of protection provided include:

- Software protection
- Error protection
- Boot program protection.

43.11 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

Pin name	I/O	Applicable modes	Function
MD	Input	SCI boot mode USB boot mode*1 (serial programming mode)	Selection of operating mode
P110/RXD9	Input	SCI boot mode	For host communication, to receive data through the SCI
P109/TXD9	Output		For host communication, to transmit data through the SCI
USB_DP, USB_DM	I/O	USB boot mode*1	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.
Note 1. USB boot mode does not exist with 32-pin products.

43.11.1 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.

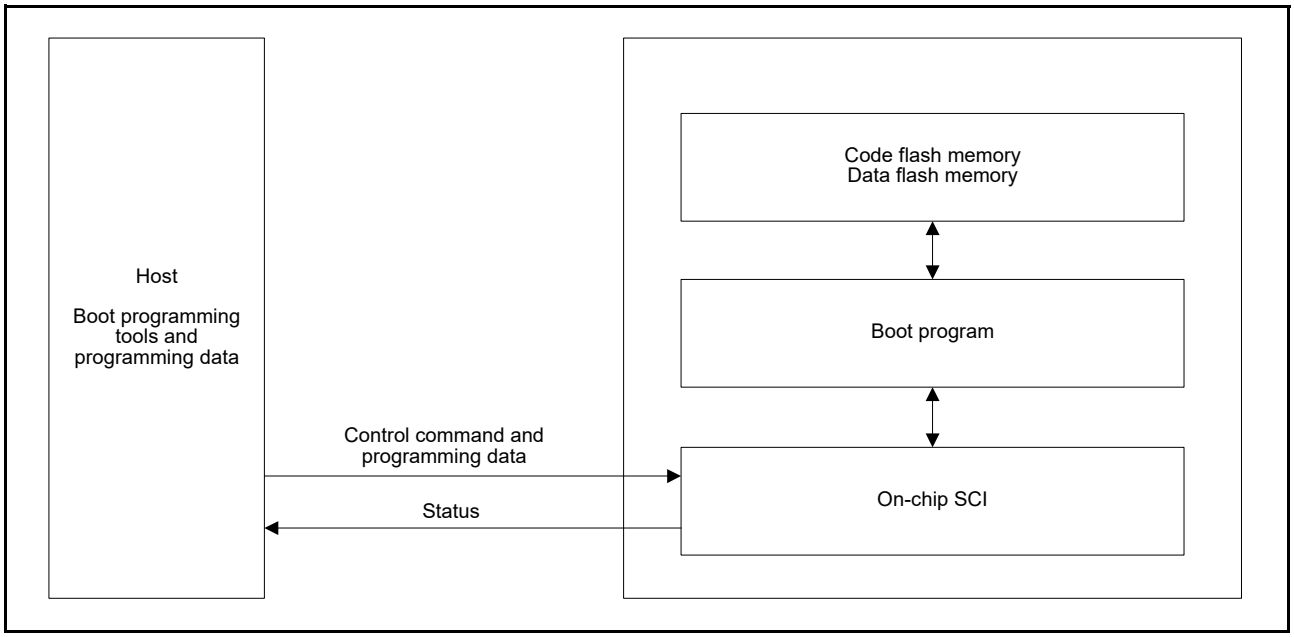


Figure 43.9 System configuration in SCI boot mode

43.11.2 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.

Note: USB boot mode does not exist with 32-pin products.

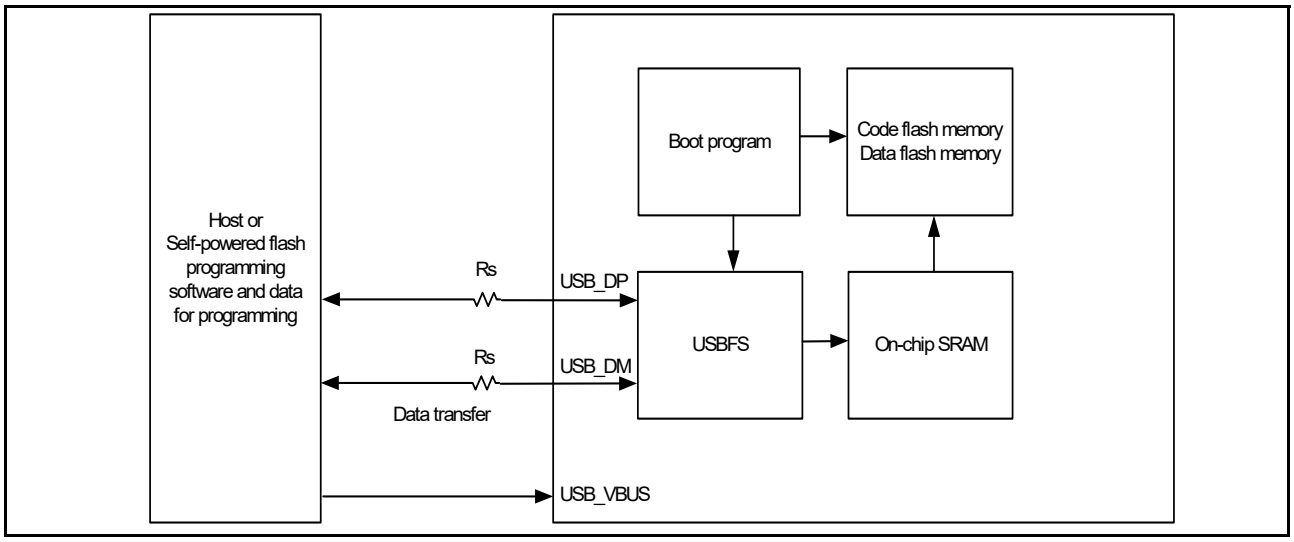


Figure 43.10 System configuration in USB boot mode

43.12 Using a Serial Programmer

A dedicated flash memory programmer can be used to program the flash memory in serial programming mode.

43.12.1 Serial Programming

The MCU is mounted on the system board for serial programming. A connector to the board allows programming by the flash memory programmer.

Figure 43.11 shows the environments recommended by Renesas for programming the flash memory of the MCU with data.

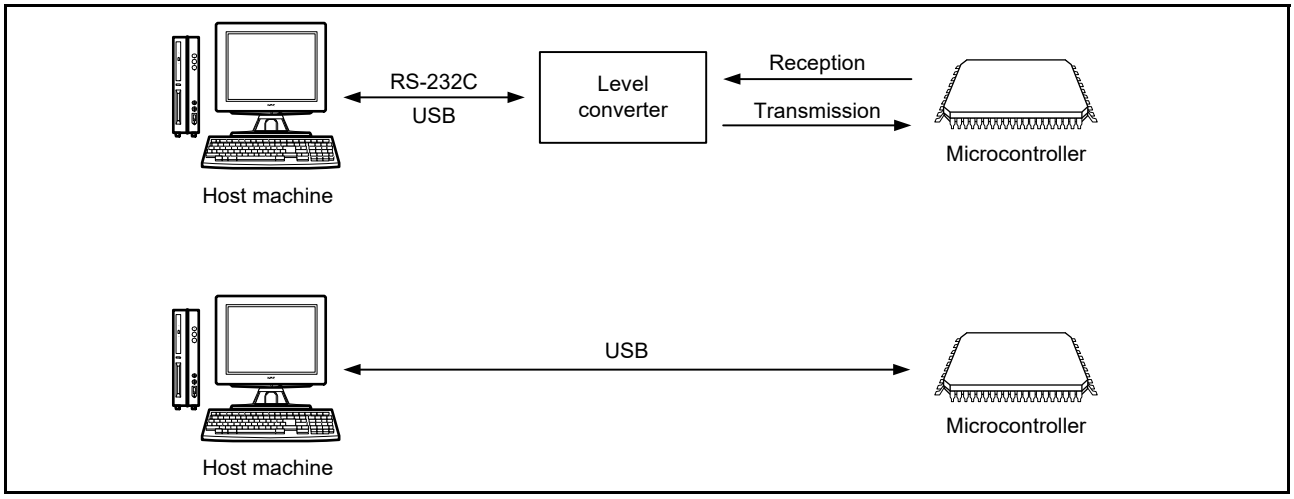


Figure 43.11 Environments for writing programs to the flash memory

43.13 Self-Programming

43.13.1 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.

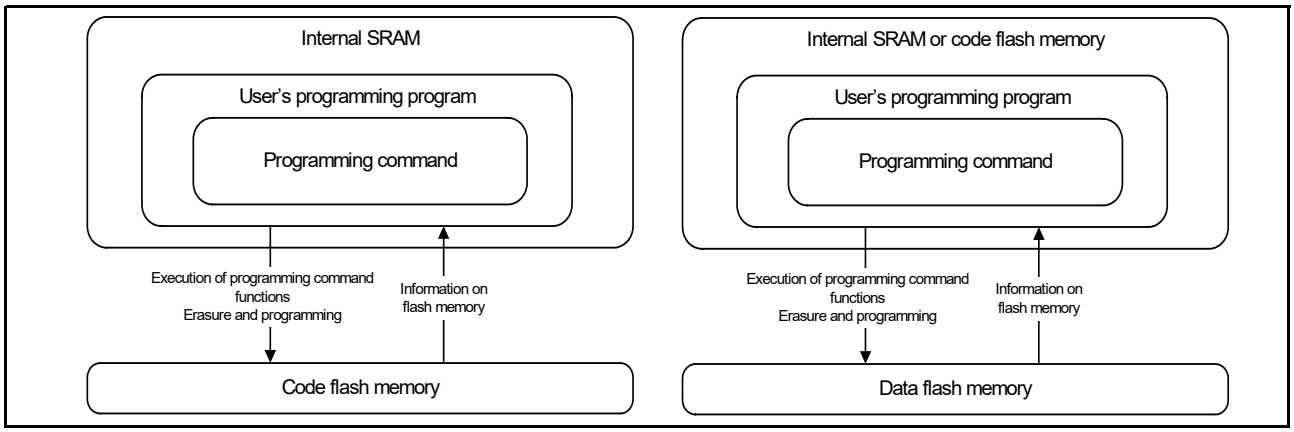


Figure 43.12 Schematic view of self-programming

43.13.2 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

43.14 Reading the Flash Memory

43.14.1 Reading the Code Flash Memory

No special settings are required to read the code flash memory in Normal mode. Data can be read by accessing 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.

43.14.2 Reading the Data Flash Memory

No special settings are required to read the data flash memory in Normal mode except when issuing a reset that causes the data flash access disable mode to disable reading. In this case, the application must transfer back to the data flash read mode. When reading data flash memory that is erased but not yet reprogrammed, such as data flash in the non-programmed state, all bits are read as 1s.

43.15 Usage Notes

43.15.1 Erase Suspended Area

Data in areas where an erase operation is suspended, is undefined. To avoid malfunctions caused by reading undefined data, do not execute commands and read data in the area where erase operation is suspended.

43.15.2 Suspension by Erase Suspend Commands

When suspending an erase operation with the erase suspend command, complete the operation with a resume command.

43.15.3 Restrictions 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.

43.15.4 Reset during Programming and Erasure

If inputting a reset from the RES pin, release the reset after a reset input time of at least t_{RESW} (see section 47.3.3, Reset Timing) within the range of the operating voltage defined in the electrical characteristics.

The IWDT reset and software reset do not require a t_{RESW} input time.

43.15.5 Non-Maskable Interrupt Disabled during Programming and Erasure

When a non-maskable interrupt*1 occurs during a programming or erasure operation, the vectors are fetched from the code flash memory, and undefined data is read. Therefore, do not generate a non-maskable interrupts during programming and erasure operations in the code flash memory. This constraint applies only to the code flash memory.

Note 1. A non-maskable interrupt is an NMI pin interrupt, oscillation stop detection interrupt, WDT underflow or refresh error, IWDT underflow or refresh error, voltage monitor 1 interrupt, voltage monitor 2 interrupt, SRAM parity error, SRAM ECC error, MPU bus slave error, MPU bus master error, or CPU stack pointer monitor.

43.15.6 Location of Interrupt Vectors during Programming and Erasure

When an interrupt occurs during a programming and erasure operation, the vector can be fetched from the code flash memory. 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.

43.15.7 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.

43.15.8 Abnormal Termination during Programming and Erasure

When the voltage exceeds the range of the operating voltage during a programming and erasure operation, or when a programming or erasure operation did not complete successfully because of a reset or prohibited actions as described in section 43.15.9, Actions Prohibited during Programming and Erasure, erase the area again.

43.15.9 Actions Prohibited during Programming and Erasure

To prevent damage to the flash memory, comply with the following instructions during programming and erasure:

- Do not use an MCU power supply that is outside the operating voltage range
- Do not update the OPCCR.OPCM[1:0] bit value
- Do not update the SOPCCR.SOPCM bit value
- Do not change the division ratio of the flash interface clock (FCLK)
- Do not place the MCU in Software Standby mode
- Do not access the data flash memory during a program or erase operation to the code flash memory
- Do not change the data flash access control setting during a program or erase operation to the data flash memory.

44. AES Engine

Regarding the public release of this information, a non-disclosure agreement is required. For details, contact your Renesas sales office.

45. True Random Number Generator (TRNG)

Regarding the public release of this information, a non-disclosure agreement is required. For details, contact your Renesas sales office.

46. Internal Voltage Regulator

46.1 Overview

The MCU includes a linear regulator (LDO) that supplies voltage to the internal circuit and memory, except for I/O and the analog domain.

46.2 Operation

Table 46.1 lists the LDO mode pin settings, and Figure 46.1 shows the LDO mode settings. In LDO mode, the internal voltage is generated from VCC.

Table 46.1 LDO mode pin settings

Pin	Settings
All VCC pins	• Connect each pin to the system power supply • Connect each pin to VSS through a 0.1-µF multilayer ceramic capacitor. Place the capacitor close to the pin.
VCL pin	• Connect each pin to VSS through a 4.7-µF multilayer ceramic capacitor. Place the capacitor close to the pin.

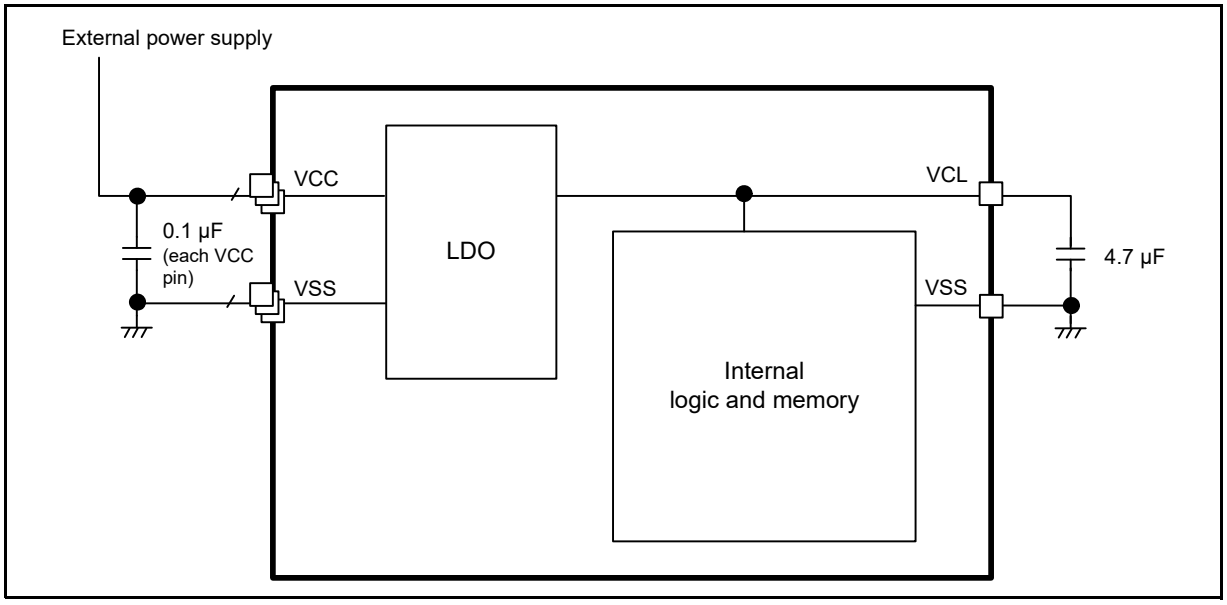


Figure 46.1 LDO mode settings

47. Electrical Characteristics

Unless otherwise specified, the electrical characteristics of the MCU are defined under the following conditions:

- VCC*1 = AVCC0 = AVCC1 = VCC_USB*2 = VCC_USB_LDO*2 = 1.6 to 5.5 V
- VREFH = VREFH0 = 1.6 to AVCC0
- VSS = AVSS0 = AVSS1 = VREFL = VREFL0 = VSS_USB = 0 V
- Ta = T_{opr}.

Note 1. The typical condition is set to VCC = 3.3 V.
Note 2. When USBFS is not used.

Figure 47.1 shows the timing conditions.

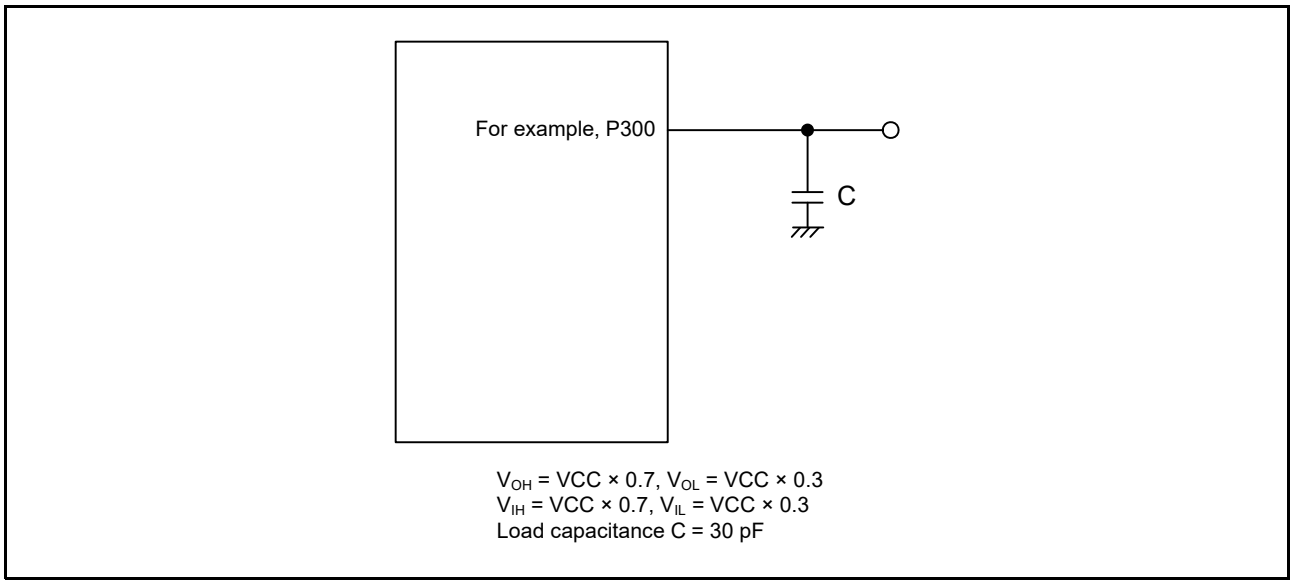


Figure 47.1 Input or output timing measurement conditions

The measurement conditions for the timing specifications of each peripheral are recommended for the best peripheral operation. However, make sure to adjust driving abilities of each pin to meet the conditions of your system.

Each function pin used for the same function must select the same drive ability. If the I/O drive ability of each function pin is mixed, the A/C specification of each function is not guaranteed.

47.1 Absolute Maximum Ratings

Table 47.1 Absolute maximum ratings (1 of 2)

Parameter		Symbol	Value	Unit
Power supply voltage		VCC	-0.5 to +6.5	V
Input voltage	5 V-tolerant ports*1	V_{in}	-0.3 to +6.5	V
	P002, P003, P012 to P015, P500 to P502	V_{in}	-0.3 to AVCC0 + 0.3	V
	P100 to P107	V_{in}	-0.3 to AVCC1 + 0.3	V
	Others	V_{in}	-0.3 to VCC + 0.3	V
Reference power supply voltage		VREFH0	-0.3 to +6.5	V
		VREFH	-0.3 to +6.5	V
		VREFI	-0.3 to AVCC1 + 0.3	V
Analog power supply voltage		AVCC0, AVCC1*5	-0.5 to +6.5	V

Table 47.1 Absolute maximum ratings (2 of 2)

Parameter		Symbol	Value	Unit
USB power supply voltage		VCC_USB	-0.5 to +6.5	V
		VCC_USB_LDO	-0.5 to +6.5	V
Analog input voltage	When AN000 to AN008 are used	V_{AN}	-0.3 to AVCC0 + 0.3	V
	When AN016 to AN023 are used		-0.3 to AVCC1 + 0.3	V
	When ANSD0P to ANSD3P and ANSD0N to ANSD3N are used		-0.3 to AVCC1 + 0.3	V
Operating temperature*2 *3 *4		T_{opr}	-40 to +85 -40 to +105	°C
Storage temperature		T_{stg}	-55 to +125	°C

Note 1. Ports P000, P111, P112, P205, P206, P301, P401, P407, and P409 are 5 V tolerant.
Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up might cause malfunction and the abnormal current that passes in the device at this time might cause degradation of internal elements.

Note 2. See section 47.2.1, Tj/Ta Definition.

Note 3. Contact Renesas Electronics sales office for information on derating operation when Ta = +85°C to +105°C. Derating is the systematic reduction of load for improved reliability.

Note 4. The upper limit of the operating temperature is 85°C or 105°C, depending on the product. For details, see section 1.3, Part Numbering.

Note 5. Use AVCC0 and AVCC1 under the same conditions:
AVCC0 = AVCC1

Caution: Permanent damage to the MCU may result if absolute maximum ratings are exceeded.
To preclude any malfunctions due to noise interference, insert capacitors with high frequency characteristics between the VCC and VSS pins, between the AVCC0 and AVSS0 pins, between the AVCC1 and AVSS1 pins, between the VCC_USB and VSS_USB pins, between the VREFH and VREFL pins, and between the VREFH0 and VREFL0 pins when VREFH0 is selected as the high potential reference voltage for the ADC16. Place capacitors of the following value as close as possible to every power supply pin and use the shortest and heaviest possible traces:
- VCC and VSS: about 0.1 µF
- AVCC0 and AVSS0: about 0.1 µF
- AVCC1 and AVSS1: about 0.1 µF
- VREFH and VREFL: about 0.1 µF
- VREFH0 and VREFL0: about 10 µF.
Also, connect capacitors as stabilization capacitance.
Connect the VCL pin to a VSS pin by a 4.7 µF capacitor. Connect the VREFH0 pin to a VREFL0 pin by 1 µF (-25% to +25%) capacitor when VREFADC is selected as the high potential reference voltage of the ADC16. Connect the ADREG pin to a AVSS1 pin by a 0.47 µF (-50% to +20%) capacitor. Connect the SBIAS/VREFI pin to a AVSS1 pin by a 0.22 µF (-20% to +20%) capacitor. Every capacitor must be placed close to the pin.

Table 47.2 Recommended operating conditions (1 of 2)

Parameter	Symbol	Value	Min	Typ	Max	Unit
Power supply voltages	VCC*1, *2	When USBFS is not used	1.6	-	5.5	V
		When USBFS is used USB Regulator Disable	VCC_USB	-	3.6	V
		When USBFS is used USB Regulator Enable	VCC_USB _LDO	-	5.5	V
	VSS		-	0	-	V

Table 47.2 Recommended operating conditions (2 of 2)

Parameter	Symbol	Value	Min	Typ	Max	Unit
USB power supply voltages	VCC_USB	When USBFS is not used	-	VCC	-	V
		When USBFS is used USB Regulator Disable (Input)	3.0	3.3	3.6	V
	VCC_USB_LDO	When USBFS is not used	-	VCC	-	V
		When USBFS is used USB Regulator Disable	-	VCC	-	V
		When USBFS is used USB Regulator Enable	3.8	-	5.5	V
	VSS_USB		-	0	-	V
Analog power supply voltages	AVCC0*1, *2		1.6	-	5.5	V
	AVSS0		-	0	-	V
	AVCC1*1, *2		-	AVCC0	-	V
	AVSS1		-	0	-	V
	VREFH0	When used as ADC16 Reference	1.7	-	AVCC0	V
	VREFL0		-	0	-	V
	VREFH	When used as DAC12 Reference	1.7	-	AVCC0	V
	VREFL		-	0	-	V
	VREFI	When used as SDADC24 Reference*3	0.8	-	2.4	V

Note 1. Use AVCC0, AVCC1, and VCC under the following conditions:
AVCC0, AVCC1, and VCC can be set individually within the operating range when VCC ≥ 2.2 V and AVCC0 = AVCC1 ≥ 2.2 V.
AVCC0 = AVCC1 = VCC when VCC < 2.2 V or AVCC0 = AVCC1 < 2.2 V.

Note 2. When powering on the VCC and AVCC0 and AVCC1 pins, power them on at the same time or the VCC pin first and then the AVCC0 and AVCC1 pins.

Note 3. The condition when using external input for the reference voltage of SDADC24.

47.2 DC Characteristics

47.2.1 Tj/Ta Definition

Table 47.3 DC characteristics

Conditions: Products with operating temperature (T_a) -40 to +105°C

Parameter	Symbol	Typ	Max	Unit	Test conditions
Permissible junction temperature	Tj	-	125	°C	High-speed mode Middle-speed mode Low-voltage mode Low-speed mode SubOSC-speed mode
			105*1		

Note: Make sure that Tj = T_a + θja × total power consumption (W), where total power consumption = (VCC - V_{OH}) × ΣI_{OH} + V_{OL} × ΣI_{OL} + I_{CC}max × VCC.

Note 1. The upper limit of operating temperature is 85°C or 105°C, depending on the product. For details, see section 1.3, Part Numbering. If the part number shows the operation temperature at 85°C, then the maximum value of Tj is 105°C, otherwise it is 125°C.

47.2.2 I/O V_{IH}, V_{IL}

Table 47.4 I/O V_{IH}, V_{IL}

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = VCC_USB_LDO = 1.6 to 5.5 V

Parameter		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	IIC (except for SMBus)*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)	IIC (SMBus)*2	V_{IH}	2.2	-	-		VCC = 3.6 to 5.5 V
		V_{IH}	2.0	-	-		VCC =2.7 to 3.6 V
		V_{IL}	-	-	0.8		VCC = 2.7 to 5.5 V
	5 V-tolerant ports*3	V_{IH}	VCC × 0.8	-	5.8		-
		V_{IL}	-	-	VCC × 0.2		
	P002, P003, P012 to P015, P500 to P502	V_{IH}	AVCC0 × 0.8	-	-		
		V_{IL}	-	-	AVCC0 × 0.2		
	P100 to P107	V_{IH}	AVCC1 × 0.8	-	-		
		V_{IL}	-	-	AVCC1 × 0.2		
	P914, P915	V_{IH}	VCC_USB × 0.8	-	VCC_USB + 0.3		
		V_{IL}	-	-	VCC_USB × 0.2		
	EXTAL Input ports pins except for P002, P003, P012 to P015, P100 to P107, P500 to P502, P914, P915	V_{IH}	VCC × 0.8	-	-		
		V_{IL}	-	-	VCC × 0.2		

Note 1. SCL0_A, SCL0_B, SCL0_C, SDA0_A, SDA0_C, SCL1_B, SCL1_C, SDA1_B, SDA1_C (total 9 pins)

Note 2. SCL0_A, SCL0_B, SCL0_C, SDA0_A, SDA0_B, SDA0_C, SCL1_A, SCL1_B, SCL1_C, SDA1_A, SDA1_B, SDA1_C, SDA1_D (total 13 pins)

Note 3. P000, P111, P112, P205, P206, P301, P401, P407, P409 (total 9 pins)

47.2.3 I/O I_{OH}, I_{OL}

Table 47.5 I/O I_{OH}, I_{OL}

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = VCC_USB_LDO = 1.6 to 5.5 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
	Ports P407, P408, P409	Low drive*1	I_{OH}	-	-	-4.0	mA
			I_{OL}	-	-	4.0	mA
		Middle drive for IIC Fast mode and SPI*4	I_{OH}	-	-	-8.0	mA
			I_{OL}	-	-	8.0	mA
		Middle drive*2 VCC = 3.0 to 5.5 V	I_{OH}	-	-	-20.0	mA
			I_{OL}	-	-	20.0	mA
	Ports P914, P915		I_{OH}	-	-	-4.0	mA
			I_{OL}	-	-	4.0	mA
	Other output pins*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
	Ports P407, P408, P409	Low drive*1	I_{OH}	-	-	-4.0	mA
			I_{OL}	-	-	4.0	mA
		Middle drive for IIC Fast mode and SPI*4	I_{OH}	-	-	-8.0	mA
			I_{OL}	-	-	8.0	mA
		Middle drive*2 VCC = 3.0 to 5.5 V	I_{OH}	-	-	-20.0	mA
			I_{OL}	-	-	20.0	mA
	Ports P914, P915		I_{OH}	-	-	-4.0	mA
			I_{OL}	-	-	4.0	mA
	Other output pins*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 P002, P003, P012 to P015, P500 to P502		$\Sigma I_{OH\ (max)}$	-	-	-30	mA
			$\Sigma I_{OL\ (max)}$	-	-	30	mA
	Total of ports P100 to P107		$\Sigma I_{OH\ (max)}$	-	-	-30	mA
			$\Sigma I_{OL\ (max)}$	-	-	30	mA
	Total of ports P914, P915		ΣI_{OH}	-	-	-4.0	mA
			ΣI_{OL}	-	-	4.0	mA
	Total of all output pin*5		$\Sigma I_{OH\ (max)}$	-	-	-60	mA
			$\Sigma I_{OL\ (max)}$	-	-	60	mA

Note 1. This is the value when low driving ability is selected with the Port Drive Capability bit in the PmnPFS register.
Note 2. This is the value when middle driving ability is selected with the Port Drive Capability bit in the 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 and SPI 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 47.12, CTSU Characteristics.

Caution: To protect the reliability of the MCU, the output current values should not exceed the values in Table 47.5. The average output current indicates the average current value measured during 100 µs.

47.2.4 I/O V_{OH}, V_{OL}, and Other Characteristics

Table 47.6 I/O V_{OH}, V_{OL} (1)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = VCC_USB_LDO = 4.0 to 5.5 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 P407, P408, P409	Low drive	V_{OH}	VCC - 0.8	-	-		I_{OH} = -2.0 mA
			V_{OL}	-	-	0.8		I_{OL} = 2.0 mA
		Middle drive for IIC Fast mode and SPI*5	V_{OH}	VCC - 0.8	-	-		I_{OH} = -4.0 mA
			V_{OL}	-	-	0.8		I_{OL} = 4.0 mA
		Middle drive*2,*3	V_{OH}	VCC - 1.0	-	-		I_{OH} = -20 mA
			V_{OL}	-	-	1.0		I_{OL} = 20 mA
	Ports P002, P003, P012 to P015, P500 to P502	Low drive	V_{OH}	AVCC0 - 0.8	-	-		I_{OH} = -2.0 mA
			V_{OL}	-	-	0.8		I_{OL} = 2.0 mA
		Middle drive	V_{OH}	AVCC0 - 0.8	-	-		I_{OH} = -4.0 mA
			V_{OL}	-	-	0.8		I_{OL} = 4.0 mA
	Ports P100 to P107	Low drive	V_{OH}	AVCC1 - 0.8	-	-		I_{OH} = -2.0 mA
			V_{OL}	-	-	0.8		I_{OL} = 2.0 mA
		Middle drive	V_{OH}	AVCC1 - 0.8	-	-		I_{OH} = -4.0 mA
			V_{OL}	-	-	0.8		I_{OL} = 4.0 mA
	Ports P914, P915		V_{OH}	VCC_USB - 0.8	-	-		I_{OH} = -2.0 mA
			V_{OL}	-	-	0.8		I_{OL} = 2.0 mA
	Other output pins*4	Low drive	V_{OH}	VCC - 0.8	-	-		I_{OH} = -2.0 mA
			V_{OL}	-	-	0.8		I_{OL} = 2.0 mA
		Middle drive*6	V_{OH}	VCC - 0.8	-	-		I_{OH} = -4.0 mA
			V_{OL}	-	-	0.8		I_{OL} = 4.0 mA

Note 1. SCL0_A, SCL0_B, SCL0_C, SDA0_A, SDA0_B, SDA0_C, SCL1_A, SCL1_B, SCL1_C, SDA1_A, SDA1_B, SDA1_C, SDA1_D (total 13 pins).
Note 2. This is the value when middle driving ability is selected with the Port Drive Capability bit in the PmnPFS register.
Note 3. Based on characterization data, not tested in production.
Note 4. Except for P200, P214, P215, which are input ports.
Note 5. This is the value when middle driving ability for IIC and SPI is selected with the Port Drive Capability bit in PmnPFS register for P407, P408, and P409.
Note 6. Except for P212, P213.

Table 47.7 I/O V_{OH}, V_{OL} (2)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = VCC_USB_LDO = 2.7 to 4.0 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 P407, P408, P409	Low drive	V_{OH}	VCC - 0.5	-	-		I_{OH} = -1.0 mA
			V_{OL}	-	-	0.5		I_{OL} = 1.0 mA
		Middle drive for IIC Fast mode and SPI*5	V_{OH}	VCC - 0.5	-	-		I_{OH} = -2.0 mA
			V_{OL}	-	-	0.5		I_{OL} = 2.0 mA
		Middle drive*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 P002, P003, P012 to P015, P500 to P502	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 P100 to P107	Low drive	V_{OH}	AVCC1 - 0.5	-	-		I_{OH} = -1.0 mA
			V_{OL}	-	-	0.5		I_{OL} = 1.0 mA
		Middle drive	V_{OH}	AVCC1 - 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	Low drive	V_{OH}	VCC - 0.5	-	-		I_{OH} = -1.0 mA
			V_{OL}	-	-	0.5		I_{OL} = 1.0 mA
		Middle drive*6	V_{OH}	VCC - 0.5	-	-		I_{OH} = -2.0 mA
			V_{OL}	-	-	0.5		I_{OL} = 2.0 mA

Note 1. SCL0_A, SCL0_B, SCL0_C, SDA0_A, SDA0_B, SDA0_C, SCL1_A, SCL1_B, SCL1_C, SDA1_A, SDA1_B, SDA1_C, SDA1_D (total 13 pins).

Note 2. This is the value when middle driving ability is selected with the Port Drive Capability bit in the PmnPFS register.

Note 3. Based on characterization data, not tested in production.

Note 4. Except for P200, P214, P215, which are input ports.

Note 5. This is the value when middle driving ability for IIC and SPI is selected with the Port Drive Capability bit in PmnPFS register for P407, P408, and P409.

Note 6. Except for P212, P213.

Table 47.8 I/O V_{OH}, V_{OL} (3)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = VCC_USB_LDO = 1.6 to 2.7 V

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage	Ports P407, P408, P409	Low drive	V_{OH}	VCC - 0.3	-	-	V	I_{OH} = -0.5 mA
			V_{OL}	-	-	0.3		I_{OL} = 0.5 mA
		Middle drive for IIC Fast mode and SPI*2	V_{OH}	VCC - 0.3	-	-		I_{OH} = -1.0 mA
			V_{OL}	-	-	0.3		I_{OL} = 1.0 mA
	Ports P002, P003, P012 to P015, P500 to P502	Low drive	V_{OH}	AVCC0 - 0.3	-	-		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 P100 to P107	Low drive	V_{OH}	AVCC0 - 0.3	-	-		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	Low drive	V_{OH}	VCC - 0.3	-	-		I_{OH} = -0.5 mA
			V_{OL}	-	-	0.3		I_{OL} = 0.5 mA
		Middle drive*3	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. This is the value when middle driving ability for IIC and SPI is selected with the Port Drive Capability bit in the PmnPFS register for P407, P408, and P409.

Note 3. Except for P212, P213.

Table 47.9 I/O other characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = VCC_USB_LDO = 1.6 to 5.5 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input leakage current	RES, ports P200, P214, P215	$\lvert I_{in} \rvert$	-	-	1.0	μA	V_{in} = 0 V V_{in} = VCC
Three-state leakage current (off state)	5 V-tolerant ports	$\lvert I_{TSI} \rvert$	-	-	1.0	μA	V_{in} = 0 V V_{in} = 5.8 V
	Other ports		-	-	1.0		V_{in} = 0 V V_{in} = VCC
Input pull-up resistor	All ports (except for P200, P214, P215, P914, P915)	R_U	10	20	50	kΩ	V_{in} = 0 V
Input capacitance	P012 to P015, P200, P502, P914, P915	C_{in}	-	-	30	pF	V_{in} = 0 V f = 1 MHz T_a = 25°C
	Other input pins		-	-	15		

47.2.5 Output Characteristics for I/O Pins (Low Drive Capacity)

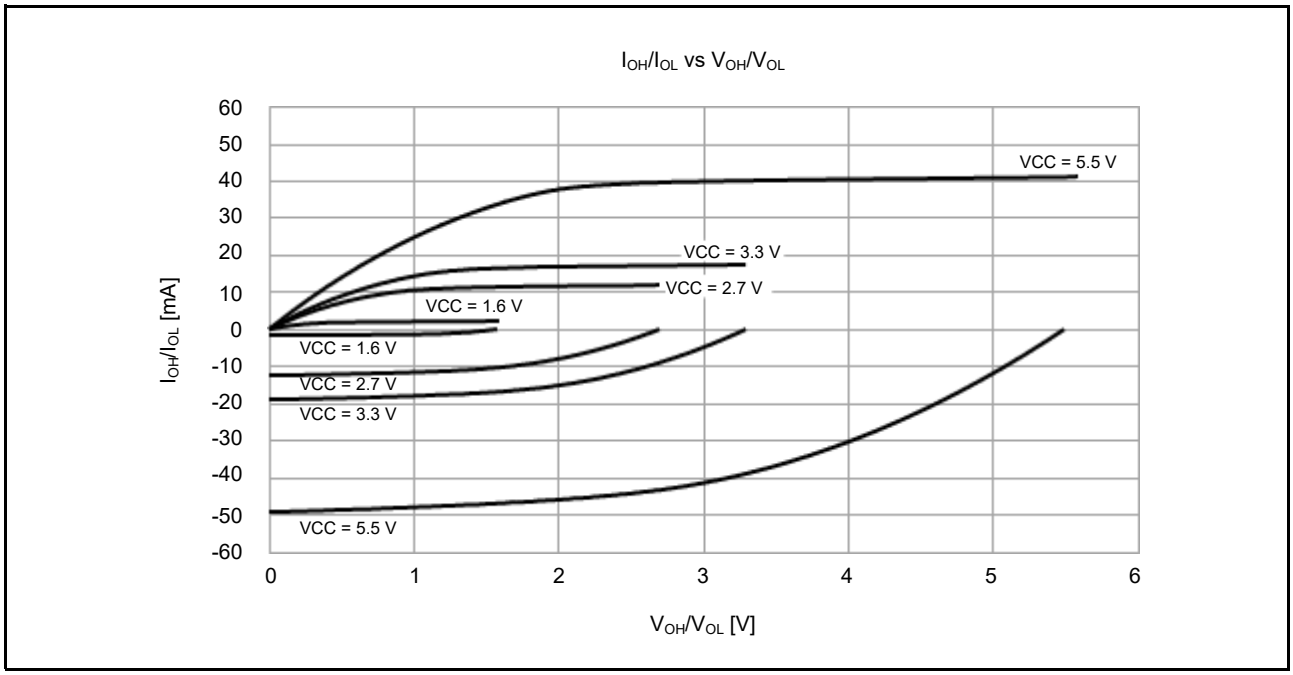


Figure 47.2 **V_{OH}/V_{OL} and I_{OH}/I_{OL} voltage characteristics at Ta = 25°C when low drive output is selected (reference data, except for P914 and P915)**

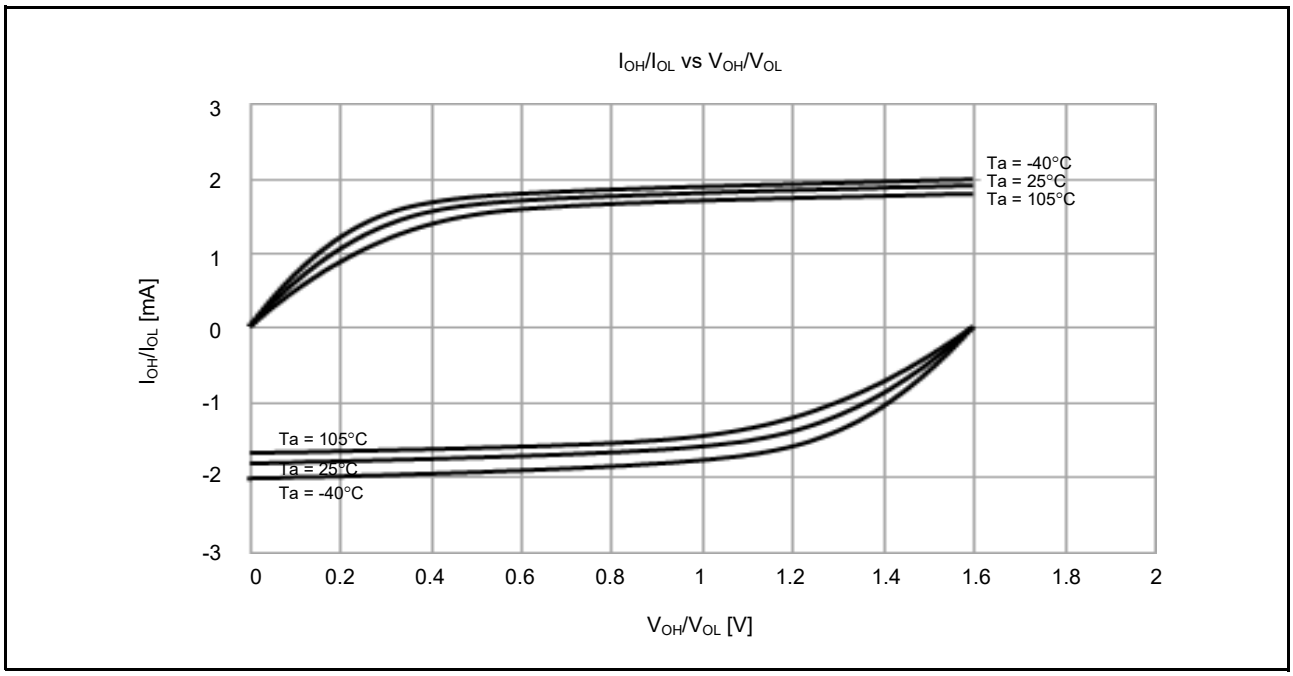


Figure 47.3 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 1.6 V when low drive output is selected (reference data, except for P914 and P915)**

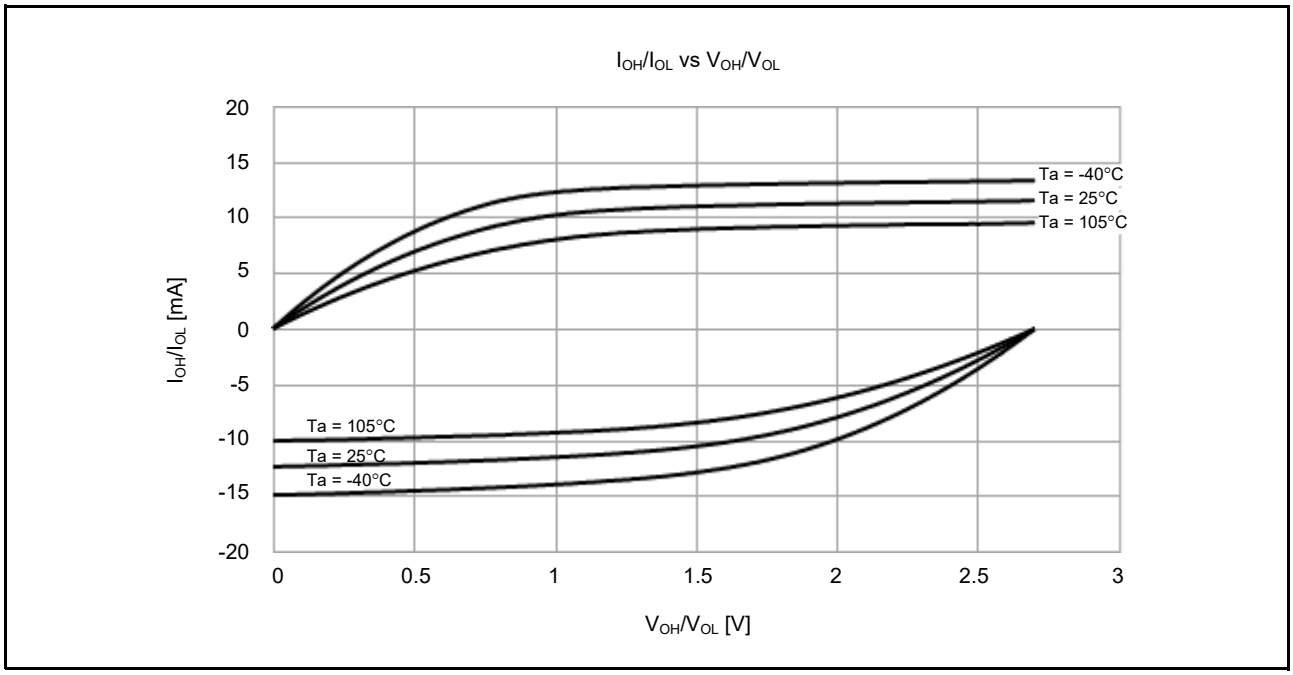


Figure 47.4 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 2.7 V when low drive output is selected (reference data, except for P914 and P915)**

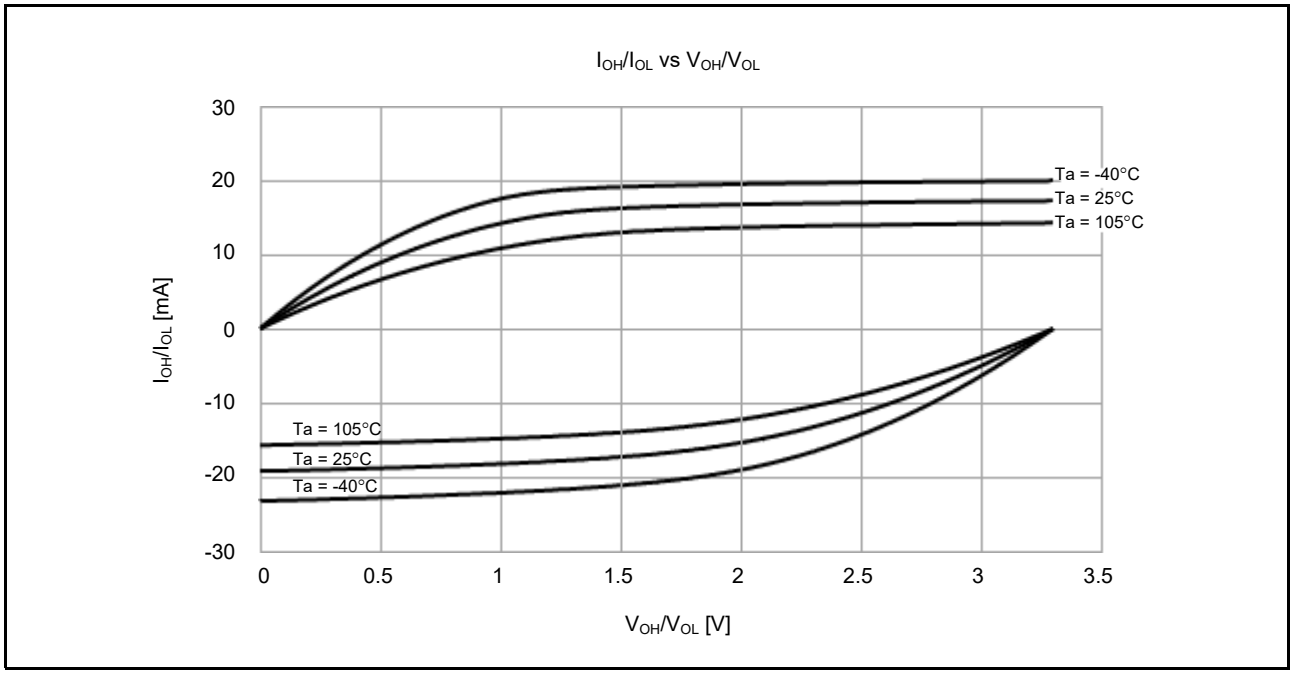


Figure 47.5 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 3.3 V when low drive output is selected (reference data, except for P914 and P915)**

