

RE01 Group

Products with 256-KB Flash Memory
User's Manual: Hardware

RE Family / RE0 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>).

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(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.4.0-1 November 2017)

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.

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

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

1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity.

Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).

7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Preface

1. About this document

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

2. Audience

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

3. Renesas Publications

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

Component	Document Type	Description
Microcontrollers	Data sheet	Features, overview, and electrical characteristics of the MCU
	User's Manual: Hardware	MCU specifications such as pin assignments, memory maps, peripheral functions, electrical characteristics, timing diagrams, and operation descriptions
	Application Notes	Technical notes, board design guidelines, and software migration information
	Technical Update (TU)	Preliminary reports on product specifications such as restriction and errata

4. Numbering Notation

The following numbering notation is used throughout this manual:

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

5. Typographic Notation

The following typographic notation is used throughout this manual:

Example	Description
WDT.WDTRCR.RSTIRQS	Periods separated a function module symbol (WDT), register symbol (WDTRCR), and bit field symbol (RSTIRQS).
WDT.WDTRCR	A period separated a function module symbol (WDT) and register symbol (WDTRCR).
WDTRCR.RSTIRQS	A period separated a register symbol (WDTRCR) and bit field symbol (RSTIRQS).
CKS[3:0]	Numbers in brackets expresses a bit number. For example, CKS[3:0] occupies bits 3 to 0 of the WDT Control Register (WDTCR) register.

6. Unit and Unit Prefix

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

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

7. Special Terms

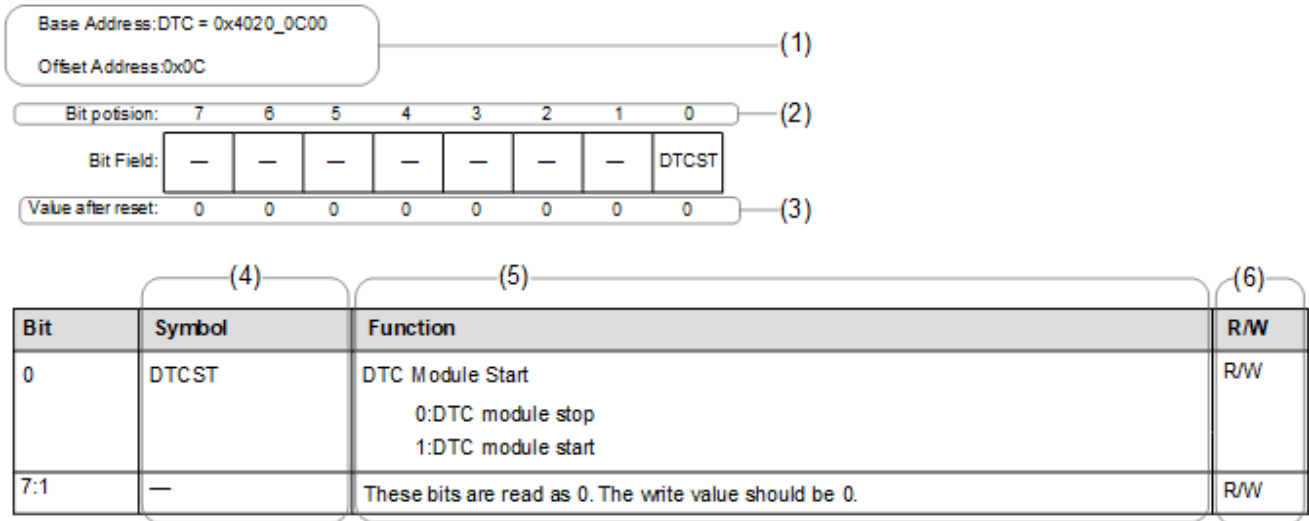
The following terms have special meanings.

Term	Description
NC	Not connected pin. NC means that 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.

XX.XX DTCST : DTC Module Start Register



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

Function module symbol, register symbol, and address assignment of this register are generally expressed. Base Address and Offset Address mean DTC Module Start Register (DTCST) of Data Transfer Controller (DTC) is assigned to address 0x4020_0C00.

(2) Bit number

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

(3) Value after reset

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

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

(4) Symbol

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

(5) Function

Function indicates the full name of the bit field and enumerated values.

(6) R/W

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

R/W: The bit field is readable and writable.

R: The bit field is readable only. Writing to this bit field has no effect.

W: The bit field is writable only. The read value is the same as after a reset unless specified otherwise.

9. Abbreviations

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

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

10. Proprietary Notice

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

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

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

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

11. Feedback on the product

If you have any comments or suggestions about this product, go to [Contact Us](#).

Contents

Features	44
1. Overview.....	45
1.1 Function Outline	45
1.2 Block Diagram	50
1.3 Part Numbering	51
1.4 Function Comparison	51
1.5 Pin Functions.....	54
1.6 Pin Assignments	59
1.7 Pin Lists	63
2. CPU	66
2.1 Overview.....	66
2.1.1 CPU	66
2.1.2 Debug	66
2.1.3 Operating Frequency	66
2.1.4 Block Diagram	66
2.2 Implementation Options.....	67
2.3 SWD Interface	67
2.4 Debug Function	68
2.4.1 Debug Mode Definition	68
2.4.2 Debug Mode Effects	68
2.5 Programmers Model	69
2.5.1 Address Spaces.....	69
2.5.2 Cortex-M0+ Peripheral Address Map	69
2.5.3 External Debug Address Map	70
2.5.4 CoreSight ROM Table	70
2.5.5 DBGREG Module	71
2.5.6 OCDREG Module	73
2.6 SysTick Timer	76
2.7 OCD Emulator Connection	76
2.7.1 Unlock ID Code.....	76
2.7.2 Restrictions on Connecting an OCD emulator	76
2.8 References	79
3. Startup Modes.....	80
3.1 Types and Selection of Startup Mode.....	80
3.2 Details of Startup Modes	80
3.2.1 Normal Startup and Energy Harvesting Startup Modes.....	80
3.2.2 Serial Programming Mode	80

3.2.3	On-chip Debug Mode.....	80
3.3	Startup Modes Transitions.....	80
3.3.1	Startup Mode Determined by the Mode Setting Pins.....	80
3.3.2	Power-up Sequence	81
4.	Address Space.....	82
4.1	Address Space	82
4.2	External Address Space	82
5.	I/O Registers	84
5.1	Address Information	84
5.2	Access Cycle	85
6.	Resets.....	87
6.1	Overview.....	87
6.2	Register Descriptions	90
6.2.1	RSTSR0 : Reset Status Register 0.....	90
6.2.2	RSTSR1 : Reset Status Register 1.....	92
6.2.3	RSTSR2 : Reset Status Register 2.....	94
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	Deep Software Standby Reset	97
6.3.5	Independent Watchdog Timer Reset	97
6.3.6	Watchdog Timer Reset	98
6.3.7	Software Reset	98
6.3.8	MINPWON Mode Reset.....	98
6.3.9	Determination of Cold/Warm Start.....	98
6.3.10	Determination of Reset Generation Source.....	99
7.	Option-Setting Memory.....	101
7.1	Overview.....	101
7.2	Register Descriptions	101
7.2.1	OFS0 : Option Function Select Register 0.....	101
7.2.2	OFS1 : Option Function Select Register 1.....	105
7.2.3	MPU Registers.....	106
7.2.4	AWS : Access Window Setting Register.....	107
7.2.5	OSIS : OCD/Serial Programmer ID Setting Register.....	109
7.3	Setting Option-Setting Memory	109
7.3.1	Allocation of Data in Option-Setting Memory	109
7.3.2	Setting Data for Programming Option-Setting Memory	110
7.4	Usage Notes.....	110

7.4.1	Data for Programming Reserved Areas and Reserved Bits in the Option-Setting Memory ...	110
7.4.2	Note on FSPR Bit	110
7.4.3	Note on the ID code protection	110
8.	Low Voltage Detection (LVD)	111
8.1	Overview.....	111
8.2	Register Descriptions	113
8.2.1	LVCMPCR : Voltage Monitor Circuit Control Register	113
8.2.2	LVDLVLR : Voltage Detection Level Select Register	113
8.2.3	LVD1CR0 : Voltage Monitor 1 Circuit Control Register 0	114
8.2.4	LVDBATCR0 : Voltage Monitor BAT Circuit Control Register 0	115
8.2.5	LVD1CR1 : Voltage Monitor 1 Circuit Control Register	116
8.2.6	LVD1SR : Voltage Monitor 1 Circuit Status Register	117
8.2.7	LVDBATCR1 : Voltage Monitor BAT Circuit Control Register 1	117
8.2.8	LVDBATSR : Voltage Monitor BAT Circuit Status Register	118
8.3	VCC Input Voltage Monitor	118
8.3.1	Monitoring Vdet0.....	118
8.3.2	Monitoring Vdet1.....	118
8.3.3	Monitoring VdetBAT	119
8.4	Reset from Voltage Monitor 0	119
8.5	Interrupt and Reset from Voltage Monitor 1.....	120
8.6	Interrupt and Reset from Voltage Monitor BAT	122
8.7	Event Link Controller (ELC) Output.....	125
8.7.1	Interrupt Handling and Event Linking.....	125
8.8	Usage Notes.....	125
8.8.1	Register Write Protection	125
9.	Clock Generation Circuit	126
9.1	Overview.....	126
9.2	Register Descriptions	129
9.2.1	SCKDIVCR : System Clock Division Control Register.....	129
9.2.2	SCKSCR : System Clock Source Control Register	130
9.2.3	MOSCCR : Main Clock Oscillator Control Register	131
9.2.4	SOSCCR : Sub-Clock Oscillator Control Register	131
9.2.5	LOCOCR : Low-Speed On-Chip Oscillator Control Register	132
9.2.6	HOCOCR : High-Speed On-Chip Oscillator Control Register	133
9.2.7	HOCOMCR : High-Speed On-Chip Oscillator Mode Control Register	134
9.2.8	MOCOCR : Middle-Speed On-Chip Oscillator Control Register	135
9.2.9	FLLCR1 : FLL Control Register1	135
9.2.10	OSCSF : Oscillation Stabilization Flag Register	136
9.2.11	OSTDCR : Oscillation Stop Detection Control Register	137

9.2.12	OSTDSR : Oscillation Stop Detection Status Register	138
9.2.13	MOSCWTCR : Main Clock Oscillator Wait Control Register	139
9.2.14	MOMCR : Main Clock Oscillator Mode Oscillation Control Register	139
9.2.15	SOMCR : Sub-Clock Oscillator Mode Control Register	140
9.2.16	CKOCR : Clock Out Control Register	141
9.2.17	CKO32CR : Clock Output 32-kHz Control Register	142
9.3	Main Clock Oscillator	142
9.3.1	Connecting a Crystal Resonator	143
9.3.2	External Clock Input	143
9.3.3	Notes on External Clock Input	144
9.4	Sub-Clock Oscillator	144
9.4.1	Connecting a 32.768-kHz Crystal Resonator	144
9.4.2	Pin Handling When the Sub-Clock Oscillator Is Not Used	145
9.5	Oscillation Stop Detection Function	145
9.5.1	Oscillation Stop Detection and Operation after Detection	145
9.5.2	Oscillation Stop Detection Interrupts	146
9.6	Internal Clock	146
9.6.1	System Clock (ICLK)/Peripheral Module Clock (PCLKA)	147
9.6.2	Peripheral Module Clock (PCLKB)	148
9.6.3	CAC Clock (CACCLK)	148
9.6.4	SOSC Clock (SOSC), LOCO Clock (LOCOCLK)	148
9.6.5	IWDT-Dedicated Clock (IWDTCLK)	148
9.6.6	WDT Clock (WDTCLK)	149
9.6.7	AGT-Dedicated Clock (AGTSCLK, AGTLCLK)	149
9.6.8	SysTick Timer-Dedicated Clock (SYSTICCLK)	149
9.6.9	External Pin Output Clock (CLKOUT)	149
9.6.10	External Pin Output Sub-Clock (CLKOUT32K)	149
9.6.11	CCC-Dedicated Sub-Clock (CCC32K)	149
9.6.12	JTAG Clock (JTAGTCK)	149
9.6.13	Serial Wire Debug Clock (SWCLK)	149
9.7	Usage Notes	149
9.7.1	Notes on register access	149
9.7.2	Notes on Clock Generation Circuit	149
9.7.3	Notes on Resonator and Oscillator	150
9.7.4	Notes on Board Design	150
9.7.5	Notes on Resonator Connect Pin	150
9.7.6	Notes on using the debugger	150
10.	Clock Correction Circuit (CCC)	152
10.1	Overview	152
10.2	Register Descriptions	153

10.2.1	ADJUSTR : Adjustment Register	153
10.2.2	R128CNT : 128-Hz Counter	153
10.2.3	R128CTRL : 128-Hz Counter Control Register	154
10.3	Operation	157
10.3.1	Correction of the Sub-Clock Signal	157
10.3.2	Interrupt source	157
10.3.3	Procedure for Reading from 128-Hz Counter	159
10.3.4	Event Link Output	159
10.4	Usage Notes	160
10.4.1	Precautions for Using Periodic Interrupts	160
10.4.2	Notes on CCCOUT Output	160
10.4.3	Notes on Transition to Low Power Consumption Modes after Setting Registers	161
10.4.4	Notes on Reading and Writing Registers	161
10.4.5	Module-Stop Function Settings	161
11.	Clock Frequency Accuracy Measurement Circuit (CAC)	162
11.1	Overview	162
11.2	Register Descriptions	163
11.2.1	CACR0 : CAC Control Register 0	163
11.2.2	CACR1 : CAC Control Register 1	164
11.2.3	CACR2 : CAC Control Register 2	164
11.2.4	CAICR : CAC Interrupt Control Register	165
11.2.5	CASTR : CAC Status Register	166
11.2.6	CAULVR : CAC Upper-Limit Value Setting Register	167
11.2.7	CALLVR : CAC Lower-Limit Value Setting Register	168
11.2.8	CACNTBR : CAC Counter Buffer Register	168
11.3	Operation	168
11.3.1	Measuring Clock Frequency	168
11.3.2	Digital Filtering of Signals on CACREF Pin	170
11.4	Interrupt Requests	170
11.5	Usage Notes	170
11.5.1	Settings for the Module-Stop Function	170
12.	Key Interrupt Function (KINT)	171
12.1	Overview	171
12.2	Register Descriptions	171
12.2.1	KRCTL : Key Return Control Register	171
12.2.2	KRF : Key Return Flag Register	172
12.2.3	KRM : Key Return Mode Register	172
12.3	Operation	172
12.3.1	Operation When Not Using the Key Interrupt Flags (KRCTL.KRMD = 0)	172