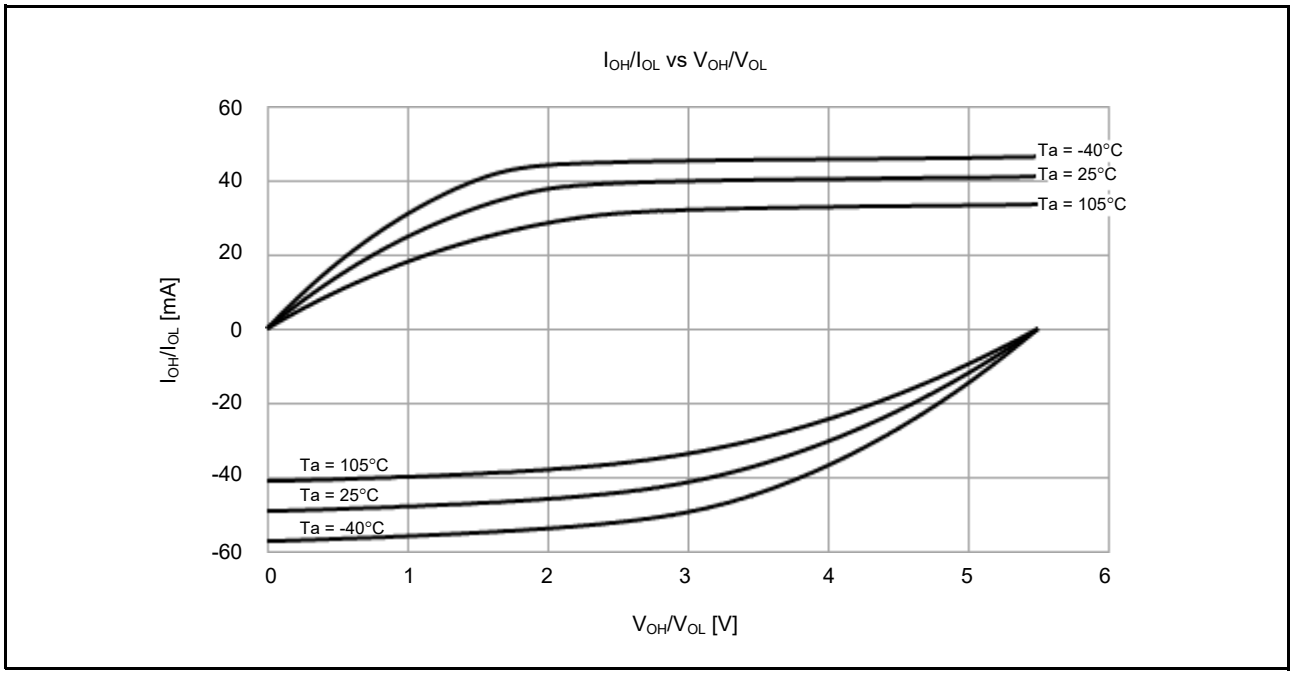


Figure 47.6 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 5.5 V when low drive output is selected (reference data, except for P914 and P915)**

47.2.6 Output Characteristics for I/O Pins (Middle Drive Capacity)

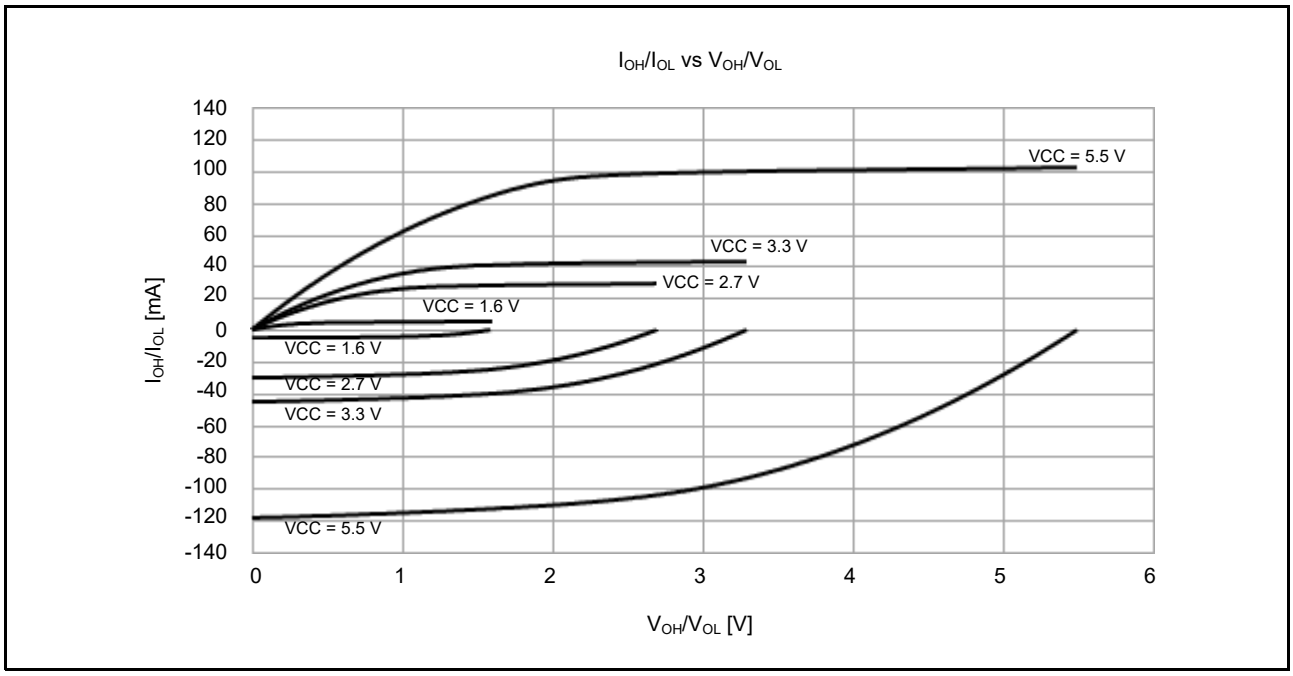


Figure 47.7 **V_{OH}/V_{OL} and I_{OH}/I_{OL} voltage characteristics at Ta = 25°C when middle drive output is selected (reference data, except for P914 and P915)**

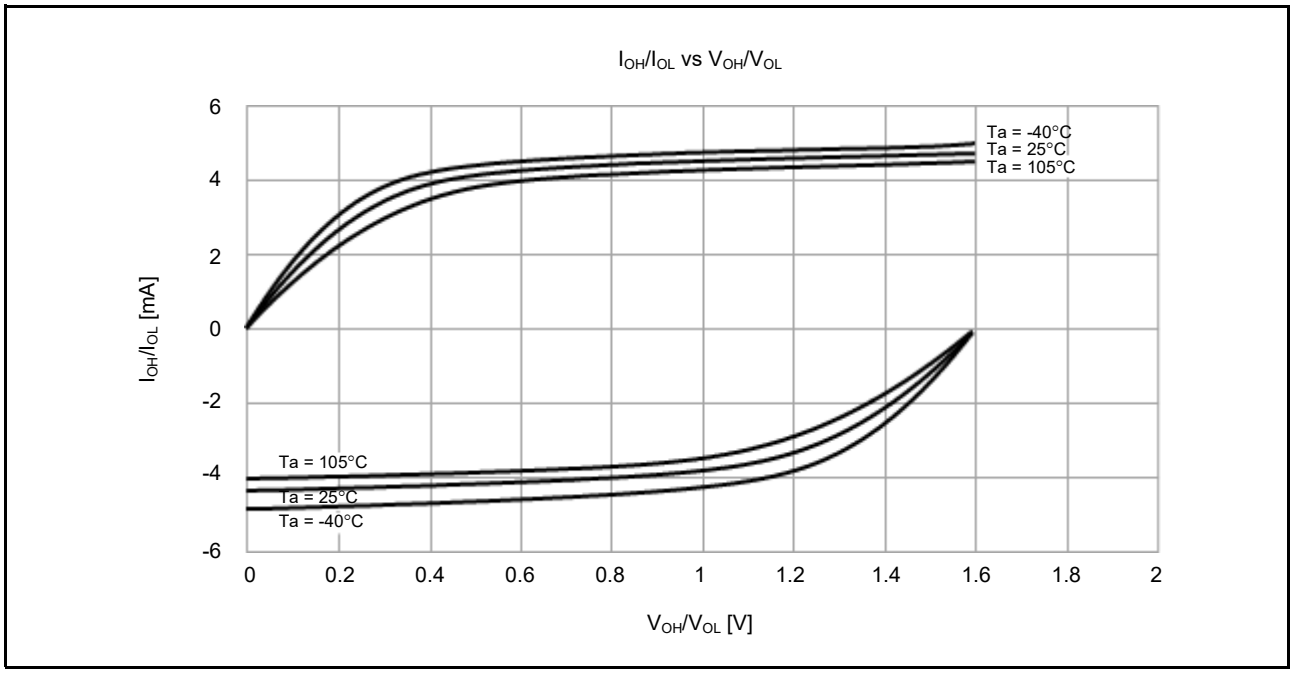


Figure 47.8 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 1.6 V when middle drive output is selected (reference data, except for P914 and P915)**

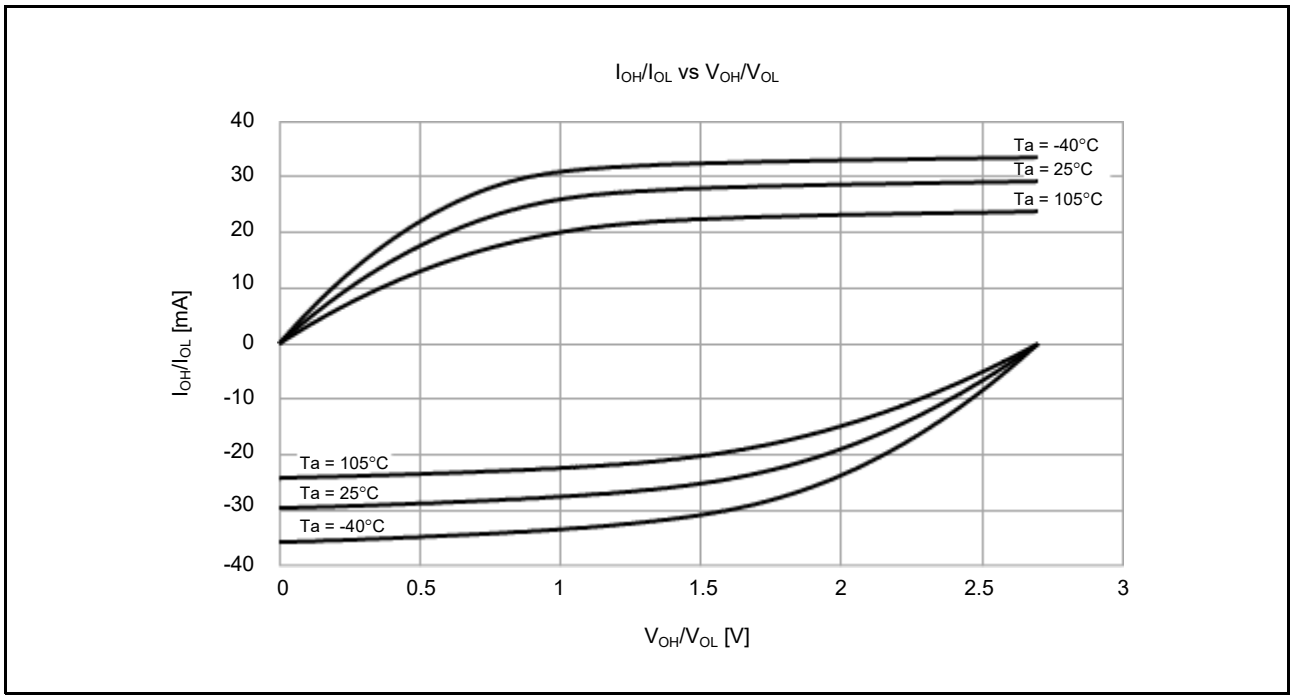


Figure 47.9 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 2.7 V when middle drive output is selected (reference data, except for P914 and P915)**

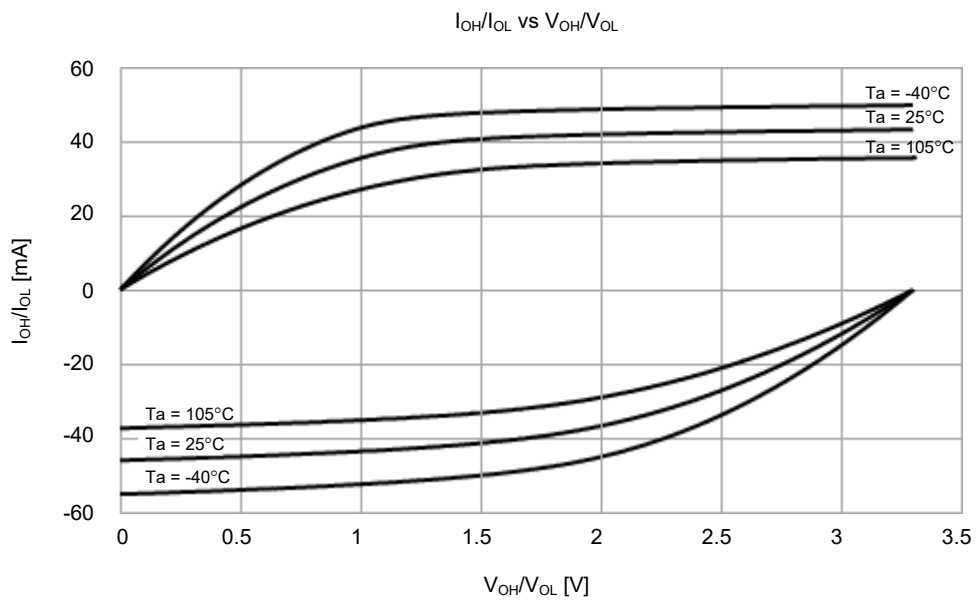


Figure 47.10 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 3.3 V when middle drive output is selected (reference data, except for P914 and P915)**

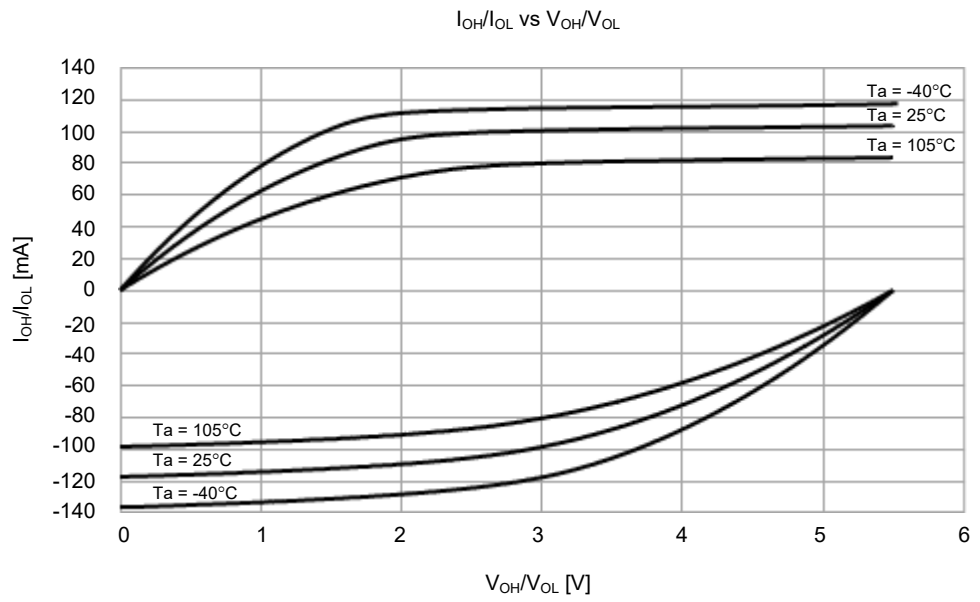


Figure 47.11 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 5.5 V when middle drive output is selected (reference data, except for P914 and P915)**

47.2.7 Output Characteristics for P407, P408 and P409 I/O Pins (Middle Drive Capacity)

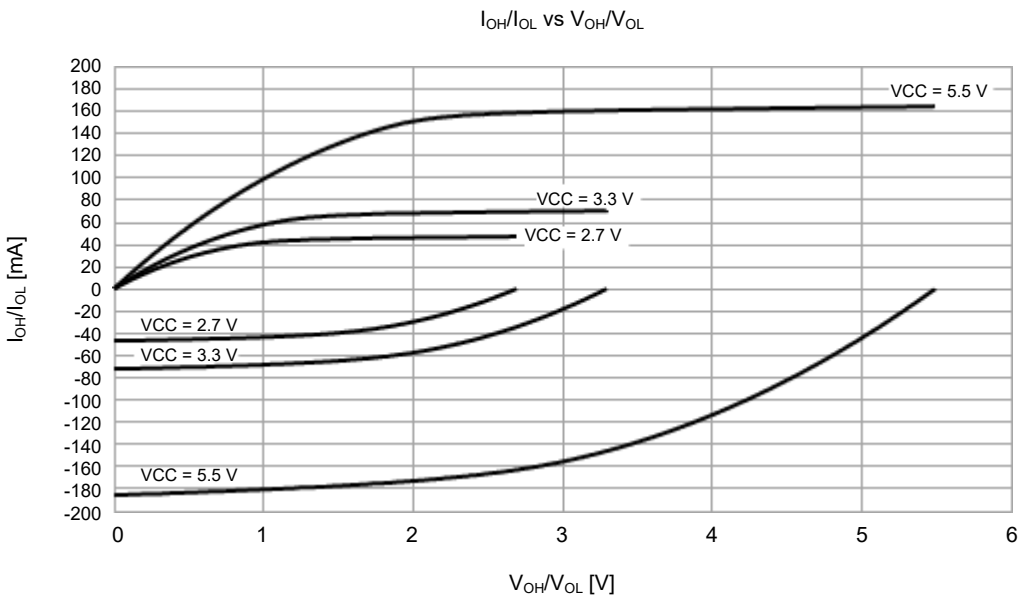


Figure 47.12 **V_{OH}/V_{OL} and I_{OH}/I_{OL} voltage characteristics at Ta = 25°C when middle drive output is selected (reference data)**

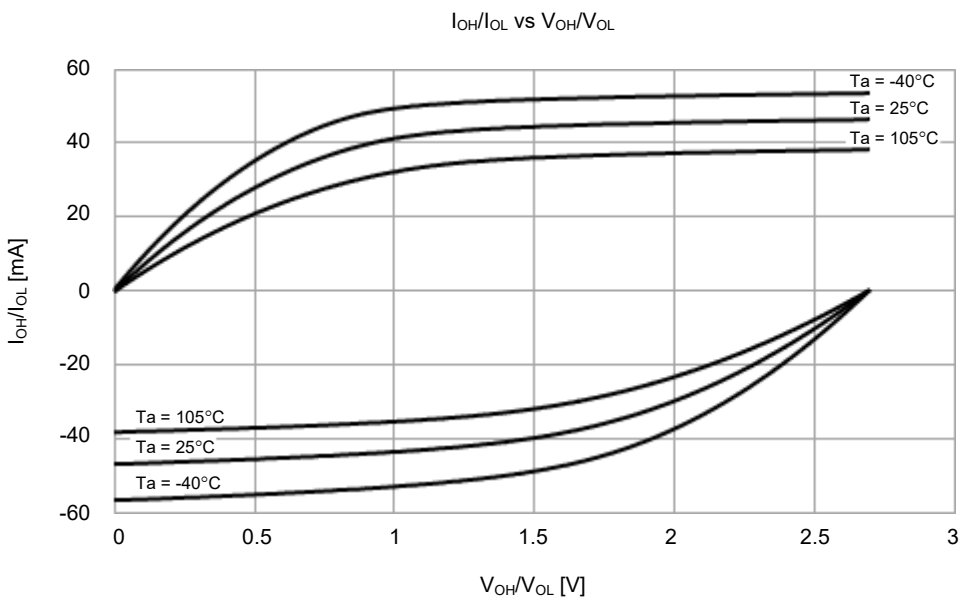


Figure 47.13 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 2.7 V when middle drive output is selected (reference data)**

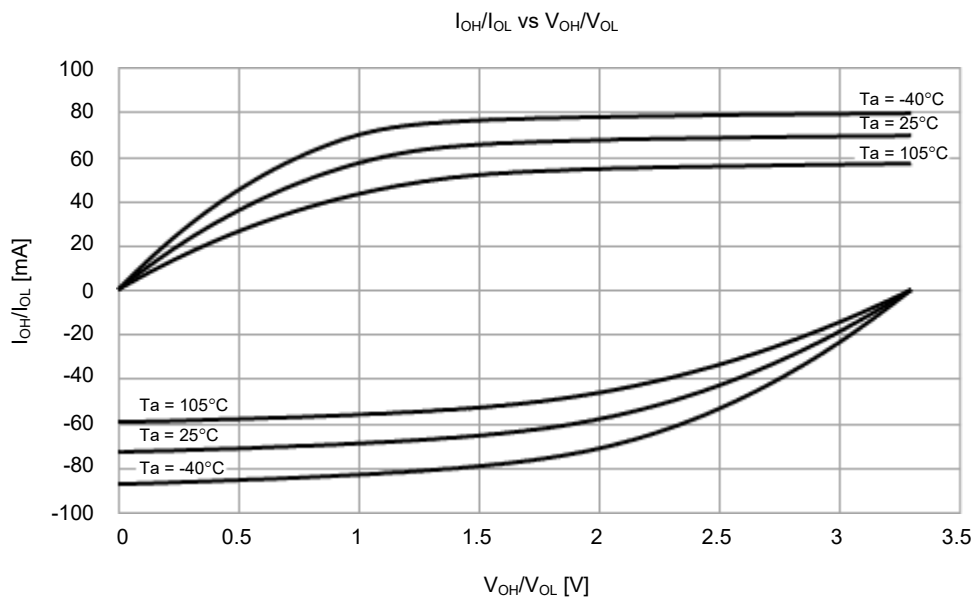


Figure 47.14 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 3.3 V when middle drive output is selected (reference data)**

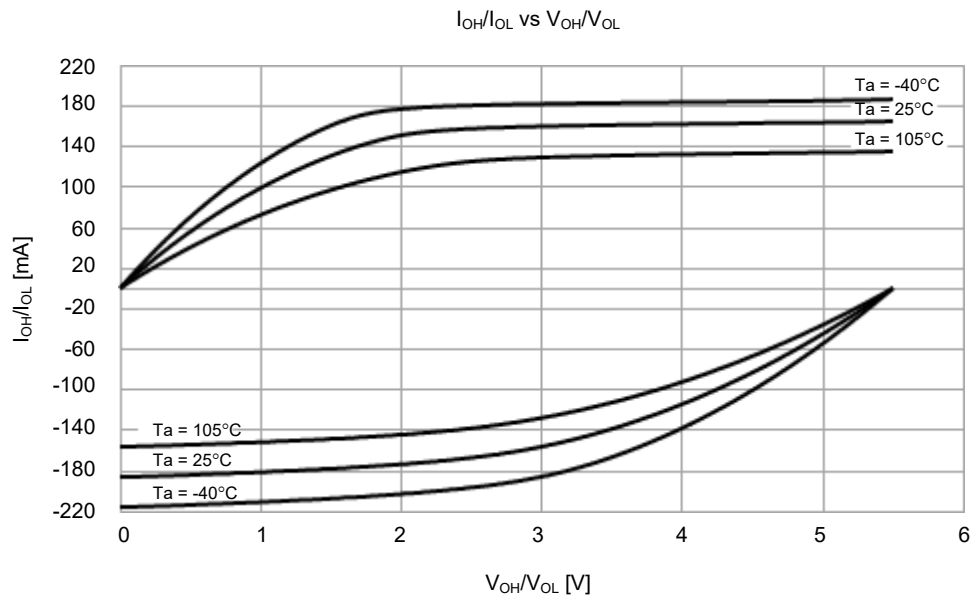


Figure 47.15 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 5.5 V when middle drive output is selected (reference data)**

47.2.8 Output Characteristics for IIC I/O Pins

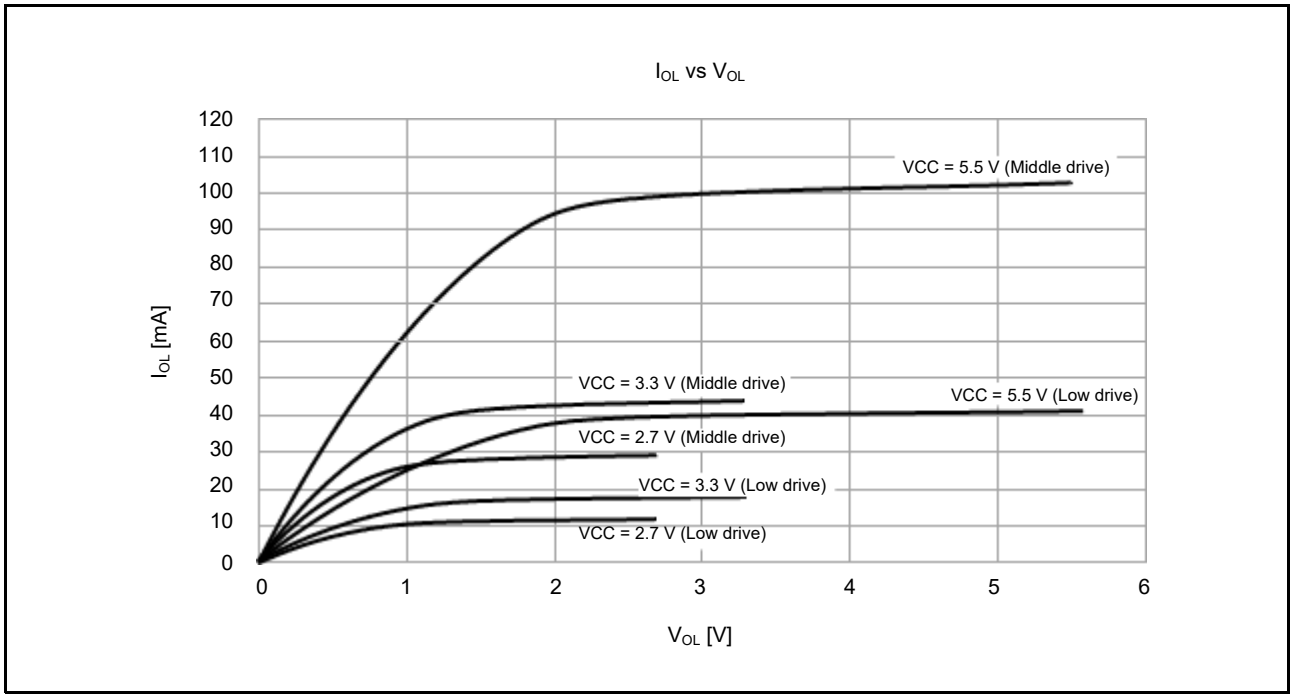


Figure 47.16 V_{OH}/V_{OL} and I_{OH}/I_{OL} voltage characteristics at Ta = 25°C

47.2.9 Operating and Standby Current

Table 47.10 Operating and standby current (1) (1 of 2)

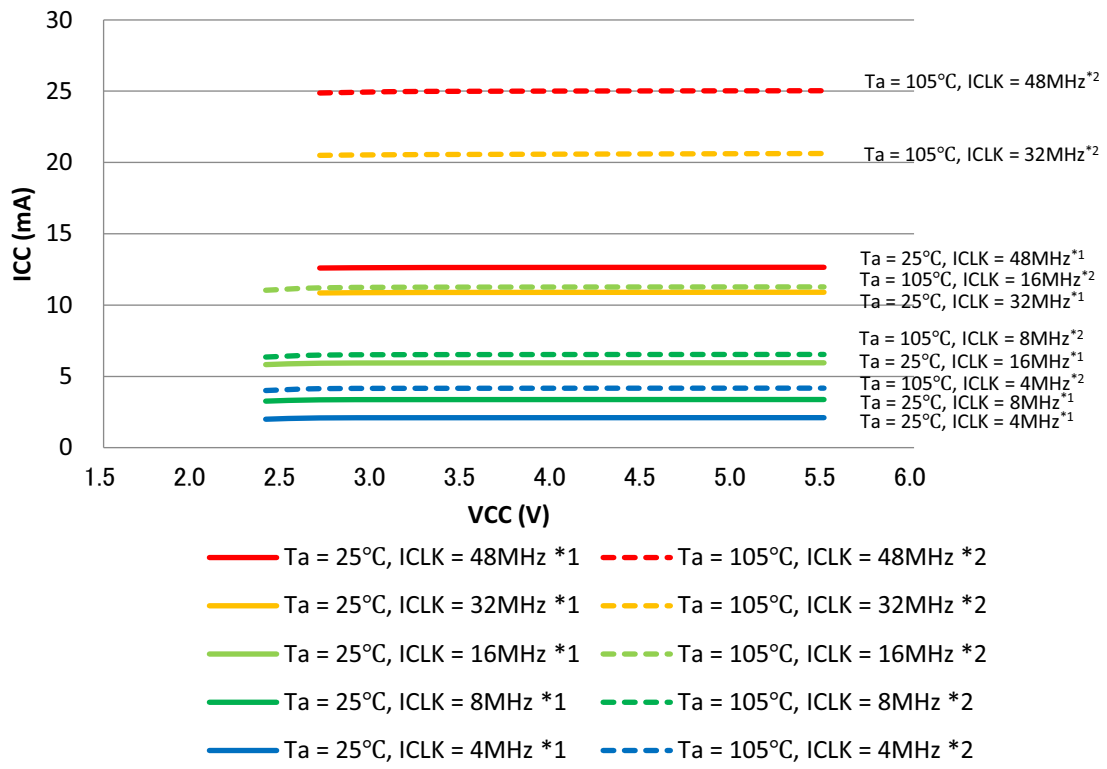
Conditions: VCC = AVCC0 = AVCC1 = 1.6 to 5.5 V

Parameter					Symbol	Typ*10	Max	Unit	Test Conditions
Supply current*1	High-speed mode*2	Normal mode	All peripheral clocks disabled, while (1) code executing from flash*5	ICLK = 48 MHz	I_{CC}	5.2	-	mA	*7, *11
				ICLK = 32 MHz		3.8	-		
				ICLK = 16 MHz		2.3	-		
				ICLK = 8 MHz		1.6	-		
			All peripheral clocks disabled, CoreMark code executing from flash*5	ICLK = 48 MHz		12.1	-		
				ICLK = 32 MHz		8.3	-		
				ICLK = 16 MHz		4.6	-		
				ICLK = 8 MHz		2.8	-		
			All peripheral clocks enabled, while (1) code executing from flash*5	ICLK = 48 MHz		12.6	-		*9, *11
				ICLK = 32 MHz		10.9	-		*8, *11
				ICLK = 16 MHz		5.9	-		
				ICLK = 8 MHz		3.4	-		
			All peripheral clocks enabled, code executing from flash*5	ICLK = 48 MHz		-	28.5		*9, *11
		Sleep mode	All peripheral clocks disabled*5	ICLK = 48 MHz		2.7	-		*7
				ICLK = 32 MHz		2.1	-		
				ICLK = 16 MHz		1.5	-		
				ICLK = 8 MHz		1.1	-		
			All peripheral clocks enabled*5	ICLK = 48 MHz		9.8	-		*9
				ICLK = 32 MHz		8.9	-		*8
				ICLK = 16 MHz		5.0	-		
				ICLK = 8 MHz		2.9	-		
		Increase during BGO operation*6				2.5	-		-
	Middle-speed mode*2	Normal mode	All peripheral clocks disabled, while (1) code executing from flash*5	ICLK = 12 MHz	I_{CC}	1.6	-	mA	*7, *11
				ICLK = 8 MHz		1.3	-		
			All peripheral clocks disabled, CoreMark code executing from flash*5	ICLK = 12 MHz		3.4	-		
				ICLK = 8 MHz		2.6	-		
			All peripheral clocks enabled, while (1) code executing from flash*5	ICLK = 12 MHz		4.3	-		*8, *11
				ICLK = 8 MHz		3.1	-		
			All peripheral clocks enabled, code executing from flash*5	ICLK = 12 MHz		-	12.6		
		Sleep mode	All peripheral clocks disabled*5	ICLK = 12 MHz		1.0	-		*7
				ICLK = 8 MHz		0.9	-		
			All peripheral clocks enabled*5	ICLK = 12 MHz		3.6	-		*8
				ICLK = 8 MHz		2.7	-		
		Increase during BGO operation*6				2.5	-		-

Table 47.10 Operating and standby current (1) (2 of 2)
Conditions: VCC = AVCC0 = AVCC1 = 1.6 to 5.5 V

Parameter					Symbol	Typ*10	Max	Unit	Test Conditions
Supply current*1	Low-speed mode*3	Normal mode	All peripheral clocks disabled, while (1) code executing from flash*5	ICLK = 1 MHz	I_{CC}	0.3	-	mA	*7, *11
			All peripheral clocks disabled, CoreMark code executing from flash*5	ICLK = 1 MHz		0.4	-		
			All peripheral clocks enabled, while (1) code executing from flash*5	ICLK = 1 MHz		0.5	-		*8, *11
			All peripheral clocks enabled, code executing from flash*5	ICLK = 1 MHz		-	2.5		
		Sleep mode	All peripheral clocks disabled*5	ICLK = 1 MHz		0.2	-		*7
			All peripheral clocks enabled*5	ICLK = 1 MHz		0.4	-		*8
	Low-voltage mode*3	Normal mode	All peripheral clocks disabled, while (1) code executing from flash*5	ICLK = 4 MHz	I_{CC}	1.5	-	mA	*7, *11
			All peripheral clocks disabled, CoreMark code executing from flash*5	ICLK = 4 MHz		2.2	-		
			All peripheral clocks enabled, while (1) code executing from flash*5	ICLK = 4 MHz		2.5	-		*8, *11
			All peripheral clocks enabled, code executing from flash*5	ICLK = 4 MHz		-	7.0		
		Sleep mode	All peripheral clocks disabled*5	ICLK = 4 MHz		1.3	-		*7
			All peripheral clocks enabled*5	ICLK = 4 MHz		2.3	-		*8
	Subosc-speed mode*4	Normal mode	All peripheral clocks disabled, while (1) code executing from flash*5	ICLK = 32.768 kHz	I_{CC}	6.5	-	μA	*8, *11
			All peripheral clocks enabled, while (1) code executing from flash*5	ICLK = 32.768 kHz		12.1	-		
			All peripheral clocks enabled, code executing from flash*5	ICLK = 32.768 kHz		-	190.0		
		Sleep mode	All peripheral clocks disabled*5	ICLK = 32.768 kHz		4.5	-		*8
			All peripheral clocks enabled*5	ICLK = 32.768 kHz		10.2	-		*8

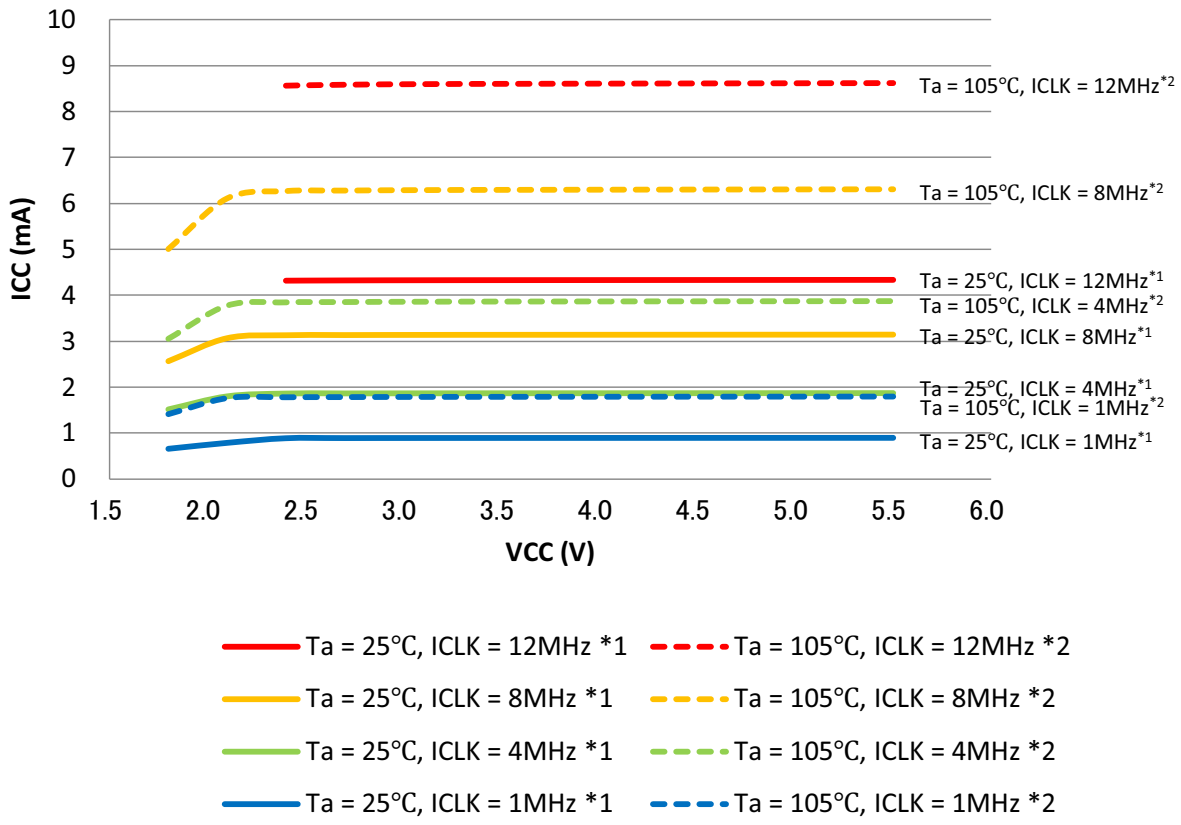
Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.
Note 2. The clock source is HOCO.
Note 3. The clock source is MOCO.
Note 4. The clock source is the sub-clock oscillator.
Note 5. This does not 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, PCLKB, and PCLKD are set to divided by 64.
Note 8. FCLK, PCLKB, and PCLKD are the same frequency as that of ICLK.
Note 9. FCLK and PCLKB are set to be divided by 2 and PCLKD is the same frequency as that of ICLK.
Note 10. VCC = 3.3 V.
Note 11. The flash cache is operating.



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.

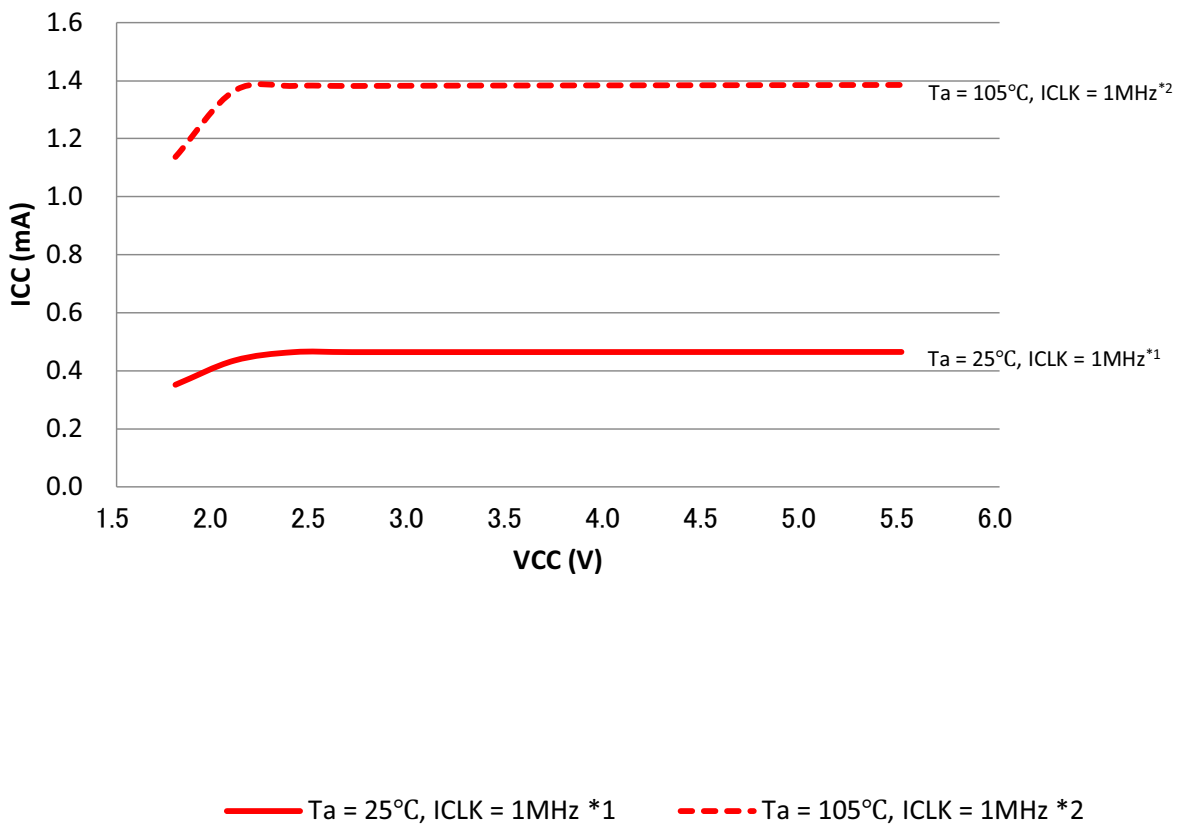
Figure 47.17 Voltage dependency in high-speed operating mode (reference data)



Note 1. All peripheral operations except any BGO operation are operating normally. This is the average of the actual measurements of the sample cores during product evaluation.

Note 2. All peripheral operations except any BGO operation are operating at maximum. This is the average of the actual measurements for the upper limit samples during product evaluation.

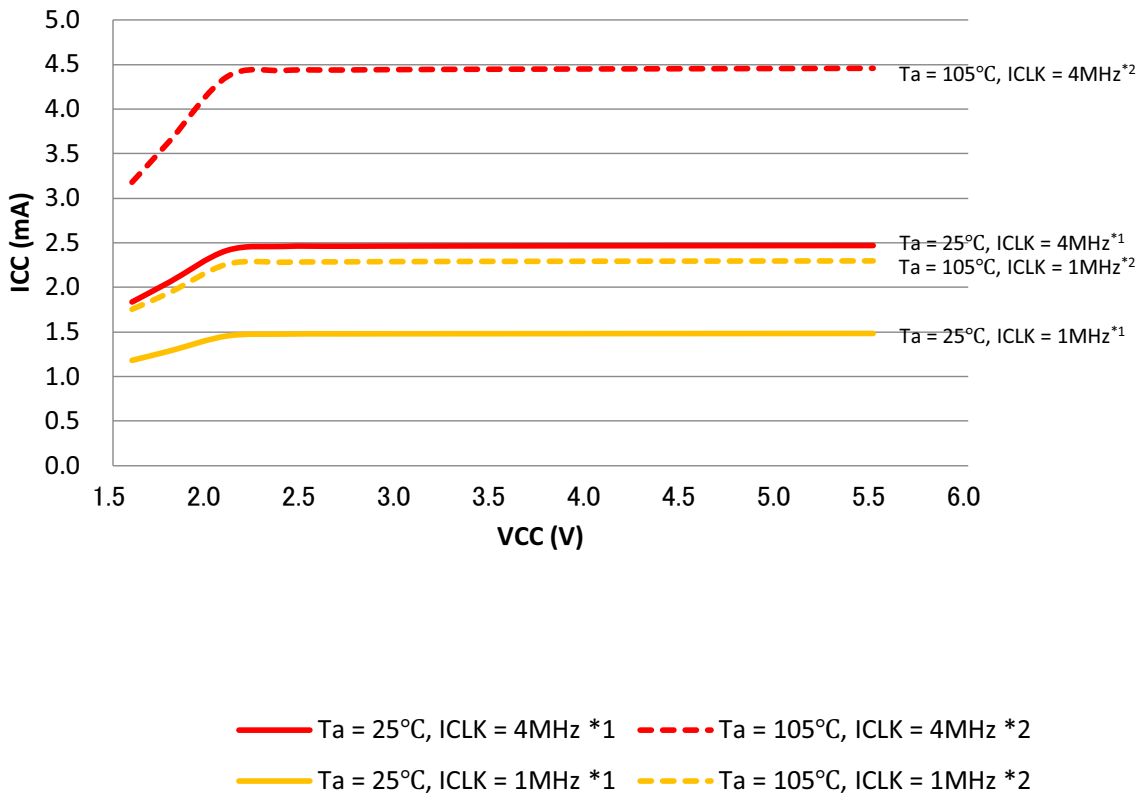
Figure 47.18 Voltage dependency in middle-speed operating mode (reference data)



Note 1. All peripheral operations except any BGO operation are operating normally. This is the average of the actual measurements of the sample cores during product evaluation.

Note 2. All peripheral operations except any BGO operation are operating at maximum. This is the average of the actual measurements for the upper limit samples during product evaluation.

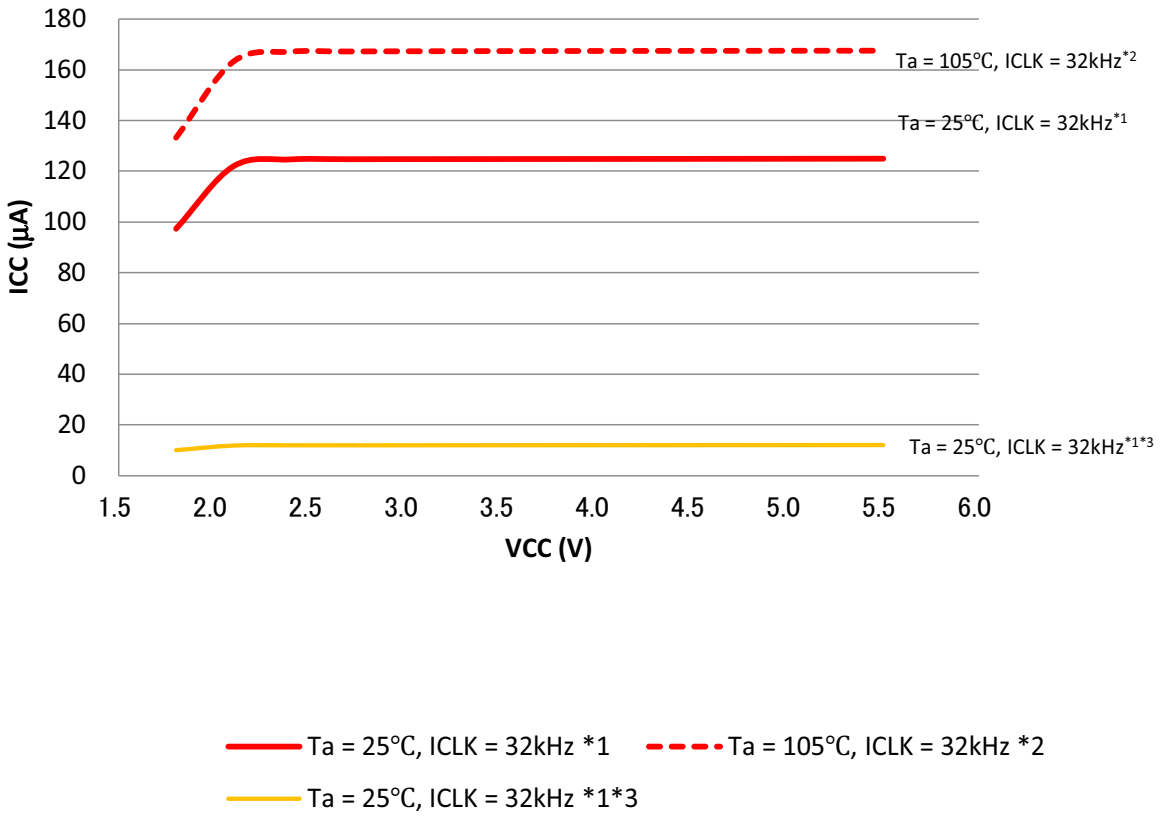
Figure 47.19 Voltage dependency in low-speed operating mode (reference data)



Note 1. All peripheral operations except any BGO operation are operating normally. This is the average of the actual measurements of the sample cores during product evaluation.

Note 2. All peripheral operations except any BGO operation are operating at maximum. This is the average of the actual measurements for the upper limit samples during product evaluation.

Figure 47.20 Voltage dependency in low-voltage operating mode (reference data)



Note 1. All peripheral operations except any BGO operation are operating normally. This is the average of the actual measurements of the sample cores during product evaluation.
Note 2. All peripheral operations except any BGO operation are operating at maximum. This is the average of the actual measurements for the upper limit samples during product evaluation.
Note 3. MOCO and DAC are stopped.