12.3.2	Operation When Using the Key Interrupt Flags (KRCTL.KRMD = 1)	173
12.4	Usage Notes	175
13.	Power-Saving Functions	176
13.1	Overview	176
13.2	Register Descriptions	190
13.2.1	MSTPCRA : Module Stop Control Register A	190
13.2.2	MSTPCRB : Module Stop Control Register B	191
13.2.3	MSTPCRC : Module Stop Control Register C	192
13.2.4	MSTPCRD : Module Stop Control Register D	193
13.2.5	FSTPCR : Function Stop Control Register	194
13.2.6	PWSTCR : Power Supply State Control Register	195
13.2.7	PWSTF : Power Supply State Flag Register	197
13.2.8	OPCCR : Operating Power Control Register	197
13.2.9	SBYCR : Standby Control Register	198
13.2.10	RAMSDCR : RAM Cutoff Control Register	199
13.2.11	SNZCR : Snooze Control Register	200
13.2.12	SNZEDCR0 : Snooze End Control Register 0	201
13.2.13	SNZREQCR0 : Snooze Request Control Register 0	202
13.2.14	DPSBYCR : Deep Software Standby Control Register	204
13.2.15	DPSIER0 : Deep Software Standby Interrupt Enable Register 0	205
13.2.16	DPSIER1 : Deep Software Standby Interrupt Enable Register 1	205
13.2.17	DPSIFR0 : Deep Software Standby Interrupt Flag Register 0	206
13.2.18	DPSIFR1 : Deep Software Standby Interrupt Flag Register 1	207
13.2.19	DPSIEGR0 : Deep Software Standby Interrupt Edge Register 0	208
13.2.20	DPSIEGR1 : Deep Software Standby Interrupt Edge Register 1	209
13.2.21	SYOCDRC : System Control OCD Control Register	209
13.2.22	VOCR : Power Supply Open Control Register	210
13.2.23	LDOCR : Regulator Control Register	213
13.2.24	VBBCR : Back Bias Voltage Control Register	214
13.2.25	VBBST : Back Bias Voltage Status Register	215
13.3	Reducing Power Consumption by Switching Clock Signals	216
13.4	Module-Stop Function	216
13.5	Functions for Reducing Power	216
13.5.1	Setting Power Supply Mode (ALLPWON/EXFPWON/MINPWON)	216
13.5.2	Setting Power Control Mode (BOOST/NORMAL/VBB)	223
13.6	Low Power Consumption Mode	236
13.6.1	Sleep Mode	236
13.6.2	Software Standby Mode	237
13.6.3	Snooze Mode	239
13.6.4	Deep Software Standby Mode	244

13.7	Usage Notes	247
13.7.1	Register Access	247
13.7.2	I/O Port pin states	249
13.7.3	Module-Stop State of DMAC, DTC	249
13.7.4	Internal Interrupt Sources	249
13.7.5	Transition to Low Power Modes.....	249
13.7.6	Input Buffer Control by DIRQnE Bit (n = 0 to 3).....	249
13.7.7	Timing of WFI Instruction	249
13.7.8	Writing WDT/IWDT Registers by DMAC or DTC in Sleep Mode or Snooze Mode.....	249
13.7.9	Oscillators in Snooze Mode	249
13.7.10	Snooze Mode Entry by RXD0 Falling Edge	250
13.7.11	Using SCIO in Snooze Mode.....	250
13.7.12	Conditions of A/D Conversion Start in Snooze Mode	250
13.7.13	ELC Event in Snooze Mode.....	250
13.7.14	Notes on Switching Power Supply, Power Control, and Low Power Consumption Modes	250
13.7.15	Notes on mode transition	250
14.	Energy Harvesting Controller (EHC).....	258
14.1	Overview.....	258
14.2	Register Descriptions	259
14.2.1	EHCCR0 : EHC Control Register 0	259
14.2.2	EHCCR1 : EHC Control Register 1	261
14.2.3	EHCRMRR : EHC resistance Monitoring Register	262
14.3	Operation.....	263
14.3.1	Basic Operation	263
14.3.2	Initial Settings	265
14.3.3	Checking the Voltage on the VBAT_EHC Pin	265
14.3.4	Processing in Response to Excessive Dropping of the Secondary Battery Voltage.....	267
14.3.5	Secondary Battery Overcharging Protection	267
14.3.6	Reverse Current Flow Prevention.....	268
14.3.7	Level of Power Generation Detection	268
14.4	Interrupt Sources	269
14.4.1	Secondary Battery Charging Detection Interrupt.....	269
14.4.2	Storage Capacitor Charging Detection	269
14.5	Usage Notes	269
14.5.1	Register Write Protection	269
14.5.2	Handling of the VCC, IOVCC0/1, and AVCC0 Pins when using the EHC	269
14.5.3	Handling of Pins when the EHC is not Used	270
14.5.4	Selecting a Power Generating Element.....	271
14.5.5	Selecting a Secondary Battery	271
14.5.6	Selecting a Secondary Battery when the MLCD is Used.....	271

14.5.7	Setting the Frequency for the Secondary Battery Charging Period	272
14.5.8	Reset range	272
15.	Register Write Protection	273
15.1	Overview.....	273
15.2	Register Descriptions	273
15.2.1	PRCR : Protect Register	273
16.	Interrupt Controller Unit (ICU)	275
16.1	Overview.....	275
16.2	Register Descriptions	276
16.2.1	IRQCRi : IRQ Control Register (i = 0 to 9)	276
16.2.2	NMISR : Non-Maskable Interrupt Status Register	277
16.2.3	NMIER : Non-Maskable Interrupt Enable Register	279
16.2.4	NMICLR : Non-Maskable Interrupt Status Clear Register	281
16.2.5	NMICR : NMI Pin Interrupt Control Register	282
16.2.6	IELSRn : ICU Event Link Setting Register n (n = 0 to 31)	282
16.2.7	DELSRn : DMAC Event Link Setting Register n (n = 0 to 3)	284
16.2.8	SELSR0 : SYS Event Link Setting Register	284
16.2.9	WUPEN : Wake Up Interrupt Enable Register	285
16.3	Vector Table	287
16.3.1	Interrupt Vector Table	287
16.3.2	Event Table	289
16.3.3	ICU and DTC Event Number	294
16.4	Interrupt Operation	301
16.4.1	Interrupt detection selection.....	301
16.4.2	Detecting Interrupts	301
16.5	Interrupt setting procedure	301
16.5.1	Enabling Interrupt Requests	301
16.5.2	Disabling Interrupt Requests	302
16.5.3	Polling for interrupts.....	302
16.5.4	Selecting Interrupt Request Destinations	302
16.5.5	External Pin Interrupts	303
16.5.6	Non-Maskable Interrupt Operation	303
16.6	Return from Low Power Modes	304
16.6.1	Return from Sleep Mode.....	304
16.6.2	Return from Software Standby Mode	304
16.6.3	Return from Snooze Mode.....	304
16.7	Usage Notes	305
16.7.1	When using WFI instruction with non-maskable interrupt.....	305
16.7.2	Notes on PCLKB.....	305

16.7.3	Notes on DMAC/DTC activation source setting	305
16.7.4	Interrupt at Module-stop	305
16.8	Reference	305
17.	Buses	306
17.1	Overview	306
17.2	Description of Buses	307
17.2.1	Bus Master	307
17.2.2	Bus Slave	307
17.2.3	Parallel Operations	307
17.3	Register Descriptions	308
17.3.1	BUSMCNTx : Bus Master Control Register x (x = SYS, DMA)	308
17.3.2	BUSnERRADD : Bus Error Address Register n (n = 3, 4)	309
17.3.3	BUSnERRSTAT : BUS Error Status Register n (n = 3, 4)	309
17.4	Bus Error Monitoring Section	310
17.4.1	Error Type that Occurs by Bus	310
17.4.2	Operation when a Bus Error Occurs	310
17.4.3	Conditions for issuing illegal Address Access Errors	310
17.5	References	311
18.	Memory Protection Unit (MPU)	312
18.1	Overview	312
18.2	CPU Stack Pointer Monitor	312
18.2.1	Protecting the Registers	315
18.2.2	Overflow and Underflow Errors	315
18.2.3	Register Descriptions	316
18.3	Arm MPU	320
18.3.1	Priorities to the Protection Region	320
18.4	Bus Master MPU	320
18.4.1	Register Descriptions	321
18.4.2	Operation	324
18.5	Bus Slave MPU	327
18.5.1	Register Descriptions	328
18.5.2	Functions	334
18.6	Security MPU	334
18.6.1	Register Descriptions (Option-Setting Memory)	335
18.6.2	Memory Protection	338
18.7	Usage Notes	340
18.7.1	Notes on the Use of a Debugger	340
18.7.2	Notes on Setting the Registers	340
18.8	References	341

19. DMA Controller (DMAC)	342
19.1 Overview	342
19.2 Register Descriptions	343
19.2.1 DMSAR : DMA Source Address Register	343
19.2.2 DMDAR : DMA Destination Address Register	344
19.2.3 DMCRA : DMA Transfer Count Register	344
19.2.4 DMCRB : DMA Block Transfer Count Register	345
19.2.5 DMTMD : DMA Transfer Mode Register	345
19.2.6 DMINT : DMA Interrupt Setting Register	346
19.2.7 DMAMD : DMA Address Mode Register	348
19.2.8 DMOFR : DMA Offset Register	350
19.2.9 DMCNT : DMA Transfer Enable Register	350
19.2.10 DMREQ : DMA Software Start Register	351
19.2.11 DMSTS : DMA Status Register	352
19.2.12 DMAST : DMA Module Activation Register	353
19.3 Operation	353
19.3.1 Transfer Mode	353
19.3.2 Extended Repeat Area Function	356
19.3.3 Address Update Function using Offset	358
19.3.4 Activation Sources	362
19.3.5 Operation Timing	362
19.3.6 DMAC Execution Cycles	363
19.3.7 Activating the DMAC	364
19.3.8 Starting DMA Transfer	365
19.3.9 Registers during DMA Transfer	365
19.3.10 Channel Priority	366
19.4 Ending DMA Transfer	366
19.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations	366
19.4.2 Transfer End by Repeat Size End Interrupt	367
19.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow	367
19.5 Interrupts	367
19.5.1 Transfer End Interrupt	367
19.6 Event Link	369
19.7 Low-Power Consumption Function	369
19.8 Usage Notes	370
19.8.1 Access to the Registers during DMA Transfer	370
19.8.2 DMA Transfer to Reserved Areas	370
19.8.3 Setting of DMAC Event Link Setting Register of the Interrupt Controller Unit (ICU.DELSRn n = 0 to 3)	370
19.8.4 Suspending or Restarting DMAC Activation	370

19.8.5	Precautions for Resuming DMA Transfer	370
20.	Data Transfer Controller (DTC).....	372
20.1	Overview.....	372
20.2	Register Descriptions	373
20.2.1	MRA : DTC Mode Register A.....	374
20.2.2	MRB : DTC Mode Register B.....	374
20.2.3	SAR : DTC Transfer Source Register	376
20.2.4	DAR : DTC Transfer Destination Register	376
20.2.5	CRA : DTC Transfer Count Register A	376
20.2.6	CRB : DTC Transfer Count Register B	377
20.2.7	DTCCR : DTC Control Register.....	377
20.2.8	DTCVBR : DTC Vector Base Register.....	378
20.2.9	DTCST : DTC Module Start Register.....	378
20.2.10	DTCSTS : DTC Status Register	379
20.3	Activation Sources.....	379
20.3.1	Allocating Transfer Information and DTC Vector Table.....	380
20.4	Operation.....	381
20.4.1	Transfer Information Read Skip Function	383
20.4.2	Transfer Information Write-Back Skip Function	383
20.4.3	Normal Transfer Mode	384
20.4.4	Repeat Transfer Mode	385
20.4.5	Block Transfer Mode.....	386
20.4.6	Chain Transfer	387
20.4.7	Operation Timing	388
20.4.8	Execution Cycles of DTC.....	390
20.4.9	DTC Bus Mastership Release Timing.....	391
20.5	DTC Setting Procedure	391
20.6	Examples of DTC Usage	392
20.6.1	Normal Transfer	392
20.6.2	Chain transfer	392
20.6.3	Chain Transfer when Counter = 0.....	394
20.7	Interrupt	396
20.7.1	Interrupt Sources	396
20.8	Event Link.....	396
20.9	Low Power Consumption Function.....	396
20.10	Usage Notes.....	397
20.10.1	Transfer Information Start Address.....	397
21.	Event Link Controller (ELC).....	398
21.1	Overview.....	398

21.2	Register Descriptions	399
21.2.1	ELCR : Event Link Control Registers.....	399
21.2.2	ELSRn : Event Link Setting Register n (n = 0 to 3, 6 to 8, 14, 15, 18, 19).....	399
21.2.3	ELSEGRn : Event Link Software Event Generation Register n (n = 0, 1)	402
21.2.4	ELOPA : Event Link Option Setting Register A.....	402
21.3	Operation.....	403
21.3.1	Relation between Interrupt Handling and Event Linking.....	403
21.3.2	Linking Events	403
21.3.3	Example of Operation Setting Procedure for Linking Events.....	404
21.4	Usage Notes.....	404
21.4.1	Linking DMAC or DTC Transfer End Signals as Events.....	404
21.4.2	Setting Clocks.....	404
21.4.3	Notes on event settings	404
21.4.4	Settings for the Module-Stop Function	405
22.	I/O Ports.....	406
22.1	Overview.....	406
22.2	Register Descriptions	408
22.2.1	PCNTR1/PODR/PDR : Port Control Register 1	408
22.2.2	PCNTR2/EIDR/PIDR : Port Control Register 2.....	409
22.2.3	PCNTR3/PORR/POSR : Port Control Register 3	410
22.2.4	PCNTR4/EORR/EOSR : Port Control Register 4	411
22.2.5	PWPR : Write-Protect Register.....	411
22.2.6	PmnPFS : Port mn Pin Function Select Register (m = 0 to 8, n = 00 to 15).....	412
22.3	Operation.....	413
22.3.1	General I/O Ports.....	413
22.3.2	Port Function Select	413
22.3.3	Port Group Function for ELC	414
22.3.4	Peripheral Select Settings for Each Product.....	416
22.4	Handling of Unused Pins	423
22.5	Usage Notes.....	424
22.5.1	Port Output Data Register (PODR) Summary	424
22.5.2	Procedure for Specifying Input/Output Pin Function.....	424
22.5.3	Procedure for Using Port Group Input	424
22.5.4	Notes on Using of Analog Functions	425
23.	Port Output Enable for GPT (POE).....	426
23.1	Overview.....	426
23.2	Register Descriptions	427
23.2.1	POEGGn : POE Group n Setting Register (n = A, B)	427
23.3	Output-Disable Control Operation	428

23.3.1	Pin Input Level Detection Operation	429
23.3.2	Output-Disable Requests from the GPT	429
23.3.3	Output-Disable Control Using Detection of Stopped Oscillation	429
23.3.4	Output-Disable Control Using Registers	429
23.3.5	Release from Output-Disable	430
23.4	Interrupt Sources	430
23.5	External Trigger Output to the GPT	431
23.6	Usage Notes	431
23.6.1	Transition to Software Standby Mode	431
23.6.2	Specifying Pins Associated with the GPT	431
23.6.3	Settings for the Module-Stop Function	432
24.	General PWM Timer (GPT)	433
24.1	Overview	433
24.2	Register Descriptions	436
24.2.1	GTWP : General PWM Timer Write-Protection Register	436
24.2.2	GTSTR : General PWM Timer Software Start Register	437
24.2.3	GTSTP : General PWM Timer Software Stop Register	437
24.2.4	GTCLR : General PWM Timer Software Clear Register	438
24.2.5	GTSSR : General PWM Timer Start Source Select Register	438
24.2.6	GTSPSR : General PWM Timer Stop Source Select Register	441
24.2.7	GTCSR : General PWM Timer Clear Source Select Register	443
24.2.8	GTUPSR : General PWM Timer Up Count Source Select Register	446
24.2.9	GTDNSR : General PWM Timer Down Count Source Select Register	448
24.2.10	GTICASR : General PWM Timer Input Capture Source Select Register A	451
24.2.11	GTICBSR : General PWM Timer Input Capture Source Select Register B	453
24.2.12	GTCR : General PWM Timer Control Register	456
24.2.13	GTUDDTYC : General PWM Timer Count Direction and Duty Setting Register	458
24.2.14	GTIOR : General PWM Timer I/O Control Register	460
24.2.15	GTINTAD : General PWM Timer Interrupt Output Setting Register	464
24.2.16	GTST : General PWM Timer Status Register	465
24.2.17	GTBER : General PWM Timer Buffer Enable Register	469
24.2.18	GTCNT : General PWM Timer Counter	470
24.2.19	GTCCRn : General PWM Timer Compare Capture Register n (n = A to F)	470
24.2.20	GTPR : General PWM Timer Cycle Setting Register	471
24.2.21	GTPBR : General PWM Timer Cycle Setting Buffer Register	471
24.2.22	GTDTCR : General PWM Timer Dead Time Control Register	471
24.2.23	GTDVU : General PWM Timer Dead Time Value Register U	472
24.2.24	OPSCR : Output Phase Switching Control Register	473
24.3	Operation	475
24.3.1	Basic Operation	475