Figure 47.21 Voltage dependency in subosc-speed operating mode (reference data)

Table 47.11 Operating and standby current (2)
Conditions: VCC = AVCC0 = AVCC1 = 1.6 to 5.5 V

Parameter			Symbol	Typ*3	Max	Unit	Test conditions
Supply current*1	Software Standby mode*2	T_a = 25°C	I_{CC}	0.5	2.0	µA	-
		T_a = 55°C		0.8	7.0		
		T_a = 85°C		1.8	17.0		
		T_a = 105°C		4.4	45.0		
	Increment for RTC operation with low-speed on-chip oscillator*4			0.4	-		-
	Increment for RTC operation with sub-clock oscillator*4			0.5	-		SOMCR.SODRV[1:0] are 11b (Low power mode 3)
				1.3	-		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 MOS transistors are in the off state.
Note 2. The IWDT and LVD are not operating.
Note 3. VCC = 3.3 V.
Note 4. Includes the low-speed on-chip oscillator or sub-oscillation circuit current.

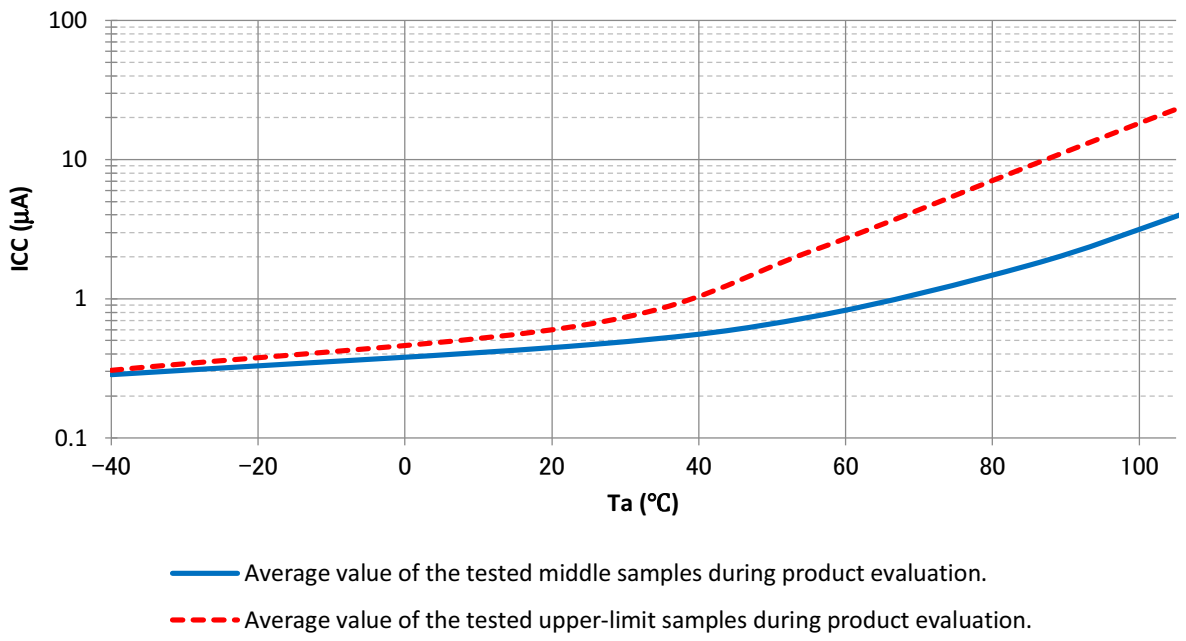
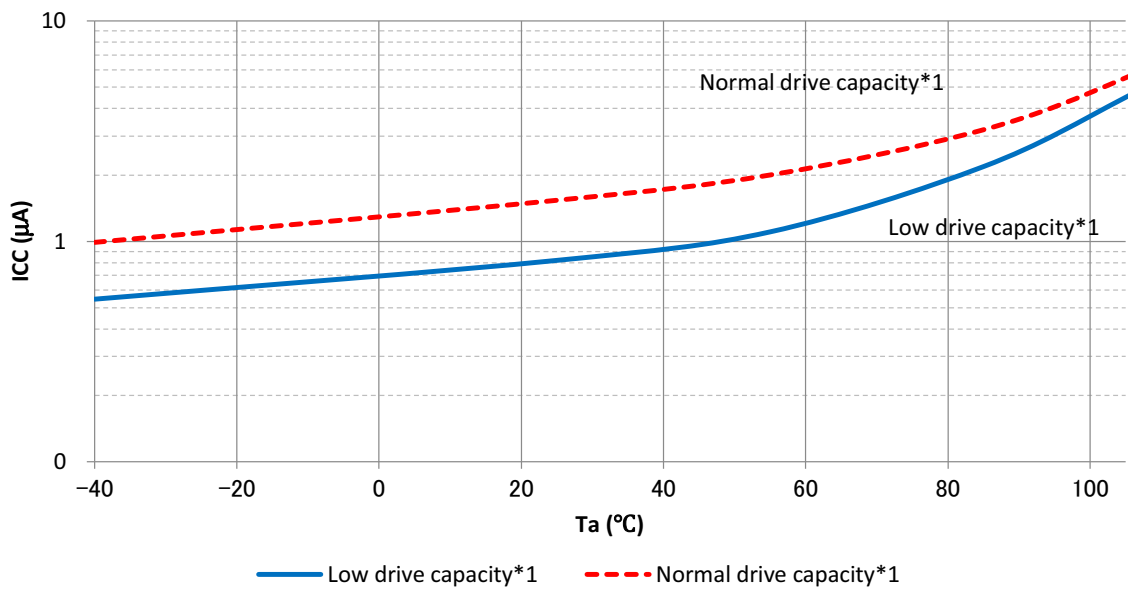


Figure 47.22 Temperature dependency in Software Standby mode (reference data)



Note: Average value of the tested middle samples during product evaluation.

Figure 47.23 Temperature dependency of RTC operation (reference data)

Table 47.12 Operating and standby current (3)
Conditions: VCC = AVCC0 = AVCC1 = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Analog power supply current	During 16-bit A/D conversion		I_{AVCC0}	-	-	1.5	mA	-
	During 8-bit D/A conversion (per channel) *1			-	-	1.6	mA	-
	During 12-bit D/A conversion (per channel) *1			-	-	0.9	mA	-
	Waiting for 16-bit A/D, 8-bit D/A and 12-bit D/A conversion (all units) *5			-	-	2.0	μA	-
	During 24-bit sigma-delta A/D conversion (at normal mode)		I_{AVCC1}	-	-	1.29	mA	-
	During 24-bit sigma-delta A/D conversion (at low-power conversion)			-	-	1.06	mA	G_{SET1} = 8, or G_{TOTAL} = 24,32
				-	-	0.9	mA	G_{SET1}, G_{TOTAL} = the others
	Waiting for 24-bit sigma-delta A/D conversion*6			-	-	1.0	μA	-
Reference power supply current	During 16-bit A/D conversion		I_{REFH0}	-	-	80	μA	-
	Waiting for 16-bit A/D conversion			-	-	60	nA	-
	During 12-bit D/A conversion		I_{REFH}	-	-	650	μA	-
	Waiting for 12-bit D/A conversion			-	-	100	nA	-
	During 24-bit sigma-delta A/D conversion		I_{REFI}	-	-	30	μA	External VREF mode
Temperature Sensor (TSN) operating current			I_{TNS}	-	75	-	μA	-
Low-power Analog Comparator (ACMPLP) operating current	Window comparator (high-speed mode)		I_{CMPLP}	-	15	-	μA	-
	Comparator (high-speed mode)			-	10	-	μA	-
	Comparator (low-speed mode)			-	2	-	μA	-
High-speed analog comparator (ACMPHS) operating current			I_{CPMHS}	-	70	100	μA	AVCC0 ≥ 2.7 V
Operational Amplifier (OPAMP) operating current	Low power mode	1 unit operating	I_{AMP}	-	10	16	μA	-
		2 unit operating		-	19	30	μA	-
		3 unit operating		-	28	44	μA	-
	Middle speed mode	1 unit operating		-	280	360	μA	-
		2 unit operating		-	530	690	μA	-
		3 unit operating		-	770	1020	μA	-
	High speed mode	1 unit operating		-	0.74	0.91	mA	-
		2 unit operating		-	1.41	1.74	mA	-
		3 unit operating		-	2.07	2.57	mA	-
Internal reference voltage for ADC16 operating current			$I_{VREFADC}$	-	65	130	μA	-
USBFS operating current	During USB communication under the following settings and conditions: • Function controller is in Full-Speed mode and - Bulk OUT transfer is (64 bytes) × 1 - Bulk IN transfer is (64 bytes) × 1 • Host device is connected by a 1-meter USB cable from the USB port.		I_{USBF}*2	-	3.6 (VCC) 1.1 (VCC_USB)*4	-	mA	-
	During suspended state under the following setting and conditions: • Function controller is in Full-Speed mode (the USB_DP pin is pulled up) • Software Standby mode • Host device is connected through 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 is 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 in the suspended state.
Note 4. When VCC = VCC_USB = 3.3 V.
Note 5. When the MCU is in Software Standby mode or the MSTPCRD.MSTPD16 (ADC160 module-stop bit) is in the module-stop

state.

Note 6. When the MCU is in the MSTPCRD.MSTPD17 (SDADC24 module-stop bit) is in the module-stop state.

47.2.10 VCC Rise and Fall Gradient and Ripple Frequency

Table 47.13 Rise and fall gradient characteristics

Conditions: VCC = AVCC0 = AVCC1 = 0 to 5.5 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Power-on VCC rising gradient	Voltage monitor 0 reset disabled at startup	SrVCC	0.02	-	2	ms/V	-
	Voltage monitor 0 reset enabled at startup*1, *2				-		
	SCI/USB boot mode*2				2		

Note 1. When OFS1.LVDAS = 0.

Note 2. At boot mode, the reset from voltage monitor 0 is disabled regardless of the value of OFS1.LVDAS bit.

Table 47.14 Rising and falling gradient and ripple frequency characteristics

Conditions: VCC = AVCC0 = AVCC1 = 1.6 to 5.5 V

The ripple voltage must meet the allowable ripple frequency $f_{r(VCC)}$ within the range between the VCC upper limit (5.5 V) and lower limit (1.6 V).

When the VCC change exceeds VCC ± 10%, the allowable voltage change rising and falling gradient dt/dVCC must be met.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Allowable ripple frequency	$f_{r(VCC)}$	-	-	10	kHz	Figure 47.24 $V_{r\ (VCC)} \leq$ VCC × 0.2
		-	-	1	MHz	Figure 47.24 $V_{r\ (VCC)} \leq$ VCC × 0.08
		-	-	10	MHz	Figure 47.24 $V_{r\ (VCC)} \leq$ VCC × 0.06
Allowable voltage change rising and falling gradient	dt/dVCC	1.0	-	-	ms/V	When VCC change exceeds VCC ± 10%

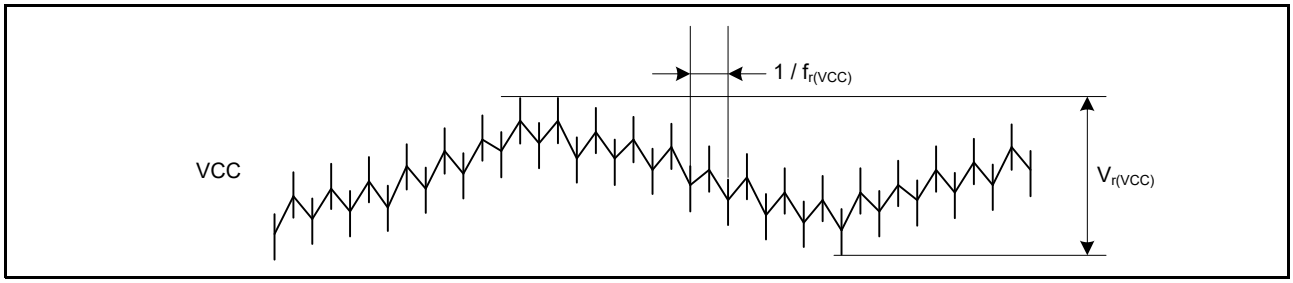


Figure 47.24 Ripple waveform

47.3 AC Characteristics

47.3.1 Frequency

Table 47.15 Operation frequency in high-speed operating mode
Conditions: VCC = AVCC0 = AVCC1 = 2.4 to 5.5 V

Parameter			Symbol	Min	Typ	Max*7	Unit
Operation frequency	System clock (ICLK)*6	2.7 to 5.5 V	f	0.032768	-	48	MHz
		2.4 to 2.7 V		0.032768	-	16	
	FlashIF clock (FCLK)*1,*2,*6	2.7 to 5.5 V		0.032768	-	32	
		2.4 to 2.7 V		0.032768	-	16	
	Peripheral module clock (PCLKB)*5,*6	2.7 to 5.5 V		-	-	32	
		2.4 to 2.7 V		-	-	16	
	Peripheral module clock (PCLKD)*3,*6	2.7 to 5.5 V		-	-	64*4	
		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% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.
Note 3. The lower-limit frequency of PCLKD is 1 MHz when the ADC16 is in use.
Note 4. The upper-limit frequency of PCLKD is 32 MHz when the ADC16 is in use.
Note 5. The lower-limit frequency of PCLKB is 1 MHz when the SDADC24 is in use.
Note 6. See section 9, Clock Generation Circuit for the relationship of frequencies between ICLK, PCLKB, PCLKD, and FCLK.
Note 7. The maximum value of operation frequency does not include internal oscillator errors. For details on the range for guaranteed operation, see Table 47.20, Clock timing.

Table 47.16 Operation frequency in middle-speed operating mode
Conditions: VCC = AVCC0 = AVCC1 = 1.8 to 5.5 V

Parameter			Symbol	Min	Typ	Max*6	Unit
Operation frequency	System clock (ICLK)*5	2.7 to 5.5 V	f	0.032768	-	12	MHz
		2.4 to 2.7 V		0.032768	-	12	
		1.8 to 2.4 V		0.032768	-	8	
	FlashIF clock (FCLK)*1,*2,*5	2.7 to 5.5 V		0.032768	-	12	
		2.4 to 2.7 V		0.032768	-	12	
		1.8 to 2.4 V		0.032768	-	8	
	Peripheral module clock (PCLKB)*4,*5	2.7 to 5.5 V		-	-	12	
		2.4 to 2.7 V		-	-	12	
		1.8 to 2.4 V		-	-	8	
	Peripheral module clock (PCLKD)*3,*5	2.7 to 5.5 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 PCLKD is 1 MHz when the ADC16 is in use.
Note 4. The lower-limit frequency of PCLKB is 1 MHz when the SDADC24 is in use.
Note 5. See section 9, Clock Generation Circuit for the relationship of frequencies between ICLK, PCLKB, PCLKD, and

FCLK.

Note 6. The maximum value of operation frequency does not include internal oscillator errors. For details on the range for guaranteed operation, see Table 47.20, Clock timing.

Table 47.17 Operation frequency in low-speed operating mode

Conditions: VCC = AVCC0 = AVCC1 = 1.8 to 5.5 V

Parameter			Symbol	Min	Typ	Max*6	Unit
Operation frequency	System clock (ICLK)*5	1.8 to 5.5 V	f	0.032768	-	1	MHz
	FlashIF clock (FCLK) *1,*2,*5	1.8 to 5.5 V		0.032768	-	1	
	Peripheral module clock (PCLKB)*4,*5	1.8 to 5.5 V		-	-	1	
	Peripheral module clock (PCLKD)*3,*5	1.8 to 5.5 V		-	-	1	

Note 1. The lower-limit frequency of FCLK is 1 MHz while programming or erasing the flash memory.
Note 2. The 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 PCLKD is 1 MHz when the ADC16 is in use.
Note 4. The lower-limit frequency of PCLKB is 1 MHz when the SDADC24 is in use.
Note 5. See section 9, Clock Generation Circuit for the relationship of frequencies between ICLK, PCLKB, PCLKD, and FCLK.
Note 6. The maximum value of operation frequency does not include internal oscillator errors. For details on the range for guaranteed operation, see Table 47.20, Clock timing.

Table 47.18 Operation frequency in low-voltage operating mode

Conditions: VCC = AVCC0 = AVCC1 = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max*6	Unit
Operation frequency	System clock (ICLK)*5	1.6 to 5.5 V	f	0.032768	-	4	MHz
	FlashIF clock (FCLK)*1,*2,*5	1.6 to 5.5 V		0.032768	-	4	
	Peripheral module clock (PCLKB)*4,*5	1.6 to 5.5 V		-	-	4	
	Peripheral module clock (PCLKD)*3,*5	1.6 to 5.5 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% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.
Note 3. The lower-limit frequency of PCLKD is 1 MHz when the ADC16 is in use.
Note 4. The lower-limit frequency of PCLKB is 1 MHz when the SDADC24 is in use.
Note 5. See section 9, Clock Generation Circuit for the relationship of frequencies between ICLK, PCLKB, PCLKD, and FCLK.
Note 6. The maximum value of operation frequency does not include internal oscillator errors. For details on the range for guaranteed operation, see Table 47.20, Clock timing.

Table 47.19 Operation frequency in Subosc-speed operating mode

Conditions: VCC = AVCC0 = AVCC1 = 1.8 to 5.5 V

Parameter			Symbol	Min	Typ	Max	Unit
Operation frequency	System clock (ICLK)*4	1.8 to 5.5 V	f	27.8528	32.768	37.6832	kHz
	FlashIF clock (FCLK)*1,*4	1.8 to 5.5 V		27.8528	32.768	37.6832	
	Peripheral module clock (PCLKB)*3,*4	1.8 to 5.5 V		-	-	37.6832	
	Peripheral module clock (PCLKD)*2,*4	1.8 to 5.5 V		-	-	37.6832	

Note 1. Programming and erasing the flash memory is not possible.
Note 2. The ADC16 cannot be used.
Note 3. The SDADC24 cannot be used.
Note 4. See section 9, Clock Generation Circuit for the relationship of frequencies between ICLK, PCLKB, PCLKD, and FCLK.

47.3.2 Clock Timing

Table 47.20 Clock timing (1 of 2)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
EXTAL external clock input cycle time		t_{Xcyc}	50	-	-	ns	Figure 47.25
EXTAL external clock input high pulse width		t_{XH}	20	-	-	ns	
EXTAL external clock input low pulse width		t_{XL}	20	-	-	ns	
EXTAL external clock rising time		t_{Xr}	-	-	5	ns	
EXTAL external clock falling time		t_{Xf}	-	-	5	ns	
EXTAL external clock input wait time*1		t_{EXWT}	0.3	-	-	μs	-
EXTAL external clock input frequency		f_{EXTAL}	-	-	20	MHz	2.4 ≤ VCC ≤ 5.5
			-	-	8		1.8 ≤ VCC < 2.4
			-	-	1		1.6 ≤ VCC < 1.8
Main clock oscillator oscillation frequency		f_{MAIN}	1	-	20	MHz	2.4 ≤ VCC ≤ 5.5
			1	-	8		1.8 ≤ VCC < 2.4
			1	-	4		1.6 ≤ VCC < 1.8
LOCO clock oscillation frequency		f_{LOCO}	27.8528	32.768	37.6832	kHz	-
LOCO clock oscillation stabilization time		t_{LOCO}	-	-	100	μs	Figure 47.26
IWDT-dedicated clock oscillation frequency		f_{ILOCO}	12.75	15	17.25	kHz	-
MOCO clock oscillation frequency		f_{MOCO}	6.8	8	9.2	MHz	-
MOCO clock oscillation stabilization time		t_{MOCO}	-	-	1	μs	-
HOCO clock oscillation frequency		f_{HOCO24}	23.64	24	24.36	MHz	Ta = -40 to -20°C 1.8 ≤ VCC ≤ 5.5
			22.68	24	25.32		Ta = -40 to 85°C 1.6 ≤ VCC < 1.8
			23.76	24	24.24		Ta = -20 to 85°C 1.8 ≤ VCC ≤ 5.5
			23.52	24	24.48		Ta = 85 to 105°C 2.4 ≤ VCC ≤ 5.5
		f_{HOCO32}	31.52	32	32.48		Ta = -40 to -20°C 1.8 ≤ VCC ≤ 5.5
			30.24	32	33.76		Ta = -40 to 85°C 1.6 ≤ VCC < 1.8
			31.68	32	32.32		Ta = -20 to 85°C 1.8 ≤ VCC ≤ 5.5
			31.36	32	32.64		Ta = 85 to 105°C 2.4 ≤ VCC ≤ 5.5
		f_{HOCO48}*3	47.28	48	48.72		Ta = -40 to -20°C 1.8 ≤ VCC ≤ 5.5
			47.52	48	48.48		Ta = -20 to 85°C 1.8 ≤ VCC ≤ 5.5
			47.04	48	48.96		Ta = 85 to 105°C 2.4 ≤ VCC ≤ 5.5
		f_{HOCO64}*4	63.04	64	64.96		Ta = -40 to -20°C 2.4 ≤ VCC ≤ 5.5
			63.36	64	64.64		Ta = -20 to 85°C 2.4 ≤ VCC ≤ 5.5
			62.72	64	65.28		Ta = 85 to 105°C 2.4 ≤ VCC ≤ 5.5
HOCO clock oscillation stabilization time*5, *6	Except low-voltage mode	t_{HOCO24} t_{HOCO32}	-	-	37.1	μs	Figure 47.27
		t_{HOCO48}	-	-	43.3		
		t_{HOCO64}	-	-	80.6		
	Low-voltage mode	t_{HOCO24} t_{HOCO32} t_{HOCO48} t_{HOCO64}	-	-	100.9		
Sub-clock oscillator oscillation frequency		f_{SUB}	-	32.768	-	kHz	-

Table 47.20 Clock timing (2 of 2)

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Sub-clock oscillation stabilization time*2	t_{SUBOSC}	-	0.5	-	s	Figure 47.28

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. After changing the setting of the SOSCCR.SOSTP bit to start sub-clock oscillator operation, only start using the sub-clock oscillator after the sub-clock oscillation stabilization wait time elapsed. Use the oscillator wait time value recommended by the oscillator manufacturer.
Note 3. The 48-MHz HOCO can be used within a VCC range of 1.8 V to 5.5 V.
Note 4. The 64-MHz HOCO can be used within a VCC range of 2.4 V to 5.5 V.
Note 5. This is a characteristic when the HOCOCR.HCSTP bit is cleared to 0 (oscillation) in the MOCO stop state.
When the HOCOCR.HCSTP bit is set to 0 (oscillation) during MOCO oscillation, this specification is shortened by 1 μs.
Note 6. Check OSCSF.HOCOSF to confirm whether stabilization time has elapsed.

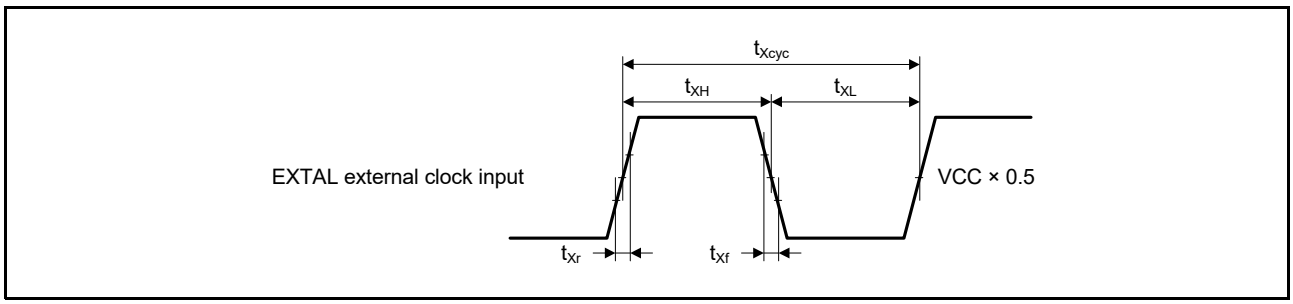


Figure 47.25 EXTAL external clock input timing

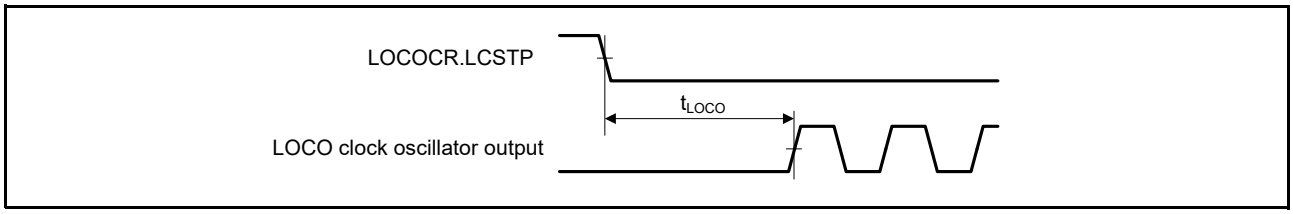


Figure 47.26 LOCO clock oscillation start timing

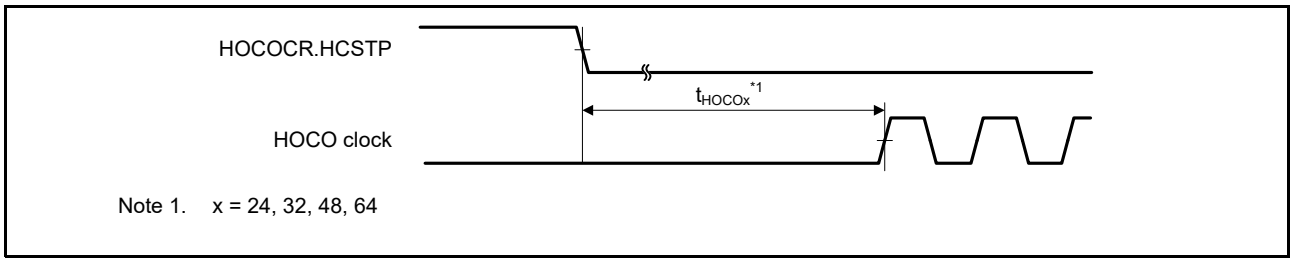


Figure 47.27 HOCO clock oscillation start timing (started by setting the HOCOCR.HCSTP bit)

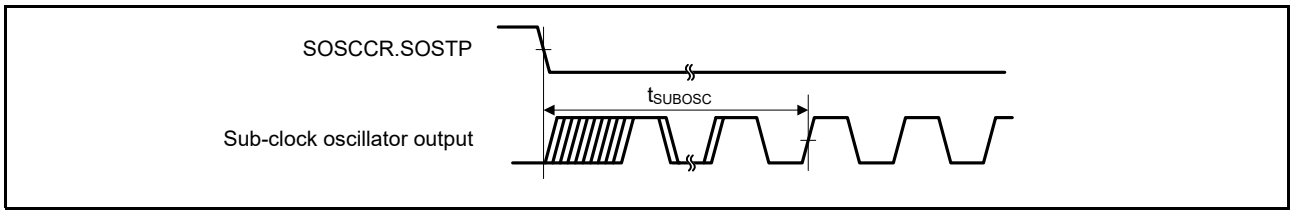


Figure 47.28 Sub-clock oscillation start timing

47.3.3 Reset Timing

Table 47.21 Reset timing

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
RES pulse width	At power-on	t_{RESWP}	3	-	-	ms	Figure 47.29
	Not at power-on	t_{RESW}	30	-	-	μs	Figure 47.30
Wait time after RES cancellation (at power-on)	LVD0 enabled*1	t_{RESWT}	-	0.7	-	ms	Figure 47.29
	LVD0 disabled*2		-	0.3	-		
Wait time after RES cancellation (during powered-on state)	LVD0 enabled*1	t_{RESWT2}	-	0.5	-	ms	Figure 47.30
	LVD0 disabled*2		-	0.1	-		
Wait time after internal reset cancellation (Watchdog timer reset, SRAM parity error reset, SRAM ECC error reset, bus master MPU error reset, bus slave MPU error reset, stack pointer error reset, software reset)	LVD0 enabled*1	t_{RESWT3}	-	0.6	-	ms	Figure 47.31
	LVD0 disabled*2		-	0.15	-		

Note 1. When OFS1.LVDAS = 0.
Note 2. When OFS1.LVDAS = 1.

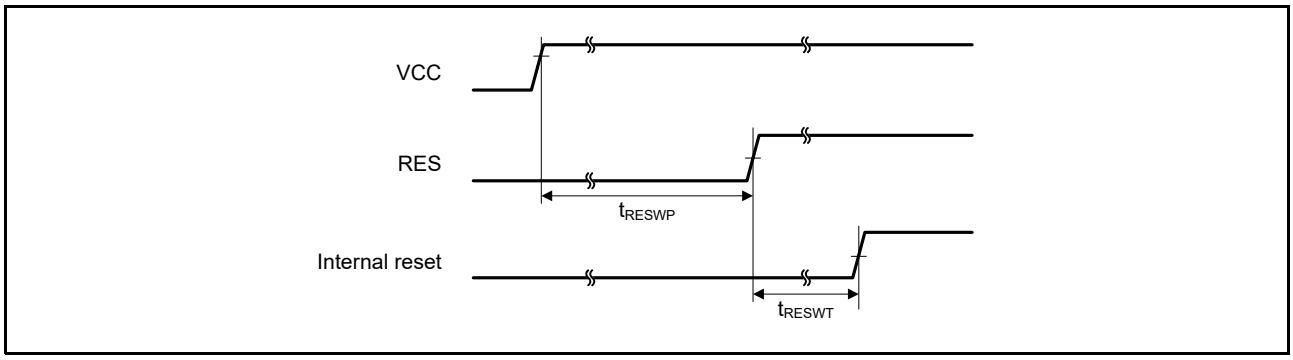


Figure 47.29 Reset input timing at power-on

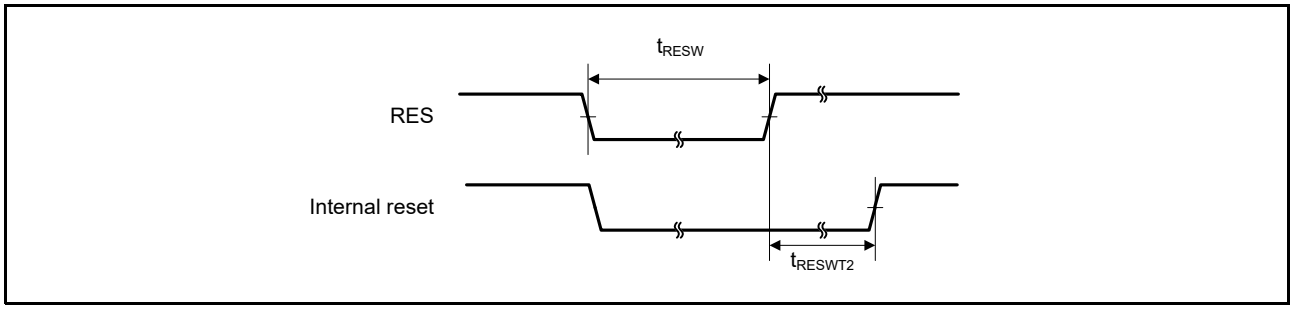


Figure 47.30 Reset input timing (1)

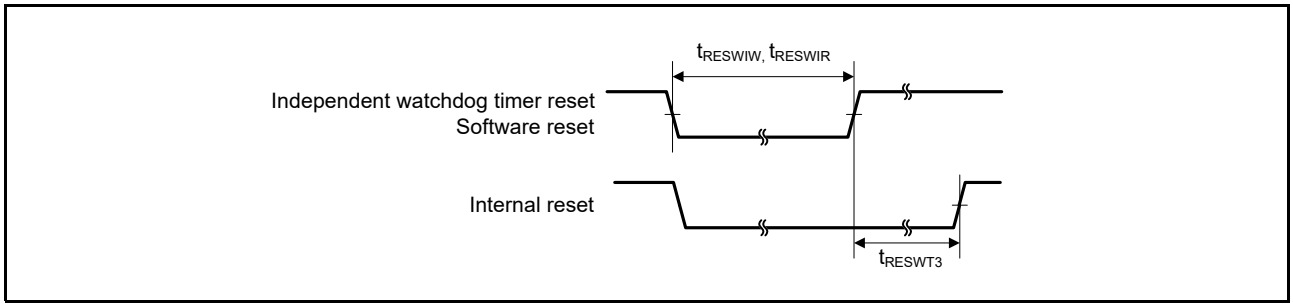


Figure 47.31 Reset input timing (2)

47.3.4 Wakeup Time

Table 47.22 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 47.32
		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 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, 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 47.23 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 47.32
		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 HOCO*4		t_{SBYHO}	-	38	50	μs	
		System clock source is MOCO (8 MHz)		t_{SBYMO}	-	3.5	5.5	μs	

Note 1. The division ratio of ICK, FCK, and PCKx is the minimum division ratio within the allowable frequency range. The recovery time is determined by the system clock source.
Note 2. The Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 05h.
Note 3. The Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 00h.
Note 4. The system clock is 12 MHz.

Table 47.24 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 47.32
		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 (1 MHz)		t_{SBYMO}	-	25	35	μs	

Note 1. The division ratio of ICK, 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 47.25 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 47.32
		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 (4 MHz)		t_{SBYHO}	-	108	130	μs	

Note 1. The division ratio of ICK, 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 47.26 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 47.32
		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 oscillating in Software Standby mode during Subosc-speed mode.

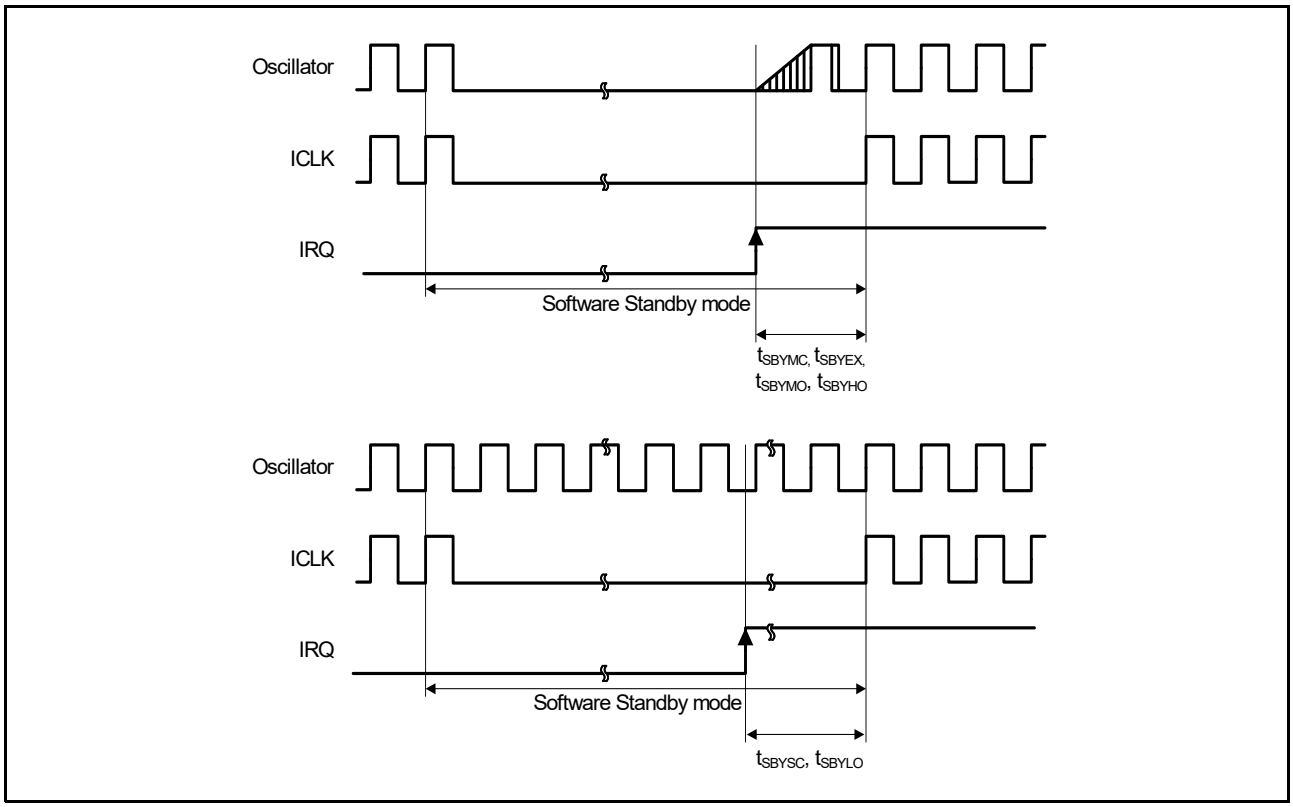
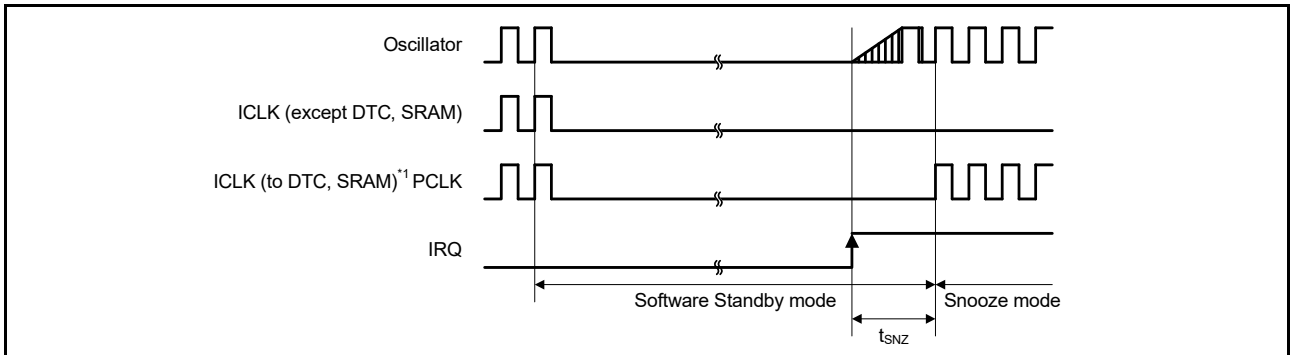


Figure 47.32 Software Standby mode cancellation timing

Table 47.27 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 47.33
	Middle-speed mode System clock source is MOCO (8 MHz)	t_{SNZ}	-	1.3	3.6	μs	
	Low-speed mode System clock source is MOCO (1 MHz)	t_{SNZ}	-	10	13	μs	
	Low-voltage mode System clock source is HOCO (4 MHz)	t_{SNZ}	-	87	110	μs	



Note 1. When SNZCR.SNZDTCEN bit is set to 1, ICLK is supplied to DTC and SRAM.

Figure 47.33 Recovery timing from Software Standby mode to Snooze mode

47.3.5 NMI and IRQ Noise Filter

Table 47.28 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 PCLKB cycle.
Note 2. t_{NMICK} indicates the cycle of the NMI digital filter sampling clock.
Note 3. t_{IRQCK} indicates the cycle of the IRQi digital filter sampling clock (i = 0 to 7).

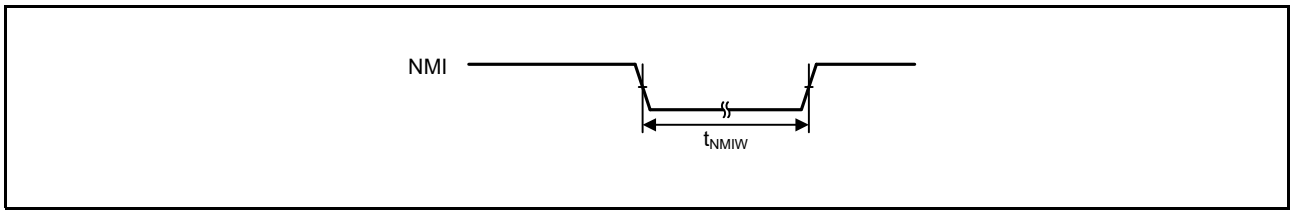


Figure 47.34 NMI interrupt input timing

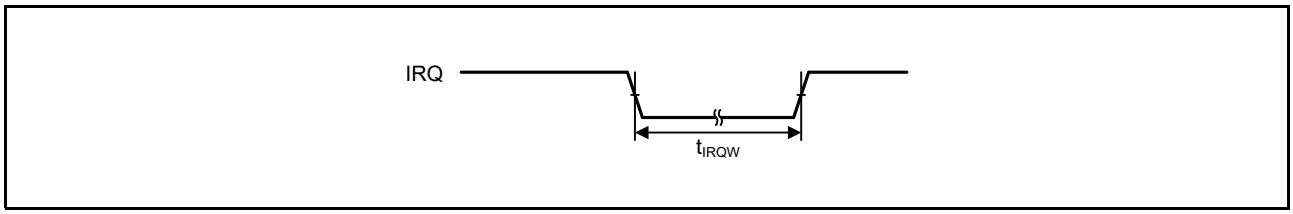


Figure 47.35 IRQ interrupt input timing

47.3.6 I/O Ports, POEG, GPT, AGT, KINT, and ADC16 Trigger Timing

Note:

Table 47.29 I/O Ports, POEG, GPT, AGT, KINT, and ADC16 trigger timing

Parameter			Symbol	Min	Max	Unit	Test conditions
I/O Ports	Input data pulse width		t_{PRW}	1.5	-	t_{Pcyc}	Figure 47.36
POEG	POEG input trigger pulse width		t_{POEW}	3	-	t_{Pcyc}	Figure 47.37
GPT	Input capture pulse width	Single edge	t_{GTICW}	1.5	-	t_{PDcyc}	Figure 47.38
		Dual edge		2.5	-		
AGT	AGTIO, AGTEE input cycle	2.7 V ≤ VCC ≤ 5.5 V	t_{ACYC}*1	250	-	ns	Figure 47.39
		2.4 V ≤ VCC < 2.7 V		500	-	ns	
		1.8 V ≤ VCC < 2.4 V		1000	-	ns	
		1.6 V ≤ VCC < 1.8 V		2000	-	ns	
	AGTIO, AGTEE input high-level width, low-level width	2.7 V ≤ VCC ≤ 5.5 V	t_{ACKWH}, t_{ACKWL}	100	-	ns	
		2.4 V ≤ VCC < 2.7 V		200	-	ns	
		1.8 V ≤ VCC < 2.4 V		400	-	ns	
		1.6 V ≤ VCC < 1.8 V		800	-	ns	
	AGTIO, AGTO, AGTOA, AGTOB output cycle	2.7 V ≤ VCC ≤ 5.5 V	t_{ACYC2}	62.5	-	ns	Figure 47.39
		2.4 V ≤ VCC < 2.7 V		125	-	ns	
		1.8 V ≤ VCC < 2.4 V		250	-	ns	
		1.6 V ≤ VCC < 1.8 V		500	-	ns	
ADC16	16-bit A/D converter trigger input pulse width		t_{TRGW}	1.5	-	t_{Pcyc}	Figure 47.40
KINT	KRn (n = 00 to 07) pulse width		t_{KR}	250	-	ns	Figure 47.41

Note: t_{Pcyc}: PCLKB cycle, t_{PDcyc}: PCLKD cycle.

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.

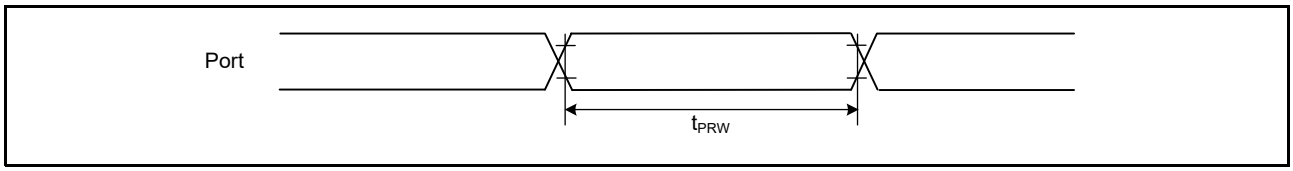


Figure 47.36 I/O ports input timing

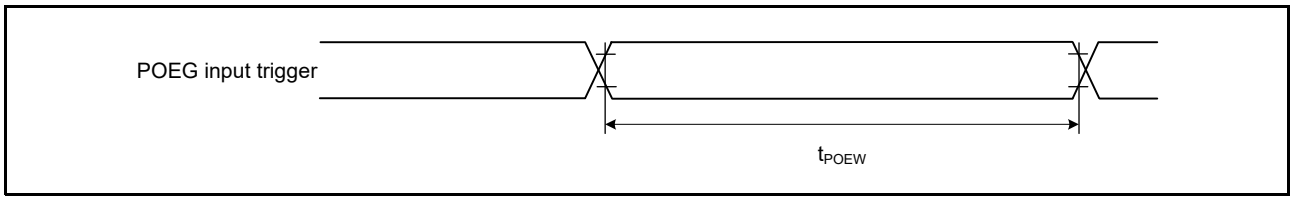


Figure 47.37 POEG input trigger timing

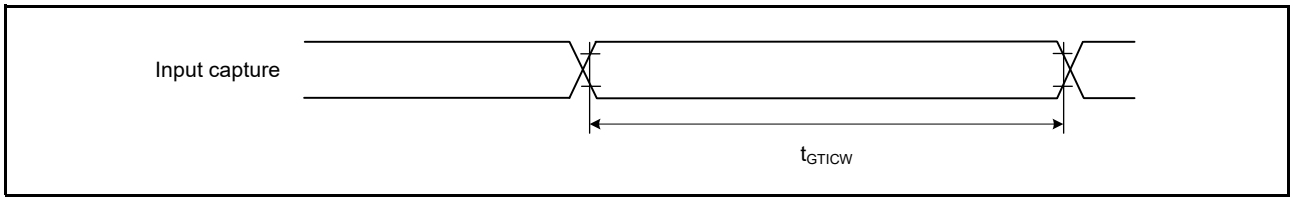


Figure 47.38 GPT input capture timing

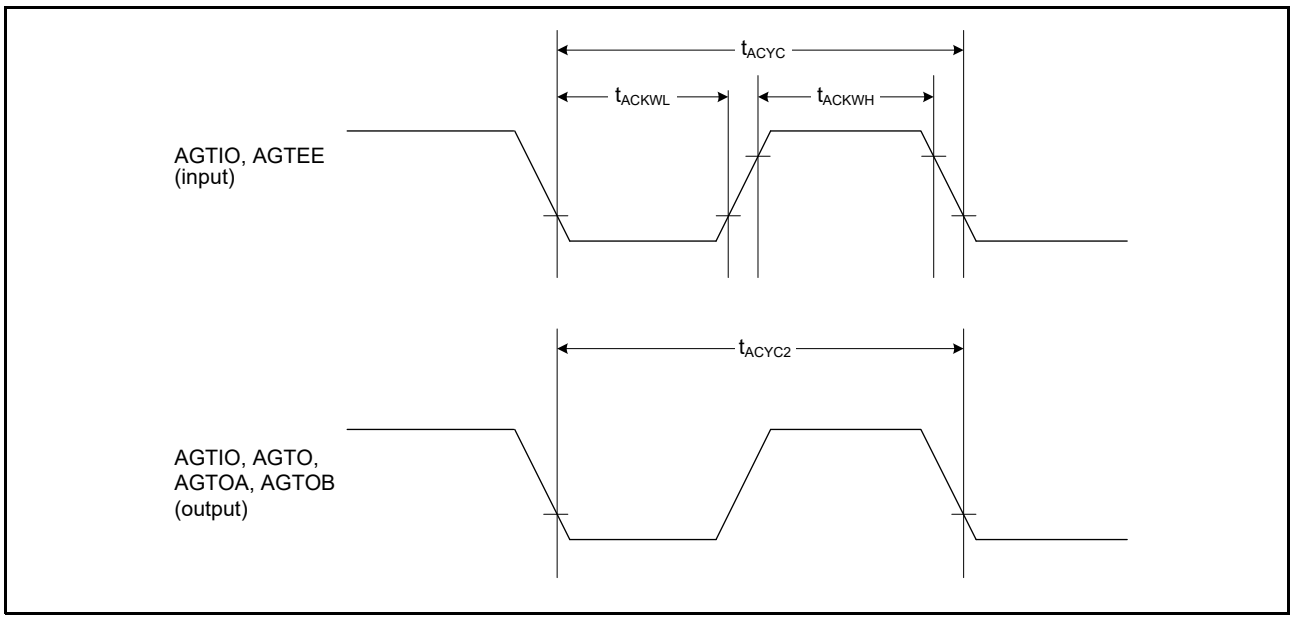


Figure 47.39 AGT I/O timing

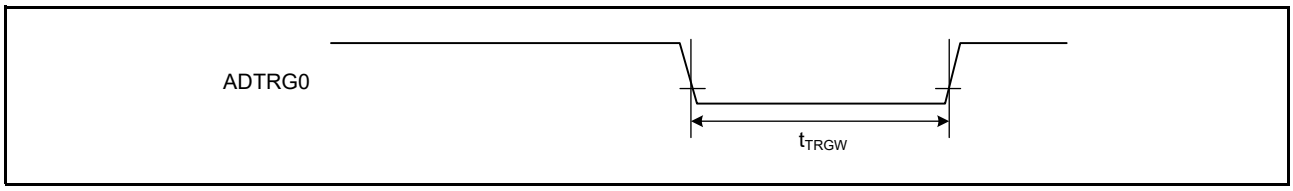


Figure 47.40 ADC16 trigger input timing

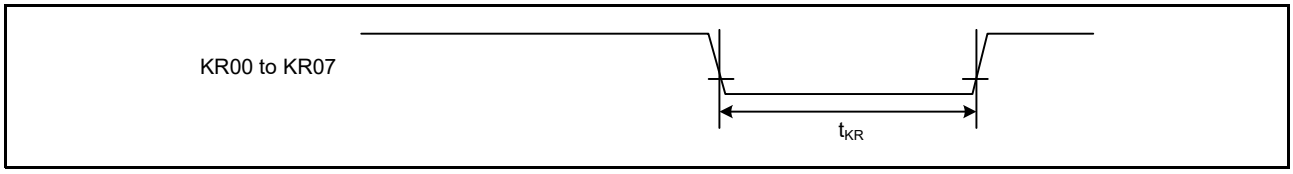


Figure 47.41 Key interrupt input timing

47.3.7 CAC Timing

Table 47.30 CAC timing
Conditions: VCC = AVCC0 = AVCC1 = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
CAC	CACREF input pulse width	t_{Pcyc} *1 ≤ t_{cac} *2	t_{CACREF}	4.5 × t_{cac} + 3 × t_{Pcyc}	-	-	ns	-
		t_{Pcyc} *1 > t_{cac} *2		5 × t_{cac} + 6.5 × t_{Pcyc}	-	-	ns	

Note 1. t_{Pcyc}: PCLKB cycle.