24.3.2	Buffer Operation.....	483
24.3.3	PWM Output Operating Mode	490
24.3.4	Automatic Dead Time Setting Function	499
24.3.5	Count Direction Changing Function	503
24.3.6	Function of Output Duty 0% and 100%	503
24.3.7	Hardware Count Start/Count Stop and Clear Operation.....	505
24.3.8	Synchronized Operation	510
24.3.9	PWM Output Operation Examples.....	514
24.3.10	Phase Counting Function	519
24.3.11	Output Phase Switching (GPT_OPS).....	529
24.4	Interrupt Sources	536
24.4.1	Interrupt Sources	536
24.4.2	DMAC and DTC Activation	538
24.5	Operations Linked by ELC.....	538
24.5.1	Event Signal Output to ELC.....	538
24.5.2	Event Signal Inputs from ELC.....	538
24.6	Noise Filter Function.....	539
24.7	Protection Function.....	539
24.7.1	Write-Protection for Registers.....	539
24.7.2	Disabling of Buffer Operation.....	539
24.7.3	GTIOCnm Pin Output Negate Control (n = 0 to 5, m = A, B).....	540
24.8	Initialization Method of Output Pins.....	541
24.8.1	Pin Settings after Reset	541
24.8.2	Pin Initialization Due to Error during Operation	542
24.9	Usage Notes.....	542
24.9.1	Module-Stop Function Setting	542
24.9.2	GTCCRn Settings during Compare Match Operation (n = A to F).....	542
24.9.3	Setting Range for GTCNT Counter.....	543
24.9.4	Starting and Stopping the GTCNT Counter	543
24.9.5	Priority Order of Each Event	544
25.	Low Power Asynchronous General Purpose Timer (AGT, AGTW).....	545
25.1	Overview.....	545
25.1.1	Difference between AGT and AGTW.....	547
25.2	Register Descriptions	547
25.2.1	AGT : AGT Counter Register.....	547
25.2.2	AGTCMA : AGT Compare Match A Register.....	548
25.2.3	AGTCMB : AGT Compare Match B Register.....	548
25.2.4	AGTCR : AGT Control Register.....	549
25.2.5	AGTMR1 : AGT Mode Register 1	550
25.2.6	AGTMR2 : AGT Mode Register 2	551

25.2.7	AGTIOC : AGT I/O Control Register	553
25.2.8	AGTISR : AGT Event Pin Select Register	554
25.2.9	AGTCMSR : AGT Compare Match Function Select Register	554
25.2.10	AGTIOSEL : AGT Pin Select Register	555
25.3	Operation	556
25.3.1	Reload Register and Counter Rewrite Operation	556
25.3.2	Reload Register and AGT Compare Match A/B Register Rewrite Operation	557
25.3.3	Timer Mode	558
25.3.4	Pulse Output Mode	559
25.3.5	Event Counter Mode	560
25.3.6	Pulse Width Measurement Mode	561
25.3.7	Pulse Period Measurement Mode	562
25.3.8	Compare Match function	563
25.3.9	Output Settings for Each Mode	564
25.3.10	Standby Mode	565
25.3.11	Interrupt Sources	566
25.3.12	Event Signal Output to ELC	567
25.4	Usage Notes	567
25.4.1	Count Operation Start and Stop Control	567
25.4.2	Access to Counter Register	568
25.4.3	When Changing Mode	568
25.4.4	Output pin setting	568
25.4.5	Digital Filter	568
25.4.6	How to Calculate Event Number, Pulse Width, and Pulse Period	568
25.4.7	When Count is Forcibly Stopped by TSTOP Bit	568
25.4.8	When Selecting AGT0 Underflow as the Count Source	568
25.4.9	Module-stop function	569
26.	8-Bit Timers (TMR)	570
26.1	Overview	570
26.2	Register Descriptions	573
26.2.1	TCNT : Timer Counter	573
26.2.2	TCORA : Time Constant Register A	573
26.2.3	TCORB : Time Constant Register B	574
26.2.4	TCR : Timer Control Register	574
26.2.5	TCCR : Timer Counter Control Register	575
26.2.6	TCSR : Timer Control/Status Register	576
26.3	Operation	577
26.3.1	Pulse Output	577
26.3.2	Delayed Pulse Output Caused by External Counter Reset Input	578
26.4	Operation Timing	579

26.4.1	TCNT Count Timing	579
26.4.2	Timing of Interrupt on Compare Match	579
26.4.3	Timing of Signal Output on Compare Match	580
26.4.4	Timing of Counter Clearing by Compare Match	580
26.4.5	Timing of TCNT Counter Clearing by External Counter Reset	581
26.4.6	Timing of Interrupt on Overflow	582
26.5	Operation with Cascaded Connection	582
26.5.1	16-Bit Count Mode	582
26.5.2	Compare Match Count Mode	582
26.6	Interrupt Sources	582
26.6.1	Interrupt Sources and DMAC/DTC Activation	582
26.6.2	A/D Converter Activation	583
26.7	Link Operation by the ELC	583
26.7.1	Event Signal Output to the ELC	583
26.8	Usage Notes	583
26.8.1	Settings for the Module-Stop Function	583
26.8.2	Notes on Cycle Setting	583
26.8.3	Conflict between Writing to and Clearing of the TCNT Counter	584
26.8.4	Conflict between Writing to and Incrementing of the TCNT Counter	584
26.8.5	Conflict between Writing to the TCORA or TCORB Register and Compare Match	585
26.8.6	Conflict between Compare Matches A and B	585
26.8.7	Switching of Internal Clocks and TCNT Operation	585
26.8.8	Clock Source Setting with Cascaded Connection	587
26.8.9	Continuous Output of Compare Match Interrupt Signal	587
27.	Wake Up Timer (WUPT)	589
27.1	Overview	589
27.2	Register Descriptions	590
27.2.1	TCMn : Timer Compare Match Register (n = 0 to 3)	590
27.2.2	TCR : Timer Control Register	590
27.3	Operation	591
27.3.1	Pulse Output	591
27.3.2	Operation Timing	593
27.4	Interrupt Sources	594
27.4.1	Interrupt Sources	594
27.5	Link Operation by the ELC	594
27.6	Usage Notes	594
27.6.1	Module-Stop Function Settings	594
27.6.2	Note on Cycle Setting	594
28.	Realtime Clock (RTC)	595

28.1	Overview.....	595
28.2	Register Descriptions	596
28.2.1	R64CNT : 64-Hz Counter	597
28.2.2	RSECCNT : Second Counter (in Calendar Count Mode).....	597
28.2.3	RMINCNT : Minute Counter (in Calendar Count Mode)	598
28.2.4	RHRCNT : Hour Counter (in Calendar Count Mode).....	598
28.2.5	RWKCNT : Day-of-Week Counter (in Calendar Count Mode)	599
28.2.6	BCNTn : Binary Counter n (n = 0 to 3) (in Binary Count Mode and 32-kHz Count Mode).....	599
28.2.7	RDAYCNT : Day Counter.....	600
28.2.8	RMONCNT : Month Counter.....	600
28.2.9	RYRCNT : Year Counter	601
28.2.10	RSECAR : Second Alarm Register (in Calendar Count Mode)	601
28.2.11	RMINAR : Minute Alarm Register (in Calendar Count Mode).....	602
28.2.12	RHRAR : Hour Alarm Register (in Calendar Count Mode)	602
28.2.13	RWKAR : Day-of-Week Alarm Register (in Calendar Count Mode)	603
28.2.14	BCNTnAR : Binary Counter n Alarm Register (n = 0 to 3) (in Binary Count Mode and 32-kHz Count Mode).....	603
28.2.15	RDAYAR : Date Alarm Register (in Calendar Count Mode).....	604
28.2.16	RMONAR : Month Alarm Register (in Calendar Count Mode).....	604
28.2.17	RYRAR : Year Alarm Register (in Calendar Count Mode).....	605
28.2.18	RYRAREN : Year Alarm Enable Register (in Calendar Count Mode).....	605
28.2.19	BCNTnAER : Binary Counter n Alarm Enable Register (n = 0, 1) (in Binary Count Mode and 32-kHz Count Mode).....	605
28.2.20	BCNT2AER : Binary Counter 2 Alarm Enable Register (in Binary Count Mode and 32-kHz Count Mode)	606
28.2.21	BCNT3AER : Binary Counter 3 Alarm Enable Register (in Binary Count Mode and 32-kHz Count Mode)	606
28.2.22	RCR1 : RTC Control Register 1	607
28.2.23	RCR2 : RTC Control Register 2 (in Calendar Count Mode)	608
28.2.24	RCR2 : RTC Control Register 2 (in Binary Count Mode and 32-kHz Count Mode).....	610
28.2.25	RCR4 : RTC Control Register 4.....	611
28.2.26	RADJ : Time Error Adjustment Register	612
28.2.27	RCPE : RTC Time Capture Enable Register	612
28.2.28	RTCCRn : Time Capture Control Register n (n = 0 to 2)	613
28.2.29	RSECCPn : Second Capture Register n (n = 0 to 2) (in Calendar Count Mode).....	614
28.2.30	RMINCPn : Minute Capture Register n (n = 0 to 2) (in Calendar Count Mode).....	615
28.2.31	RHRCPn : Hour Capture Register n (n = 0 to 2) (in Calendar Count Mode)	615
28.2.32	RDAYCPn : Date Capture Register n (n = 0 to 2) (in Calendar Count Mode)	616
28.2.33	RMONCPn : Month Capture Register n (n = 0 to 2) (in Calendar Count Mode).....	616
28.2.34	BCNTnCPm : BCNTn Capture Register m (n= 0 to 3, m = 0 to 2) (in Binary Count Mode) ...	617
28.3	Operation.....	617

28.3.1	Outline of Initial Settings of Registers after Power On.....	617
28.3.2	Clock and Count Mode Setting Procedure	617
28.3.3	Setting the Time	618
28.3.4	30-Second Adjustment	619
28.3.5	Reading 64-Hz Counter and Time	620
28.3.6	Alarm Function.....	622
28.3.7	Procedure for Disabling Alarm Interrupt	623
28.3.8	Time Error Adjustment Function	623
28.3.9	Capturing the time	625
28.4	Interrupt Sources	626
28.5	Event Link Output	627
28.5.1	Interrupt Handling and Event Linking.....	628
28.6	Usage Notes.....	628
28.6.1	Register Writing during Counting	628
28.6.2	Use of Periodic Interrupts	628
28.6.3	RTCOUNT (1-Hz/64-Hz) Clock Output.....	629
28.6.4	Transitions to Low Power Modes after Setting Registers	629
28.6.5	Notes on Writing to and Reading from Registers.....	629
28.6.6	Changing the Count Mode.....	629
28.6.7	Time Capture Operation While Count is Running or Stopping	629
28.6.8	Initialization	630
29.	Low-Speed Clock Timer (LST).....	631
29.1	Overview.....	631
29.2	Register Descriptions	632
29.2.1	LSTCNT : Low-Speed Clock Timer Count Register.....	632
29.2.2	LSTCTRL : Low-Speed Clock Timer Counter Control Register.....	632
29.3	Operation.....	633
29.3.1	Basic Operation	633
29.4	Usage Notes.....	633
29.4.1	Settings for the Module-Stop Function	633
29.4.2	Limits in the Bus Clock Specifications	633
29.4.3	Settings for the Clock Source	633
30.	Watchdog Timer (WDT)	634
30.1	Overview.....	634
30.2	Register Descriptions	635
30.2.1	WDTRR : WDT Refresh Register	635
30.2.2	WDTCR : WDT Control Register	636
30.2.3	WDTSR : WDT Status Register	638
30.2.4	WDTRCR : WDT Reset Control Register	639

30.2.5	WDTCSSTPR : WDT Count Stop Control Register	640
30.2.6	Option Function Select Register 0 (OFS0)	640
30.3	Operation	641
30.3.1	Count Operation in each Start Mode	641
30.3.2	Controlling Writes to the WDTCSR, WDTRCR, and WDTCSSTPR Registers.....	643
30.3.3	Refresh Operation	644
30.3.4	Status Flags	645
30.3.5	Reset Output.....	645
30.3.6	Interrupt Sources	645
30.3.7	Reading the Down-Counter Value	646
30.3.8	Association between Option Function Select Register 0 (OFS0) and WDT Registers.....	646
30.4	Output to the Event Link Controller (ELC)	647
30.5	Usage Notes.....	647
30.5.1	ICU Event Link Setting Register n (IELSRn) Setting	647
30.5.2	Transition to MINPWON	647
31.	Independent Watchdog Timer (IWDTC).....	648
31.1	Overview.....	648
31.2	Register Descriptions	649
31.2.1	IWDTCRR : IWDTC Refresh Register	649
31.2.2	IWDTCR : IWDTC Status Register.....	650
31.3	Operation	651
31.3.1	Auto Start Mode	651
31.3.2	Refresh Operation	652
31.3.3	Status Flags	653
31.3.4	Reset Output.....	654
31.3.5	Interrupt Sources	654
31.3.6	Reading the Down-Counter Value	654
31.4	Output to the Event Link Controller (ELC)	654
31.5	Usage Notes.....	655
31.5.1	Refresh Operations.....	655
31.5.2	Clock Division Ratio Setting.....	655
31.5.3	Constraints on the ICU Event Link Setting Register n (IELSRn) Setting	655
32.	Serial Communications Interface (SCI)	656
32.1	Overview.....	656
32.2	Register Descriptions	658
32.2.1	RSR : Receive Shift Register.....	658
32.2.2	RDR : Receive Data Register	659
32.2.3	RDRHL : Receive Data Register	659
32.2.4	FRDRHL/FRDRH/FRDRL : Receive FIFO Data Register	659

32.2.5	TDR : Transmit Data Register.....	661
32.2.6	TDRHL : Transmit Data Register	661
32.2.7	FTDRHL/FTDRH/FTDRL : Transmit FIFO Data Register.....	662
32.2.8	TSR : Transmit Shift Register	662
32.2.9	SMR : Serial Mode Register for Non-Smart Card Interface Mode (SCMR.SMIF = 0)	663
32.2.10	SMR_SMCI : Serial Mode Register for Smart Card Interface Mode (SCMR.SMIF = 1)	664
32.2.11	SCR : Serial Control Register for Non-Smart Card Interface Mode (SCMR.SMIF = 0)	666
32.2.12	SCR_SMCI : Serial Control Register for Smart Card Interface Mode (SCMR.SMIF = 1)	667
32.2.13	SSR : Serial Status Register for Non-Smart Card Interface and Non-FIFO Mode (SCMR.SMIF = 0, FCR.FM = 0)	669
32.2.14	SSR_FIFO : Serial Status Register for Non-Smart Card Interface and FIFO Mode (SCMR.SMIF = 0, FCR.FM = 1)	671
32.2.15	SSR_SMCI : Serial Status Register for Smart Card Interface Mode (SCMR.SMIF = 1)	674
32.2.16	SCMR : Smart Card Mode Register	676
32.2.17	BRR : Bit Rate Register	678
32.2.18	MDDR : Modulation Duty Register.....	687
32.2.19	SEMR : Serial Extended Mode Register.....	690
32.2.20	SNFR : Noise Filter Setting Register	692
32.2.21	SIMR1 : IIC Mode Register 1	692
32.2.22	SIMR2 : IIC Mode Register 2	693
32.2.23	SIMR3 : IIC Mode Register 3	694
32.2.24	SISR : IIC Status Register	696
32.2.25	SPMR : SPI Mode Register	696
32.2.26	FCR : FIFO Control Register	698
32.2.27	FDR : FIFO Data Count Register.....	699
32.2.28	LSR : Line Status Register	700
32.2.29	CDR : Compare Match Data Register	700
32.2.30	DCCR : Data Compare Match Control Register	701
32.2.31	SPTR : Serial Port Register	702
32.3	Operation in Asynchronous Mode	703
32.3.1	Serial Data Transfer Format	704
32.3.2	Receive Data Sampling Timing and Reception Margin in Asynchronous Mode.....	705
32.3.3	Clock.....	705
32.3.4	Double-Speed Operation and Frequency of 6 Times the Bit Rate.....	706
32.3.5	CTS and RTS Functions	706
32.3.6	Address Match (Receive Data Match Detection) Function	707
32.3.7	SCI Initialization in Asynchronous Mode	709
32.3.8	Serial Data Transmission in Asynchronous Mode	711
32.3.9	Serial Data Reception in Asynchronous Mode	716
32.4	Multi-Processor Communication Function.....	729
32.4.1	Multi-Processor Serial Data Transmission.....	731