Note 2. t_{cac}: CAC count clock source cycle.

47.3.8 SCI Timing

Table 47.31 SCI timing (1)

Conditions: VCC = AVCC0 = AVCC1 = 1.6 to 5.5 V

Parameter				Symbol	Min	Max	Unit*1	Test conditions
SCI	Input clock cycle	Asynchronous		t_{Scyc}	4	-	t_{Pcyc}	Figure 47.42
		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 ≤ VCC ≤ 5.5 V	t_{SCKr}	-	20	ns	
			1.6 V ≤ VCC < 1.8 V		-	30		
	Output clock fall time		1.8 V ≤ VCC ≤ 5.5 V	t_{SCKf}	-	20	ns	
			1.6 V ≤ VCC < 1.8 V		-	30		
	Transmit data delay (master)	Clock synchronous	1.8 V ≤ VCC ≤ 5.5 V	t_{TXD}	-	40	ns	Figure 47.43
			1.6 V ≤ VCC < 1.8 V		-	45		
	Transmit data delay (slave)	Clock synchronous	2.7 V ≤ VCC ≤ 5.5 V		-	55	ns	
			2.4 V ≤ VCC < 2.7 V		-	60		
			1.8 V ≤ VCC < 2.4 V		-	100		
			1.6 V ≤ VCC < 1.8 V		-	125		
	Receive data setup time (master)	Clock synchronous	2.7 V ≤ VCC ≤ 5.5 V	t_{RXS}	45	-	ns	
			2.4 V ≤ VCC < 2.7 V		55	-		
			1.8 V ≤ VCC < 2.4 V		90	-		
			1.6 V ≤ VCC < 1.8 V		110	-		
	Receive data setup time (slave)	Clock synchronous	2.7 V ≤ VCC ≤ 5.5 V		40	-	ns	
			1.6 V ≤ VCC < 2.7 V		45	-		
	Receive data hold time (master)	Clock synchronous		t_{RXH}	5	-	ns	
	Receive data hold time (slave)	Clock synchronous		t_{RXH}	40	-	ns	

Note 1. t_{Pcyc}: PCLKB cycle.

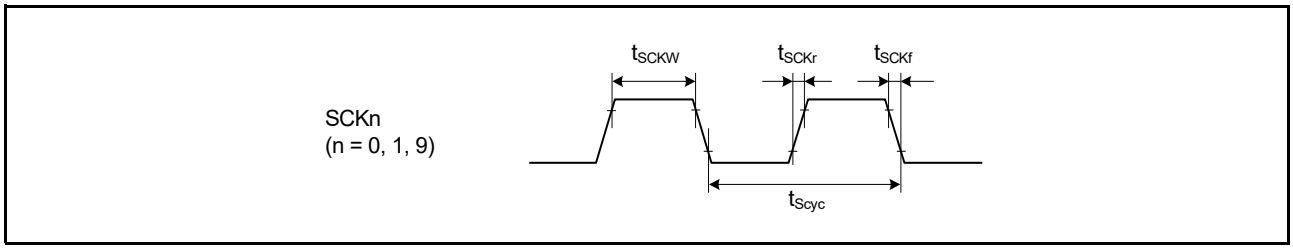


Figure 47.42 SCK clock input timing

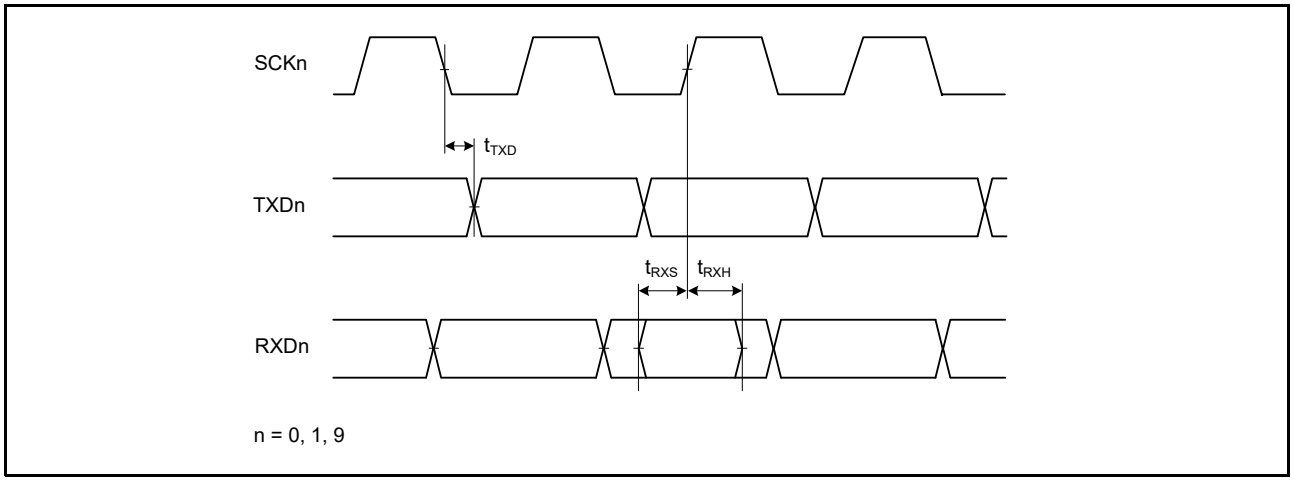


Figure 47.43 SCI input/output timing in clock synchronous mode

Table 47.32 SCI timing (2)

Conditions: VCC = AVCC0 = AVCC1 = 1.6 to 5.5 V

Parameter				Symbol	Min	Max	Unit*1	Test conditions
Simple SPI	SCK clock cycle output (master)			t_{SPcyc}	4	65536	t_{Pcyc}	Figure 47.44
	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 ≤ VCC ≤ 5.5 V	t_{SPCKr}, t_{SPCKf}	-	20	ns	
			1.6 V ≤ VCC < 1.8 V		-	30		
	Data input setup time	Master	2.7 V ≤ VCC ≤ 5.5 V	t_{SU}	45	-	ns	Figure 47.45 to Figure 47.48
			2.4 V ≤ VCC < 2.7 V		55	-		
			1.8 V ≤ VCC < 2.4 V		80	-		
			1.6 V ≤ VCC < 1.8 V		110	-		
		Slave	2.7 V ≤ VCC ≤ 5.5 V		40	-		
			1.6 V ≤ VCC < 2.7 V		45	-		
	Data input hold time	Master		t_H	33.3	-	ns	
		Slave			40	-		
	SS input setup time			t_{LEAD}	1	-	t_{SPcyc}	
	SS input hold time			t_{LAG}	1	-	t_{SPcyc}	
	Data output delay	Master	1.8 V ≤ VCC ≤ 5.5 V	t_{OD}	-	40	ns	
			1.6 V ≤ VCC < 1.8 V		-	50		
		Slave	2.4 V ≤ VCC ≤ 5.5 V		-	65		
			1.8 V ≤ VCC < 2.4 V		-	100		
			1.6 V ≤ VCC < 1.8 V		-	125		
	Data output hold time	Master	2.7 V ≤ VCC ≤ 5.5 V	t_{OH}	-10	-	ns	
			2.4 V ≤ VCC < 2.7 V		-20	-		
			1.8 V ≤ VCC < 2.4 V		-30	-		
			1.6 V ≤ VCC < 1.8 V		-40	-		
		Slave			-10	-		
	Data rise and fall time	Master	1.8 V ≤ VCC ≤ 5.5 V	t_{Dr}, t_{Df}	-	20	ns	
			1.6 V ≤ VCC < 1.8 V		-	30		
		Slave	1.8 V ≤ VCC ≤ 5.5 V		-	20		
			1.6 V ≤ VCC < 1.8 V		-	30		
Simple SPI	Slave access time			t_{SA}	-	6	t_{Pcyc}	Figure 47.48
	Slave output release time			t_{REL}	-	6	t_{Pcyc}	

Note 1. t_{Pcyc}: PCLKB cycle.

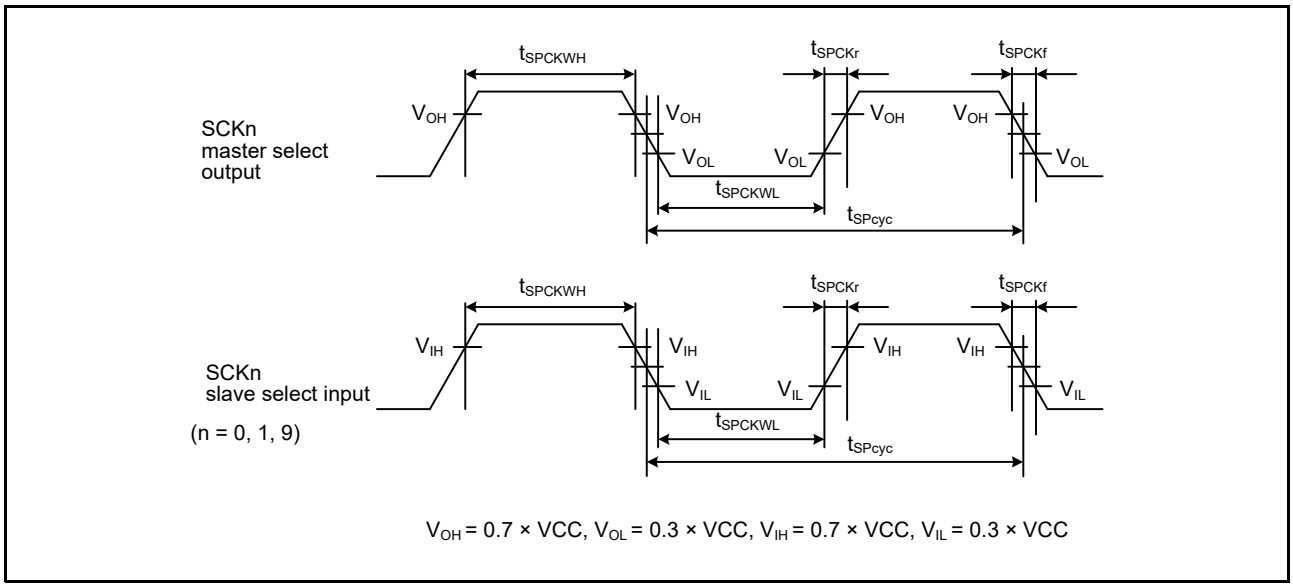


Figure 47.44 SCI simple SPI mode clock timing

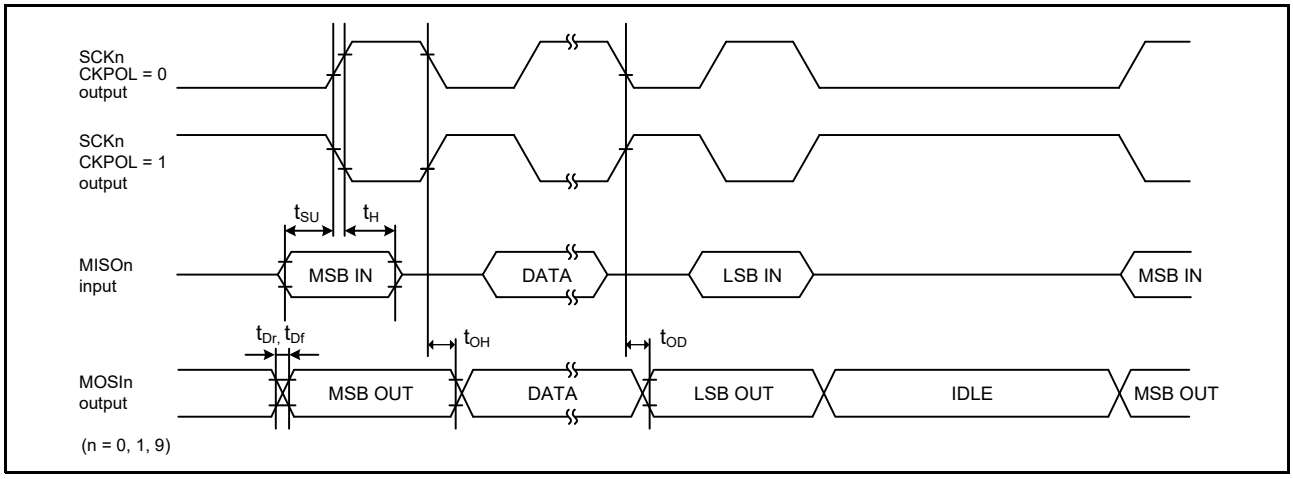


Figure 47.45 SCI simple SPI mode timing (master, CKPH = 1)

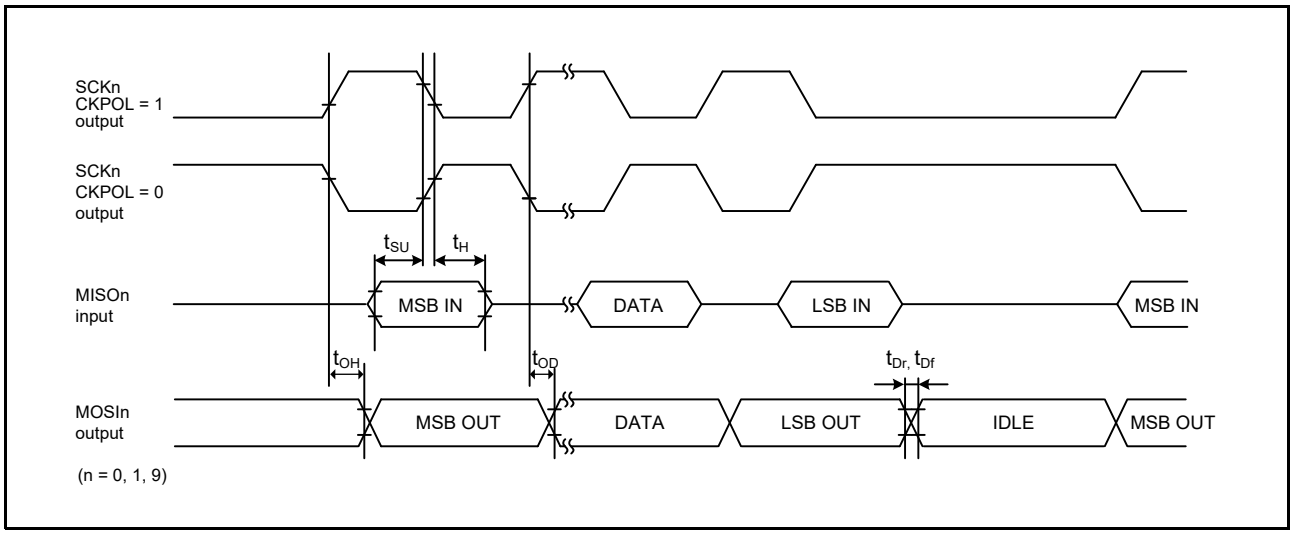


Figure 47.46 SCI simple SPI mode timing (master, CKPH = 0)

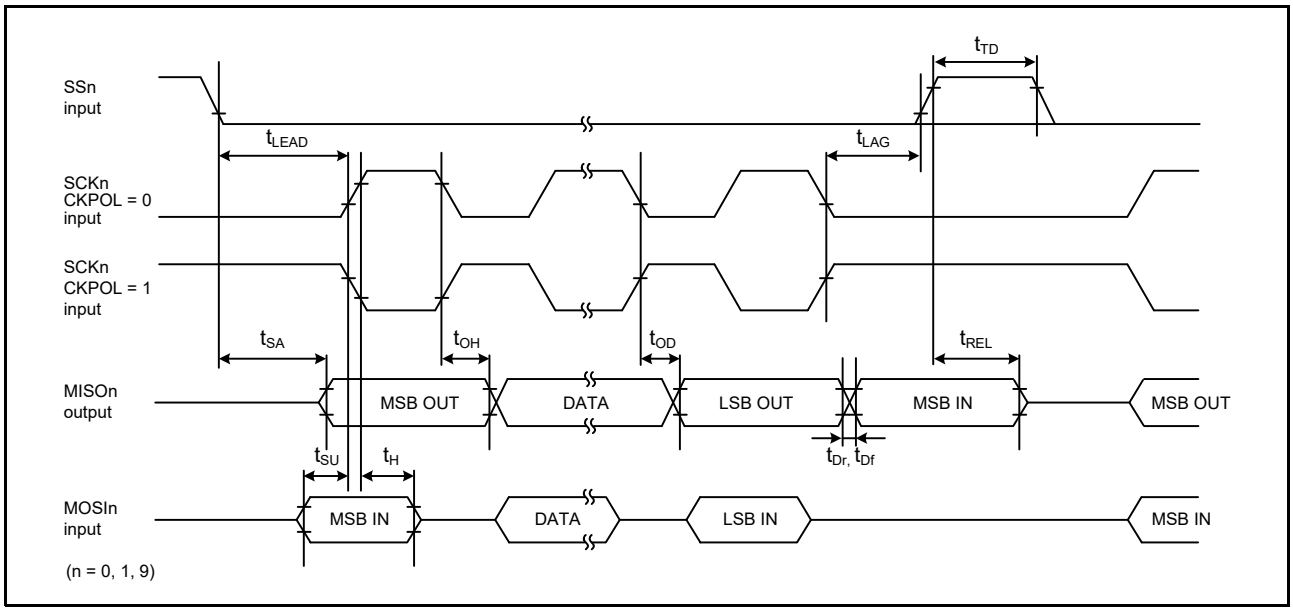


Figure 47.47 SCI simple SPI mode timing (slave, CKPH = 1)

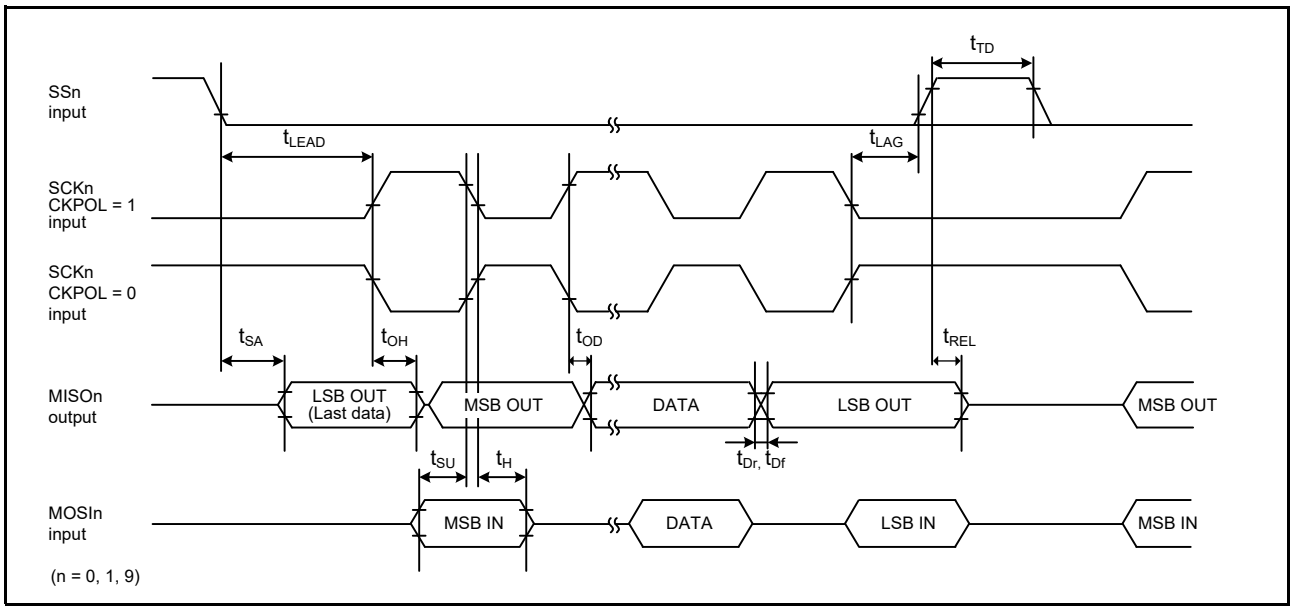


Figure 47.48 SCI simple SPI mode timing (slave, CKPH = 0)

Table 47.33 SCI timing (3)

Conditions: VCC = AVCC0 = AVCC1 = 2.7 to 5.5 V

Parameter		Symbol	Min	Max	Unit	Test conditions
Simple IIC (Standard mode)	SDA input rise time	t_{Sr}	-	1000	ns	Figure 47.49
	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	

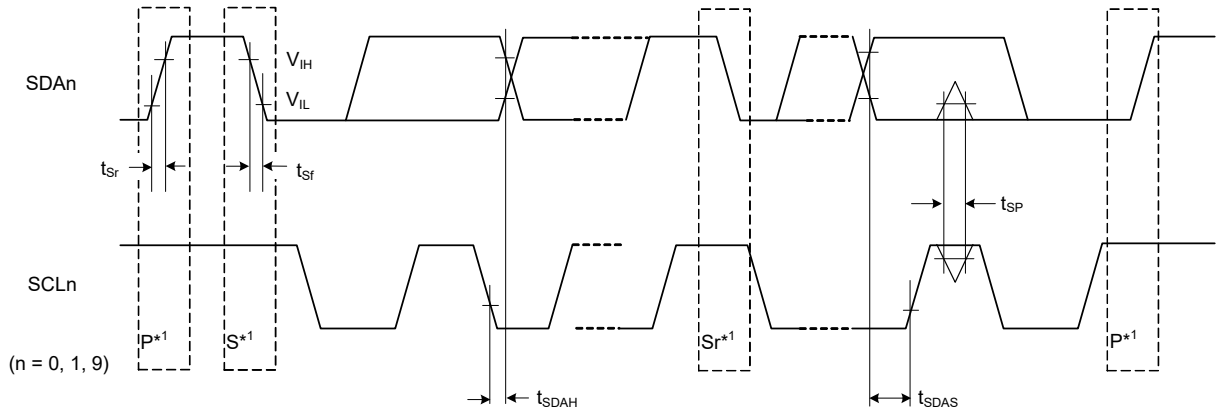
Table 47.33 SCI timing (3)

Conditions: VCC = AVCC0 = AVCC1 = 2.7 to 5.5 V

Parameter		Symbol	Min	Max	Unit	Test conditions
Simple IIC (Fast mode)	SDA input rise time	t_{Sr}	-	300	ns	Figure 47.49
	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}	100	-	ns	
	Data input hold time	t_{SDAH}	0	-	ns	
	SCL, SDA capacitive load	C_b*2	-	400	pF	

Note 1. t_{IICcyc}: Clock cycle selected by the SMR.CKS[1:0] bits.

Note 2. C_b indicates the total capacity of the bus line.



Test conditions:
V_{IH} = VCC × 0.7, V_{IL} = VCC × 0.3
V_{OL} = 0.6 V, I_{OL} = 6 mA

Note 1. S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 47.49 SCI simple IIC mode timing

47.3.9 SPI Timing

Table 47.34 SPI timing (1 of 2)

Conditions: Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter				Symbol	Min	Max	Unit*1	Test conditions
SPI	RSPCK clock cycle	Master		t_{SPcyc}	2	4096	t_{Pcyc}	Figure 47.50 C = 30 pF
		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 ≤ VCC ≤ 5.5 V	t_{SPCKr}, t_{SPCKf}	-	10	ns	
			2.4 V ≤ VCC < 2.7 V		-	15		
			1.8 V ≤ VCC ≤ 2.4 V		-	20		
			1.6 V ≤ VCC < 1.8 V		-	30		
		Input			-	1	μs	
	Data input setup time	Master		t_{SU}	10	-	ns	Figure 47.51 to Figure 47.56 C = 30 pF
		Slave	2.4 V ≤ VCC ≤ 5.5 V		10	-		
			1.8 V ≤ VCC < 2.4 V		15	-		
			1.6 V ≤ VCC < 1.8 V		20	-		
	Data input hold time	Master (RSPCK is PCLKB/2)		t_{HF}	0	-	ns	
		Master (RSPCK is not PCLKB/2)		t_H	t_{Pcyc}	-		
		Slave		t_H	20	-		
	SSL setup time	Master	1.8 V ≤ VCC ≤ 5.5 V	t_{LEAD}	-30 + N × t_{Spcyc}*2	-	ns	
			1.6 V ≤ VCC < 1.8 V		-50 + N × t_{Spcyc}*2	-		
		Slave			6 × t_{Pcyc}	-	ns	
	SSL hold time	Master		t_{LAG}	-30 + N × t_{Spcyc}*3	-	ns	
		Slave			6 × t_{Pcyc}	-	ns	
	Data output delay	Master	2.7 V ≤ VCC ≤ 5.5 V	t_{OD}	-	14	ns	
			2.4 V ≤ VCC < 2.7 V		-	20		
			1.8 V ≤ VCC < 2.4 V		-	25		
			1.6 V ≤ VCC < 1.8 V		-	30		
		Slave	2.7 V ≤ VCC ≤ 5.5 V		-	50		
			2.4 V ≤ VCC < 2.7 V		-	60		
			1.8 V ≤ VCC < 2.4 V		-	85		
			1.6 V ≤ VCC < 1.8 V		-	110		
	Data output hold time	Master		t_{OH}	0	-	ns	
		Slave			0	-		
	Successive transmission delay	Master		t_{TD}	t_{SPcyc} + 2 × t_{Pcyc}	8 × t_{SPcyc} + 2 × t_{Pcyc}	ns	
		Slave			6 × t_{Pcyc}	-		

Table 47.34 SPI timing (2 of 2)

Conditions: Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter				Symbol	Min	Max	Unit*1	Test conditions
SPI	MOSI and MISO rise and fall time	Output	2.7 V ≤ VCC ≤ 5.5 V	t_{Dr}, t_{Df}	-	10	ns	Figure 47.51 to Figure 47.56 C = 30 pF
			2.4 V ≤ VCC < 2.7 V		-	15		
			1.8 V ≤ VCC < 2.4 V		-	20		
			1.6 V ≤ VCC < 1.8 V		-	30		
		Input			-	1	μs	
	SSL rise and fall time	Output	2.7 V ≤ VCC ≤ 5.5 V	t_{SSLr}, t_{SSLf}	-	10	ns	
			2.4 V ≤ VCC < 2.7 V		-	15		
			1.8 V ≤ VCC < 2.4 V		-	20		
			1.6 V ≤ VCC < 1.8 V		-	30		
		Input			-	1	μs	
	Slave access time		2.4 V ≤ VCC ≤ 5.5 V	t_{SA}	-	2 × t_{Pcyc} + 100	ns	Figure 47.55 and Figure 47.56 C = 30 pF
			1.8 V ≤ VCC < 2.4 V		-	2 × t_{Pcyc} + 140		
			1.6 V ≤ VCC < 1.8 V		-	2 × t_{Pcyc} + 180		
	Slave output release time		2.4 V ≤ VCC ≤ 5.5 V	t_{REL}	-	2 × t_{Pcyc} + 100	ns	
			1.8 V ≤ VCC < 2.4 V		-	2 × t_{Pcyc} + 140		
			1.6 V ≤ VCC < 1.8 V		-	2 × t_{Pcyc} + 180		

Note 1. t_{Pcyc}: PCLKB cycle.
Note 2. N is set as an integer from 1 to 8 by the SPCKD register.
Note 3. N is set as an integer from 1 to 8 by the SSLND register.

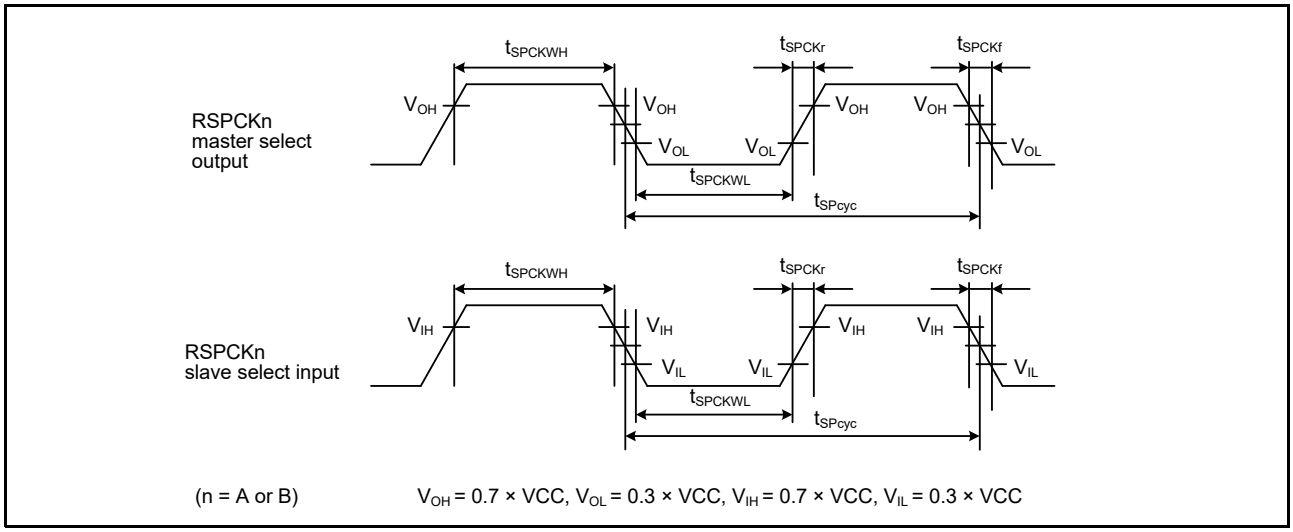


Figure 47.50 SPI clock timing

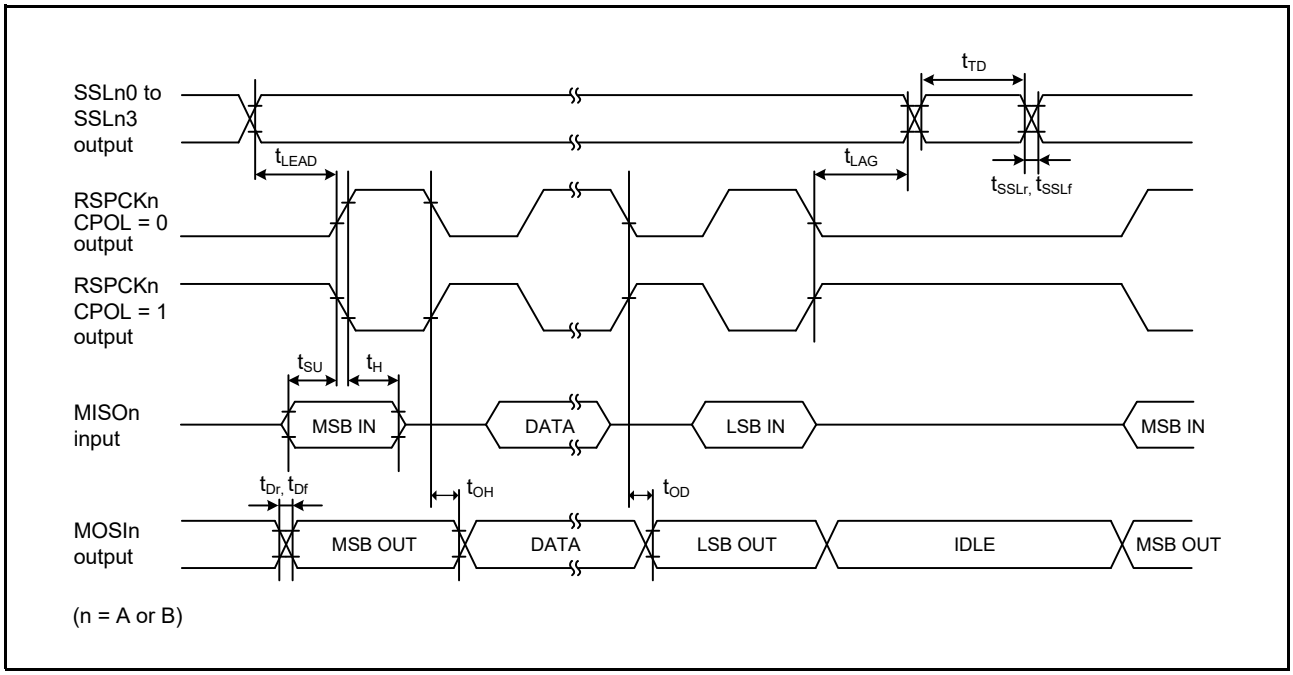


Figure 47.51 SPI timing (master, CPHA = 0) (bit rate: PCLKB division ratio is set to any value other than 1/2)

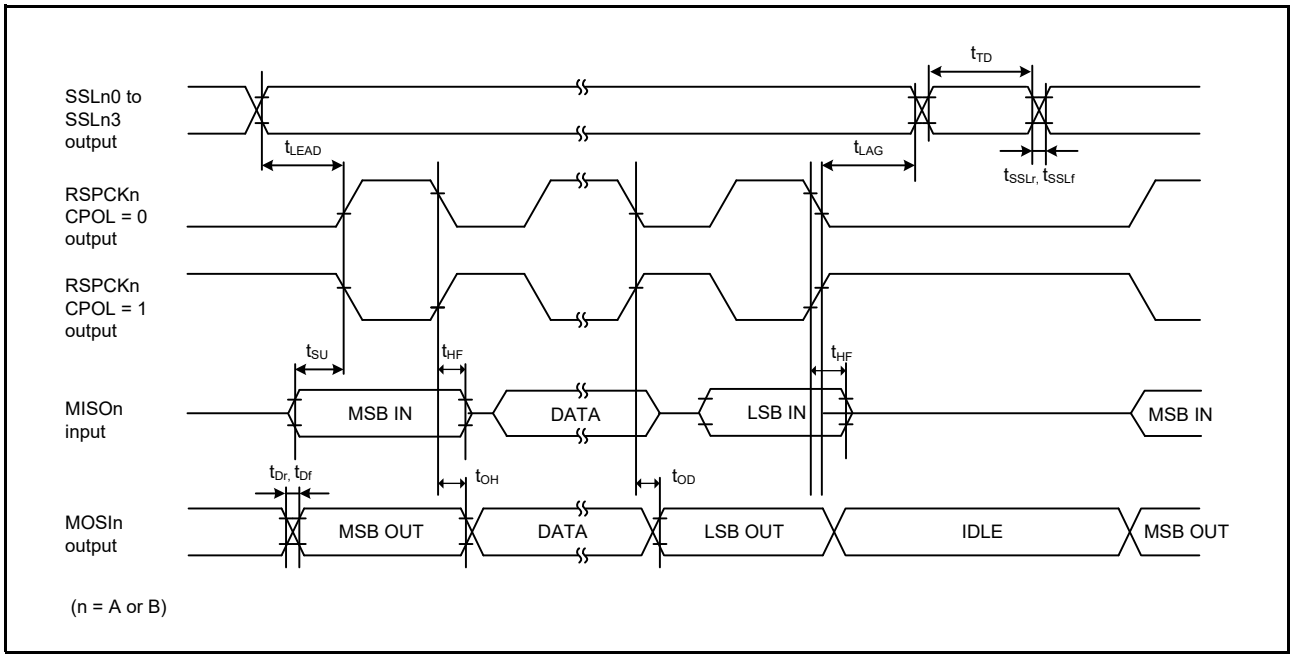


Figure 47.52 SPI timing (master, CPHA = 0) (bit rate: PCLKB division ratio is set to 1/2)

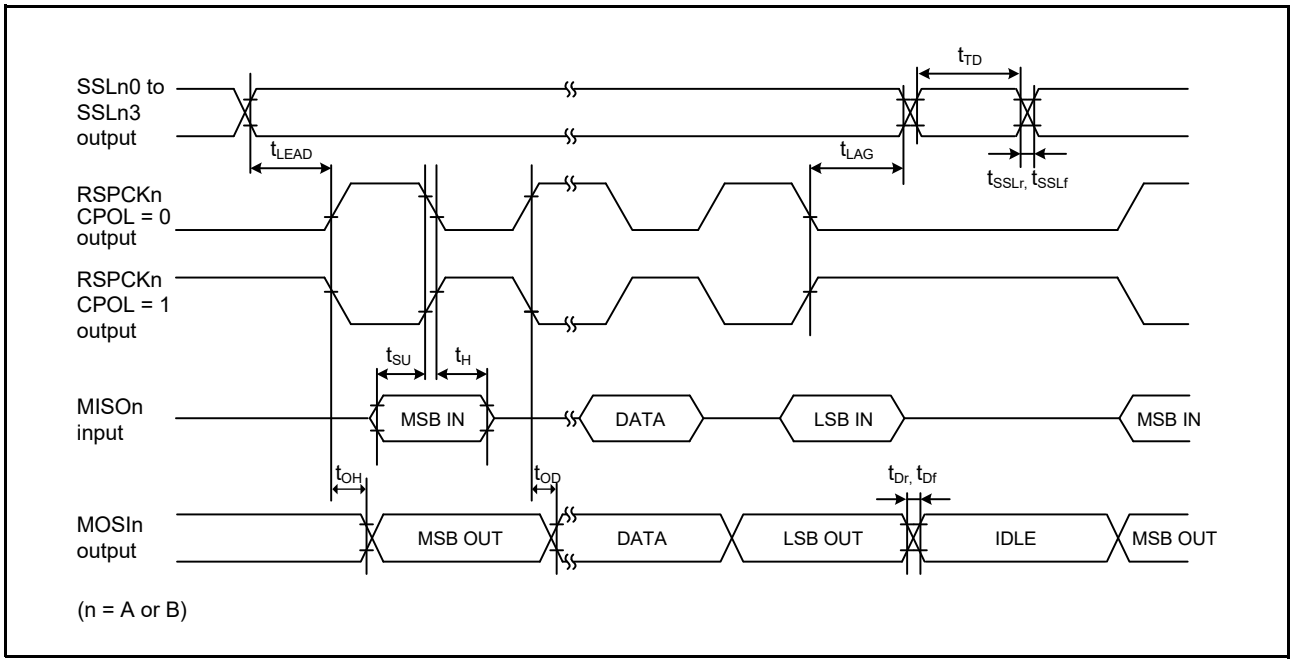


Figure 47.53 SPI timing (master, CPHA = 1) (bit rate: PCLKB division ratio is set to any value other than 1/2)

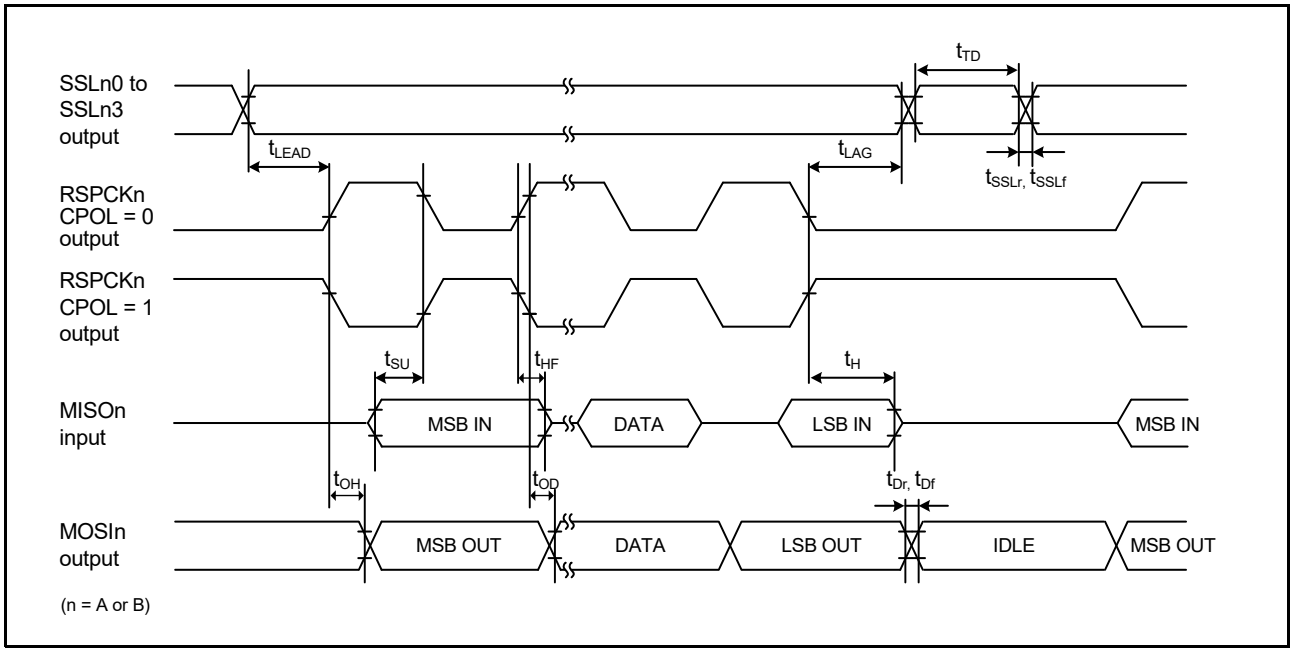


Figure 47.54 SPI timing (master, CPHA = 1) (bit rate: PCLKB division ratio is set to 1/2)

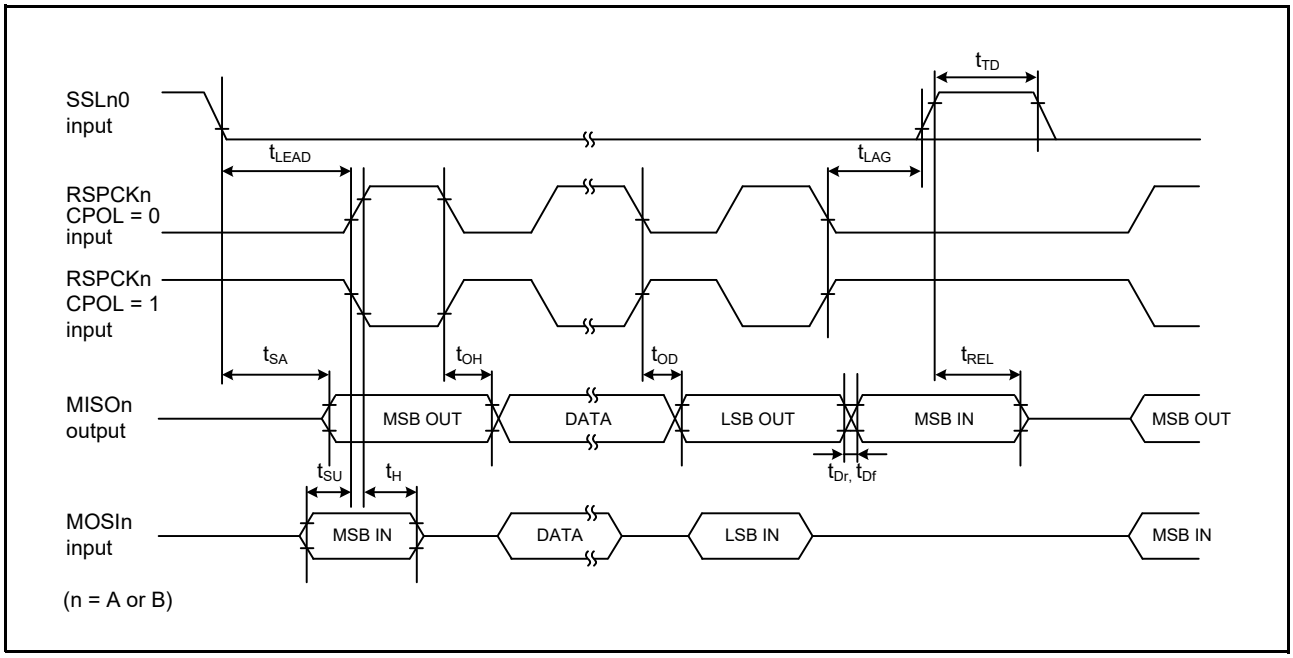


Figure 47.55 SPI timing (slave, CPHA = 0)

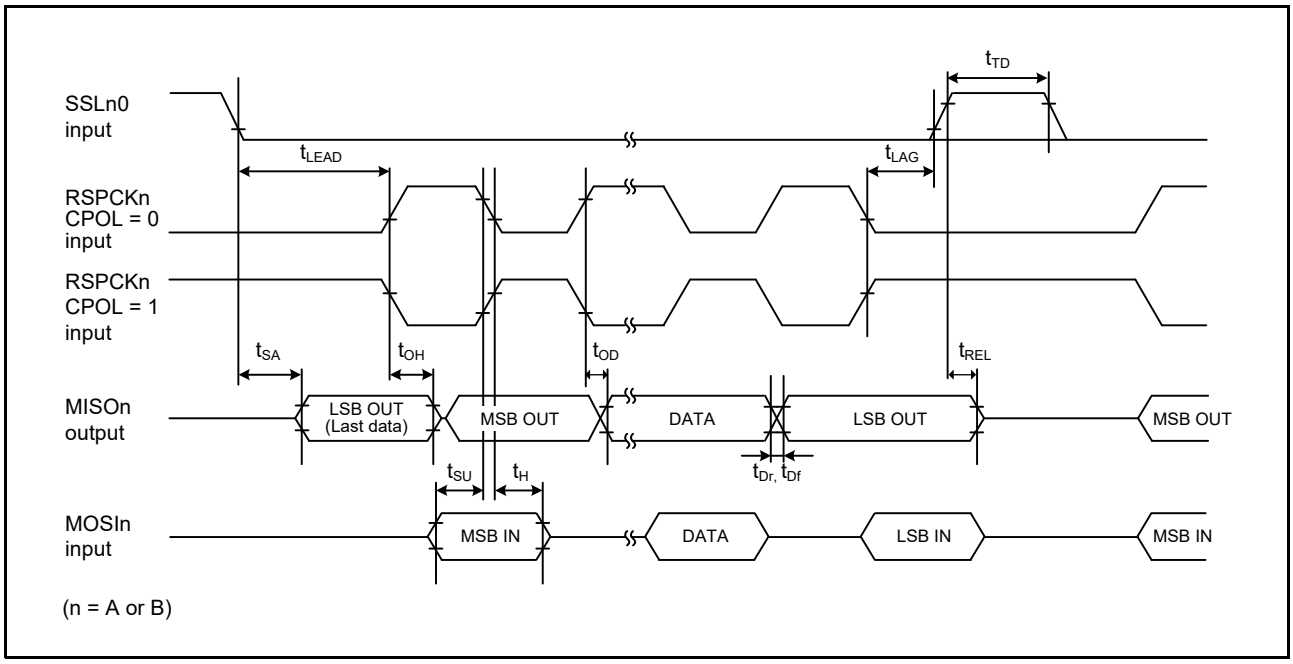


Figure 47.56 SPI timing (slave, CPHA = 1)

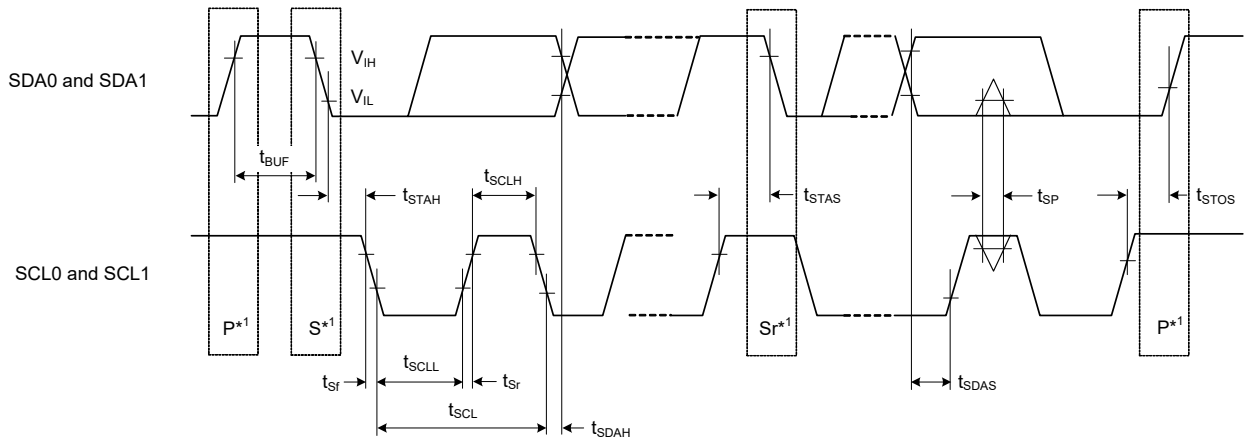
47.3.10 IIC Timing

Table 47.35 IIC timing
Conditions: VCC = AVCC0 = AVCC1 = 2.7 to 5.5 V

Parameter		Symbol	Min*1	Max	Unit	Test conditions
IIC (Standard mode, SMBus)	SCL input cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 1300	-	ns	Figure 47.57
	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 47.57
	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}	-	300	ns	
	SCL, SDA input fall time	t_{Sf}	-	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA input bus free time (When wakeup function is disabled)	t_{BUF}	3 (6) × t_{IICcyc} + 300	-	ns	
	SDA input bus free time (When wakeup function is enabled)	t_{BUF}	3 (6) × t_{IICcyc} + 4 × t_{Pcyc} + 300	-	ns	
	START condition input hold time (When wakeup function is disabled)	t_{STAH}	t_{IICcyc} + 300	-	ns	
	START condition input hold time (When wakeup function is enabled)	t_{STAH}	1 (5) × t_{IICcyc} + t_{Pcyc} + 300	-	ns	
	Repeated START condition input setup time	t_{STAS}	300	-	ns	
	STOP condition input setup time	t_{STOS}	300	-	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	-	ns	
	Data input hold time	t_{SDAH}	0	-	ns	
	SCL, SDA capacitive load	C_b	-	400	pF	

Note: t_{IICcyc}: IIC internal reference clock (IICφ) cycle, t_{Pcyc}: PCLKB cycle

Note 1. Values in parentheses apply when ICMR3.NF[1:0] is set to 11b while the digital filter is enabled with ICFER.NFE set to 1.



Note 1. S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 47.57 I²C bus interface input/output timing

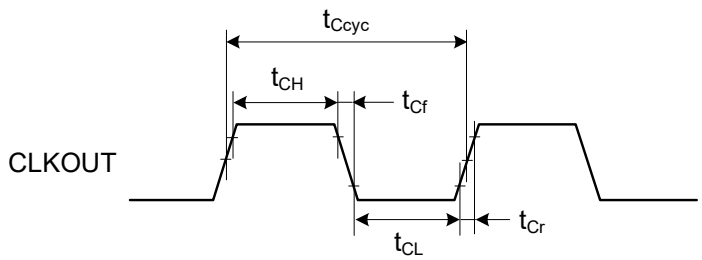
47.3.11 CLKOUT Timing

Table 47.36 CLKOUT timing

Parameter			Symbol	Min	Max	Unit	Test conditions
CLKOUT	CLKOUT pin output cycle*1	2.7 V ≤ VCC ≤ 5.5 V	t_{Ccyc}	62.5	-	ns	Figure 47.58
		1.8 V ≤ VCC < 2.7 V		125	-		
		1.6 V ≤ VCC < 1.8 V		250	-		
	CLKOUT pin high pulse width*2	2.7 V ≤ VCC ≤ 5.5 V	t_{CH}	15	-	ns	
		1.8 V ≤ VCC < 2.7 V		30	-		
		1.6 V ≤ VCC < 1.8 V		150	-		
	CLKOUT pin low pulse width*2	2.7 V ≤ VCC ≤ 5.5 V	t_{CL}	15	-	ns	
		1.8 V ≤ VCC < 2.7 V		30	-		
		1.6 V ≤ VCC < 1.8 V		150	-		
	CLKOUT pin output rise time	2.7 V ≤ VCC ≤ 5.5 V	t_{Cr}	-	12	ns	
		1.8 V ≤ VCC < 2.7 V		-	25		
		1.6 V ≤ VCC < 1.8 V		-	50		
	CLKOUT pin output fall time	2.7 V ≤ VCC ≤ 5.5 V	t_{Cf}	-	12	ns	
		1.8 V ≤ VCC < 2.7 V		-	25		
		1.6 V ≤ VCC < 1.8 V		-	50		

Note 1. When the EXTAL external clock input or an oscillator is used with division by 1 (the CKOCR.CKOSEL[2:0] bits are 011b and the CKOCR.CKODIV[2:0] bits are 000b) to output from CLKOUT, specifications in Table 47.36 should be satisfied with 45% to 55% of input duty cycle.

Note 2. When MOCO is selected as the clock output source (the CKOCR.CKOSEL[2:0] bits are 001b), set the clock output division ratio to be divided by 2 (the CKOCR.CKODIV[2:0] bits are 001b).



Test conditions: V_{OH} = VCC × 0.7, V_{OL} = VCC × 0.3, I_{OH} = -1.0 mA, I_{OL} = 1.0 mA, C = 30 pF

Figure 47.58 CLKOUT output timing

47.4 USB Characteristics

47.4.1 USBFS Timing

Table 47.37 USB characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = 3.0 to 3.6 V, Ta = -20 to +85°C

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 47.59, Figure 47.60, Figure 47.61
	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 version 1.2	D+ sink current		I_{DP_SINK}	25	175	μA	-
	D- sink current		I_{DM_SINK}	25	175	μA	-
	DCD source current		I_{DP_SRC}	7	13	μA	-
	Data detection voltage		V_{DAT_REF}	0.25	0.4	V	-
	D+ source voltage		V_{DP_SRC}	0.5	0.7	V	Output current = 250 μA
	D- source voltage		V_{DM_SRC}	0.5	0.7	V	Output current = 250 μA

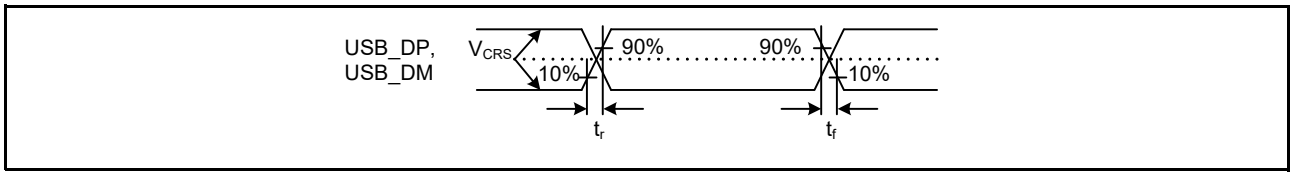


Figure 47.59 USB_DP and USB_DM output timing

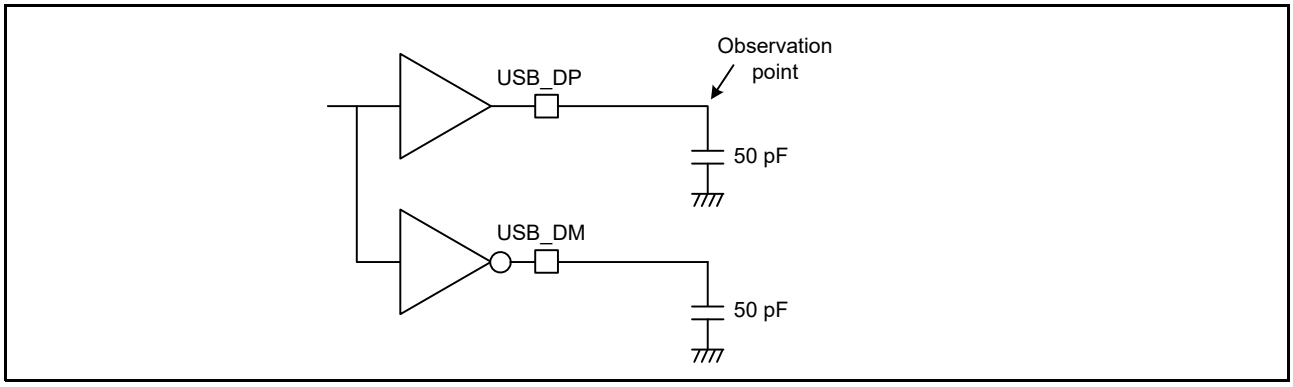


Figure 47.60 Test circuit for Full-Speed (FS) connection

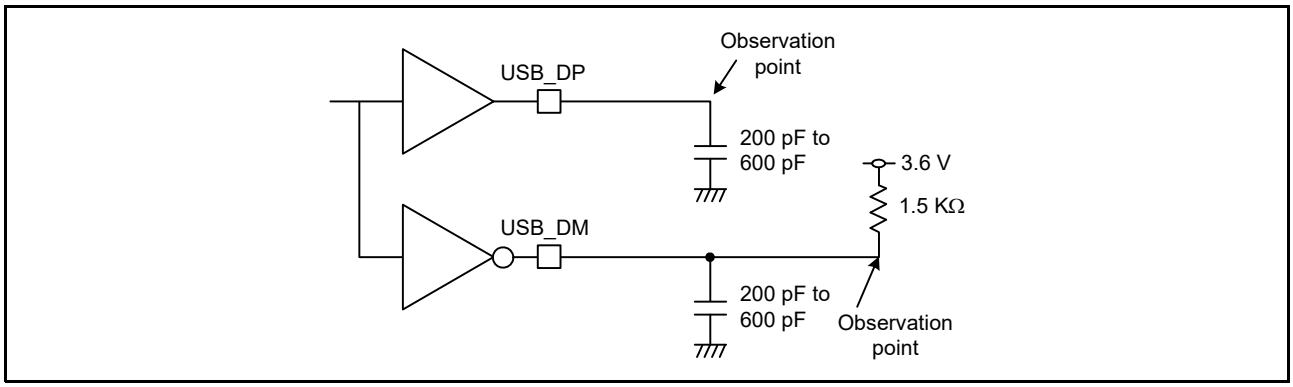


Figure 47.61 Test circuit for Low-Speed (LS) connection

47.4.2 USB External Supply

Table 47.38 USB regulator

Parameter		Min	Typ	Max	Unit	Test conditions
VCC_USB supply current	3.8 V ≤ VCC_USB_LDO < 4.5 V	-	-	50	mA	-
	4.5 V ≤ VCC_USB_LDO ≤ 5.5 V	-	-	100	mA	-
VCC_USB supply voltage		3.0	-	3.6	V	-

47.5 ADC16 Characteristics

Table 47.39 16-bit A/D conversion, power supply, and input range conditions

Conditions: VCC = AVCC0 = AVCC1 = 1.7 to 5.5 V, VREFH0 = 1.7 to 5.5 V, VSS = AVSS0 = AVSS1 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
High-potential reference voltage		1.5	3.3	AVCC0	V	-
Low-potential reference voltage		-	AVSS0	-	V	-
Analog input voltage range		0	-	VREFH0	V	-
Input common-mode range	Acm	0	VREFH0/2	VREFH0	V	Differential analog input
Analog input capacitance*2	Cs	-	-	4.3	pF	-
Analog input resistance*1	Rs	-	-	0.7	kΩ	High-precision channel 2.7 V ≤ AVCC0 ≤ 5.5 V
		-	-	1.5		High-precision channel 1.7 V ≤ AVCC0 < 2.7 V
		-	-	2.5		Normal-precision channel 2.7 V ≤ AVCC0 ≤ 5.5 V
		-	-	3.8		Normal-precision channel 1.7 V ≤ AVCC0 < 2.7 V

Note 1. These values are based on simulation. They are not production tested.
Note 2. Except for I/O input capacitance (Cin), see section 47.2.4, I/O VOH, VOL, and Other Characteristics.

Figure 47.62 shows the equivalent circuit for analog input.

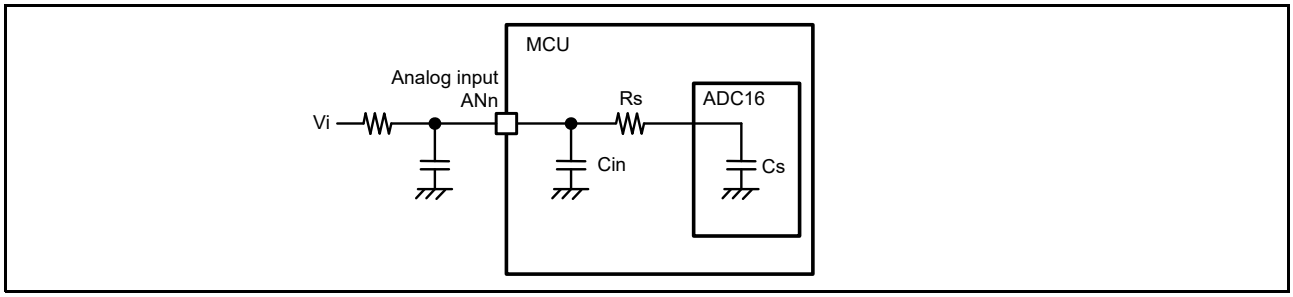


Figure 47.62 Equivalent circuit for analog input

Table 47.40 16-bit A/D conversion, timing parameters (1 of 2)

Conditions: VCC = AVCC0 = AVCC1 = 1.7 to 5.5 V, VREFH0 = 1.7 to 5.5 V, VSS = AVSS0 = AVSS1 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Frequency	ADCLK	1	-	32	MHz	3.0 V ≤ AVCC0 ≤ 5.5 V, 3.0 V ≤ VREFH0
		1	-	24		2.7 V ≤ AVCC0 ≤ 5.5 V, 2.7 V ≤ VREFH0
		1	-	16		2.4 ≤ AVCC0 ≤ 5.5 V, 1.5 V ≤ VREFH0
		1	-	8		1.8 V ≤ AVCC0 ≤ 5.5 V, 1.5 V ≤ VREFH0
		1	-	4		1.7 V ≤ AVCC0 ≤ 5.5 V, 1.5 V ≤ VREFH0
Conversion rate	Fs	-	-	1 / (tSPL + 18 / ADCLK)	S/s	-

Table 47.40 16-bit A/D conversion, timing parameters (2 of 2)
Conditions: VCC = AVCC0 = AVCC1 = 1.7 to 5.5 V, VREFH0 = 1.7 to 5.5 V, VSS = AVSS0 = AVSS1 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Sampling time*1	Permissible signal source impedance Max = 0.5 kΩ	tSPL	0.25	-	-	μs	High-precision channel 2.7 V ≤ AVCC0 ≤ 5.5 V
			3	-	-		High-precision channel 1.7 V ≤ AVCC0 < 2.7 V
			3	-	-		Normal-precision channel 2.7 V ≤ AVCC0 ≤ 5.5 V
			10	-	-		Normal-precision channel 1.7 V ≤ AVCC0 < 2.7 V
Settling time*1		tSTART	-	-	1	μs	2.7 V ≤ AVCC0 ≤ 5.5 V
			-	-	3.2		1.8 V ≤ AVCC0 < 2.7 V
			-	-	8.9		1.7 V ≤ AVCC0 < 1.8 V

Note 1. These values are based on simulation. They are not production tested.

Table 47.41 16-bit A/D conversion, linearity parameters
Conditions: VCC = AVCC0 = AVCC1 = 1.7 to 5.5 V, VREFH0 = 1.7 to 5.5 V, VSS = AVSS0 = AVSS1 = VREFL0 = 0 V
External clock input used. Reference voltage range applied to the VREFH0 and VREFL0.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Resolution	-	-	16	-	Bit	-
Integral non-linearity *1	INL	-	± 4	± 8	LSB	2.7 V ≤ AVCC0 ≤ 5.5 V, 2.7 V ≤ VREFH0
		-		± 16		1.7 V ≤ AVCC0 < 2.7 V
Differential non-linearity*1	DNL	-	-1 to +2	-	LSB	-
Offset error*1	Ofst	-	± 4	-	LSB	-
Gain error*1	Gerr	-	-	±0.1	%	2.7 V ≤ VREFH0

Note: The characteristics apply when no pin functions other than 16-bit A/D converter input are used. Offset error, full-scale error, DNL differential non-linearity error, and INL integral non-linearity error do not include quantization errors.
Note 1. These values are based on simulation. They are not production tested.

Table 47.42 16-bit A/D conversion, dynamic parameters (1) (1 of 2)
Conditions: VCC = AVCC0 = AVCC1 = 1.7 to 5.5 V, VREFH0 = 1.7 to 5.5 V, VSS = AVSS0 = AVSS1 = VREFL0 = 0 V
External clock input used. Reference voltage range applied to VREFH0 and VREFL0.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Signal-to-noise and distortion*2	SINAD	67	81	-	dB	Differential input, Fin = 1 kHz, VREFH0 = 1.7 V to 5.5 V, AVCC0 = 1.7 V to 5.5 V
		78	81	-		Differential input, Fin = 1 kHz, VREFH0 = 3.3 V, AVCC0 = 3.3 V
		-	92	-		Differential input, Fin = 1 kHz, VREFH0 = 3.3 V, AVCC0 = 3.3 V, ADADC.ADC[2:0] = 101b
		61	75	-		Single input, Fin = 1 kHz, VREFH0 = 1.7 V to 5.5 V, AVCC0 = 1.7 V to 5.5 V
		72	75	-		Single input, Fin = 1 kHz, VREFH0 = 3.3 V, AVCC0 = 3.3 V

Table 47.42 16-bit A/D conversion, dynamic parameters (1) (2 of 2)

Conditions: VCC = AVCC0 = AVCC1 = 1.7 to 5.5 V, VREFH0 = 1.7 to 5.5 V, VSS = AVSS0 = AVSS1 = VREFL0 = 0 V
External clock input used. Reference voltage range applied to VREFH0 and VREFL0.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Effective number of bits*2	ENOB	11	13.2	-	bit	Differential input, Fin = 1 kHz, VREFH0 = 1.7 V to 5.5 V, AVCC0 = 1.7 V to 5.5 V
		12.7	13.2	-		Differential input, Fin = 1 kHz, VREFH0 = 3.3 V, AVCC0 = 3.3 V
		-	15	-		Differential input, Fin = 1 kHz, VREFH0 = 3.3 V, AVCC0 = 3.3 V, ADADC.ADC[2:0] = 101b
		10	12.2	-		Single input, Fin = 1 kHz, VREFH0 = 1.7 V to 5.5 V, AVCC0 = 1.7 V to 5.5 V
		11.7	12.2	-		Single input, Fin = 1 kHz, VREFH0 = 3.3 V, AVCC0 = 3.3 V
Total harmonic distortion*1, *2	THD	-	-100	-	dB	Differential input, Fin = 1 kHz, AVCC0 = 3.3 V
		-	-90	-		Single input, Fin = 1 kHz, AVCC0 = 3.3 V
Common mode rejection ratio*2	CMRR	-	100	-	dB	Differential input, Acm = 0 to VREFH0 at 1 kHz, AVCC0 = 3.3 V

Note: The characteristics apply when no pin functions other than 16-bit A/D converter input are used.
Note 1. THD = HD2 + HD3 + HD4 + HD5.
Note 2. These values are based on simulation. They are not production tested.

Table 47.43 16-bit A/D conversion, dynamic parameters (2)

Conditions: VCC = AVCC0 = AVCC1 = 1.7 to 5.5 V, VSS = AVSS0 = AVSS1 = VREFL0 = 0 V
External clock input used.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Signal-to-noise and distortion*1	SINAD	-	78.6	-	dB	Differential input, Fin = 1 kHz, AVCC0 = 3.3 V, VREFADC output = 2.5 V
		-	76.6	-		Differential input, Fin = 1 kHz, AVCC0 = 3.3 V, VREFADC output = 2.0 V
		-	74.2	-		Differential input, Fin = 1 kHz, AVCC0 = 3.3 V, VREFADC output = 1.5 V
Effective number of bits*1	ENOB	-	12.8	-	bit	Differential input, Fin = 1 kHz, AVCC0 = 3.3 V, VREFADC output = 2.5 V
		-	12.4	-		Differential input, Fin = 1 kHz, AVCC0 = 3.3 V, VREFADC output = 2.0 V
		-	12.0	-		Differential input, Fin = 1 kHz, AVCC0 = 3.3 V, VREFADC output = 1.5 V

Note: The characteristics apply when no pin functions other than 16-bit A/D converter input are used.
Note 1. These values are based on simulation. They are not production tested.

Table 47.44 16-bit A/D converter channel classification

Classification	Channel	Conditions
High-precision channel	AN000 to AN008	AVCC0 = 1.7 to 5.5 V
Normal-precision channel	AN016 to AN023	
Internal reference voltage input channel	Internal reference voltage	AVCC0 = 2.0 to 5.5 V
Temperature sensor input channel	Temperature sensor output	AVCC0 = 2.0 to 5.5 V

Table 47.45 Internal reference voltage for 16-bit ADC (VREFADC) characteristics
Conditions: VCC = AVCC0 = AVCC1 = 1.7 V to 5.5 V, VSS = AVSS0 = AVSS1 = VREFL0 = 0 V

Parameter	Min	Typ	Max	Unit	Test conditions
Output voltage range	1.41	1.5	1.59	V	VREFAMPCNT.VREFADCG[1:0] = 00b AVCC0 ≥ 1.7 V
	1.88	2	2.12		VREFAMPCNT.VREFADCG[1:0] = 10b AVCC0 ≥ 2.2 V
	2.35	2.5	2.65		VREFAMPCNT.VREFADCG[1:0] = 11b AVCC0 ≥ 2.7 V
BGR stabilization time*2 (after BGR is enabled)	-	-	150	μs	VREFAMPCNT.BGREN = 1
VREF AMP stabilization time*2 (after VREFAMP is enabled)	-	-	1500	μs	VREFAMPCNT.VREFADCEN = 1
Detect over current*2	-	20	40	mA	-
Load capacitance*1	0.75	1	1.25	μF	-

Note 1. Connect capacitors as stabilization capacitance between the VREFH0 and VREFL0 pins when VREFADC is used.
Note 2. These values are based on simulation. They are not production tested.

Table 47.46 A/D internal reference voltage characteristics
Conditions: VCC = AVCC0 = AVCC1 = VREFH0 = 2.0 to 5.5 V*1

Parameter	Min	Typ	Max	Unit	Test conditions
Internal reference voltage input channel*2	1.36	1.43	1.50	V	-
Sampling time*3	5.0	-	-	μs	-

Note 1. The internal reference voltage cannot be selected for input channels when AVCC0 < 2.0 V.
Note 2. The 16-bit A/D internal reference voltage indicates the voltage when the internal reference voltage is input to the 16-bit A/D converter.
Note 3. This is a parameter for ADC16 when the internal reference voltage is selected for an analog input channel in ADC16.

47.6 SDADC24 Characteristics

Table 47.47 Analog inputs characteristics (1 of 2)
Conditions: VCC = AVCC0 = AVCC1 = 2.7 to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Full-scale range		F_{SR}	-	± 0.8 / G_{TOTAL}	-	V	-
Analog input in differential input mode	Differential input voltage range	V_{ID}	-0.8 / G_{TOTAL}	-	0.8 / G_{TOTAL}	V	V_{ID} = ANSDnP - ANSDnN, or AMP0O - AMP1O (n = 0 to 3), d_{OFR} = 0 mV
	Input voltage range	V_I	0.2	-	1.8	V	V_I = ANSDnP, ANSDnN, AMP0O, or AMP1O (n = 0 to 3)
	Common mode Input voltage range	V_{COM}	0.2 + (|V_{ID}| × G_{SET1}) / 2	1.0	1.8 - (|V_{ID}| × G_{SET1}) / 2	V	d_{OFR} = 0 mV

Table 47.47 Analog inputs characteristics (2 of 2)
Conditions: VCC = AVCC0 = AVCC1 = 2.7 to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Analog Input in single-ended input mode	Input voltage range*1	V_I	0.2	-	1.8	V	V_I = ANSDnP, ANSDnN, AMP0O, or AMP1O (n = 0 to 3), V_{COM} = 1.0 V, d_{OFR} = 0 mV, G_{SET1} = 1, G_{SET2} = 1, OSR = 256

Note 1. The single-ended input mode supports only d_{OFR} = 0 mV, G_{SET1} = 1, G_{SET2} = 1 and OSR = 256.

Table 47.48 Programmable gain instrumentation amplifier and sigma-delta A/D converter (1)
Conditions: VCC = AVCC0 = AVCC1 = 2.7 to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Resolution		RES	-	24	-	bits	-
Over sampling frequency	Normal A/D conversion mode	Fos	-	1	-	MHz	-
	Low-power A/D conversion mode		-	0.125	-		
Output data rate		f_{DATA1}	0.48828	-	15.625	ksps	Normal A/D conversion mode
		f_{DATA2}	61.03615	-	1953.125	sps	Low-power A/D conversion mode
Gain Setting range		G_{TOTAL}	1	-	32	V/V	$G_{TOTAL} = G_{SET1} \times G_{SET2}$
1st Gain Setting range		G_{SET1}	-	1, 2, 3, 4, 8	-	V/V	-
2nd Gain Setting range		G_{SET2}	-	1, 2, 4, 8	-	V/V	-
Offset adjust bit range		d_{OFB}	-	5	-	bits	-
Offset adjust range		d_{OFR}	-164.06 / G_{SET1}	-	+164.06 / G_{SET1}	mV	Referred to input
Offset adjust step		d_{OFS}	-	350 / 32 / G_{SET1}	-	mV	Referred to input

Table 47.49 Programmable gain instrumentation amplifier and sigma-delta A/D converter (2)
Conditions: VCC = AVCC0 = AVCC1 = 2.7 to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V
The electrical specifications are applied at differential input mode, external clock input used, F_{OS} = 1 MHz, dOFR = 0 mV, unless otherwise specified.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions	
Signal to Noise Ratio*1,*3 V_{ID} = 0 V	SNR	83	86	-	dB	G_{SET1} = 1, G_{SET2} = 1	OSR = 256
		81	84	-	dB	G_{SET1} = 8, G_{SET2} = 4	OSR = 1024
Signal to Noise and Distortion Ratio*1, *2,*3 fin = 50 Hz	SINAD	82	85	-	dB	G_{SET1} = 1, G_{SET2} = 1	OSR = 256
		79	82	-	dB	G_{SET1} = 8, G_{SET2} = 4	OSR = 1024
		74	80	-	dB	G_{SET1} = 1, G_{SET2} = 1	OSR = 256, Single-ended input mode

Note: The characteristics apply when no pin functions other than 24-bit sigma-delta A/D converter input are used.
Note 1. SNR and SINAD are the ratio to Full-Scale Range (FSR) of analog inputs. These do not include the noise of analog inputs.
Note 2. When V_{ID} is equal to ± 0.8 / G_{TOTAL} actually, the digital output may overflow due to Gain Error (E_G), Offset Error (E_{OS}), and so forth. As a result, SINAD is degraded. See Table 33.7 for the relation between analog input and digital output.
Note 3. Not production tested but is guaranteed by the design and characterization.

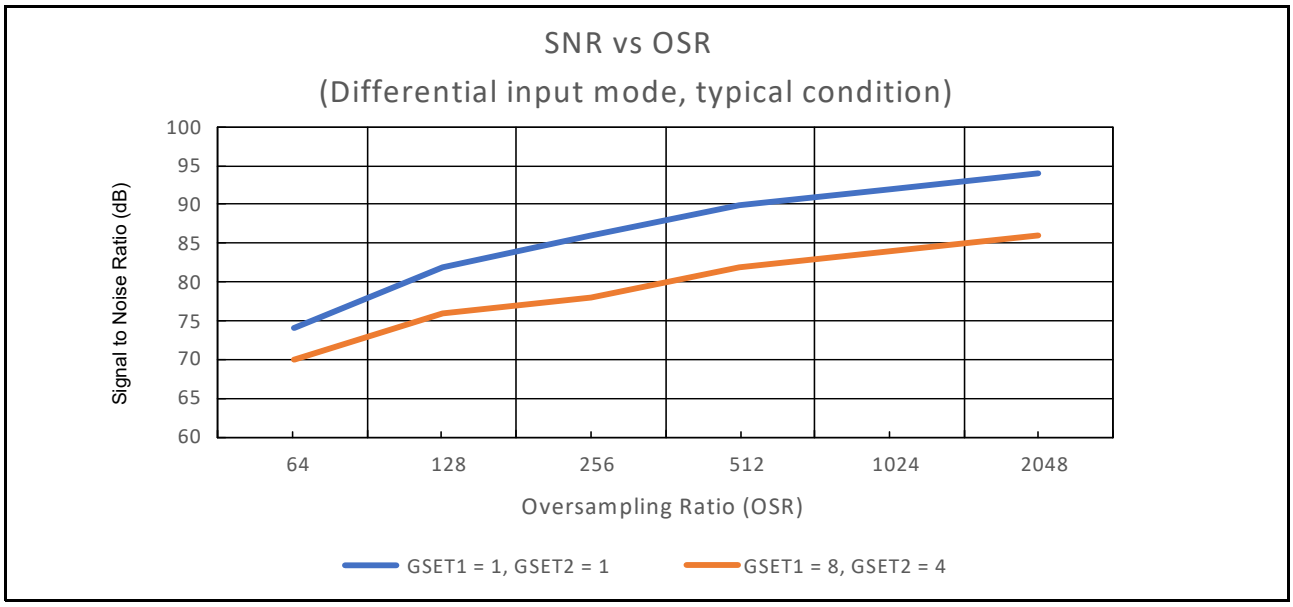


Figure 47.63 SNR vs. OSR (reference data)

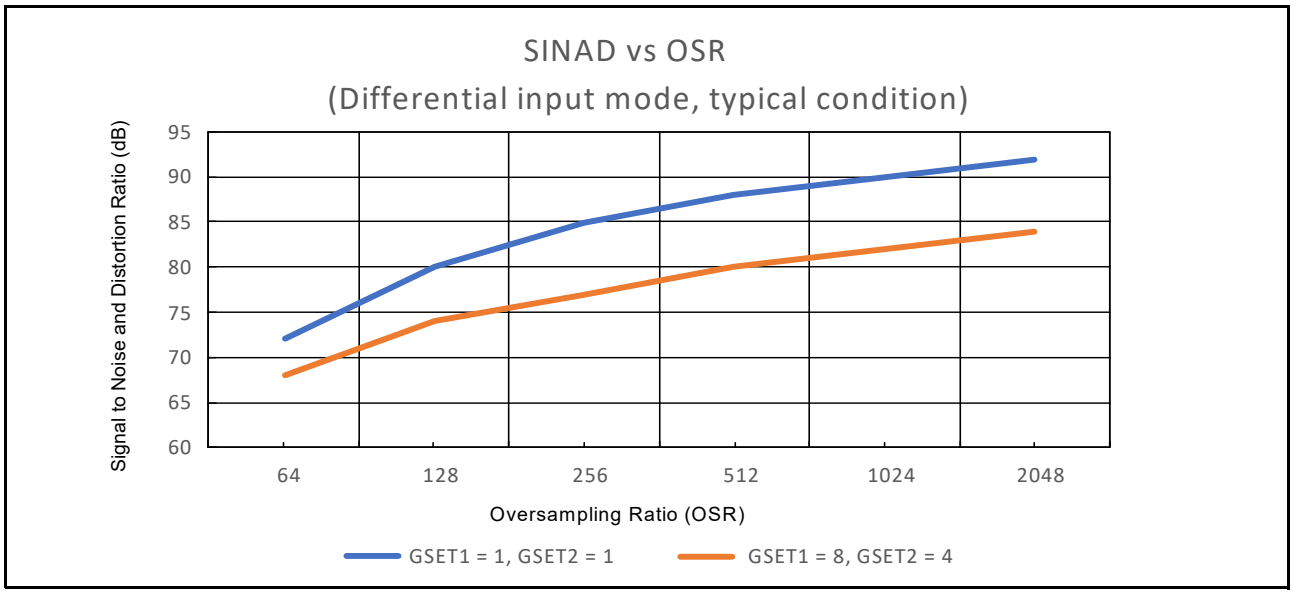


Figure 47.64 SINAD vs. OSR (reference data)

Table 47.50 Programmable gain instrumentation amplifier and sigma-delta A/D converter (3) (1 of 2)

Conditions: VCC = AVCC0 = AVCC1 = 2.7 to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V

The electrical specifications are applied at the differential input mode, with external clock input used, Fos = 1 MHz, OSR = 256, and d_{OFR} = 0 mV, unless otherwise specified.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Gain error*2 (excluding SINC3 frequency response characteristic)	E_G	-0.5	-	0.5	%	After internal calibration, excluding SBIAS error or VREFI error, G_{SET1} = 1, G_{SET2} = 1
		-3	-	3		Single-ended input mode, excluding SBIAS error or VREFI error, G_{SET1} = 1, G_{SET2} = 1

Table 47.50 Programmable gain instrumentation amplifier and sigma-delta A/D converter (3) (2 of 2)

Conditions: VCC = AVCC0 = AVCC1 = 2.7 to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V
The electrical specifications are applied at the differential input mode, with external clock input used, Fos = 1 MHz, OSR = 256, and d_{OFR} = 0 mV, unless otherwise specified.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Gain drift*1, *2	dE_G	-	6	22	ppm/°C	Excluding SBIAS error or VREFI error, G_{SET1} = 1, G_{SET2} = 1
Offset error*2	E_{OS}	-1	-	1	mV	After internal calibration, G_{SET1} = 1, G_{SET2} = 1, referred to input
		-50	-	50		Single-ended input mode, including SBIAS error, G_{SET1} = 1, G_{SET2} = 1, referred to input
Offset drift*1, *2	dE_{OS}	-	2	6	μV/°C	Referred to input
		-	-	120		Single-ended input mode, including SBIAS error, G_{SET1} = 1, G_{SET2} = 1
Integral non-linearity*2	INL	-	15	-	ppm of FSR	Input = DC, OSR = 2048
Common mode Rejection ratio*2	CMRR	-	80	-	dB	V_{COM} = 1.0 ± 0.8 V, f_{in} = 50 Hz, G_{SET1} = 1, G_{SET2} = 1
Power supply Rejection ratio*2	PSRR	-	70	-	dB	AVCC1 = 5.0 V + 0.1 V_{pp_ripple}, f_{in} = 50 Hz, G_{SET1} = 1, G_{SET2} = 1, excluding SBIAS error or VREFI error
Input absolute current*2	I_{IN}	-	2	-	nA	V_I = 1 V
Input offset current*2	I_{INOFR}	-	1	-	nA	V_{ID} = 0 V, V_{COM} = 1 V
Input impedance*2	Z_{IN}	-	500	-	Mohm	V_{ID} = 1 V, V_{COM} = 1 V
Offset adjust gain error*2	d_{OFGE}	-5	-	5	%	Including SBIAS error, d_{OFR} ≠ 0 mV
Offset adjust integral non-linearity*2	dOFINL	-0.5	-	0.5	LSB	d_{OFR} ≠ 0 mV

Note: The characteristics apply when no pin functions other than 24-bit sigma-delta A/D converter input are used.
Note 1. Gain drift is calculated by (Max (EG (T (-40°C) to T (125°C))) - Min (EG (T (-40°C) to T (125°C)))) / (125°C - (-40°C))
Offset drift is calculated by (Max (EOS (T (-40°C) to T (125°C))) - Min (EOS (T (-40°C) to T (125°C)))) / (125°C - (-40°C)).
Note 2. Not production tested but is guaranteed by the design and characterization.

Table 47.51 2.1 V LDO linear regulator for ADC (ADREG) characteristics

Conditions: VCC = AVCC0 = AVCC1 = 2.7 to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V
Connect the ADREG pin to a AVSS1 pin by a 0.47 μF (-50% to +20%) capacitor.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
ADREG output voltage	V_{ADREG}	-	2.1	-	V	-

Table 47.52 ADC external reference voltage (VREFI) characteristics

Conditions: VCC = AVCC0 = AVCC1 = 2.7 to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
External reference voltage range*1	VREFI	0.8	-	2.4	V	SDADCSTC1.VREFSEL = 1
External reference voltage step	VR_{STEP}	-	0.2	-	V	SDADCSTC1.VREFSEL = 1
External reference voltage accuracy	VR_A	-3	-	3	%	SDADCSTC1.VREFSEL = 1

Note 1. Select the reference voltage input value with STC1.VSBIAS[3:0].

Table 47.53 Sensor bias (SBIAS) characteristics
Conditions: VCC = AVCC0 = AVCC1 = 2.7 to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V
Connect the SBIAS/VREFI pin to a AVSS1 pin by a 0.22 μF (-20% to +20%)

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage range*2	SBIAS	0.8	-	2.2	V	-
Output voltage step	SV_{STEP}	-	0.2	-	V	-
Output voltage accuracy*1	SV_A	-3	-	3	%	SI_{OUT} = 1 mA
Output current*1	SI_{OUT}	-	-	10	mA	-
Short current*1	SI_{SHORT}	-	35	65	mA	SBIAS = 0 V
Load regulation*1	SL_R	-	-	15	mV	1 mA ≤ SI_{OUT} ≤ 5 mA
		-	-	20	mV	1 mA ≤ SI_{OUT} ≤ 10 mA
Power supply rejection ratio*1	SPSRR	-	50	-	dB	AVCC1 = 5.0 V + 0.1 V_{pp_ripple}, f = 100 Hz, SI_{OUT} = 2.5 mA
Transition time of one step*1,*3	ST_{TS}	-	-	80	μs	SBIAS < SV_A ± 3%
						1 mA ≤ SI_{OUT} ≤ SI_{OUT_MAX}

Note 1. Not production tested but is guaranteed by the design and characterization.
Note 2. Select the reference voltage output value for the sensor with STC1.VSBIAS[3:0].
Note 3. The load current of more than 1 mA is required because the output stage of SBIAS is Pch open drain. When the original load current is small, additional external load resistance is required.

47.7 DAC12 Characteristics

Table 47.54 12-bit D/A conversion characteristics
Conditions: VCC = AVCC0 = AVCC1 = 1.7 V to 5.5 V, VREFH = 1.7 V to 5.5 V, VSS = AVSS0 = AVSS1 = VREFL = 0 V

Parameter		Min	Typ	Max	Unit	Test conditions
Resolution		-	-	12	bit	-
Charge pump stabilization time*1		-	-	100	μs	-
SW stabilization time*1		-	-	50	μs	-
Conversion time*1	DAC Ref. = AVCC or VREFH ≥ 2.7 V	-	-	1.0	μs	Cload = 38 pF, @ 1 LSB step Cload = 8 pF, @ full range
	DAC Ref. = AVCC or VREFH < 2.7 V	-	-	1.2		-
Wake-up time*1		-	-	1.0	μs	-
Absolute accuracy		-	-	± 12	LSB	2-MΩ resistive load
DNL differential non-linearity error	DAC Ref. = AVCC or VREFH ≥ 2.7 V	-	-	±1.0	LSB	-
	DAC Ref. = AVCC or VREFH < 2.7 V	-	-	±2.0		-
INL integral non-linearity error		-	-	±7.0	LSB	-
RO output resistance		-	3.5	-	kΩ	-
Load resistance		2	2	-	MΩ	-
Load capacitance	1 LSB step	-	38	-	pF	-
	Full range	-	8	-		-

Note 1. These values are based on simulation. They are not production tested.

47.8 DAC8 Characteristics

Table 47.55 8-bit D/A conversion characteristics (1 of 2)
Conditions: VCC = AVCC0 = AVCC1 = 1.7 V to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V

Parameter	Min	Typ	Max	Unit	Test conditions
Resolution	-	-	8	bit	-
Charge pump stabilization time*1	-	-	100	μs	-

Table 47.55 8-bit D/A conversion characteristics (2 of 2)
Conditions: VCC = AVCC0 = AVCC1 = 1.7 V to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V

Parameter		Min	Typ	Max	Unit	Test conditions
Switch stabilization time*1		-	-	50	μs	-
Conversion time*1	AVCC0 = 2.7 to 5.5 V	-	-	3.0	μs	35-pF capacitive load
	AVCC0 = 1.7 to 2.7 V	-	-	6.0	μs	
Absolute accuracy	AVCC0 = 2.7 to 5.5 V	-	-	± 3.0	LSB	2-MΩ resistive load
	AVCC0 = 1.7 to 2.7 V	-	-	± 3.5		
	AVCC0 = 2.7 to 5.5 V	-	-	± 2.0	LSB	4-MΩ resistive load
	AVCC0 = 1.7 to 2.7 V	-	-	± 2.5		
RO output resistance		-	7.4	-	kΩ	-

Note 1. These values are based on simulation. They are not production tested.

47.9 TSN Characteristics

Table 47.56 TSN characteristics
Conditions: VCC = AVCC0 = AVCC1 = 2.0 to 5.5 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Relative accuracy	-	-	± 1.5	-	°C	2.4 V or above
		-	± 2.0	-	°C	Below 2.4 V
Temperature slope	-	-	-3.65	-	mV/°C	-
Output voltage (at 25°C)	-	-	1.05	-	V	VCC = 3.3 V
Temperature sensor start time	t_{START}	-	-	5	μs	-
Sampling time	-	5	-	-	μs	

47.10 OSC Stop Detect Characteristics

Table 47.57 Oscillation stop detection circuit characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Detection time	t_{dr}	-	-	1	ms	Figure 47.65

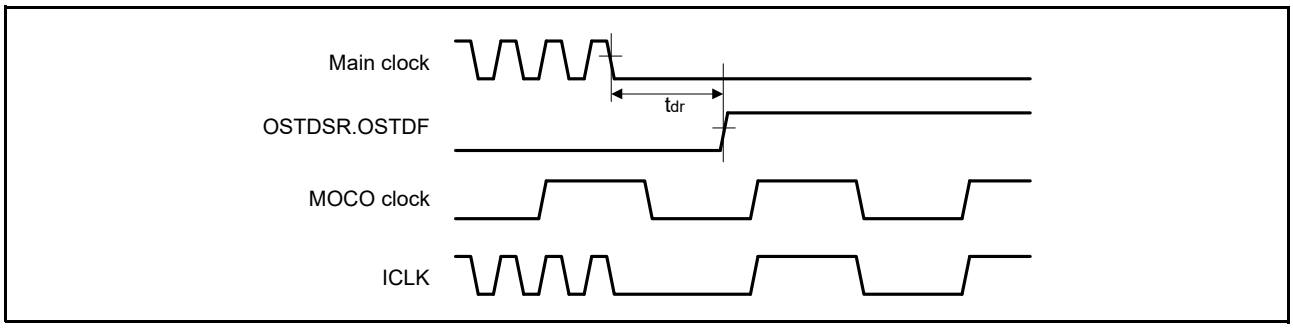


Figure 47.65 Oscillation stop detection timing

47.11 POR and LVD Characteristics

Table 47.58 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 47.66, Figure 47.67
	Voltage detection circuit (LVD0)*2	V_{det0_0}	3.68	3.85	4.00	V	Figure 47.68 At falling edge VCC
		V_{det0_1}	2.68	2.85	2.96		
		V_{det0_2}	2.38	2.53	2.64		
		V_{det0_3}	1.78	1.90	2.02		
		V_{det0_4}	1.60	1.69	1.82		
	Voltage detection circuit (LVD1)*3	V_{det1_0}	4.13	4.29	4.45	V	Figure 47.69 At falling edge VCC
		V_{det1_1}	3.98	4.16	4.30		
		V_{det1_2}	3.86	4.03	4.18		
		V_{det1_3}	3.68	3.86	4.00		
		V_{det1_4}	2.98	3.10	3.22		
		V_{det1_5}	2.89	3.00	3.11		
		V_{det1_6}	2.79	2.90	3.01		
		V_{det1_7}	2.68	2.79	2.90		
		V_{det1_8}	2.58	2.68	2.78		
		V_{det1_9}	2.48	2.58	2.68		
		V_{det1_A}	2.38	2.48	2.58		
		V_{det1_B}	2.10	2.20	2.30		
		V_{det1_C}	1.84	1.96	2.05		
		V_{det1_D}	1.74	1.86	1.95		
		V_{det1_E}	1.63	1.75	1.84		
		V_{det1_F}	1.60	1.65	1.73		
	Voltage detection circuit (LVD2)*4	V_{det2_0}	4.11	4.31	4.48	V	Figure 47.70 At falling edge VCC
		V_{det2_1}	3.97	4.17	4.34		
		V_{det2_2}	3.83	4.03	4.20		
		V_{det2_3}	3.64	3.84	4.01		

Note 1. These characteristics apply when noise is not superimposed on the power supply. When a setting causes this voltage detection level to overlap with that of the voltage detection circuit, it cannot be specified whether LVD1 or LVD2 is used for voltage detection.

Note 2. # in the symbol $V_{det0_\#}$ denotes the value of the OFS1.VDSEL1[2:0] bits.

Note 3. # in the symbol $V_{det1_\#}$ denotes the value of the LVDLVLR.LVD1LVL[4:0] bits.

Note 4. # in the symbol $V_{det2_\#}$ denotes the value of the LVDLVLR.LVD2LVL[2:0] bits.

Table 47.59 Power-on reset circuit and voltage detection circuit characteristics (2)

Parameter		Symbol	Min	Typ	Max	Unit	Test Conditions
Wait time after power-on reset cancellation	LVD0: enable	t_{POR}	-	1.7	-	ms	-
	LVD0: disable	t_{POR}	-	1.3	-	ms	-
Wait time after voltage monitor 0,1,2 reset cancellation	LVD0: enable*1	$t_{LVD0,1,2}$	-	0.6	-	ms	-
	LVD0: disable*2	$t_{LVD1,2}$	-	0.2	-	ms	-
Response delay*3		t_{det}	-	-	350	μs	Figure 47.66, Figure 47.67
Minimum VCC down time		t_{VOFF}	450	-	-	μs	Figure 47.66, VCC = 1.0 V or above
Power-on reset enable time		$t_{W\ (POR)}$	1	-	-	ms	Figure 47.67, VCC = below 1.0 V
LVD operation stabilization time (after LVD is enabled)		$T_{d\ (E-A)}$	-	-	300	μs	Figure 47.69, Figure 47.70
Hysteresis width (POR)		V_{PORH}	-	110	-	mV	-
Hysteresis width (LVD0, LVD1 and LVD2)		V_{LVH}	-	60	-	mV	LVD0 selected
			-	100	-		V_{det1_0} to V_{det1_2} selected
			-	60	-		V_{det1_3} to V_{det1_9} selected
			-	50	-		V_{det1_A} to V_{det1_B} selected
			-	40	-		V_{det1_C} to V_{det1_F} selected
			-	60	-		LVD2 selected

Note 1. When OFS1.LVDAS = 0.
Note 2. When OFS1.LVDAS = 1.
Note 3. The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels V_{POR}, V_{det0}, V_{det1}, and V_{det2} for the POR/LVD.

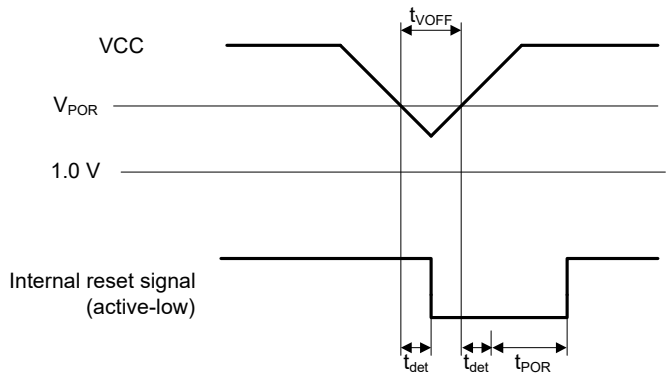
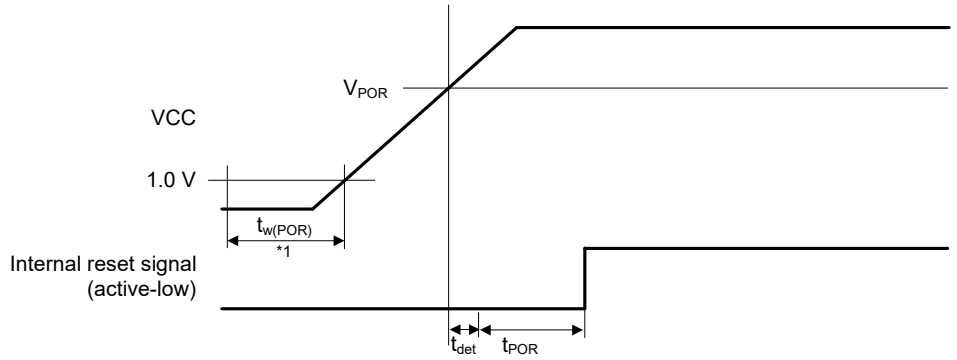


Figure 47.66 Voltage detection reset timing



Note 1. $t_{w(POR)}$ is the time required for a power-on reset to be enabled while the external power VCC is being held below the valid voltage (1.0 V).
When VCC turns on, maintain $t_{w(POR)}$ for 1.0 ms or more.