32.4.2	Multi-Processor Serial Data Reception	734
32.5	Operation in Clock Synchronous Mode	739
32.5.1	Clock	740
32.5.2	CTS and RTS Functions	740
32.5.3	SCI Initialization in Clock Synchronous Mode	741
32.5.4	Serial Data Transmission in Clock Synchronous Mode	742
32.5.5	Serial Data Reception in Clock Synchronous Mode	746
32.5.6	Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode	751
32.6	Operation in Smart Card Interface Mode	754
32.6.1	Example Connection	754
32.6.2	Data Format (Except in Block Transfer Mode)	754
32.6.3	Block Transfer Mode	756
32.6.4	Receive Data Sampling Timing and Reception Margin	756
32.6.5	SCI Initialization (Smart Card Interface Mode)	757
32.6.6	Serial Data Transmission (Except in Block Transfer Mode)	758
32.6.7	Serial Data Reception (Except in Block Transfer Mode)	760
32.6.8	Clock Output Control	762
32.7	Operation in Simple IIC Mode	763
32.7.1	Generation of Start, Restart, and Stop Conditions	764
32.7.2	Clock Synchronization	765
32.7.3	SSDAn Output Delay	766
32.7.4	SCI Initialization in Simple IIC Mode	766
32.7.5	Operation in Master Transmission in Simple IIC Mode	767
32.7.6	Master Reception in Simple IIC Mode	770
32.8	Operation in Simple SPI Mode	771
32.8.1	States of Pins in Master and Slave Modes	772
32.8.2	SS Function in Master Mode	773
32.8.3	SS Function in Slave Mode	773
32.8.4	Relationship between Clock and Transmit/Receive Data	773
32.8.5	SCI Initialization in Simple SPI Mode	774
32.8.6	Transmission and Reception of Serial Data in Simple SPI Mode	774
32.9	Bit Rate Modulation Function	774
32.10	Interrupt Sources	775
32.10.1	Buffer Operation for SCIn_TXI and SCIn_RXI Interrupts (non-FIFO selected)	775
32.10.2	Buffer Operation for SCIn_TXI and SCIn_RXI Interrupts (FIFO selected)	775
32.10.3	Interrupts in Asynchronous, Clock Synchronous, and Simple SPI Modes	775
32.10.4	Interrupts in Smart Card Interface Mode	777
32.10.5	Interrupts in Simple IIC Mode	778
32.11	Event Linking	778
32.12	Address Non-match Event Output (SCI0_DCUF)	779

32.13	Noise Cancellation Function.....	779
32.14	Usage Notes.....	780
32.14.1	Settings for the Module-Stop Function	780
32.14.2	SCI Operation during Low Power State.....	780
32.14.3	Break Detection and Processing	785
32.14.4	Mark State and Production of Breaks	786
32.14.5	Receive Error Flags and Transmit Operation in Clock Synchronous Mode and Simple SPI Mode.....	786
32.14.6	Restrictions on Clock Synchronous Transmission in Clock Synchronous Mode and Simple SPI Mode	786
32.14.7	Restrictions on Using DTC or DMAC.....	787
32.14.8	Notes on Starting Transfer.....	788
32.14.9	External Clock Input in Clock Synchronous Mode and Simple SPI Mode	788
32.14.10	Limitations on Simple SPI Mode	788
32.14.11	Notes on Transmitt Enable bit (SCR.TE).....	789
32.14.12	Note on Stopping Reception When Using the RTS Function in Asynchronous Mode.....	789
33.	IrDA Interface (IrDA).....	790
33.1	Overview.....	790
33.2	Register Descriptions	790
33.2.1	IRCR : IrDA Control Register.....	790
33.3	Operation.....	791
33.3.1	IrDA Interface Setup Procedure.....	791
33.3.2	Transmission.....	791
33.3.3	Reception.....	792
33.4	Usage Notes.....	792
33.4.1	Settings for the Module-Stop Function	792
33.4.2	Asynchronous Reference Clock for SCI1	792
34.	I2C Bus Interface (IIC)	793
34.1	Overview.....	793
34.2	Register Descriptions	795
34.2.1	ICCR1 : I ² C Bus Control Register 1.....	795
34.2.2	ICCR2 : I ² C Bus Control Register 2.....	797
34.2.3	ICMR1 : I ² C Bus Mode Register 1	800
34.2.4	ICMR2 : I ² C Bus Mode Register 2.....	801
34.2.5	ICMR3 : I ² C Bus Mode Register 3.....	802
34.2.6	ICFER : I ² C Bus Function Enable Register	804
34.2.7	ICSER : I ² C Bus Status Enable Register.....	805

34.2.8	ICIER : I ² C Bus Interrupt Enable Register	806
34.2.9	ICSR1 : I ² C Bus Status Register 1	808
34.2.10	ICSR2 : I ² C Bus Status Register 2	810
34.2.11	SARLy : Slave Address Register Ly (y = 0 to 2)	813
34.2.12	SARUy : Slave Address Register Uy (y = 0 to 2).....	814
34.2.13	ICBRL : I ² C Bus Bit Rate Low-Level Register	814
34.2.14	ICBRH : I ² C Bus Bit Rate High-Level Register	815
34.2.15	ICDRT : I ² C Bus Transmit Data Register.....	816
34.2.16	ICDRR : I ² C Bus Receive Data Register	817
34.2.17	ICDRS : I ² C Bus Shift Register.....	817
34.3	Operation.....	817
34.3.1	Communication Data Format.....	817
34.3.2	Initial Settings	818
34.3.3	Master Transmit Operation	819
34.3.4	Master Receive Operation	823
34.3.5	Slave Transmit Operation	828
34.3.6	Slave Receive Operation	830
34.4	SCL Synchronization Circuit.....	832
34.5	SDA Output Delay Function	833
34.6	Digital Noise Filter Circuits	834
34.7	Address Match Detection	835
34.7.1	Slave-Address Match Detection	835
34.7.2	Detection of General Call Address	838
34.7.3	Device-ID Address Detection.....	839
34.7.4	Host Address Detection	840
34.8	Automatic Low-Hold Function for SCL	841
34.8.1	Function to Prevent Wrong Transmission of Transmit Data	841
34.8.2	NACK Reception Transfer Suspension Function.....	842
34.8.3	Function to Prevent Failure to Receive Data	843
34.9	Arbitration-Lost Detection Functions	844
34.9.1	Master Arbitration-Lost Detection (MALE Bit).....	844
34.9.2	Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)	846
34.9.3	Slave Arbitration-Lost Detection (SALE Bit)	847
34.10	Start, Restart, and Stop Condition Issuing Function.....	848
34.10.1	Issuing a Start Condition.....	848
34.10.2	Issuing a Restart Condition.....	848
34.10.3	Issuing a Stop Condition	850
34.11	Bus Hanging	850
34.11.1	Timeout Function	850