Figure 47.67 Power-on reset timing

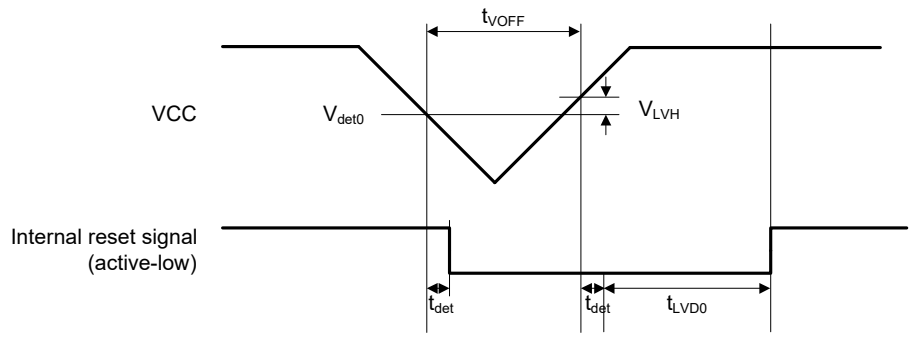


Figure 47.68 Voltage detection circuit timing (V_{det0})

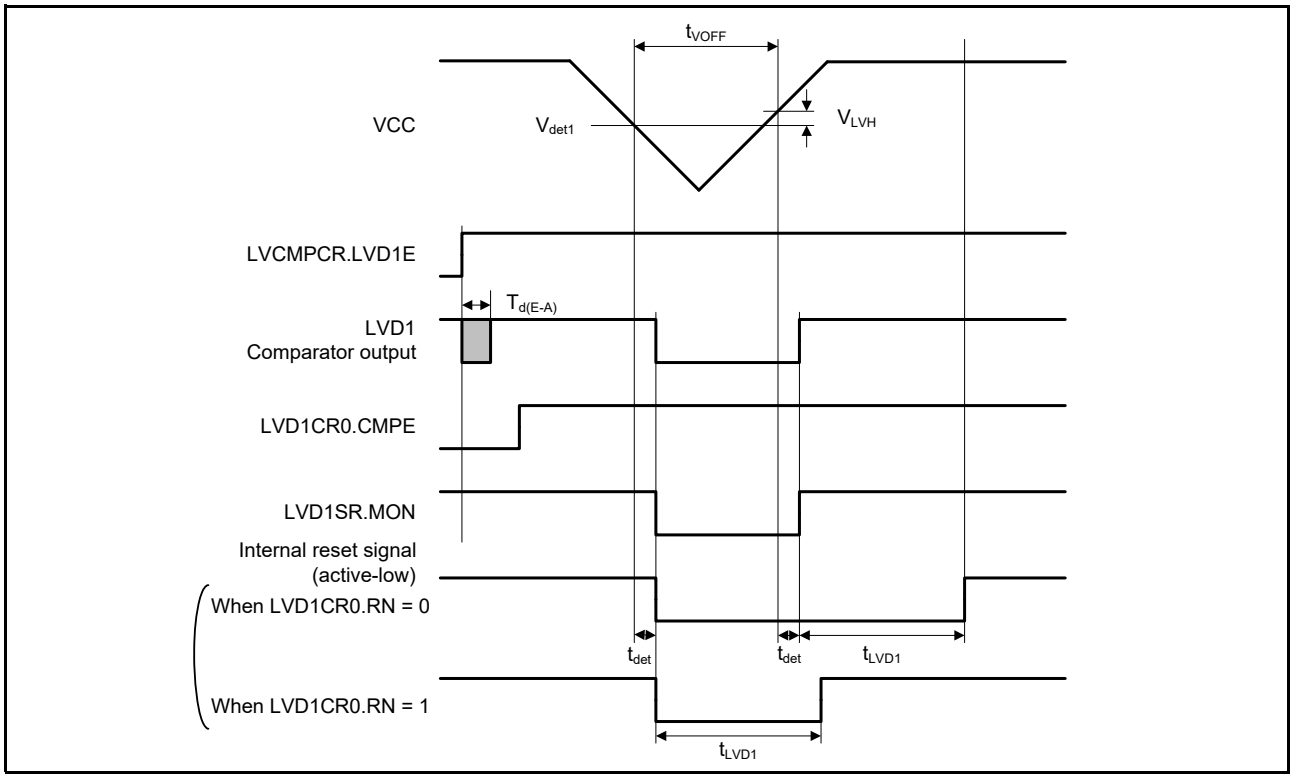


Figure 47.69 Voltage detection circuit timing (V_{det1})

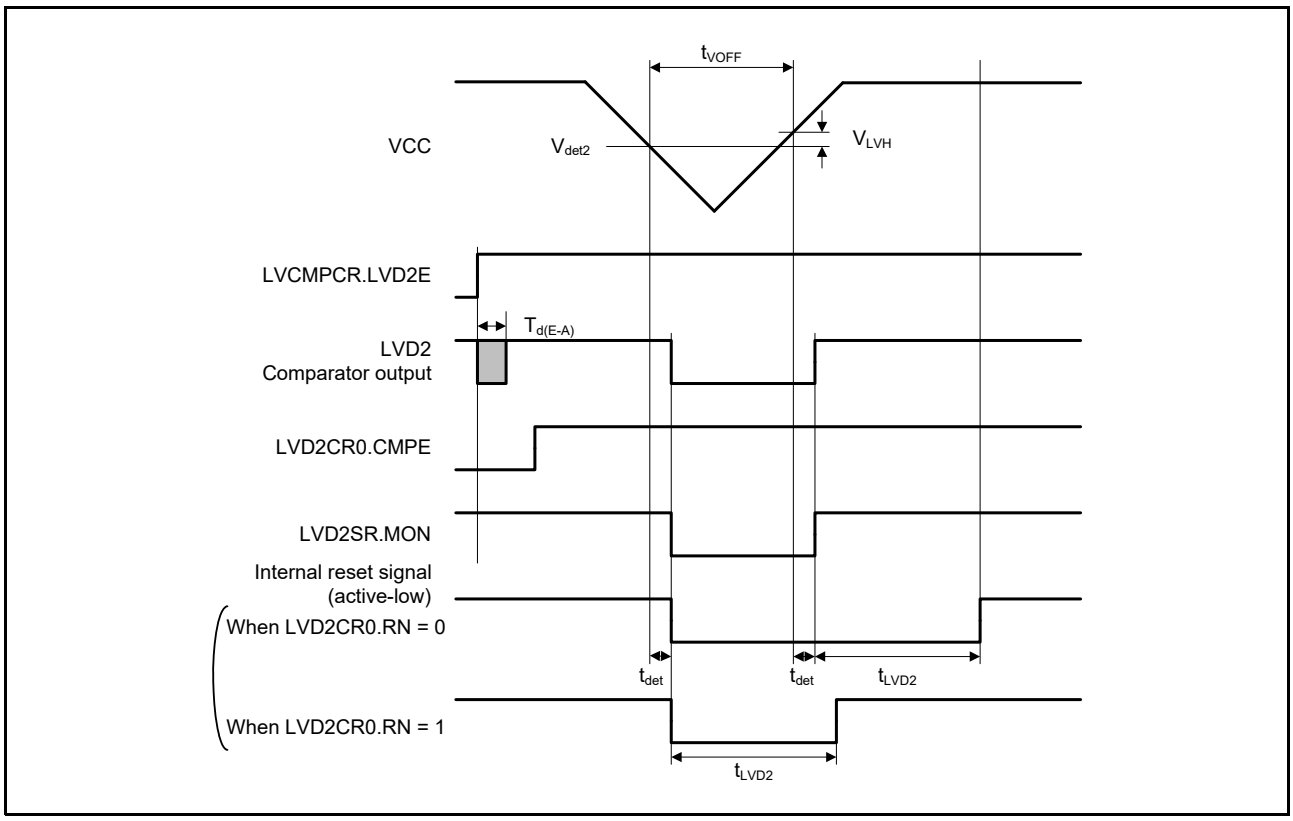


Figure 47.70 Voltage detection circuit timing (V_{det2})

47.12 CTSU Characteristics

Table 47.60 CTSU characteristics
Conditions: VCC = AVCC0 = AVCC1 = 1.8 to 5.5 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
External capacitance connected to TSCAP pin	C_{tscap}	9	10	11	nF	-
TS pin capacitive load	C_{base}	-	-	50	pF	-
Permissible output high current	ΣI_{OH}	-	-	-24	mA	When the mutual capacitance method is applied and TS07 to TS14 are not used for transmit channel
		-	-	-14		When the mutual capacitance method is applied and TS07 to TS14 are used for transmit channel

47.13 Comparator Characteristics

Table 47.61 ACMPHS characteristics
Conditions: VCC = AVCC0 = AVCC1 = 2.7 to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Input offset voltage	V_{IOCMP}	-	± 5	± 40	mV	-
Input voltage range	V_{ICPM}	0	-	AVCC0	V	-
Internal reference voltage input*3	Vref	1.36	1.43	1.50	V	AVCC0 ≥ 2.0 V
Input signal cycle	t_{PCMP}	10	-	-	μs	-
Output delay time	T_d	-	50	100	ns	Input amplitude ± 100 mV
Stabilization wait time during input channel switching*1	T_{WAIT}	300	-	-	ns	Input amplitude ± 100 mV
Operation stabilization wait time*2	T_{cmp}	1	-	-	μs	3.3 V ≤ AVCC0 ≤ 5.5 V
		3	-	-	μs	2.7 V ≤ AVCC0 < 3.3 V

Note 1. Period from when the comparator input channel is switched until the switched result reflects in its output.
Note 2. Period from when comparator operation is enabled (CPMCTL.HCMPON = 1) until the comparator satisfies the DC/AC characteristics.
Note 3. The internal reference voltage cannot be selected for input channels when AVCC0 < 2.0 V.

Table 47.62 ACMPLP characteristics
Conditions: VCC = AVCC0 = AVCC1 = 1.8 to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input voltage range	IVREF0	V_{REF}	0	-	VCC - 1.4*1	V	-
	IVREF1 (Standard mode)		0	-	VCC - 1.4	V	
	IVREF1 (Window mode)		1.4*1	-	VCC	V	
	IVCMP0, IVCMP1	V_I	0	-	VCC	V	
Internal reference voltage*2		-	1.36	1.43	1.50	V	VCC ≥ 2.0 V
Output delay	Comparator high-speed mode (Standard mode)	T_d	-	-	1.2	μs	VCC = 3.0 V Slew rate of input signal > 50 mV/μs
	Comparator high-speed mode (Window mode)		-	-	2.0	μs	
	Comparator low-speed mode (Standard mode)		-	-	5.0	μs	

Table 47.62 ACMPLP characteristics
Conditions: VCC = AVCC0 = AVCC1 = 1.8 to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Offset voltage	Comparator high-speed mode (Standard mode)	-	-	-	50	mV	-
	Comparator high-speed mode (Window mode)		-	-	60	mV	
	Comparator low-speed mode (Standard mode)		-	-	40	mV	
Operation stabilization wait time		T_{cmp}	100	-	-	μs	-

Note 1. In window mode, be sure to satisfy the following condition: $V_{IVREF1} - V_{IVREF0} \geq 0.2$ V.
Note 2. The internal reference voltage cannot be selected for input channels when VCC < 2.0 V.

47.14 OPAMP Characteristics

Table 47.63 OPAMP characteristics (1 of 3)
Conditions: VCC = AVCC0 = AVCC1 = 1.7 V to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Supply voltage range	AVCC0	Low power mode	1.7	-	5.5	V
		Middle-speed mode	2.1	-	5.5	V
		High-speed mode	2.4	-	5.5	V
Charge pump stabilization time*1	-	-	-	-	100	μs
SW stabilization time*1	-	-	-	-	50	μs
Input voltage range	V_{icm1}	Low power mode	AVSS0	-	AVCC0	V
	V_{icm2}	Middle-speed mode				
	V_{icm3}	High-speed mode				
Output voltage range	V_{olh1}	Low power mode, Ilode = 100 μA	AVSS0	-	AVCC0	V
	V_{olh2}	Middle-speed mode, Iload = 100 μA				
	V_{olh3}	High-speed mode, Iload = 100 μA				
Input offset trimming range*1	$V_{offadj2l}$	Middle-speed mode, Vin = 0.1 V, Tj = 25°C	-3	-	3	mV
	$V_{offadj2h}$	Middle-speed mode, Vin = AVCC0 - 0.1 V, Tj = 25°C				
	$V_{offadj3l}$	High-speed mode, Vin = 0.1 V, Tj = 25°C				
	$V_{offadj3h}$	High-speed mode, Vin = AVCC0 - 0.1 V, Tj = 25°C				

Table 47.63 OPAMP characteristics (2 of 3)
Conditions: VCC = AVCC0 = AVCC1 = 1.7 V to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Input offset*1	V_{ioff1a}	Low power mode, Vin < AVCC0 - 1.0 V	-5.0	-	5.0	mV
	V_{ioff1b}	Low power mode, Vin ≥ AVCC0 - 1.0 V	-8.0	-	8.0	
	V_{ioff2a}	Middle-speed mode, Vin < AVCC0 - 1.2 V	-3.0	-	3.0	
	V_{ioff2b}	Middle-speed mode, Vin ≥ AVCC0 - 1.2 V	-3.0	-	3.0	
	V_{ioff3a}	High-speed mode, Vin < AVCC0 - 1.2 V	-2.5	-	2.5	
	V_{ioff3b}	High-speed mode, Vin ≥ AVCC0 - 1.2 V	-2.5	-	2.5	
Offset drift*1	Drift1a	Low power mode, Vin < AVCC0 - 1.0 V	-70	-	70	μV/°C
	Drift1b	Low power mode, Vin ≥ AVCC0 - 1.0 V	-70	-	70	
	Drift2a	Middle-speed mode, Vin < AVCC0 - 1.2 V	-30	-	30	
	Drift2b	Middle-speed mode, Vin ≥ AVCC0 - 1.2 V	-30	-	30	
	Drift3a	High-speed mode, Vin < AVCC0 - 1.2 V	-30	-	30	
	Drift3b	High-speed mode, Vin ≥ AVCC0 - 1.2 V	-30	-	30	
Open gain*1	Av1	Low power mode	70	130	-	dB
	Av2	Middle-speed mode	70	120	-	
	Av3	High-speed mode	60	130	-	
Gain bandwidth product*1	GBW1	Low power mode	-	90	-	kHz
	GBW2	Middle-speed mode	-	2	-	MHz
	GBW3	High-speed mode	-	4.8	-	MHz
Phase margin*1	PM1	Low power mode	35	-	-	deg
	PM2	Middle-speed mode	35	-	-	
	PM3	High-speed mode	35	-	-	
Gain margin*1	GM1	Low power mode	10	-	-	dB
	GM2	Middle-speed mode	10	-	-	
	GM3	High-speed mode	10	-	-	
Input noise density*1	V_{ind11}	Low power mode, f = 10 Hz	-	860	-	nV/√Hz
	V_{ind12}	Low power mode, f = 1 kHz	-	260	-	
	V_{ind21}	Middle-speed mode, f = 1 kHz	-	50	-	
	V_{ind22}	Middle-speed mode, f = 100 kHz	-	30	-	
	V_{ind31}	High-speed mode, f = 1 kHz	-	40	-	
	V_{ind32}	High-speed mode, f = 100 kHz	-	20	-	

Table 47.63 OPAMP characteristics (3 of 3)
Conditions: VCC = AVCC0 = AVCC1 = 1.7 V to 5.5 V, VSS = AVSS0 = AVSS1 = 0 V

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Power supply rejection ratio*1	PSRR1	Low power mode	-	90	-	dB
	PSRR2	Middle-speed mode	-	90	-	
	PSRR3	High-speed mode	-	90	-	
Common mode rejection ratio*1	CMRR1	Low power mode	-	90	-	dB
	CMRR2	Middle-speed mode	-	90	-	
	CMRR3	High-speed mode	-	90	-	
Settling time*1	T_{set1}	Low power mode	-	70	200	μS
	T_{set2}	Middle-speed mode	-	2.8	8	
	T_{set3}	High-speed mode	-	1.2	3.2	
Slew rate*1	SR1	Low power mode	0.02	0.05	-	V/μS
	SR2	Middle-speed mode	0.8	1.3	-	
	SR3	High-speed mode	1.8	3.0	-	
Turn on time*1	T_{turn1}	Low power mode, AMPENx = 0 → 1, IREFEN = 0 → 1	-	80	220	μS
	T_{turn2}	Middle-speed mode, AMPENx = 0 → 1, IREFEN = 0 → 1	-	3	10	
	T_{turn3}	High-speed mode, AMPENx = 0 → 1, IREFEN = 0 → 1	-	1.3	4	
Input offset trimming step*1	$V_{ioffst2}$	Middle-speed mode, Vin < AVCC0 - 1.2 V	0.3	0.459	0.58	mV/code
		Middle-speed mode, Vin ≥ AVCC0 - 1.2 V	0.24	-	0.56	
	$V_{ioffst3}$	High-speed mode, Vin < AVCC0 - 1.2 V	0.35	0.52	0.65	
		High-speed mode, Vin ≥ AVCC0 - 1.2 V	0.28	-	0.61	
Wait time after trimming*1	T_{turn_tm2}	Middle-speed mode	-	-	1.5	μS
	T_{turn_tm3}	High-speed mode	-	-	1	
Load current	I_{load}	-	-	-	100	μA
Load capacitance	C_L	-	-	-	20	pF

Note 1. These values are based on simulation. They are not production tested.

47.15 Flash Memory Characteristics

47.15.1 Code Flash Memory Characteristics

Table 47.64 Code flash characteristics (1)

Parameter		Symbol	Min	Typ	Max	Unit	Conditions
Reprogramming/erasure cycle*1		N_{PEC}	1000	-	-	Times	-
Data hold time	After 1000 times 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 performed n times for each block. For instance, when 4-byte programming is performed 256 times for different addresses in 1-KB blocks, and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasure is not enabled (overwriting is prohibited).
Note 2. 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 47.65 Code flash characteristics (2)
High-speed operating mode
Conditions: VCC = AVCC0 = AVCC1 = 2.7 to 5.5 V

Parameter		Symbol	FCLK = 1 MHz			FCLK = 32 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	8-byte	t_{P8}	-	116	998	-	54	506	μs
Erasure time	2-KB	t_{E2K}	-	9.03	287	-	5.67	222	ms
Blank check time	8-byte	t_{BC8}	-	-	56.8	-	-	16.6	μs
	2-KB	t_{BC2K}	-	-	1899	-	-	140	μs
Erase suspended time		t_{SED}	-	-	22.5	-	-	10.7	μs
Startup area switching setting time		t_{SAS}	-	21.9	585	-	12.1	447	ms
Access window time		t_{AWS}	-	21.9	585	-	12.1	447	ms
OCD/serial programmer ID setting time		t_{OSIS}	-	21.9	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% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.

Table 47.66 Code flash characteristics (3)
Middle-speed operating mode
Conditions: VCC = AVCC0 = AVCC1 = 1.8 to 5.5 V, Ta = -40 to +85°C

Parameter		Symbol	FCLK = 1 MHz			FCLK = 8 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	8-byte	t_{P8}	-	157	1411	-	101	966	μs
Erasure time	2-KB	t_{E2K}	-	9.10	289	-	6.10	228	ms
Blank check time	8-byte	t_{BC8}	-	-	87.7	-	-	52.5	μs
	2-KB	t_{BC2K}	-	-	1930	-	-	414	μs
Erase suspended time		t_{SED}	-	-	32.7	-	-	21.6	μs
Startup area switching setting time		t_{SAS}	-	22.8	592	-	14.2	465	ms
Access window time		t_{AWS}	-	22.8	592	-	14.2	465	ms
OCD/serial programmer ID setting time		t_{OSIS}	-	22.8	592	-	14.2	465	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% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.

47.15.2 Data Flash Memory Characteristics

Table 47.67 Data flash characteristics (1)

Parameter		Symbol	Min	Typ	Max	Unit	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 47.68 Data flash characteristics (2)
High-speed operating mode
Conditions: VCC = AVCC0 = AVCC1 = 2.7 to 5.5 V

Parameter		Symbol	FCLK = 4 MHz			FCLK = 32 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	1-byte	t_{DP1}	-	52.4	463	-	42.1	387	μs
Erasure time	1-KB	t_{DE1K}	-	8.98	286	-	6.42	237	ms
Blank check time	1-byte	t_{DBC1}	-	-	24.3	-	-	16.6	μs
	1-KB	t_{DBC1K}	-	-	1872	-	-	512	μs
Suspended time during erasing		t_{DSED}	-	-	13.0	-	-	10.7	μs
Data flash STOP recovery time		t_{DSTOP}	5	-	-	5	-	-	μs

Note: 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% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.

Table 47.69 Data flash characteristics (3)
Middle-speed operating mode
Conditions: VCC = AVCC0 = AVCC1 = 1.8 to 5.5 V, Ta = -40 to +85°C

Parameter		Symbol	FCLK = 4 MHz			FCLK = 8 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	1-byte	t_{DP1}	-	94.7	886	-	89.3	849	μs
Erasure time	1-KB	t_{DE1K}	-	9.59	299	-	8.29	273	ms
Blank check time	1-byte	t_{DBC1}	-	-	56.2	-	-	52.5	μs
	1-KB	t_{DBC1K}	-	-	2.17	-	-	1.51	ms
Suspended time during erasing		t_{DSED}	-	-	23.0	-	-	21.7	μs
Data flash STOP recovery time		t_{DSTOP}	720	-	-	720	-	-	ns

Note: 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% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.

47.15.3 Serial Wire Debug (SWD)

Table 47.70 SWD characteristics (1) (1 of 2)
Conditions: VCC = AVCC0 = AVCC1 = 2.4 to 5.5 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
SWCLK clock cycle time	$t_{SWCKcyc}$	80	-	-	ns	Figure 47.71
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	

Table 47.70 SWD characteristics (1) (2 of 2)
Conditions: VCC = AVCC0 = AVCC1 = 2.4 to 5.5 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
SWDIO setup time	t_{SWDS}	16	-	-	ns	Figure 47.72
SWDIO hold time	t_{SWDH}	16	-	-	ns	
SWDIO data delay time	t_{SWDD}	2	-	70	ns	

Table 47.71 SWD characteristics (2)
Conditions: VCC = AVCC0 = AVCC1 = 1.6 to 2.4 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
SWCLK clock cycle time	$t_{SWCKcyc}$	250	-	-	ns	Figure 47.71
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 47.72
SWDIO hold time	t_{SWDH}	50	-	-	ns	
SWDIO data delay time	t_{SWDD}	2	-	150	ns	

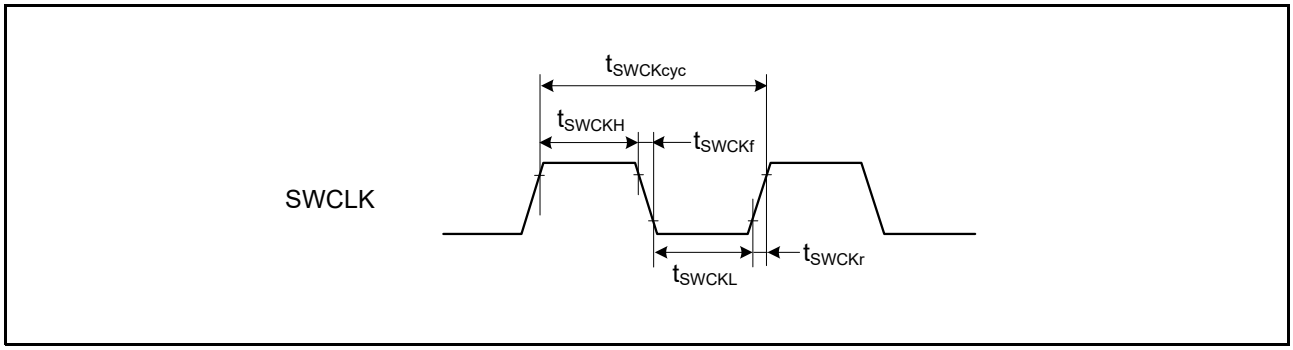


Figure 47.71 SWD SWCLK timing

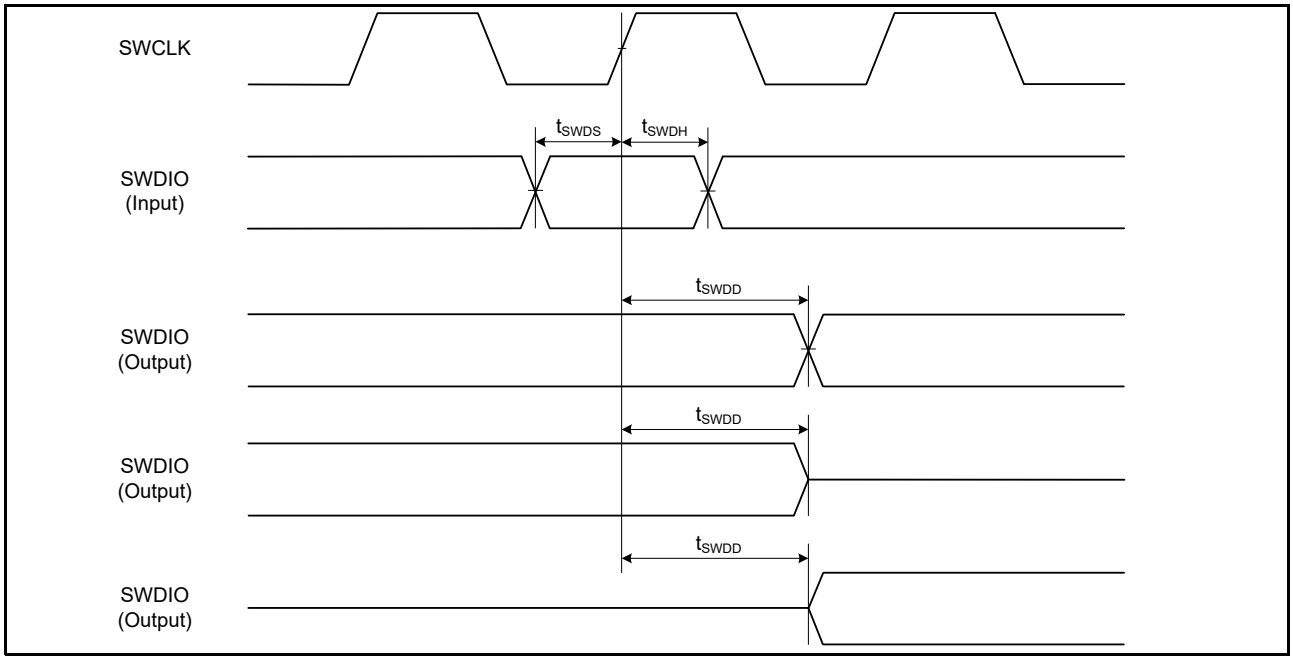


Figure 47.72 SWD input/output timing

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
P000/KR00/IRQ4/AGTIO1_A/SCL0_C	Hi-Z	Keep-O*1
P001/IRQ0/RTCOUT_D	Hi-Z	[RTCOUT selected] RTCOUT output [Other than the above] Keep-O*1
P002/DA8_1	Hi-Z	[DA8_1 output (DACE1 = 1)] DA8_1 output retained [Other than the above (DACE1 = 0)] Keep-O
P003	Hi-Z	Keep-O
P012/AMP2O	Hi-Z	[AMP2O selected] AMP2O output [Other than the above] Keep-O
P013/DA8_0	Hi-Z	[DA8_0 output (DACE0 = 1)] DA8_0 output retained [Other than the above (DACE0 = 0)] Keep-O
P014	Hi-Z	Keep-O
P015/AMP1O	Hi-Z	[AMP1O selected] AMP1O output [Other than the above] Keep-O
P100/IRQ4, P101/IRQ5	Hi-Z	Keep-O*1
P102, P103	Hi-Z	Keep-O
P104/IRQ6, P105/IRQ7	Hi-Z	Keep-O*1
P106, P107	Hi-Z	Keep-O
P108/SWDIO	Pull-up	Keep-O
P109/CMPREF0/KR01/IRQ3/AGTOA0_A/ VCOUT_A	Hi-Z	[AGTOA0_A selected] AGTOA0_A output*2 [ACMPLP or ACMPHS selected] VCOUT_A output [Other than the above] Keep-O*1
P110/IRQ2/AGTOB0_A/CLKOUT_A	Hi-Z	[AGTOB0_A selected] AGTOB0_A output*2 [CLKOUT selected] CLKOUT output [Other than the above] Keep-O*1
P111/IRQ6/RTCOUT_B	Hi-Z	[RTCOUT selected] RTCOUT output [Other than the above] Keep-O*1
P112/IRQ7/CLKOUT_B	Hi-Z	[CLKOUT selected] CLKOUT output [Other than the above] Keep-O*1
P200/NMI	Hi-Z	Hi-Z
P201	Pull-up	Keep-O
P204/RXD0_C	Hi-Z	Keep-O*1
P205/IRQ0	Hi-Z	Keep-O*1
P206/IRQ6/AGTIO0_B	Hi-Z	[AGTIO0_B output selected] AGTIO0_B output*2 [Other than the above] Keep-O*1
P212/IRQ3/AGTIO0_A/EXTAL	Hi-Z	[AGTIO0_A output selected] AGTIO0_A output*2 [Other than the above] Keep-O*1

Table 1.1 Port states in each processing mode (2 of 2)

Port name	Reset	Software Standby Mode
P213/IRQ2/XTAL	Hi-Z	Keep-O*1
P214/XCOUT, P215/XCIN	Hi-Z	[Sub-clock Oscillator selected] Sub-clock Oscillator is operating [Other than the above] Hi-Z
P300/SWCLK	Pull-up	Keep-O
P301/KR04/IRQ5/AGTOB1_A/SDA0_A/ RTCOUT_A/RXD0_B	Hi-Z	[AGTOB1_A selected] AGTOB1_A output*2 [RTCOUT selected] RTCOUT output [Other than the above] Keep-O*1
P302/KR05/IRQ4/AGTOA1_A	Hi-Z	[AGTOA1_A selected] AGTOA1_A output*2 [Other than the above] Keep-O*1
P303/KR06, P304/KR07	Hi-Z	Keep-O*1
P400/KR02/IRQ0/CMPIN0/RTCOUT_C	Hi-Z	[RTCOUT selected] RTCOUT output [Other than the above] Keep-O*1
P401/KR03/IRQ5/SDA0_C/VCOUT_B	Hi-Z	[ACMPLP or ACMPHS selected] VCOUT_B output [Other than the above] Keep-O*1
P402, P403	Hi-Z	Keep-O
P407/IRQ1/USB_VBUS/AGTIO0_C/ SCL0_A	Hi-Z	[AGTIO0_C output selected] AGTIO0_C output*2 [Other than the above] Keep-O*1
P408/IRQ1/AGTO0_A/SDA0_B/RXD0_A	Hi-Z	[AGTO0_A selected] AGTO0_A output*2 [Other than the above] Keep-O*1
P409/IRQ7/AGTO1_A/SCL0_B	Hi-Z	[AGTO1_A selected] AGTO1_A output*2 [Other than the above] Keep-O*1
P410, P411	Hi-Z	Keep-O
P500/IRQ3/DA12_0/RXD0_D	Hi-Z	[DA12_0 output (DAOE0 = 1)] DA12_0 output retained [Other than the above (DAOE0 = 0)] Keep-O*1
P501/IRQ2	Hi-Z	Keep-O*1
P502/IRQ1/AMP0O	Hi-Z	[AMP0O selected] AMP0O output [Other than the above] Keep-O*1
P914/USB_DP	Hi-Z	Keep-O
P915/USB_DM	Hi-Z	Keep-O

Hi-Z: High-impedance
Keep-O: Output pins retain their previous values. Input pins become high-impedance.

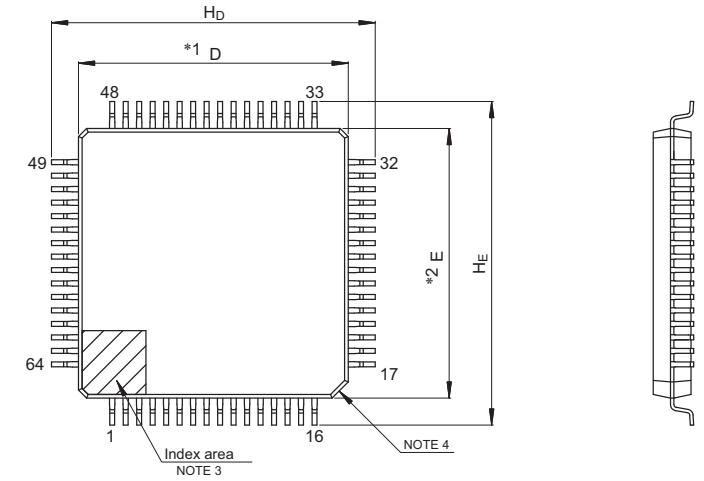
Note 1. Input is enabled if the pin is specified as the software standby canceling source while it is used as an external interrupt pin.
Note 2. AGTIO output is enabled while LOCO or SOSC is selected as a count source.

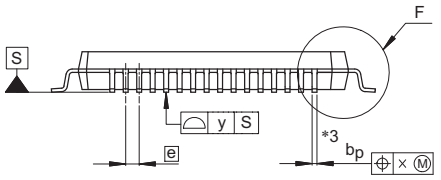
Appendix 2.Package Dimensions

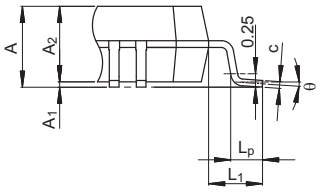
Information on the latest version of the package dimensions or mountings is displayed in “Packages” on the Renesas Electronics Corporation website.

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP64-10x10-0.50	PLQP0064KB-C	—	0.3

Unit: mm







Detail F

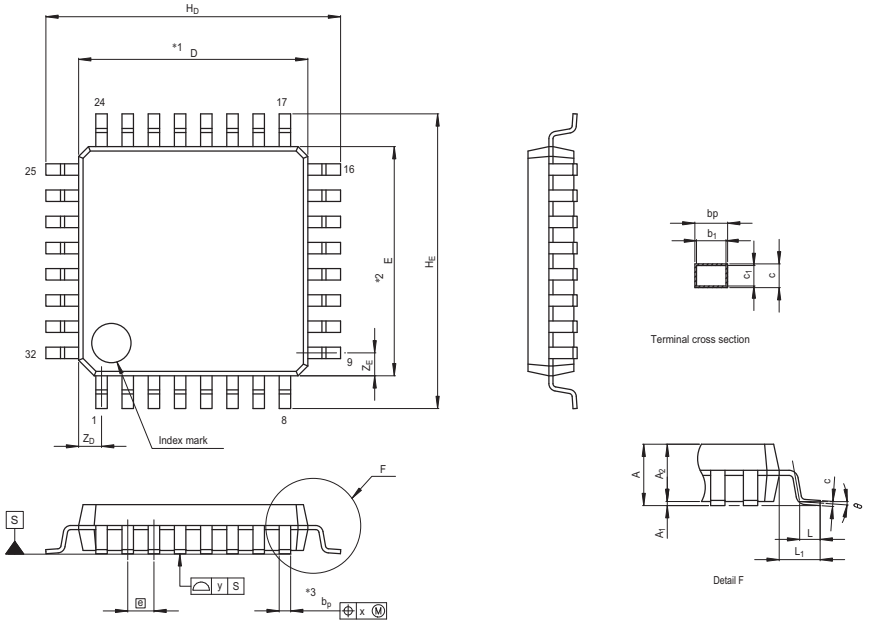
NOTE)
1. DIMENSIONS “*1” AND “*2” DO NOT INCLUDE MOLD FLASH.
2. DIMENSION “*3” DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A_2	—	1.4	—
H_D	11.8	12.0	12.2
H_E	11.8	12.0	12.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

© 2015 Renesas Electronics Corporation. All rights reserved.

Figure 2.1 LQFP 64-pin

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP32-7x7-0.80	PLQP0032GB-A	32P6U-A	0.2g

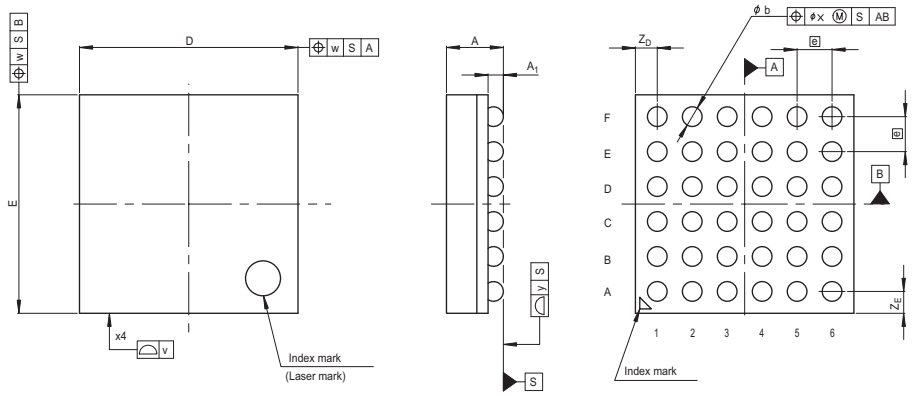


NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A_2	—	1.4	—
H_D	8.8	9.0	9.2
H_E	8.8	9.0	9.2
A	—	—	1.7
A_1	0	0.1	0.2
b_p	0.32	0.37	0.42
b_1	—	0.35	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.8	—
x	—	—	0.20
y	—	—	0.10
Z_D	—	0.7	—
Z_E	—	0.7	—
L	0.3	0.5	0.7
L_1	—	1.0	—

Figure 2.2 LQFP 32-pin

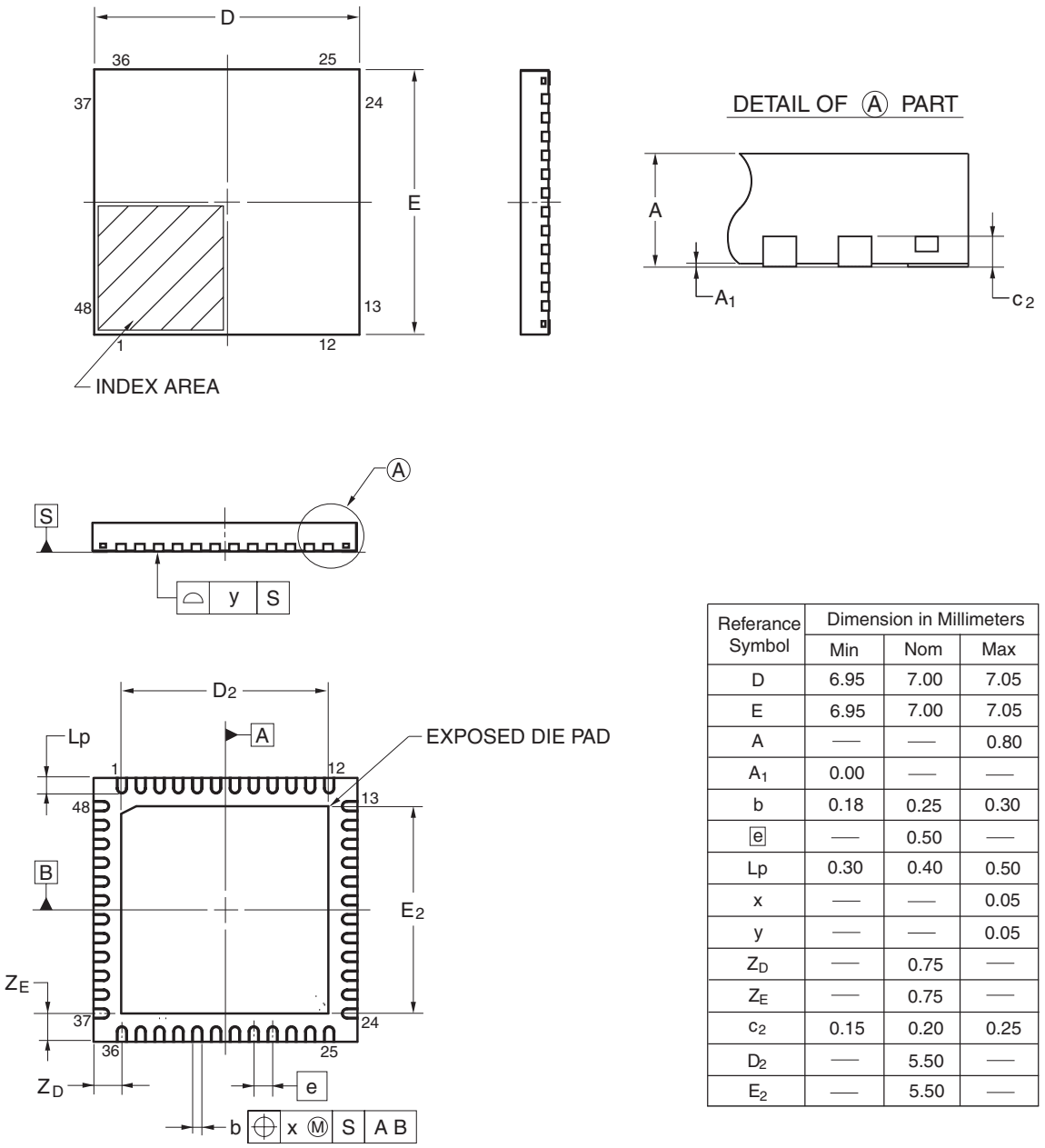
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFBGA36-5x5-0.80	PLBG0036GA-A	36FHE	0.1g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	5.0	—
E	—	5.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.4
A_1	0.3	0.35	0.4
e	—	0.8	—
b	0.4	0.45	0.5
x	—	—	0.08
y	—	—	0.10
Z_D	—	0.5	—
Z_E	—	0.5	—

Figure 2.3 BGA 36-pin

JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-HWQFN48-7x7-0.50	PWQN0048KB-A	48PJN-A P48K8-50-5B4-6	0.13

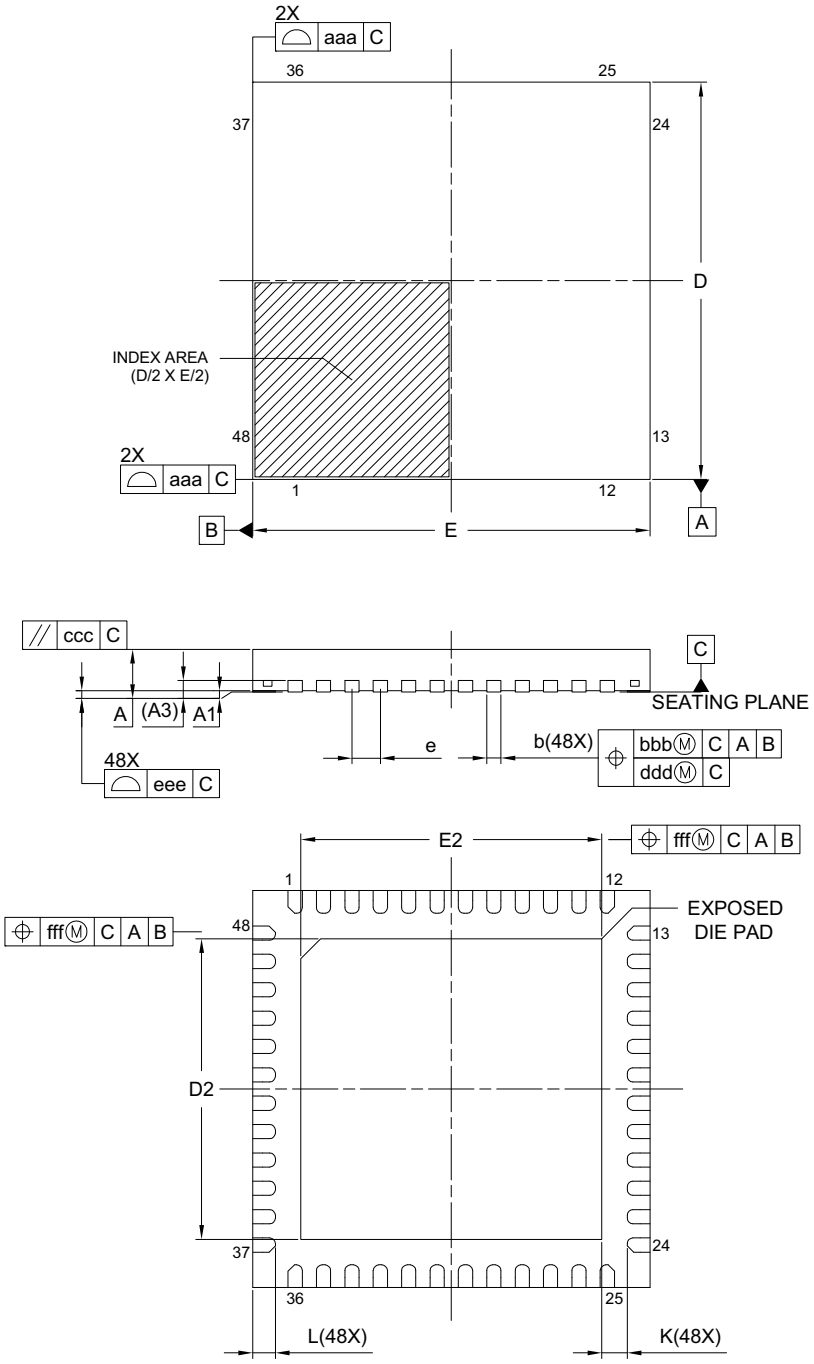


Referance Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	6.95	7.00	7.05
E	6.95	7.00	7.05
A	—	—	0.80
A_1	0.00	—	—
b	0.18	0.25	0.30
e	—	0.50	—
Lp	0.30	0.40	0.50
x	—	—	0.05
y	—	—	0.05
Z_D	—	0.75	—
Z_E	—	0.75	—
c_2	0.15	0.20	0.25
D_2	—	5.50	—
E_2	—	5.50	—

©2013 Renesas Electronics Corporation. All rights reserved.

Figure 2.4 QFN 48-pin (1)

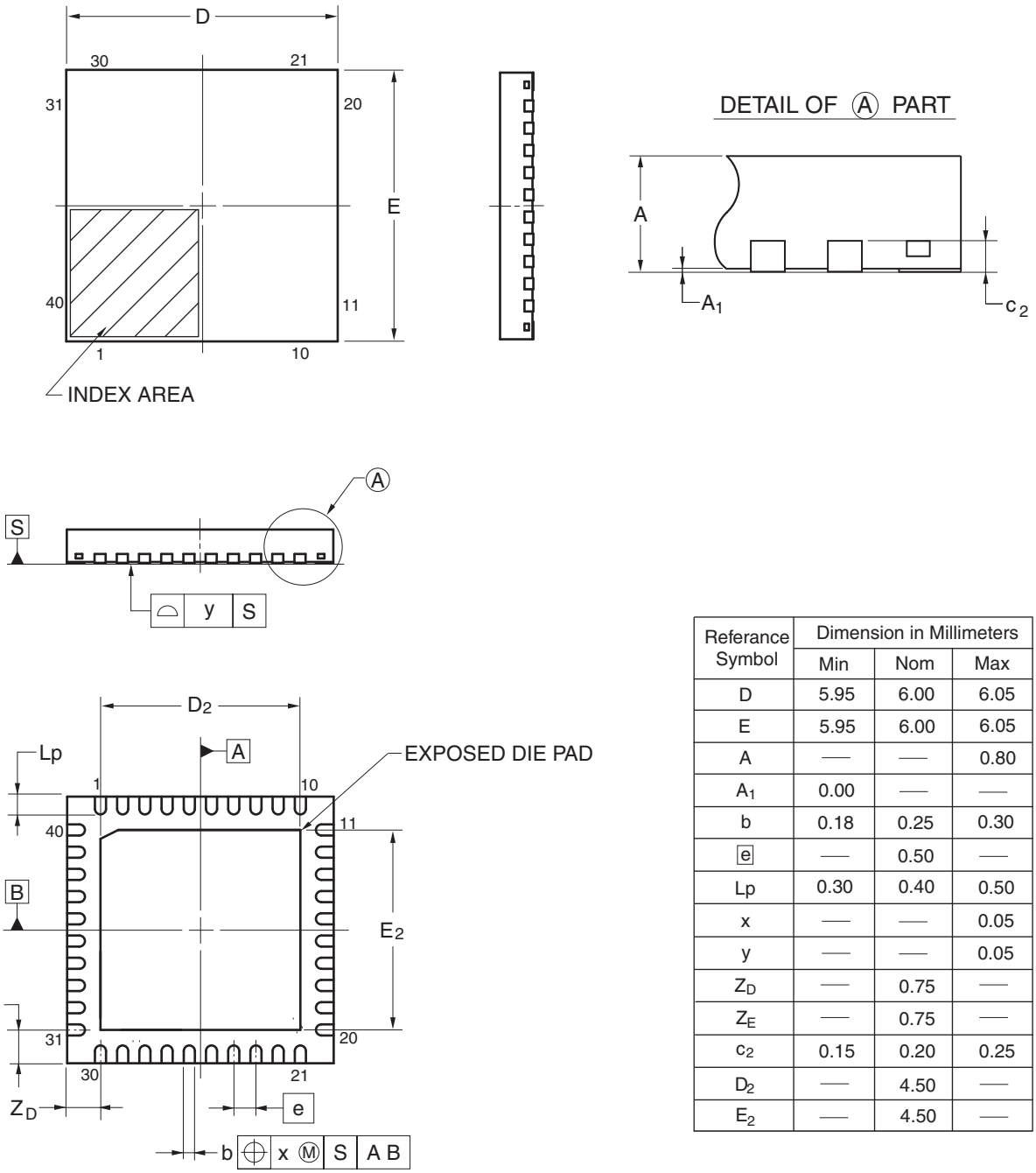
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN048-7x7-0.50	PWQN0048KC-A	0.13 g



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3	0.203 REF.		
b	0.20	0.25	0.30
D	7.00 BSC		
E	7.00 BSC		
e	0.50 BSC		
L	0.30	0.40	0.50
K	0.20	—	—
D_2	5.25	5.30	5.35
E_2	5.25	5.30	5.35
aaa	0.15		
bbb	0.10		
ccc	0.10		
ddd	0.05		
eee	0.08		
fff	0.10		

Figure 2.5 QFN 48-pin (2)

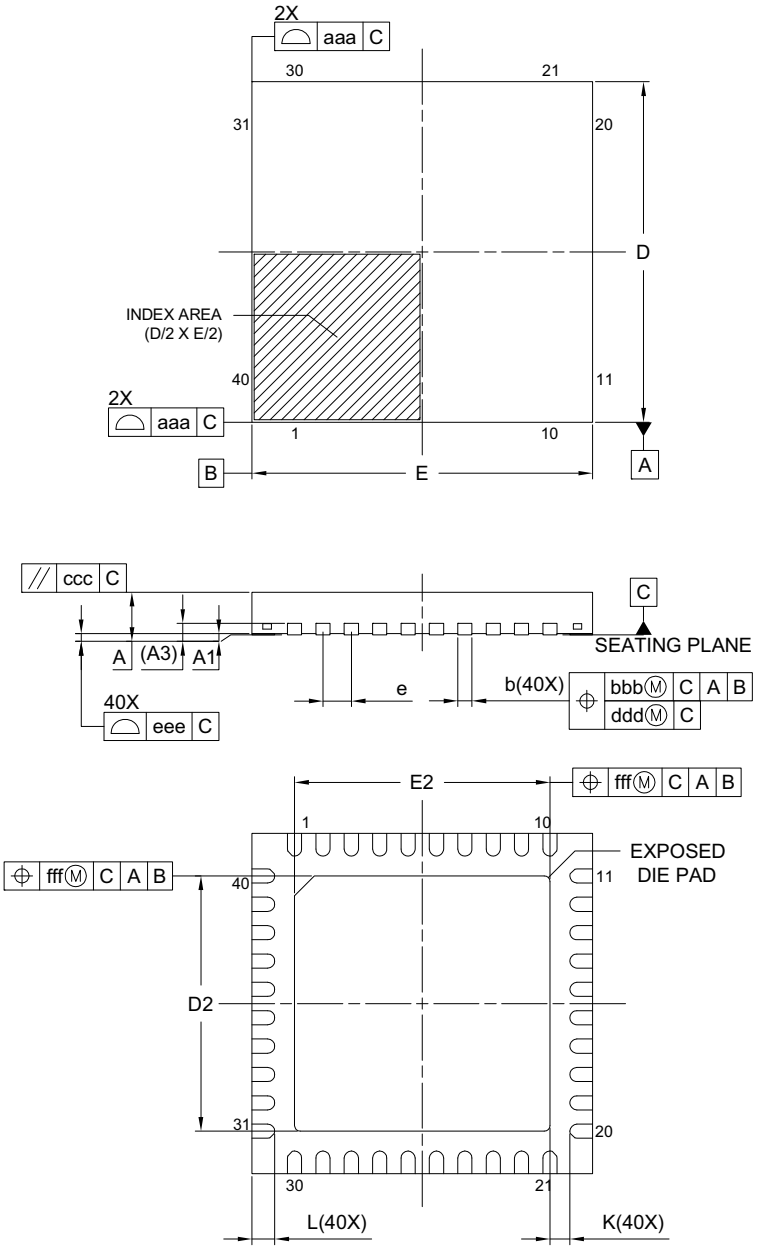
JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-HWQFN40-6x6-0.50	PWQN0040KC-A	P40K8-50-4B4-5	0.09



Referance Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	5.95	6.00	6.05
E	5.95	6.00	6.05
A	—	—	0.80
A_1	0.00	—	—
b	0.18	0.25	0.30
e	—	0.50	—
Lp	0.30	0.40	0.50
x	—	—	0.05
y	—	—	0.05
Z_D	—	0.75	—
Z_E	—	0.75	—
c_2	0.15	0.20	0.25
D_2	—	4.50	—
E_2	—	4.50	—

Figure 2.6 QFN 40-pin (1)

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN040-6x6-0.50	PWQN0040KD-A	0.08



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3	0.203 REF.		
b	0.18	0.25	0.30
D	6.00 BSC		
E	6.00 BSC		
e	0.50 BSC		
L	0.30	0.40	0.50
K	0.20	—	—
D_2	4.45	4.50	4.55
E_2	4.45	4.50	4.55
aaa	0.15		
bbb	0.10		
ccc	0.10		
ddd	0.05		
eee	0.08		
fff	0.10		

Figure 2.7 QFN 40-pin (2)

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
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
ADC160	16-bit A/D Converter	0x4005C000
DAC12	12-bit D/A Converter	0x4005E000
SCI0	Serial Communication Interface 0	0x40070000
SCI1	Serial Communication Interface 1	0x40070020
SCI9	Serial Communication Interface 9	0x40070120
SPI0	Serial Peripheral Interface 0	0x40072000
SPI1	Serial Peripheral Interface 1	0x40072100

Table 3.1 Peripheral base address (2 of 2)

Name	Description	Base address
CRC	CRC Calculator	0x40074000
GPT320	General PWM Timer 0 (32-bit)	0x40078000
GPT161	General PWM Timer 1 (16-bit)	0x40078100
GPT162	General PWM Timer 2 (16-bit)	0x40078200
GPT163	General PWM Timer 3 (16-bit)	0x40078300
GPT164	General PWM Timer 4 (16-bit)	0x40078400
GPT165	General PWM Timer 5 (16-bit)	0x40078500
GPT166	General PWM Timer 6 (16-bit)	0x40078600
GPT_OPS	Output Phase Switching Controller	0x40078FF0
KINT	Key Interrupt Function	0x40080000
CTSU	Capacitive Touch Sensing Unit	0x40081000
AGT0	Low Power Asynchronous General Purpose Timer 0	0x40084000
AGT1	Low Power Asynchronous General Purpose Timer 1	0x40084100
ACMPHS0	High-Speed Analog Comparator 0	0x40085000
ACMPLP	Low-Power Analog Comparator	0x40085E00
OPAMP	Operational Amplifier	0x40086800
USBFS	USB 2.0 Full-Speed Module	0x40090000
SDADC24	24-Bit Sigma-Delta A/D Converter	0x4009C000
DAC8	8-bit D/A Converter	0x4009E000
FLCN	Flash I/O Resister	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 master such as DTC.

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, DTC, ICU, DBG, FCACHE	4000 0000h	4001 CFFFh	2				ICLK	Memory Protection Unit, Memory Mirror Function, SRAM, Buses, Data Transfer Controller, Interrupt Controller, CPU, Flash Memory
SYSTEM	4001 E000h	4001 E3FFh	3				ICLK	Low Power Modes, Resets, Low Voltage Detection, Clock Generation Circuit, Register Write Protection
SYSTEM	4001 E400h	4001 E6FFh	7		5 to 7		PCLKB	Low Power Modes, Resets, Low Voltage Detection
PORTn, PFS, PMISC, ELC, POEG, RTC, WDT, IWDT, CAC, MSTP	4004 0000h	4004 7FFFh	3		2 to 3		PCLKB	I/O Ports, Event Link Controller, Port Output Enable for GPT, Realtime Clock, Watchdog Timer, Independent Watchdog Timer, Clock Frequency Accuracy Measurement Circuit, Module Stop Control
CAN0, IICn, DOC, ADC160,DAC12	4005 0000h	4005 EFFFh	3		2 to 3		PCLKB	Controller Area Network Module, I^{2}C Bus Interface, Data Operation Circuit, 16-Bit A/D Converter
SCIn	4007 0000h	4007 0EFFh	5		2 to 3		PCLKB	Serial Communications Interface
SPIn	4007 2000h	4007 2FFFh	5		2 to 3		PCLKB	Serial Peripheral Interface
CRC	4007 4000h	4007 4FFFh	3		2 to 3		PCLKB	CRC Calculator
GPT320,GPT16n, GPT_OPS	4007 8000h	4007 BFFFh	Refer to Table 3.3				PCLKB	General PWM Timer
KINT, CTSU	4008 0000h	4008 1FFFh	2		1 to 2		PCLKB	Key interrupt Function, Capacitive Touch Sensing Unit
AGTn	4008 4000h	4008 4FFFh	3		2 to 3		PCLKB	Low Power Asynchronous General Purpose Timer
ACMPHSn, ACMPLP, OPAMP	4008 5000h	4008 6FFFh	2		1 to 2		PCLKB	High-Speed Analog Comparator, Low-Power Analog Comparator, Operational Amplifier
USBFS	4009 0000h	4009 03FFh	4		3 to 4		PCLKB	USB 2.0 Full-Speed Module
USBFS	4009 0400h	4009 04FFh	3		2 to 3		PCLKB	USB 2.0 Full-Speed Module
SDADC24, DAC8	4009 C000h	4009 E00Fh	2		1 to 2		PCLKB	24-Bit Sigma-Delta A/D Converter, 8-Bit D/A Converter
FLCN	407E C000h	407E CFFFh	7		7		ICLK	Flash I/O Resister
TSN	407E C000h	407E CFFFh	7		7		ICLK	Temperature Sensor

Note 1. If the number of PCLK cycles is non-integer (for example 1.5), the minimum value is without the decimal point, and the maximum value is rounded up to the decimal point. For example, 1.5 to 2. 5 is 1 to 3.

Note 2. When accessing a 16-bit register (FTDRHL, FRDRHL, FCR, FDR, LSR, and CDR), access is 2 cycles more than the value shown in Table 3.2. When accessing an 8-bit register (FTDRH, FTDRL, FRDRH, and FRDRL), the access cycles are as shown in Table 3.2.

Note 3. When accessing the 32-bit register (SPDR), access is 2 cycles more than the value in Table 3.2. When accessing an 8-bit or 16-bit register (SPDR_HA), the access cycles are as shown in Table 3.2.

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 = PCLKB	5 to 6	3 to 4	PCLKB
ICLK > PCLKD > PCLKB	3 to 4	2 to 3	PCLKB
PCLKD = ICLK = PCLKB	6	4	PCLKB
PCLKD = ICLK > PCLKB	2 to 3	1 to 2	PCLKB
PCLKD > ICLK = PCLKB	4	3	PCLKB
PCLKD > ICLK > PCLKB	2 to 3	1 to 2	PCLKB

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 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
MMPU	-	-	-	MMPUCTLA	Bus Master MPU Control Register	0x000	16	read-write	0x0000	0xFFFF
				MMPUPTA	Group A Protection of Register	0x102	16	read-write	0x0000	0xFFFF
	4	0x010	0-3	MMPUACA%s	Group A Region %s Access Control Register	0x200	16	read-write	0x0000	0xFFFF
	4	0x010	0-3	MMPUSA%s	Group A Region %s Start Address Register	0x204	32	read-write	0x0000 0000	0x00000 003
	4	0x010	0-3	MMPUEA%s	Group A Region %s End Address Register	0x208	32	read-write	0x0000 0003	0x00000 003
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 SRAM	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

Table 3.4 Register description (2 of 37)

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 Monitor Start Address Register	0x08	32	read-write	0x00000000	0x00000003
				MSPMPUEA	Main Stack Pointer 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 Monitor Start Address Register	0x18	32	read-write	0x00000000	0x00000003
				PSPMPUEA	Process Stack Pointer 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 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
BUS	-	-	-	BUSMCNTSYS	Master Bus Control Register SYS	0x1008	16	read-write	0x0000	0xFFFF
				BUSMCNTDMA	Master Bus Control Register DMA	0x100C	16	read-write	0x0000	0xFFFF
				BUSSCNTFLI	Slave Bus Control Register FLI	0x1100	16	read-write	0x0000	0xFFFF
				BUSSCNTRAM0	Slave Bus Control Register RAM0	0x110C	16	read-write	0x0000	0xFFFF
	2	0x4	P0B, P2B	BUSSCNT%s	Slave Bus Control Register %s	0x1114	16	read-write	0x0000	0xFFFF
	-	-	-	BUSSCNTP4B	Slave Bus Control Register P4B	0x1120	16	read-write	0x0000	0xFFFF
	-	-	-	BUSSCNTP6B	Slave Bus Control Register P6B	0x1128	16	read-write	0x0000	0xFFFF
	-	-	-	BUSSCNTFBU	Slave Bus Control Register FBU	0x1130	16	read-write	0x0000	0xFFFF
	2	0x10	3,4	BUS%sERRADD	Bus Error Address Register %s	0x1820	32	read-only	0x0000 0000	0x00000 000
	2	0x10	3,4	BUS%sERRSTAT	Bus Error Status Register %s	0x1824	8	read-only	0x00	0xFE
DTC	-	-	-	DTCCR	DTC Control Register	0x00	8	read-write	0x08	0xFF
				DTCVBR	DTC Vector Base Register	0x04	32	read-write	0x0000 0000	0xFFFF FFFF
				DTCST	DTC Module Start Register	0x0C	8	read-write	0x00	0xFF
				DTCSTS	DTC Status Register	0x0E	16	read-only	0x0000	0xFFFF
ICU	8	0x1	0-7	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	0x0000 0000	0xFFFF FFFF
				SELSR0	SYS Event Link Setting Register	0x200	16	read-write	0x0000	0xFFFF
	32	0x4	0-31	IELSR%s	ICU Event Link Setting Register %s	0x300	32	read-write	0x0000 0000	0xFFFF FFFF
DBG	-	-	-	DBGSTR	Debug Status Register	0x00	32	read-only	0x0000 0000	0xFFFF FFFF
				DBGSTOPCR	Debug Stop Control Register	0x10	32	read-write	0x0000 0003	0xFFFF FFFF

Table 3.4 Register description (4 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
FCACHE	-	-	-	FCACHEE	Flash Cache Enable Register	0x100	16	read-write	0x0000	0xFFFF
				FCACHEIV	Flash Cache Invalidate Register	0x104	16	read-write	0x0000	0xFFFF
SYSTEM	-	-	-	SBYCR	Standby Control Register	0x00C	16	read-write	0x0000	0xFFFF
	-	-	-	MSTPCRA	Module Stop Control Register A	0x01C	32	read-write	0xFFBF FFFF	0xFFFF FFFF
				SCKDIVCR	System Clock Division Control Register	0x020	32	read-write	0x4400 0404	0xFFFF FFFF
				SCKSCR	System Clock Source Control Register	0x026	8	read-write	0x01	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
				OSTDCR	Oscillation Stop Detection Control Register	0x040	8	read-write	0x00	0xFF
				OSTDSR	Oscillation Stop Detection Status Register	0x041	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	0x0000 0000	0xFFFF FFFF
				FLSTOP	Flash Operation Control Register	0x09E	8	read-write	0x00	0xFF
				OPCCR	Operating Power Control Register	0x0A0	8	read-write	0x02	0xFF

Table 3.4 Register description (5 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SYSTEM	-	-	-	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
				SDADCCKCR	24-bit Sigma-Delta A/D Converter Clock Control Register	0x0D1	8	read-write	0x00	0xFF
				LVD1CR1	Voltage Monitor 1 Circuit Control Register 1	0x0E0	8	read-write	0x01	0xFF
				LVD1SR	Voltage Monitor 1 Circuit Status Register	0x0E1	8	read-write	0x02	0xFF
				LVD2CR1	Voltage Monitor 2 Circuit Control Register 1	0x0E2	8	read-write	0x01	0xFF
				LVD2SR	Voltage Monitor 2 Circuit Status Register	0x0E3	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

Table 3.4 Register description (6 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SYSTEM	-	-	-	LVD1CR0	Voltage Monitor 1 Circuit Control Register 0	0x41A	8	read-write	0x80	0xF7
				LVD2CR0	Voltage Monitor 2 Circuit Control Register 0	0x41B	8	read-write	0x80	0xF7
				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
PORT0,3-5,9	-	-	-	PCNTR1	Port Control Register 1	0x00	32	read-write	0x0000 0000	0xFFFF FFFF
				PODR	Output Data Register	0x00	16	read-write	0x0000	0xFFFF
				PDR	Direction Register	0x02	16	read-write	0x0000	0xFFFF
				PCNTR2	Port Control Register 2	0x04	32	read-only	0x0000 0000	0xFFFF0 000
				PIDR	Input Data Register	0x06	16	read-only	0x0000	0x0000
				PCNTR3	Port Control Register 3	0x08	32	write-only	0x0000 0000	0xFFFF FFFF
				PORR	Output Reset Register	0x08	16	write-only	0x0000	0xFFFF
				POSR	Output Set Register	0x0A	16	write-only	0x0000	0xFFFF

Table 3.4 Register description (7 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
PORT1,2	-	-	-	PCNTR1	Port Control Register 1	0x00	32	read-write	0x0000 0000	0xFFFF FFFF
				PODR	Output Data Register	0x00	16	read-write	0x0000	0xFFFF
				PDR	Direction Register	0x02	16	read-write	0x0000	0xFFFF
				PCNTR2	Port Control Register 2	0x04	32	read-only	0x0000 0000	0xFFFF0 000
				EIDR	Event Input Data Register	0x04	16	read-only	0x0000	0xFFFF
				PIDR	Input Data Register	0x06	16	read-only	0x0000	0x0000
				PCNTR3	Port Control Register 3	0x08	32	write-only	0x0000 0000	0xFFFF FFFF
				PORR	Output Reset Register	0x08	16	write-only	0x0000	0xFFFF
				POSR	Output Set Register	0x0A	16	write-only	0x0000	0xFFFF
				PCNTR4	Port Control Register 4	0x0C	32	read-write	0x0000 0000	0xFFFF FFFF
				EORR	Event Output Reset Register	0x0C	16	read-write	0x0000	0xFFFF
				EOSR	Event Output Set Register	0x0E	16	read-write	0x0000	0xFFFF

Table 3.4 Register description (8 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
PFS	-	-	-	P000PFS	P000 Pin Function Control Register	0x000	32	read-write	0x0000 0000	0xFFFF FFFF
				P000PFS_HA	P000 Pin Function Control Register	0x002	16	read-write	0x0000	0xFFFF
				P000PFS_BY	P000 Pin Function Control Register	0x003	8	read-write	0x00	0xFF
	3	0x4	1-3	P00%sPFS	P00%s Pin Function Control Register	0x004	32	read-write	0x0000 0000	0xFFFF FFFF
	3	0x4	1-3	P00%sPFS_HA	P00%s Pin Function Control Register	0x006	16	read-write	0x0000	0xFFFF
	3	0x4	1-3	P00%sPFS_BY	P00%s Pin Function Control Register	0x007	8	read-write	0x00	0xFF
	4	0x4	12-15	P0%sPFS	P0%s Pin Function Control Register	0x030	32	read-write	0x0000 0000	0xFFFF FFFF
	4	0x4	12-15	P0%sPFS_HA	P0%s Pin Function Control Register	0x032	16	read-write	0x0000	0xFFFF
	4	0x4	12-15	P0%sPFS_BY	P0%s Pin Function Control Register	0x033	8	read-write	0x00	0xFF
	-	-	-	P100PFS	P100 Pin Function Control Register	0x040	32	read-write	0x0000 0000	0xFFFF FFFF
	-	-	-	P100PFS_HA	P100 Pin Function Control Register	0x042	16	read-write	0x0000	0xFFFF
	-	-	-	P100PFS_BY	P100 Pin Function Control Register	0x043	8	read-write	0x00	0xFF
	7	0x4	1-7	P10%sPFS	P10%s Pin Function Control Register	0x044	32	read-write	0x0000 0000	0xFFFF FFFF
	7	0x4	1-7	P10%sPFS_HA	P10%s Pin Function Control Register	0x046	16	read-write	0x0000	0xFFFF
	7	0x4	1-7	P10%sPFS_BY	P10%s Pin Function Control Register	0x047	8	read-write	0x00	0xFF

Table 3.4 Register description (9 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
PFS	-	-	-	P108PFS	P108 Pin Function Control Register	0x060	32	read-write	0x0001 0010	0xFFFF FFFF
				P108PFS_HA	P108 Pin Function Control Register	0x062	16	read-write	0x0010	0xFFFF
				P108PFS_BY	P108 Pin Function Control Register	0x063	8	read-write	0x10	0xFF
				P109PFS	P109 Pin Function Control Register	0x064	32	read-write	0x0000 0000	0xFFFF FFFF
				P109PFS_HA	P109 Pin Function Control Register	0x066	16	read-write	0x0000	0xFFFF
				P109PFS_BY	P109 Pin Function Control Register	0x067	8	read-write	0x00	0xFF
	3	0x4	10-12	P1%sPFS	P1%s Pin Function Control Register	0x068	32	read-write	0x0000 0000	0xFFFF FFFF
	3	0x4	10-12	P1%sPFS_HA	P1%s Pin Function Control Register	0x06A	16	read-write	0x0000	0xFFFF
	3	0x4	10-12	P1%sPFS_BY	P1%s Pin Function Control Register	0x06B	8	read-write	0x00	0xFF
	-	-	-	P200PFS	P200 Pin Function Control Register	0x080	32	read-write	0x0000 0000	0xFFFF FFFF
				P200PFS_HA	P200 Pin Function Control Register	0x082	16	read-write	0x0000	0xFFFF
				P200PFS_BY	P200 Pin Function Control Register	0x083	8	read-write	0x00	0xFF
				P201PFS	P201 Pin Function Control Register	0x084	32	read-write	0x0000 0010	0xFFFF FFFF
				P201PFS_HA	P201 Pin Function Control Register	0x086	16	read-write	0x0010	0xFFFF
	-	-	-	P201PFS_BY	P201 Pin Function Control Register	0x087	8	read-write	0x10	0xFF
	3	0x4	4-6	P20%sPFS	P20%s Pin Function Control Register	0x090	32	read-write	0x0000 0000	0xFFFF FFFF
	3	0x4	4-6	P20%sPFS_HA	P20%s Pin Function Control Register	0x092	16	read-write	0x0000	0xFFFF
	3	0x4	4-6	P20%sPFS_BY	P20%s Pin Function Control Register	0x093	8	read-write	0x00	0xFF
	4	0x4	12-15	P2%sPFS	P2%s Pin Function Control Register	0x0B0	32	read-write	0x0000 0000	0xFFFF FFFF
	4	0x4	12-15	P2%sPFS_HA	P2%s Pin Function Control Register	0x0B2	16	read-write	0x0000	0xFFFF
	4	0x4	12-15	P2%sPFS_BY	P2%s Pin Function Control Register	0x0B3	8	read-write	0x00	0xFF

Table 3.4 Register description (10 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
PFS	-	-	-	P300PFS	P300 Pin Function Control Register	0x0C0	32	read-write	0x00010010	0xFFFFFFFF
				P300PFS_HA	P300 Pin Function Control Register	0x0C2	16	read-write	0x0010	0xFFFF
				P300PFS_BY	P300 Pin Function Control Register	0x0C3	8	read-write	0x10	0xFF
	4	0x4	1-4	P30%sPFS	P30%s Pin Function Control Register	0x0C4	32	read-write	00000000	0xFFFFFFFF
	4	0x4	1-4	P30%sPFS_HA	P30%s Pin Function Control Register	0x0C6	16	read-write	0x0000	0xFFFF
	4	0x4	1-4	P30%sPFS_BY	P30%s Pin Function Control Register	0x0C7	8	read-write	0x00	0xFF
	4	0x4	0-3	P40%sPFS	P40%s Pin Function Control Register	0x100	32	read-write	00000000	0xFFFFFFFF
	4	0x4	0-3	P40%sPFS_HA	P40%s Pin Function Control Register	0x102	16	read-write	0x0000	0xFFFF
	4	0x4	0-3	P40%sPFS_BY	P40%s Pin Function Control Register	0x103	8	read-write	0x00	0xFF
	-	-	-	P407PFS	P407 Pin Function Control Register	0x11C	32	read-write	00000000	0xFFFFFFFF
				P407PFS_HA	P407 Pin Function Control Register	0x11E	16	read-write	0x0000	0xFFFF
				P407PFS_BY	P407 Pin Function Control Register	0x11F	8	read-write	0x00	0xFF
	2	0x4	8,9	P40%sPFS	P40%s Pin Function Control Register	0x120	32	read-write	00000000	0xFFFFFFFF
	2	0x4	8,9	P40%sPFS_HA	P40%s Pin Function Control Register	0x122	16	read-write	0x0000	0xFFFF
	2	0x4	8,9	P40%sPFS_BY	P40%s Pin Function Control Register	0x123	8	read-write	0x00	0xFF
	2	0x4	10,11	P4%sPFS	P4%s Pin Function Control Register	0x128	32	read-write	00000000	0xFFFFFFFF
	2	0x4	10,11	P4%sPFS_HA	P4%s Pin Function Control Register	0x12A	16	read-write	0x0000	0xFFFF
	2	0x4	10,11	P4%sPFS_BY	P4%s Pin Function Control Register	0x12B	8	read-write	0x00	0xFF

Table 3.4 Register description (11 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
PFS	3	0x4	0-2	P50%sPFS	P50%s Pin Function Control Register	0x140	32	read-write	0x0000 0000	0xFFFF FFFF
	3	0x4	0-2	P50%sPFS_HA	P50%s Pin Function Control Register	0x142	16	read-write	0x0000	0xFFFF
	3	0x4	0-2	P50%sPFS_BY	P50%s Pin Function Control Register	0x143	8	read-write	0x00	0xFF
	-	-	-	P914PFS	P914 Pin Function Control Register	0x278	32	read-write	0x0001 0000	0xFFFF FFFF
				P914PFS_HA	P914 Pin Function Control Register	0x27A	16	read-write	0x0000	0xFFFF
				P914PFS_BY	P914 Pin Function Control Register	0x27B	8	read-write	0x00	0xFF
				P915PFS	P915 Pin Function Control Register	0x27C	32	read-write	0x0001 0000	0xFFFF FFFF
				P915PFS_HA	P915 Pin Function Control Register	0x27E	16	read-write	0x0000	0xFFFF
				P915PFS_BY	P915 Pin Function Control Register	0x27F	8	read-write	0x00	0xFF
PMISC	-	-	-	PWPR	Write-Protect Register	0x03	8	read-write	0x80	0xFF
ELC	-	-	-	ELCR	Event Link Controller Register	0x00	8	read-write	0x00	0xFF
	2	0x2	0,1	ELSEGR%s	Event Link Software Event Generation Register %s	0x02	8	read-write	0x80	0xFF
	4	0x4	0-3	ELSR%s	Event Link Setting Register %s	0x10	16	read-write	0x0000	0xFFFF
	2	0x4	8,9	ELSR%s	Event Link Setting Register %s	0x30	16	read-write	0x0000	0xFFFF
	-	-	-	ELSR12	Event Link Setting Register 12	0x40	16	read-write	0x0000	0xFFFF
	2	0x4	14,15	ELSR%s	Event Link Setting Register %s	0x48	16	read-write	0x0000	0xFFFF
	3	0x4	18-20	ELSR%s	Event Link Setting Register %s	0x58	16	read-write	0x0000	0xFFFF
	-	-	-	ELSR22	Event Link Setting Register 22	0x68	16	read-write	0x0000	0xFFFF
POEG	2	0x100	A,B	POEGG%s	POEG Group %s Setting Register	0x00	32	read-write	0x0000 0000	0xFFFF FFFF

Table 3.4 Register description (12 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
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
				RDAYAR	Date Alarm Register	0x18	8	read-write	0x00	0x00
				BCNT0AER	Binary Counter 0 Alarm Enable Register	0x18	8	read-write	0x00	0x00

Table 3.4 Register description (13 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
RTC	-	-	-	RMONAR	Month Alarm Register	0x1A	8	read-write	0x00	0x00
				BCNT1AER	Binary Counter 1 Alarm Enable Register	0x1A	8	read-write	0x00	0x00
				RYRAR	Year Alarm Register	0x1C	16	read-write	0x0000	0xFF00
				BCNT2AER	Binary Counter 2 Alarm Enable Register	0x1C	16	read-write	0x0000	0xFF00
				RYRAREN	Year Alarm Enable Register	0x1E	8	read-write	0x00	0x00
				BCNT3AER	Binary Counter 3 Alarm Enable Register	0x1E	8	read-write	0x00	0x00
				RCR1	RTC Control Register 1	0x22	8	read-write	0x00	0x0A
				RCR2	RTC Control Register 2	0x24	8	read-write	0x00	0x0E
				RCR4	RTC Control Register 4	0x28	8	read-write	0x00	0xFE
				RFRH	Frequency Register H	0x2A	16	read-write	0x0000	0xFFFE
				RFRL	Frequency Register L	0x2C	16	read-write	0x0000	0x0000
				RADJ	Time Error Adjustment Register	0x2E	8	read-write	0x00	0x00
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
IWDT	-	-	-	IWDTRR	IWDT Refresh Register	0x00	8	read-write	0xFF	0xFF
				IWDTSR	IWDT Status Register	0x04	16	read-write	0x0000	0xFFFF

Table 3.4 Register description (14 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
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	0xFFFF FFFF	0xFFFF FFFF
				MSTPCRC	Module Stop Control Register C	0x04	32	read-write	0xFFFF FFFF	0xFFFF FFFF
				MSTPCRD	Module Stop Control Register D	0x08	32	read-write	0xFFFF FFFF	0xFFFF FFFF

Table 3.4 Register description (15 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
CAN0	32	0x10	0-31	MB%s_ID	Mailbox Register	0x200	32	read-write	0x0000 0000	0x00000 000
	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
	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	0x0000 0000	0x00000 000
	2	0x4	0,1	FIDCR%s	FIFO Received ID Compare Registers	0x420	32	read-write	0x0000 0000	0x00000 000
	-	-	-	MKIVLR	Mask Invalid Register	0x428	32	read-write	0x0000 0000	0x00000 000
				MIER	Mailbox Interrupt Enable Register (normal mailbox mode)	0x42C	32	read-write	0x0000 0000	0x00000 000
				MIER_FIFO	Mailbox Interrupt Enable Register (FIFO mailbox mode)	0x42C	32	read-write	0x0000 0000	0x00000 000
	32	0x1	0-31	MCTL_TX[%s]	Message Control Register (transmit mode when the TRMREQ bit is 1 and the RECREQ bit is 0)	0x820	8	read-write	0x00	0xFF

Table 3.4 Register description (16 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
CAN0	32	0x1	0-31	MCTL_RX[%s]	Message Control Register (receive mode when the TRMREQ bit is 0 and the RECREQ bit is 1)	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	0x0000 0000	0xFFFF FFFF
				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

Table 3.4 Register description (17 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
IIC0	-	-	-	ICCR1	I2C Bus Control Register 1	0x00	8	read-write	0x1F	0xFF
				ICCR2	I2C Bus Control Register 2	0x01	8	read-write	0x00	0xFF
				ICMR1	I2C Bus Mode Register 1	0x02	8	read-write	0x08	0xFF
				ICMR2	I2C Bus Mode Register 2	0x03	8	read-write	0x06	0xFF
				ICMR3	I2C Bus Mode Register 3	0x04	8	read-write	0x00	0xFF
				ICFER	I2C Bus Function Enable Register	0x05	8	read-write	0x72	0xFF
				ICSER	I2C Bus Status Enable Register	0x06	8	read-write	0x09	0xFF
				ICIER	I2C Bus Interrupt Enable Register	0x07	8	read-write	0x00	0xFF
				ICSR1	I2C Bus Status Register 1	0x08	8	read-write	0x00	0xFF
				ICSR2	I2C Bus Status Register 2	0x09	8	read-write	0x00	0xFF
	3	0x2	0-2	SARL%s	Slave Address Register L%s	0x0A	8	read-write	0x00	0xFF
	3	0x2	0-2	SARU%s	Slave Address Register U%s	0x0B	8	read-write	0x00	0xFF
	-	-	-	ICBRL	I2C Bus Bit Rate Low-Level Register	0x10	8	read-write	0xFF	0xFF
				ICBRH	I2C Bus Bit Rate High-Level Register	0x11	8	read-write	0xFF	0xFF
				ICDRT	I2C Bus Transmit Data Register	0x12	8	read-write	0xFF	0xFF
				ICDRR	I2C Bus Receive Data Register	0x13	8	read-only	0x00	0xFF
				ICWUR	I2C Bus Wake Up Unit Register	0x16	8	read-write	0x10	0xFF
				ICWUR2	Reserved	0x17	8	read-write	0xFD	0xFF

Table 3.4 Register description (18 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
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
				ICDRR	I2C Bus Receive Data Register	0x13	8	read-only	0x00	0xFF
DOC	-	-	-	DOCR	DOC Control Register	0x00	8	read-write	0x00	0xFF
				DODIR	DOC Data Input Register	0x02	16	read-write	0x0000	0xFFFF
				DODSR	DOC Data Setting Register	0x04	16	read-write	0x0000	0xFFFF
ADC160	-	-	-	ADCSR	A/D Control Register	0x000	16	read-write	0x0000	0xFFFF
				ADANSA0	A/D Channel Select Register A0	0x004	16	read-write	0x0000	0xFFFF

Table 3.4 Register description (19 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
ADC160	-	-	-	ADANSA1	A/D Channel Select Register A1	0x006	16	read-write	0x0000	0xFFFF
				ADADS0	A/D-Converted Value Average Channel Select Register 0	0x008	16	read-write	0x0000	0xFFFF
				ADADS1	A/D-Converted Value Average Channel Select Register 1	0x00A	16	read-write	0x0000	0xFFFF
				ADADC	A/D-Converted Value 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 Duplexing 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
	9	0x2	0-8	ADDR%s	A/D Data Register %s	0x020	16	read-only	0x0000	0xFFFF
	9	0x2	16-24	ADDR%s	A/D Data Register %s	0x040	16	read-only	0x0000	0xFFFF
	-	-	-	ADDISCR	A/D Disconnection Detection Control Register	0x07A	8	read-write	0x00	0xFF

Table 3.4 Register description (20 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
ADC160	-	-	-	ADICR	A/D Interrupt Control Register	0x07D	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
				ADWINMON	A/D Compare Function Window A/B Status Monitor Register	0x08C	8	read-only	0x00	0xFF
				ADCMPCR	A/D Compare Function Control Register	0x090	16	read-write	0x0000	0xFFFF
				ADCMPANSER	A/D Compare Function Window A Extended Input Select Register	0x092	8	read-write	0x00	0xFF
				ADCMPLER	A/D Compare Function Window A Extended Input Comparison Condition Setting Register	0x093	8	read-write	0x00	0xFF
				ADCMPANSR0	A/D Compare Function Window A Channel Select Register 0	0x094	16	read-write	0x0000	0xFFFF
				ADCMPANSR1	A/D Compare Function Window A Channel Select Register 1	0x096	16	read-write	0x0000	0xFFFF
				ADCMPLR0	A/D Compare Function Window A Comparison Condition Setting Register 0	0x098	16	read-write	0x0000	0xFFFF

Table 3.4 Register description (21 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
ADC160	-	-	-	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
	9	0x1	0-8	ADSSTR0%s	A/D Sampling State Register %s	0x0E0	8	read-write	0x0D	0xFF

Table 3.4 Register description (22 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
ADC160	-	-	-	ADANIM	A/D Channel Input Mode Select Register	0x0F0	16	read-write	0x0000	0xFFFF
				ADCALEXE	A/D Calibration Execution Register	0x0F2	8	read-write	0x00	0xFF
				VREFAMPCNT	A/D Dedicated Reference Voltage Circuit Control Register	0x0F4	8	read-write	0x00	0xFF
				ADRD	A/D Self-Diagnosis Data Register	0x0F8	16	read-only	0x0000	0xFFFF
				ADRST	A/D Self-Diagnostic Status Register	0x0FA	8	read-write	0x00	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
				DAVREFCR	D/A VREF Control Register	0x07	8	read-write	0x00	0xFF
				DAPC	D/A Switch Charge Pump Control Register	0x09	8	read-write	0x00	0xFF
SCI0	-	-	-	SMR	Serial Mode Register (SCMR.SMIF = 0)	0x00	8	read-write	0x00	0xFF

Table 3.4 Register description (23 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SCI0	-	-	-	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^2C Mode Register 1	0x09	8	read-write	0x00	0xFF
				SIMR2	I^2C Mode Register 2	0x0A	8	read-write	0x00	0xFF
				SIMR3	I^2C Mode Register 3	0x0B	8	read-write	0x00	0xFF
				SISR	I^2C Status Register	0x0C	8	read-only	0x00	0xCB
				SPMR	SPI Mode Register	0x0D	8	read-write	0x00	0xFF
				TDRHL	Transmit 9-bit Data Register	0x0E	16	read-write	0xFFFF	0xFFFF
				FTDRHL	Transmit FIFO Data Register HL	0x0E	16	write-only	0xFFFF	0xFFFF

Table 3.4 Register description (24 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SCI0	-	-	-	FTDRH	Transmit FIFO Data Register H	0x0E	8	write-only	0xFF	0xFF
				FTDRL	Transmit FIFO Data Register L	0x0F	8	write-only	0xFF	0xFF
				RDRHL	Receive 9-bit Data Register	0x10	16	read-only	0x0000	0xFFFF
				FRDRHL	Receive FIFO Data Register HL	0x10	16	read-only	0x0000	0xFFFF
				FRDRH	Receive FIFO Data Register H	0x10	8	read-only	0x00	0xFF
				FRDRL	Receive FIFO Data Register L	0x11	8	read-only	0x00	0xFF
				MDDR	Modulation Duty Register	0x12	8	read-write	0xFF	0xFF
				DCCR	Data Compare Match Control Register	0x13	8	read-write	0x40	0xFF
				FCR	FIFO Control Register	0x14	16	read-write	0xF800	0xFFFF
				FDR	FIFO Data Count Register	0x16	16	read-only	0x0000	0xFFFF
				LSR	Line Status Register	0x18	16	read-only	0x0000	0xFFFF
				CDR	Compare Match Data Register	0x1A	16	read-write	0x0000	0xFFFF
				SPTR	Serial Port Register	0x1C	8	read-write	0x03	0xFF
SCI1,9	-	-	-	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

Table 3.4 Register description (25 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SCI1,9	-	-	-	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	I2C Mode Register 1	0x09	8	read-write	0x00	0xFF
				SIMR2	I2C Mode Register 2	0x0A	8	read-write	0x00	0xFF
				SIMR3	I^2C Mode Register 3	0x0B	8	read-write	0x00	0xFF
				SISR	I^2C Status Register	0x0C	8	read-only	0x00	0xCB
				SPMR	SPI Mode Register	0x0D	8	read-write	0x00	0xFF
				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

Table 3.4 Register description (26 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SPI0,1	-	-	-	SPCR	SPI Control Register	0x00	8	read-write	0x00	0xFF
				SSLP	SPI Slave Select Polarity Register	0x01	8	read-write	0x00	0xFF
				SPPCR	SPI Pin Control Register	0x02	8	read-write	0x00	0xFF
				SPSR	SPI Status Register	0x03	8	read-write	0x20	0xFF
				SPDR	SPI Data Register	0x04	32	read-write	0x0000 0000	0xFFFF FFFF
				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 Register 0	0x00	8	read-write	0x00	0xFF
				CRCCR1	CRC Control Register 1	0x01	8	read-write	0x00	0xFF
				CRCDIR	CRC Data Input Register	0x04	32	read-write	0x0000 0000	0xFFFF FFFF
				CRCDIR_BY	CRC Data Input Register (byte access)	0x04	8	read-write	0x00	0xFF
				CRCDOR	CRC Data Output Register	0x08	32	read-write	0x0000 0000	0xFFFF FFFF
				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

Table 3.4 Register description (27 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
GPT320	-	-	-	GTWP	General PWM Timer Write-Protection Register	0x00	32	read-write	0x0000 0000	0xFFFF FFFF
				GTSTR	General PWM Timer Software Start Register	0x04	32	read-write	0x0000 0000	0xFFFF FFFF
				GTSTP	General PWM Timer Software Stop Register	0x08	32	read-write	0xFFFF FFFF	0xFFFF FFFF
				GTCLR	General PWM Timer Software Clear Register	0x0C	32	write-only	0x0000 0000	0xFFFF FFFF
				GTSSR	General PWM Timer Start Source Select Register	0x10	32	read-write	0x0000 0000	0xFFFF FFFF
				GTPSR	General PWM Timer Stop Source Select Register	0x14	32	read-write	0x0000 0000	0xFFFF FFFF
				GTCSR	General PWM Timer Clear Source Select Register	0x18	32	read-write	0x0000 0000	0xFFFF FFFF
				GTUPSR	General PWM Timer Up Count Source Select Register	0x1C	32	read-write	0x0000 0000	0xFFFF FFFF
				GTDNSR	General PWM Timer Down Count Source Select Register	0x20	32	read-write	0x0000 0000	0xFFFF FFFF
				GTICASR	General PWM Timer Input Capture Source Select Register A	0x24	32	read-write	0x0000 0000	0xFFFF FFFF
				GTICBSR	General PWM Timer Input Capture Source Select Register B	0x28	32	read-write	0x0000 0000	0xFFFF FFFF
				GTCR	General PWM Timer Control Register	0x2C	32	read-write	0x0000 0000	0xFFFF FFFF
				GTUDDTYC	General PWM Timer Count Direction and Duty Setting Register	0x30	32	read-write	0x0000 0001	0xFFFF FFFF
				GTIOR	General PWM Timer I/O Control Register	0x34	32	read-write	0x0000 0000	0xFFFF FFFF

Table 3.4 Register description (28 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
GPT320	-	-	-	GTINTAD	General PWM Timer Interrupt Output Setting Register	0x38	32	read-write	0x0000 0000	0xFFFF FFFF
				GTST	General PWM Timer Status Register	0x3C	32	read-write	0x0000 8000	0xFFFF FFFF
				GTBER	General PWM Timer Buffer Enable Register	0x40	32	read-write	0x0000 0000	0xFFFF FFFF
				GTCNT	General PWM Timer Counter	0x48	32	read-write	0x0000 0000	0xFFFF FFFF
				GTCCRA	General PWM Timer Compare Capture Register A	0x4C	32	read-write	0xFFFF FFFF	0xFFFF FFFF
				GTCCRB	General PWM Timer Compare Capture Register B	0x50	32	read-write	0xFFFF FFFF	0xFFFF FFFF
				GTCCRC	General PWM Timer Compare Capture Register C	0x54	32	read-write	0xFFFF FFFF	0xFFFF FFFF
				GTCCRE	General PWM Timer Compare Capture Register E	0x58	32	read-write	0xFFFF FFFF	0xFFFF FFFF
				GTCCRD	General PWM Timer Compare Capture Register D	0x5C	32	read-write	0xFFFF FFFF	0xFFFF FFFF
				GTCCRF	General PWM Timer Compare Capture Register F	0x60	32	read-write	0xFFFF FFFF	0xFFFF FFFF
				GTPR	General PWM Timer Cycle Setting Register	0x64	32	read-write	0xFFFF FFFF	0xFFFF FFFF
				GTPBR	General PWM Timer Cycle Setting Buffer Register	0x68	32	read-write	0xFFFF FFFF	0xFFFF FFFF
				GTDTCR	General PWM Timer Dead Time Control Register	0x88	32	read-write	0x0000 0000	0xFFFF FFFF
				GTDVU	General PWM Timer Dead Time Value Register U	0x8C	32	read-write	0xFFFF FFFF	0xFFFF FFFF
GPT161-166	-	-	-	GTWP	General PWM Timer Write-Protection Register	0x00	32	read-write	0x0000 0000	0xFFFF FFFF

Table 3.4 Register description (29 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
GPT161-166	-	-	-	GTSTR	General PWM Timer Software Start Register	0x04	32	read-write	0x0000 0000	0xFFFF FFFF
				GTSTP	General PWM Timer Software Stop Register	0x08	32	read-write	0xFFFF FFFF	0xFFFF FFFF
				GTCLR	General PWM Timer Software Clear Register	0x0C	32	write-only	0x0000 0000	0xFFFF FFFF
				GTSSR	General PWM Timer Start Source Select Register	0x10	32	read-write	0x0000 0000	0xFFFF FFFF
				GTPSR	General PWM Timer Stop Source Select Register	0x14	32	read-write	0x0000 0000	0xFFFF FFFF
				GTCSR	General PWM Timer Clear Source Select Register	0x18	32	read-write	0x0000 0000	0xFFFF FFFF
				GTUPSR	General PWM Timer Up Count Source Select Register	0x1C	32	read-write	0x0000 0000	0xFFFF FFFF
				GTDNSR	General PWM Timer Down Count Source Select Register	0x20	32	read-write	0x0000 0000	0xFFFF FFFF
				GTICASR	General PWM Timer Input Capture Source Select Register A	0x24	32	read-write	0x0000 0000	0xFFFF FFFF
				GTICBSR	General PWM Timer Input Capture Source Select Register B	0x28	32	read-write	0x0000 0000	0xFFFF FFFF
				GTCR	General PWM Timer Control Register	0x2C	32	read-write	0x0000 0000	0xFFFF FFFF
				GTUDDTYC	General PWM Timer Count Direction and Duty Setting Register	0x30	32	read-write	0x0000 0001	0xFFFF FFFF
				GTIOR	General PWM Timer I/O Control Register	0x34	32	read-write	0x0000 0000	0xFFFF FFFF

Table 3.4 Register description (30 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
GPT161-166	-	-	-	GTINTAD	General PWM Timer Interrupt Output Setting Register	0x38	32	read-write	0x0000 0000	0xFFFF FFFF
				GTST	General PWM Timer Status Register	0x3C	32	read-write	0x0000 8000	0xFFFF FFFF
				GTBER	General PWM Timer Buffer Enable Register	0x40	32	read-write	0x0000 0000	0xFFFF FFFF
				GTCNT	General PWM Timer Counter	0x48	32	read-write	0x0000 0000	0xFFFF FFFF
				GTCCRA	General PWM Timer Compare Capture Register A	0x4C	32	read-write	0x0000 FFFF	0xFFFF FFFF
				GTCCRB	General PWM Timer Compare Capture Register B	0x50	32	read-write	0x0000 FFFF	0xFFFF FFFF
				GTCCRC	General PWM Timer Compare Capture Register C	0x54	32	read-write	0x0000 FFFF	0xFFFF FFFF
				GTCCRE	General PWM Timer Compare Capture Register E	0x58	32	read-write	0x0000 FFFF	0xFFFF FFFF
				GTCCRD	General PWM Timer Compare Capture Register D	0x5C	32	read-write	0x0000 FFFF	0xFFFF FFFF
				GTCCRF	General PWM Timer Compare Capture Register F	0x60	32	read-write	0x0000 FFFF	0xFFFF FFFF
				GTPR	General PWM Timer Cycle Setting Register	0x64	32	read-write	0x0000 FFFF	0xFFFF FFFF
				GTPBR	General PWM Timer Cycle Setting Buffer Register	0x68	32	read-write	0x0000 FFFF	0xFFFF FFFF
				GTDTCR	General PWM Timer Dead Time Control Register	0x88	32	read-write	0x0000 0000	0xFFFF FFFF
				GTDVU	General PWM Timer Dead Time Value Register U	0x8C	32	read-write	0x0000 FFFF	0xFFFF FFFF
GPT_OPS	-	-	-	OPSCR	Output Phase Switching Control Register	0x0	32	read-write	0x0000 0000	0xFFFF FFFF
KINT	-	-	-	KRCTL	KEY Return Control Register	0x00	8	read-write	0x00	0xFF
				KRF	KEY Return Flag Register	0x04	8	read-write	0x00	0xFF
				KRM	KEY Return Mode Register	0x08	8	read-write	0x00	0xFF

Table 3.4 Register description (31 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
CTSU	-	-	-	CTSUCR0	CTSU Control Register 0	0x00	8	read-write	0x00	0xFF
				CTSUCR1	CTSU Control Register 1	0x01	8	read-write	0x00	0xFF
				CTSUSDPRS	CTSU Synchronous Noise Reduction Setting Register	0x02	8	read-write	0x00	0xFF
				CTSUSST	CTSU Sensor Stabilization Wait Control Register	0x03	8	read-write	0x00	0xFF
				CTSUMCH0	CTSU Measurement Channel Register 0	0x04	8	read-write	0x3F	0xFF
				CTSUMCH1	CTSU Measurement Channel Register 1	0x05	8	read-only	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
				CTSUCHTRC0	CTSU Channel Transmit/Receive Control Register 0	0x0B	8	read-write	0x00	0xFF
				CTSUCHTRC1	CTSU Channel Transmit/Receive Control Register 1	0x0C	8	read-write	0x00	0xFF
				CTSUCHTRC2	CTSU Channel Transmit/Receive Control Register 2	0x0D	8	read-write	0x00	0xFF
				CTSUCHTRC3	CTSU Channel Transmit/Receive Control Register 3	0x0E	8	read-write	0x00	0xFF

Table 3.4 Register description (32 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
CTSU	-	-	-	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
AGT0,1	-	-	-	AGT	AGT Counter Register	0x00	16	read-write	0xFFFF	0xFFFF
				AGTCMA	AGT Compare Match A Register	0x02	16	read-write	0xFFFF	0xFFFF
				AGTCMB	AGT Compare Match B Register	0x04	16	read-write	0xFFFF	0xFFFF
				AGTCR	AGT Control Register	0x08	8	read-write	0x00	0xFF
				AGTMR1	AGT Mode Register 1	0x09	8	read-write	0x00	0xFF
				AGTMR2	AGT Mode Register 2	0x0A	8	read-write	0x00	0xFF
				AGTIOC	AGT I/O Control Register	0x0C	8	read-write	0x00	0xFF
				AGTISR	AGT Event Pin Select Register	0x0D	8	read-write	0x00	0xFF
				AGTCMSR	AGT Compare Match Function Select Register	0x0E	8	read-write	0x00	0xFF
				AGTIOSEL	AGT Pin Select Register	0x0F	8	read-write	0x00	0xFF
ACMPHS0	-	-	-	CMPCTL	Comparator Control Register	0x000	8	read-write	0x00	0xFF
				CMPSEL0	Comparator Input Select Register	0x004	8	read-write	0x00	0xFF
				CMPSEL1	Comparator Reference Voltage Select Register	0x008	8	read-write	0x00	0xFF
				CMPMON	Comparator Output Monitor Register	0x00C	8	read-only	0x00	0xFF
				CPIOC	Comparator Output Control Register	0x010	8	read-write	0x00	0xFF

Table 3.4 Register description (33 of 37)

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
				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	0x00	8	read-write	0x00	0xFF
				AMPTRM	Operational Amplifier Trigger Mode Control Register	0x01	8	read-write	0x00	0xFF
				AMPTRS	Operational Amplifier Activation Trigger Select Register	0x02	8	read-write	0x00	0xFF
				AMPC	Operational Amplifier Control Register	0x03	8	read-write	0x00	0xFF
				AMPMON	Operational Amplifier Monitor Register	0x04	8	read-only	0x00	0xFF
				AMP0OS	Operational Amplifier 0 Output Select Register	0x06	8	read-write	0x00	0xFF
				AMP0MS	Operational Amplifier 0 Minus Input Select Register	0x07	8	read-write	0x00	0xFF

Table 3.4 Register description (34 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
OPAMP	-	-	-	AMP0PS	Operational Amplifier 0 Plus Input Select Register	0x08	8	read-write	0x00	0xFF
				AMP1MS	Operational Amplifier 1 Minus Input Select Register	0x0A	8	read-write	0x00	0xFF
				AMP1PS	Operational Amplifier 1 Plus Input Select Register	0x0B	8	read-write	0x00	0xFF
				AMP2MS	Operational Amplifier 2 Minus Input Select Register	0x0D	8	read-write	0x00	0xFF
				AMP2PS	Operational Amplifier 2 Plus Input Select Register	0x0E	8	read-write	0x00	0xFF
				AMPCPC	Operational Amplifier Switch Charge Pump Control Register	0x12	8	read-write	0x00	0xFF
				AMPUOTE	Operational Amplifier User Offset Trimming Enable Register	0x17	8	read-write	0x00	0xFF
				AMP0OTP	Operational Amplifier 0 Offset Trimming Pch Register	0x18	8	read-write	0x00	0xE0
				AMP0OTN	Operational Amplifier 0 Offset Trimming Nch Register	0x19	8	read-write	0x00	0xE0
				AMP1OTP	Operational Amplifier 1 Offset Trimming Pch Register	0x1A	8	read-write	0x00	0xE0
				AMP1OTN	Operational Amplifier 1 Offset Trimming Nch Register	0x1B	8	read-write	0x00	0xE0
				AMP2OTP	Operational Amplifier 2 Offset Trimming Pch Register	0x1C	8	read-write	0x00	0xE0
	-	-	-	AMP2OTN	Operational Amplifier 2 Offset Trimming Nch Register	0x1D	8	read-write	0x00	0xE0

Table 3.4 Register description (35 of 37)

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	0xFFFF
				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
				CFIFOSEL	CFIFO Port Select Register	0x020	16	read-write	0x0000	0xFFFF
				CFIFOCTR	CFIFO Port Control Register	0x022	16	read-write	0x0000	0xFFFF
				INTENB0	Interrupt Enable Register 0	0x030	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
				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-only	0x0000	0xFFFF
				USBVAL	USB Request Value Register	0x056	16	read-only	0x0000	0xFFFF

Table 3.4 Register description (36 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
USBFS	-	-	-	USBINDX	USB Request Index Register	0x058	16	read-only	0x0000	0xFFFF
				USBLENG	USB Request Length Register	0x05A	16	read-only	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
	2	0x002	4,5	PIPE%sCTR	Pipe %s Control Register	0x076	16	read-write	0x0000	0xFFFF
	2	0x002	6,7	PIPE%sCTR	Pipe %s Control Register	0x07A	16	read-write	0x0000	0xFFFF
	2	0x004	4,5	PIPE%sTRE	Pipe %s Transaction Counter Enable Register	0x09C	16	read-write	0x0000	0xFFFF
	2	0x004	4,5	PIPE%sTRN	Pipe %s Transaction Counter Register	0x09E	16	read-write	0x0000	0xFFFF
	-	-	-	USBBCCTRL0	BC Control Register 0	0x0B0	16	read-write	0x0000	0xFFFF
				UCKSEL	USB Clock Selection Register	0x0C4	16	read-write	0x0000	0xFFFF
				USBMC	USB Module Control Register	0x0CC	16	read-write	0x0002	0xFFFF
SDADC24	-	-	-	STC1	Startup Control Register 1	0x00	16	read-write	0x8008	0xFFFF
				STC2	Startup Control Register 2	0x04	8	read-write	0x00	0xFF
	5	0x04	0-4	PGAC%s	Input Multiplexer %s Setting Register	0x08	32	read-write	0x0001 0040	0xFFFF FFFF

Table 3.4 Register description (37 of 37)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SDADC24	-	-	-	ADC1	Sigma-Delta A/D Converter Control Register 1	0x1C	32	read-write	0x0000 0000	0xFFFF FFFF
				ADC2	Sigma-Delta A/D Converter Control Register 2	0x20	8	read-write	0x00	0xFF
				ADCR	Sigma-Delta A/D Converter Conversion Result Register	0x24	32	read-write	0x0000 0000	0xFFFF FFFF
				ADAR	Sigma-Delta A/D Converter Average Value Register	0x28	32	read-only	0x0000 0000	0xFFFF FFFF
				CLBC	Calibration Control Register	0x30	8	read-write	0x00	0xFF
				CLBSTR	Calibration Start Control Register	0x34	8	read-write	0x00	0xFF
				CLBSSR	Calibration Status Register	0x3C	8	read-only	0x00	0xFF
				CLBPR	Calibration Control Protection Release Register	0x40	8	read-write	0x00	0xFF
	5	0x04	0-4	GCVLR%S	Gain Error Correction Factor Register %s	0x48	16	read-write	0x4492	0xFFFF
	5	0x04	0-4	OCVLR%S	Offset Error Correction Factor Register %s	0x5C	32	read-write	0x0000 0000	0xFFFF FFFF
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
				DACADSCR	D/A A/D Synchronous Start Control Register	0x06	8	read-write	0x00	0xFF
				DACPC	D/A SW Charge Pump Control Register	0x07	8	read-write	0x00	0xFF
FLCN	-	-	-	DFLCTL	DFLCTL Data flash Control Register	0x90	8	read-write	0x00	0xFF
TSN	-	-	-	TSCDRL	Temperature Sensor Calibration Data Register L	0x228	8	read-only	0x00	0x00
				TSCDRH	Temperature Sensor Calibration Data Register H	0x229	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

Appendix 4. Connection Diagram of Analog Block

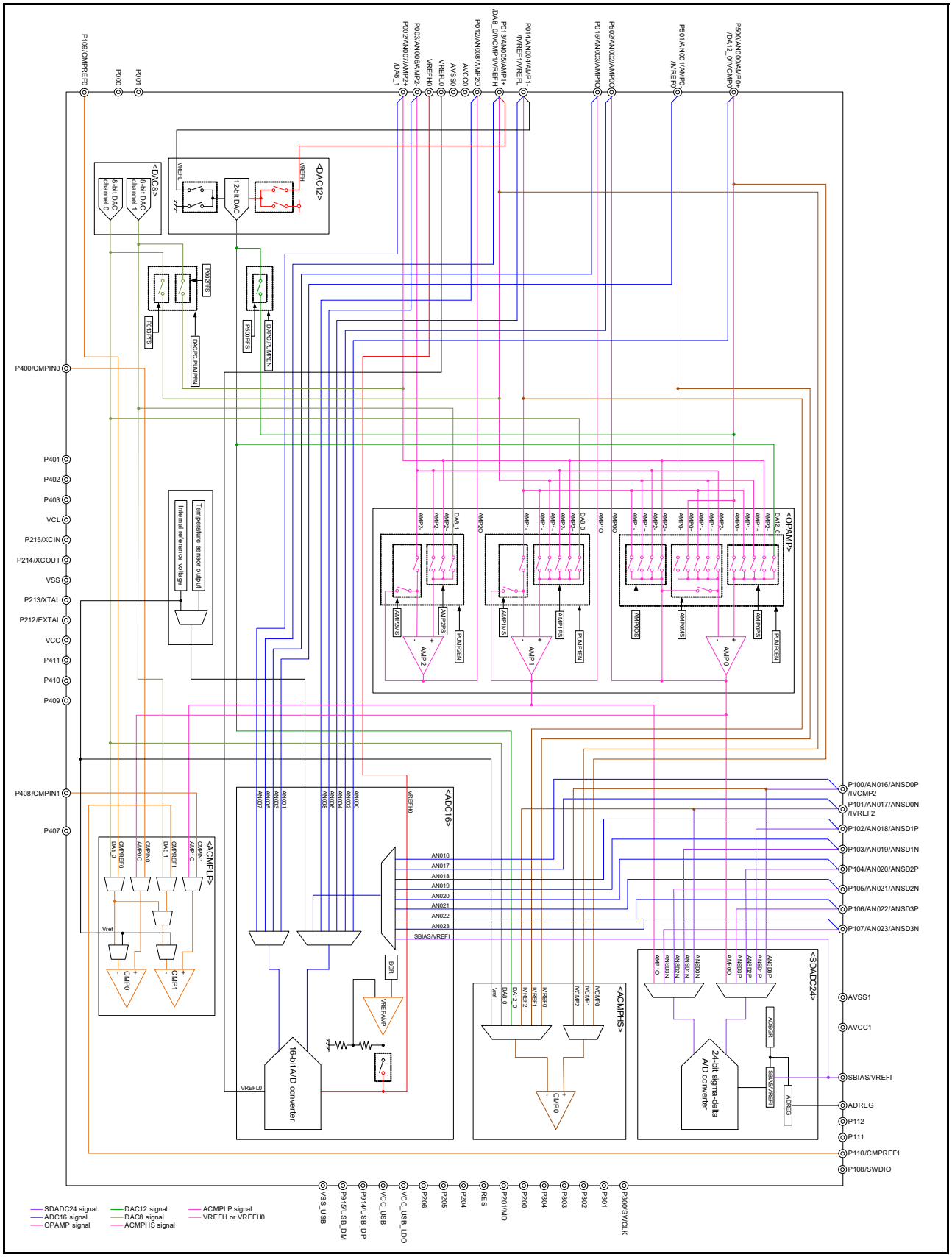


Figure 4.1 Connection diagram of analog block in 64-pin product

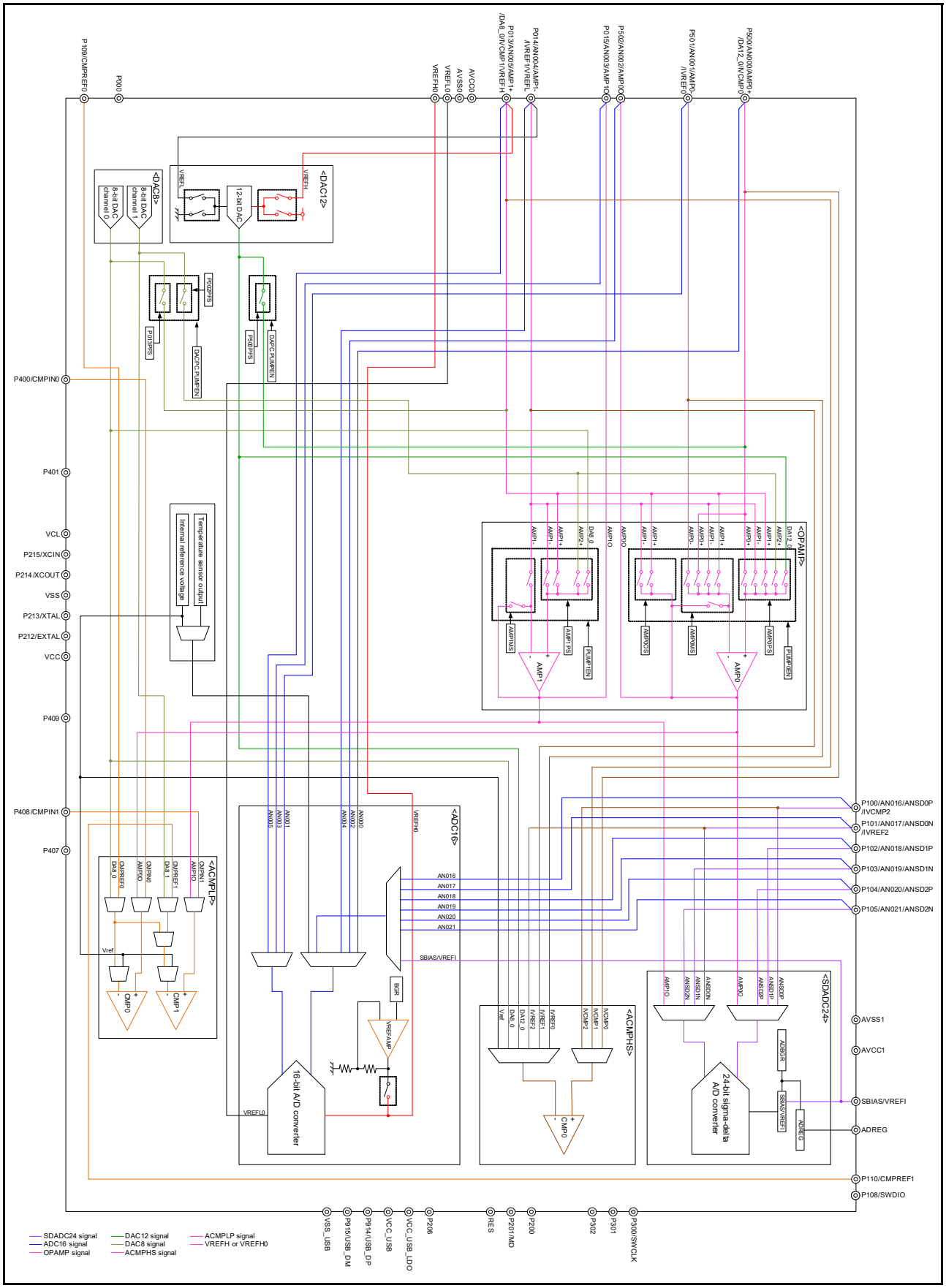


Figure 4.2 Connection diagram of analog block in 48-pin product

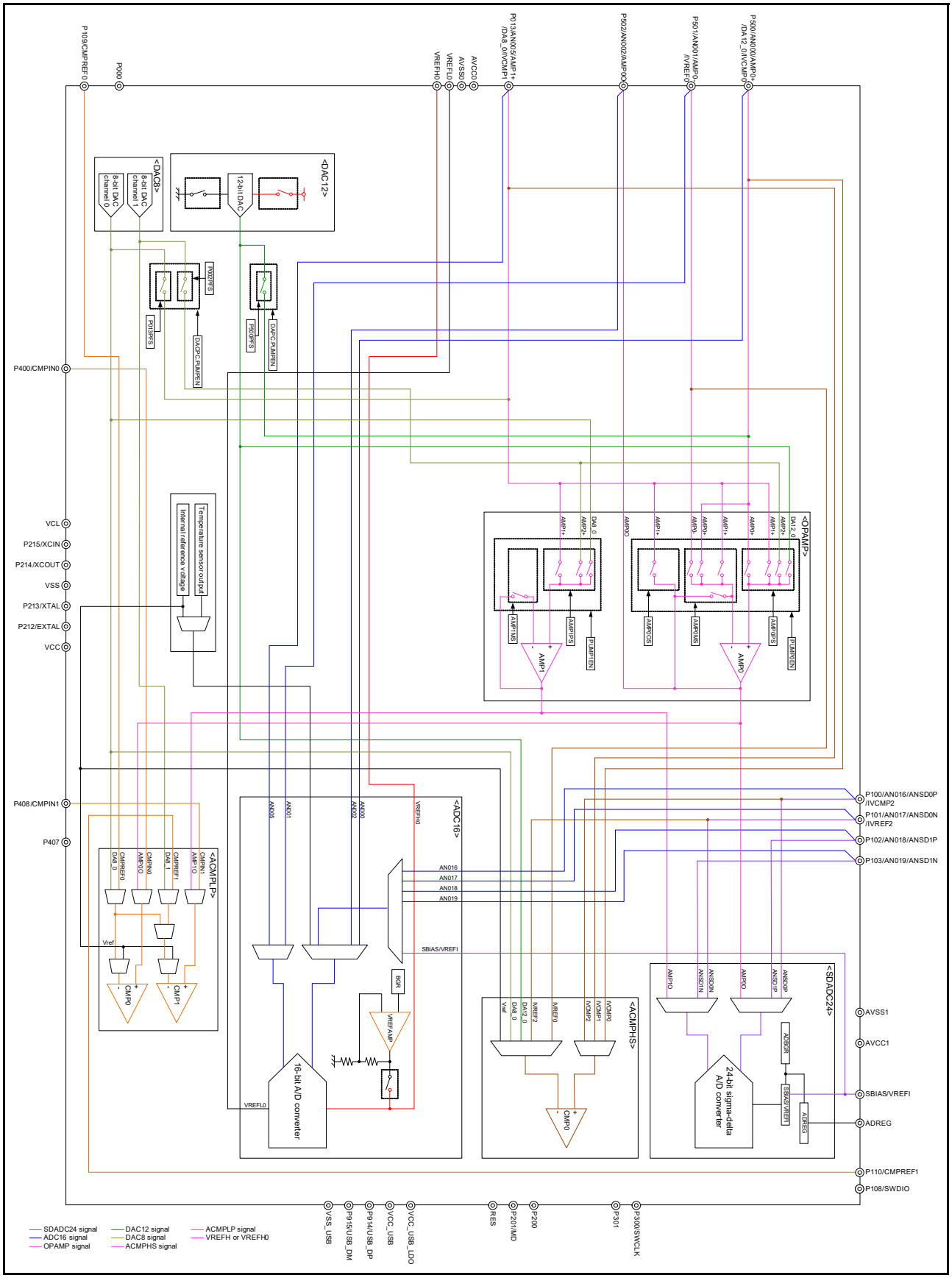


Figure 4.3 Connection diagram of analog block in 40-pin product

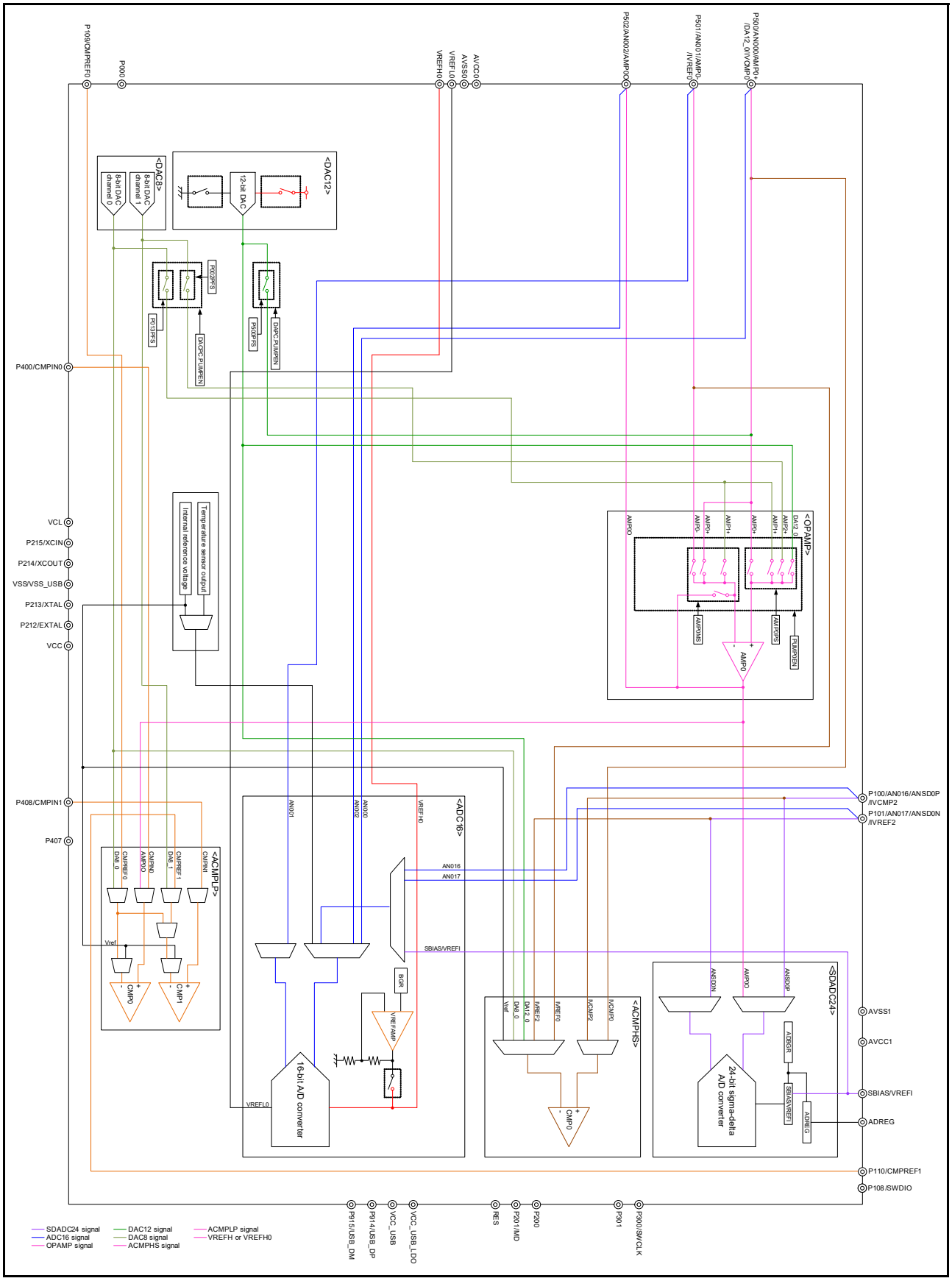


Figure 4.4 Connection diagram of analog block in 36-pin product

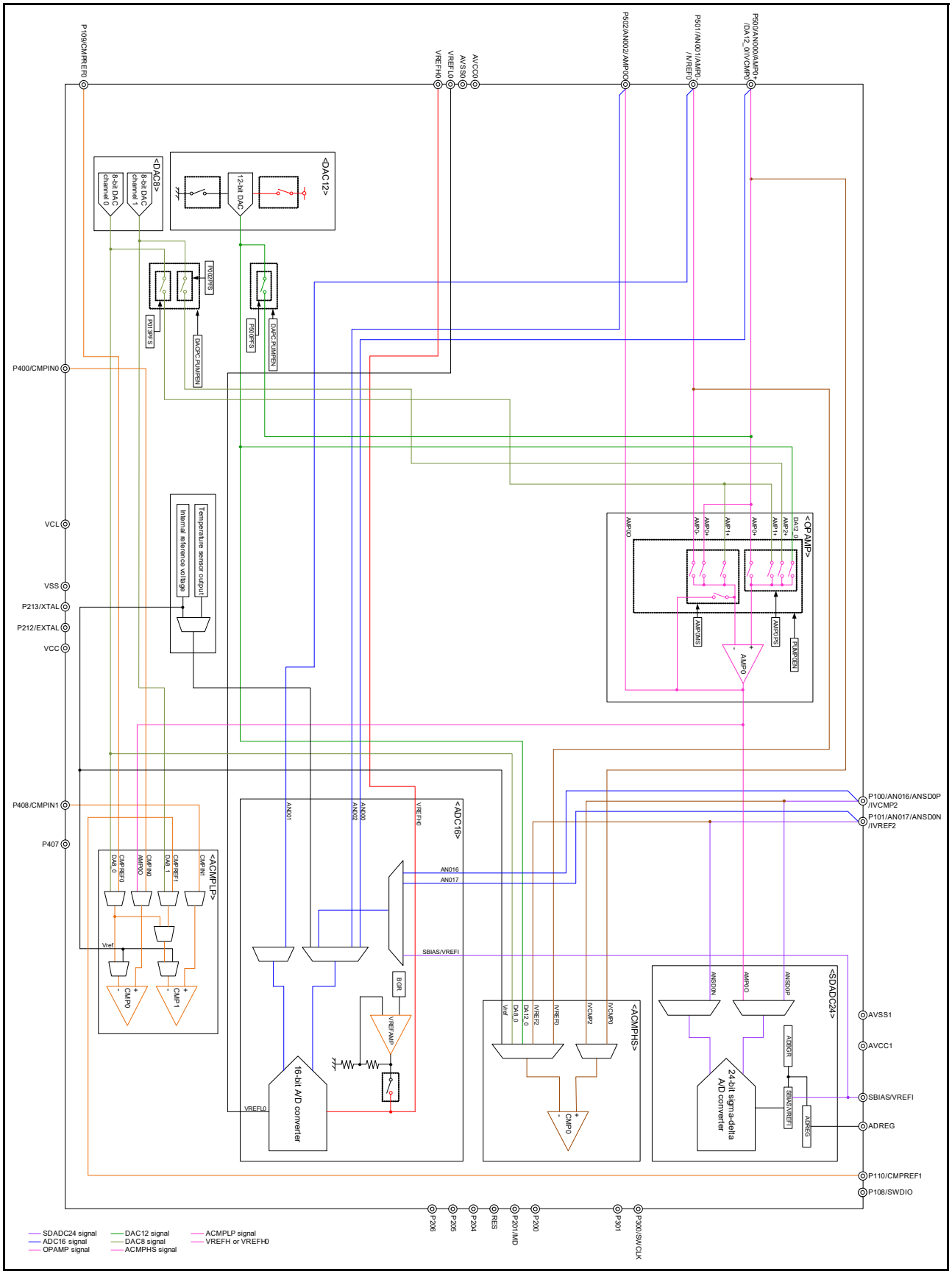


Figure 4.5 Connection diagram of analog block in 32-pin product

Revision History	RA2A1 Group User's Manual

Rev.	Date	Chapter	Summary
1.00	Oct 8, 2019	—	First Edition issued
1.10	Jul 14, 2023	—	Second Edition issued
		Features	Updated for Timers
		section 1, Overview	Updated Table 1.6, Timers
			Updated Table 1.7, Communication interfaces
			Added Table 1.12, I/O ports
			Updated Figure 1.2, Part numbering scheme
			Updated Table 1.13, Product list
			Updated Table 1.14, Function comparison
			Updated Figure 1.4, Pin assignment for QFN 48-pin
			Updated Figure 1.5, Pin assignment for QFN 40-pin
		section 2, CPU	Updated section 2.1.3, Operating Frequency
			Updated section 2.6.5.2, Debug Stop Control Register (DBGSTOPCR)
			Updated section 2.8.3.4, Connecting sequence and SWD authentication
		section 5, Memory Mirror Function (MMF)	Updated Figure 5.1, MMF operation
			Updated Figure 5.2, MMF block diagram
		section 7, Option-Setting Memory	Updated Figure 7.1, Option-setting memory area
			Updated section 7.2.1, Option Function Select Register 0 (OFS0)
			Updated section 7.2.2, Option Function Select Register 1 (OFS1)
			Updated Table 7.1, MPU registers
			Updated Figure 7.2, Access window overview
			Updated Table 7.2, Specifications for ID code protection
		section 8, Low Voltage Detection (LVD)	Updated section 8.2.3, Voltage Monitor 2 Circuit Control Register 1 (LVD2CR1)
			Updated Figure 8.4, Example of voltage monitor 0 reset operation
			Updated Table 8.4, Procedure for setting bits related to the voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring operates
			Updated Table 8.7, Procedure for setting bits related to voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitor stops
			Updated section 8.7, Event Link Output
		section 9, Clock Generation Circuit	Updated Table 9.1, Clock generation circuit specifications for the clock sources
			Updated Figure 9.1, Clock generation circuit block diagram
			Updated section 9.2.4, Main Clock Oscillator Control Register (MOSCCR)
			Updated section 9.2.5, Sub-Clock Oscillator Control Register (SOSCCR)
			Updated section 9.2.6, Low-Speed On-Chip Oscillator Control Register (LOCOCR)
			Updated section 9.2.7, High-Speed On-Chip Oscillator Control Register (HOCOCR)
			Updated section 9.2.8, Middle-Speed On-Chip Oscillator Control Register (MOCOCR)
			Updated section 9.2.9, Oscillation Stabilization Flag Register (OSCSF)
			Updated section 9.2.14, Main Clock Oscillator Mode Oscillation Control Register (MOMCR)
			Updated Figure 9.9, Clock source switching timing diagram
			Updated section 9.6.10, SysTick Timer-Dedicated Clock (SYSTICCLK)
			Updated Figure 9.10, Signal routing in board design for oscillation circuit (applicable to the main clock oscillator as well as sub-clock oscillator) title
		section 11, Low Power Modes	Updated Table 11.2, Operating conditions of each low power mode
			Updated section 11.2.5, Module Stop Control Register D (MSTPCRD)
			Updated section 11.5.1, Setting Operating Power Control Mode
			Updated section 11.6.2, Canceling Sleep Mode
			Updated section 11.7.3, Example of Software Standby Mode Application
			Updated Figure 11.7, Example of Software Standby mode application
			Updated Figure 11.12, Setting example of using SCI0 in Snooze mode entry
			Updated section 11.9.1, Register Access

Rev.	Date	Chapter	Summary
1.10	Jul 14, 2023	section 13, Interrupt Controller Unit (ICU)	Updated Figure 13.1, ICU block diagram
			Updated section 13.2.1, IRQ Control Register i (IRQCRi) (i = 0 to 7)
			Updated section 13.2.2, Non-Maskable Interrupt Status Register (NMISR)
			Updated section 13.2.3, Non-Maskable Interrupt Enable Register (NMIER)
			Updated section 13.2.5, NMI Pin Interrupt Control Register (NMICR)
			Updated section 13.2.7, SYS Event Link Setting Register (SELSR0)
			Updated section 13.4.1, Detecting Interrupts
		section 15, Memory Protection Unit (MPU)	Updated section 15.3, Arm MPU
			Updated section 15.4.1.3, Group A Region n Access Control Register (MMPUACAn) (n = 0 to 3)
			Updated section 15.6.1.1, Security MPU Program Counter Start Address Register n (SECMPUPCSn) (n = 0, 1)
			Updated section 15.6.1.2, Security MPU Program Counter End Address Register n (SECMPUPCEn) (n = 0, 1)
			Updated section 15.6.1.3, Security MPU Region 0 Start Address Register (SECMPUS0)
			Updated section 15.6.1.4, Security MPU Region 0 End Address Register (SECMPUE0)
			Updated section 15.6.1.5, Security MPU Region 1 Start Address Register (SECMPUS1)
			Updated section 15.6.1.6, Security MPU Region 1 End Address Register (SECMPUE1)
			Updated section 15.6.1.7, Security MPU Region 2 Start Address Register (SECMPUS2)
			Updated section 15.6.1.8, Security MPU Region 2 End Address Register (SECMPUE2)
			Updated section 15.6.1.9, Security MPU Region 3 Start Address Register (SECMPUS3)
			Updated section 15.6.1.10, Security MPU Region 3 End Address Register (SECMPUE3)
			Updated section 15.6.1.11, Security MPU Access Control Register (SECMPUAC)
		section 16, Data Transfer Controller (DTC)	Updated section 16.10, Module-Stop Function
		section 18, I/O Ports	Updated Table 18.2, I/O port functions
			Updated section 18.2.1, Port Control Register 1 (PCNTR1/PODR/PDR)
			Updated section 18.2.2, Port Control Register 2 (PCNTR2/EIDR/PIDR)
			Updated section 18.2.3, Port Control Register 3 (PCNTR3/PORR/POSR)
			Updated section 18.2.4, Port Control Register 4 (PCNTR4/EORR/EOSR)
			Updated section 18.2.5, Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) (m = 0 to 5, 9; n = 00 to 15)
			Updated section 18.3.2, Port Function Select
			Updated Table 18.3, Handling of unused pins
			Updated Table 18.5, Register settings for I/O pin functions (PORT0)
			Updated Table 18.6, Register settings for input/output pin function (PORT1) (1)
			Updated Table 18.7, Register settings for input/output pin function (PORT1) (2)
			Updated Table 18.8, Register settings for input/output pin function (PORT2)
			Updated Table 18.9, Register settings for input/output pin function (PORT3)
			Updated Table 18.10, Register settings for input/output pin function (PORT4)
			Updated Table 18.12, Register settings for input/output pin function (PORT9)
		section 19, Key Interrupt Function (KINT)	Updated section 19.3.2, Operation When Using Key Interrupt Flag (KRMD = 1)
		section 20, Port Output Enable for GPT (POEG)	Updated Table 20.1, POEG specifications
			Updated section 20.3, Output-Disable Control Operation
			Updated section 20.3.1.1, Digital filter
			Updated Figure 20.2, Example of digital filter operation
			Updated section 20.4, Interrupt Sources
			Updated section 20.5, External Trigger Output to the GPT
			Updated Figure 20.4, Output timing of external trigger to the GPT
		section 21, General PWM Timer (GPT)	Updated Note 1. in Table 21.2, GPT functions
			Updated section 21.2.12, General PWM Timer Control Register (GTCR)

Rev.	Date	Chapter	Summary
1.10	Jul 14, 2023	section 21, General PWM Timer (GPT)	Updated section 21.2.13, General PWM Timer Count Direction and Duty Setting Register (GTUDDTYC)
			Updated section 21.2.14, General PWM Timer I/O Control Register (GTIOR)
			Updated Note 1. in Table 21.5, Settings of GTIOA[4:0] and GTIOB[4:0] bits
			Updated section 21.2.24, Output Phase Switching Control Register (OPSCR)
			Updated section 21.3.1.1, Counter operation
			Updated Figure 21.8, Example setting for an event count operation in up-counting using hardware sources
			Updated Figure 21.10, Example setting for an event count operation in down-counting using hardware sources
			Updated section 21.3.1.3, Input capture function
			Updated Figure 21.22, Example of GTCCRA and GTCCRB buffer operation with output compare, saw waves in up-counting, high output at GTCCRA compare match, and low output at cycle end
			Updated Figure 21.43, Example setting for automatic dead time setting function in saw-wave one-shot pulse mode, and triangle-wave PWM mode 3
			Updated section 21.3.11, Output Phase Switching (GPT_OPS)
			Updated section 21.3.11.6, Event Link Controller (ELC) output
			Updated Figure 21.82, Timing of noise filtering
			Updated section 21.6, Noise Filter Function
			Updated section 21.9.1, Module-Stop Function Setting
			Updated section 21.9.5, Priority Order of Each Event
		section 22, Low Power Asynchronous General Purpose Timer (AGT)	Updated section 22, Low Power Asynchronous General Purpose Timer (AGT) title
			Updated section 22.1, Overview
			Updated Figure 22.1, AGT block diagram
			Updated Note 6. in section 22.2.5, AGT Mode Register 1 (AGTMR1)
			Updated section 22.2.7, AGT I/O Control Register (AGTIOC)
			Updated Figure 22.2, Timing of rewrite operation with TSTART, TCMEA and TCMEB bit values when compare match A register and compare match B register are invalid title
			Updated Figure 22.8, Operation example 2 in event counter mode
			Updated Figure 22.11, Operation example in compare match mode (TOPOLA = 0, TOPOLB = 0)
			Updated Table 22.9, Usable settings in Software Standby mode (AGT0)
			Updated Table 22.10, Usable settings in Software Standby mode (AGT1)
			Updated section 22.4.3, When Changing Mode
			Updated section 22.4.11, When Switching Source Clock
		section 23, Realtime Clock (RTC)	Updated Figure 23.3, Clock and count mode setting procedure
			Updated section 23.6.8, When Switching Source Clock
		section 24, Watchdog Timer (WDT)	Updated Figure 24.1, WDT block diagram
			Updated section 24.2.2, WDT Control Register (WDTCR)
			Updated Figure 24.2, RPSS[1:0] and RPES[1:0] bit settings and refresh-permitted period
			Updated section 24.2.4, WDT Reset Control Register (WDTRCR)
			Updated section 24.2.5, WDT Count Stop Control Register (WDTCSTPR)
			Updated section 24.3.1.1, Register start mode
			Updated section 24.3.1.2, Auto start mode
			Updated section 24.3.2, Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers
			Updated Figure 24.5, Control waveforms produced in response to writes to the WDTCR register
			Updated section 24.3.3, Refresh Operation
			Updated Table 24.5, Association between Option Function Select Register 0 (OFS0) and the WDT registers
			Updated section 24.5.1, ICU Event Link Setting Register n (IELSRn) Setting
		section 25, Independent Watchdog Timer (IWDT)	Updated Figure 25.1, IWDT block diagram
			Updated section 25.2.3, Option Function Select Register 0 (OFS0)
			Updated Figure 25.2, IWDTRPSS[1:0] and [IWDTRPES[1:0] bit settings and refresh-permitted period
			Updated section 25.3.1, Auto Start Mode

Rev.	Date	Chapter	Summary
1.10	Jul 14, 2023	section 25, Independent Watchdog Timer (IWDT)	Updated section 25.3.2, Refresh Operation
			Updated Figure 25.4, IWDT refresh operation waveforms when OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOPS[1:0] = 11b
			Updated Figure 25.5, Processing for reading IWDT counter value when OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOPS[1:0] = 11b
		section 26, USB 2.0 Full-Speed Module (USBFS)	Updated section 26.2.5, CFIFO Port Select Register (CFIFOSEL)
			Updated section 26.2.6, CFIFO Port Control Register (CFIFOCTR)
			Updated Note 5. in section 26.2.12, Interrupt Status Register 0 (INTSTS0)
			Updated section 26.2.27, PIPEn Control Registers (PIPEnCTR) (n = 4 to 7)
			Updated section 26.4.3, Clearing the Interrupt Status Register after Setting the Port Function
		section 27, Serial Communications Interface (SCI)	Updated Table 27.1, SCI specifications
			Updated section 27.2.11, Serial Control Register (SCR) for Non-Smart Card Interface Mode (SCMR.SMIF = 0)
			Updated section 27.2.12, Serial Control Register for Smart Card Interface Mode (SCR_SMCI) (SCMR.SMIF = 1)
			Updated section 27.2.13, Serial Status Register (SSR) for Non-Smart Card Interface and Non-FIFO Mode (SCMR.SMIF = 0 and FCR.FM = 0)
			Updated section 27.2.17, Bit Rate Register (BRR)
			Updated Note. in Table 27.17, BRR settings for different bit rates in simple IIC mode
			Updated Table 27.20, Examples of BRR and MDDR settings for different bit rates in asynchronous mode (1)
			Updated section 27.2.19, Serial Extended Mode Register (SEMR)
			Updated section 27.3.7, SCI Initialization in Asynchronous Mode
			Updated Figure 27.13, Data format written to FTDRH and FTDRL with FIFO selected
			Updated Table 27.24, Flags in SSR Status Register and receive data handling
			Updated Figure 27.19, Data format stored to FRDRH and FRDRL with FIFO selected
			Updated Figure 27.24, Data format written to FTDRH and FTDRL in multi-processor mode with FIFO selected
			Updated Figure 27.29, Data format stored to FRDRH and FRDRL in multi-processor mode with FIFO selected
			Updated section 27.5.3, SCI Initialization in Clock Synchronous Mode
			Updated Figure 27.40, Example operation of serial reception in clock synchronous mode (2) when RTS function is used
			Updated Figure 27.43, Example flow of simultaneous serial transmission and reception in clock synchronous mode with non-FIFO selected
			Updated Figure 27.44, Example flow of simultaneous serial transmission and reception in clock synchronous mode with FIFO selected
			Updated Figure 27.50, Example flow of SCI initialization in smart card interface mode
			Updated Figure 27.52, Data retransfer operation in smart card interface transmission mode title
			Updated Figure 27.55, Data retransfer operation in smart card interface reception mode title
			Updated section 27.10.3, Interrupts in Asynchronous, Clock Synchronous, and Simple SPI Modes
			Added section 27.14.11, Notes on Transmit Enable Bit (SCR.TE)
			Added section 27.14.12, Note on Stopping Reception When Using the RTS Function in Asynchronous Mode
		section 28, I^{2}C Bus Interface (IIC)	Updated section 28.2.2, I^{2}C Bus Control Register 2 (ICCR2)
			Updated section 28.2.6, I^{2}C Bus Function Enable Register (ICFER)
			Updated section 28.2.10, I^{2}C Bus Status Register 2 (ICSR2)
			Updated Table 28.5, Wakeup mode
			Updated section 28.2.12, I^{2}C Bus Wakeup Unit Register 2 (ICWUR2)
			Updated section 28.3.3, Master Transmit Operation
			Updated section 28.3.4, Master Receive Operation
			Updated section 28.3.6, Slave Receive Operation
			Updated Figure 28.23, Digital noise-filter circuit block diagram
			Updated section 28.7.3, Device ID Address Detection

Rev.	Date	Chapter	Summary
1.10	Jul 14, 2023	section 28, I²C Bus Interface (IIC)	Updated Figure 28.28, AASy/DID flag set and clear timing during reception of device ID
			Updated Figure 28.32, Timing of normal wakeup mode 1
			Updated Figure 28.34, Timing of normal wakeup mode 2
			Updated Figure 28.37, Timing of command recovery and EEP response modes
			Updated Figure 28.39, Suspension of data transfer when NACK is received (NACKE = 1)
			Updated section 28.10.1, Master Arbitration-Lost Detection (MALE Bit)
			Updated section 28.10.3, Slave Arbitration-Lost Detection (SALE Bit)
			Updated section 28.11.2, Issuing a Restart Condition
			Updated section 28.12.2, Extra SCL Clock Cycle Output Function
			Updated Figure 28.49, Extra SCL clock cycle output function using the CLO bit
			Updated Table 28.11, Register states when issuing each condition
		section 30, Serial Peripheral Interface (SPI)	Updated Table 30.1, SPI specifications
			Updated section 30.2.7, SPI Data Control Register (SPDCR)
			Updated Table 30.4, Relationship between SPCR settings and SPI modes
		section 31, Cyclic Redundancy Check (CRC) Calculator	Updated section 31.2.1, CRC Control Register 0 (CRCCR0)
		section 32, 16-Bit A/D Converter (ADC16)	Updated section 32.7.5, Settings for the Module-Stop State
		section 33, 24-Bit Sigma-Delta A/D Converter (SDADC24)	Updated section 33.2.1, Startup Control Register 1 (STC1)
			Updated section 33.3.5.1, Input Multiplexer Control Registers
			Updated section 33.3.8, Control of the Sigma-Delta A/D Converter (AUTOS-CAN)
			Updated Figure 33.17, Digital filter block diagram
			Updated section 33.5.2, Settings for the Module-Stop State
			Updated section 33.5.7, Restrictions on the Multiplexer 4 (Internal OPAMP)
		section 34, 12-Bit D/A Converter (DAC12)	Updated section 34.2.2, D/A Control Register (DACR)
			Updated section 34.6.1, Settings for the Module-Stop Function
		section 35, 8-Bit D/A Converter (DAC8)	Updated section 35.4.1, Settings for the Module-Stop Function
			Updated section 35.4.8, Connection of D/A Converter Output
		section 37, Operational Amplifier (OPAMP)	Updated section 37.1, Overview
			Updated section 37.10.4, Using the Configurable Amplifier as a D/A Converter Output Amplifier
		section 38, High-Speed Analog Comparator (ACMPHS)	Updated section 38.8.1, Settings for the Module-Stop Function
		section 39, Low-Power Analog Comparator (ACMPLP)	Updated section 39.9.1, Settings for the Module-Stop State
		section 40, Capacitive Touch Sensing Unit (CTSU)	Updated section 40.2.3, CTSU Synchronous Noise Reduction Setting Register (CTSUSDPRS)
			Updated section 40.2.5, CTSU Measurement Channel Register 0 (CTSUMCH0)
			Updated section 40.2.19, CTSU Sensor Offset Register 1 (CTSUSO1)
			Updated section 40.2.22, CTSU Error Status Register (CTSUERRS)
			Updated Figure 40.14, Software flow and operation example of self-capacitance multi-scan mode
			Updated Figure 40.16, Software flow and operation example of mutual-capacitance full scan mode
			Updated section 40.3.2.5, Mutual-capacitance full scan mode operation
			Updated section 40.3.3.2, Interrupts
			Updated section 40.4.2, Constraint on Software Trigger
		section 42, SRAM	Updated Figure 42.2, Flow of SRAM parity check when SRAM parity reset is enabled
			Updated Figure 42.3, Flow of SRAM parity check when SRAM parity interrupt is enabled
		section 43, Flash Memory	Updated section 43.4.2, Flash Cache Invalidate Register (FCACHEIV)
			Updated Table 43.6, Specifications for ID code protection
		section 47, Electrical Characteristics	Updated Table 47.16, Operation frequency in middle-speed operating mode title
			Updated Table 47.17, Operation frequency in low-speed operating mode title

Rev.	Date	Chapter	Summary
1.10	Jul 14, 2023	section 47, Electrical Characteristics	Updated Table 47.18, Operation frequency in low-voltage operating mode title
			Updated Table 47.19, Operation frequency in Subosc-speed operating mode title
		Appendix 2, Package Dimensions	Added Table 2.5, QFN 48-pin (2)
			Added Table 2.7, QFN 40-pin (2)
		Appendix 3, I/O Registers	Updated Table 3.1, Peripheral base address
			Updated Table 3.2, Access cycles for non-GPT modules
			Updated Table 3.4, Register description

RA2A1 Group User's Manual: Hardware

Publication Date: Rev.1.10 Jul 14, 2023

Published by: Renesas Electronics Corporation

RA2A1 Group

Renesas Electronics Corporation

R01UH0888EJ0110