34.11.2	Extra SCL Clock Cycle Output Function	851
34.11.3	IIC Reset and Internal Reset	852
34.12	SMBus Operation	852
34.12.1	SMBus Timeout Measurement	853
34.12.2	Packet Error Code (PEC)	854
34.12.3	SMBus Host Notification Protocol (Notify ARP Master Command)	854
34.13	Interrupt Sources	854
34.13.1	Buffer Operation for IICn_TXI and IICn_RXI Interrupts	855
34.14	State of Registers When Issuing Each Condition	855
34.15	Event Link Output	856
34.15.1	Interrupt Handling and Event Linking	856
34.16	Usage Notes	856
34.16.1	Settings for the Module-Stop Function	856
34.16.2	Notes on Starting Transfer	857
35.	Serial Peripheral Interface (SPI)	858
35.1	Overview	858
35.2	Register Descriptions	861
35.2.1	SPCR : SPI Control Register	861
35.2.2	SSLP : SPI Slave Select Polarity Register	862
35.2.3	SPPCR : SPI Pin Control Register	863
35.2.4	SPSR : SPI Status Register	864
35.2.5	SPDR/SPDR_HA/SPDR_BY : SPI Data Register	866
35.2.6	SPSCR : SPI Sequence Control Register	869
35.2.7	SPSSR : SPI Sequence Status Register	870
35.2.8	SPBR : SPI Bit Rate Register	871
35.2.9	SPDCR : SPI Data Control Register	872
35.2.10	SPCKD : SPI Clock Delay Register	873
35.2.11	SSLND : SPI Slave Select Negation Delay Register	874
35.2.12	SPND : SPI Next-Access Delay Register	875
35.2.13	SPCR2 : SPI Control Register 2	875
35.2.14	SPCMDm : SPI Command Register m (m = 0 to 7)	876
35.3	Operation	879
35.3.1	Overview of SPI Operation	879
35.3.2	Controlling the SPI Pins	880
35.3.3	SPI System Configuration Examples	881
35.3.4	Data Formats	886
35.3.5	Transfer Formats	895
35.3.6	Data Transfer Modes	897
35.3.7	Transmit Buffer Empty and Receive Buffer Full Interrupts	898
35.3.8	Error Detection	900

35.3.9	Initializing the SPI	905
35.3.10	SPI Operation	905
35.3.11	Clock Synchronous Operation	920
35.3.12	Loopback Mode	926
35.3.13	Self-Diagnosis of Parity Bit Function	927
35.3.14	Interrupt Sources	928
35.4	Event Link Controller Event Output	929
35.4.1	Receive Buffer Full Event Output.....	929
35.4.2	Transmit Buffer Empty Event Output	930
35.4.3	Mode-Fault, Underrun, Overrun, or Parity Error Event Output	930
35.4.4	SPI Idle Event Output	930
35.4.5	Transmission-Completed Event Output	930
35.5	Usage Notes.....	931
35.5.1	Settings for the Module-Stop State.....	931
35.5.2	Constraint on Low-Power Functions.....	931
35.5.3	Constraints on Starting Transfer	931
35.5.4	Constraints on Mode-Fault, Underrun, Overrun, or Parity Error Event Output	931
35.5.5	Constraints on the SPSR.SPRF and SPSR.SPTEF Flags	931
35.5.6	Constraints on Register Settings	931
35.5.7	Note on Resuming Communications in Slave Mode	931
36.	Quad Serial Peripheral Interface (QSPI)	932
36.1	Overview.....	932
36.2	Register Descriptions	933
36.2.1	SFMSMD : Transfer Mode Control Register	933
36.2.2	SFMSSC : Chip Selection Control Register.....	934
36.2.3	SFMSKC : Clock Control Register	935
36.2.4	SFMSST : Status Register.....	936
36.2.5	SFMCOM : Communication Port Register	937
36.2.6	SFMCMD : Communication Mode Control Register	938
36.2.7	SFMCST : Communication Status Register	938
36.2.8	SFMSIC : Instruction Code Register.....	939
36.2.9	SFMSAC : Address Mode Control Register.....	939
36.2.10	SFMSDC : Dummy Cycle Control Register	940
36.2.11	SFMSPC : SPI Protocol Control Register.....	941
36.2.12	SFMPMD : Port Control Register.....	941
36.2.13	SFMCNT1 : External QSPI Address Register	942
36.3	Memory Map.....	942
36.3.1	External Bus Space	942
36.3.2	Address Width of the SPI Space and SPI Bus	943
36.4	SPI Bus.....	944

36.4.1	SPI Protocol	944
36.4.2	SPI Mode	947
36.5	SPI Bus Timing Adjustment	948
36.5.1	SPI Bus Reference Cycles	948
36.5.2	QSPCLK Signal Duty Ratio	949
36.5.3	Minimum High-Level Width for the QSSL Signal	950
36.5.4	QSSL Signal Setup Time	950
36.5.5	QSSL Signal Hold Time	951
36.5.6	Hold Time of the Serial Data Output Enable	951
36.5.7	Setup Time for Serial Data Output	952
36.5.8	Hold Time for Serial Data Output	952
36.6	SPI Instruction Set Used for Serial Flash Memory Access	952
36.6.1	SPI Instructions That Are Automatically Generated	952
36.6.2	Standard Read Instruction	953
36.6.3	Fast Read Instruction	954
36.6.4	Fast Read Dual Output Instruction	955
36.6.5	Fast Read Dual I/O Instruction	956
36.6.6	Fast Read Quad Output Instruction	957
36.6.7	Fast Read Quad I/O Instruction	958
36.7	SPI Bus Cycle Arrangement	959
36.7.1	Serial Flash Memory Read Based on Individual Conversion	959
36.7.2	Serial Flash Memory Read Using the Prefetch Function	960
36.7.3	Halt of Prefetching	960
36.7.4	Direct Specification of Prefetch Destination	960
36.7.5	Prefetch State Polling	961
36.7.6	SPI Bus Cycle Extension Function	961
36.8	XIP Control	962
36.8.1	Setting XIP Mode	963
36.8.2	Releasing the XIP Mode	964
36.9	QIO2 and QIO3 Pin States	965
36.10	Direct Communication Mode	966
36.10.1	About Direct Communication	966
36.10.2	Using Direct Communication Mode	966
36.10.3	Generating the SPI Bus Cycle during Direct Communication	966
36.11	Interrupts	970
36.12	Usage Note	970
36.12.1	Settings for the Module-Stop Function	970
36.12.2	Procedure for Changing Settings in Multiple Control Registers	970
36.12.3	Setting the Serial Flash Memory	971

37.	Cyclic Redundancy Check (CRC) Calculator	972

37.1	Overview.....	972
37.2	Register Descriptions	973
37.2.1	CRCCR0 : CRC Control Register 0	973
37.2.2	CRCCR1 : CRC Control Register 1	974
37.2.3	CRCDIR/CRCDIR_BY : CRC Data Input Register	974
37.2.4	CRCDOR/CRCDOR_HA/CRCDOR_BY : CRC Data Output Register	975
37.2.5	CRCSAR : Snoop Address Register.....	975
37.3	Operation.....	975
37.3.1	Basic Operation	975
37.3.2	CRC Snoop Function	978
37.4	Usage Notes.....	979
37.4.1	Settings for the Module-Stop State	979
37.4.2	Note on Transmission	979
38.	Division Circuit (DIV).....	980
38.1	Overview.....	980
38.2	Register Descriptions	980
38.2.1	DIVCR : Division Operation Control Register	980
38.2.2	DIVIDEND : Dividend Setting Register	981
38.2.3	DIVSOR : Divisor Setting Register	982
38.2.4	QUOTIENT : Quotient Indication Register.....	982
38.2.5	REMAINDER : Remainder Register	982
38.3	Operation.....	982
38.3.1	Operation Procedure	982
38.3.2	Interrupt Sources	983
38.4	Usage Notes.....	983
38.4.1	Settings for Module-Stop Function	983
39.	Data Inversion and Logical Operation (DIL).....	984
39.1	Overview.....	984
39.2	Register Descriptions	985
39.2.1	IDRn : DIL Input Data Registers n (n = 0 to 3).....	985
39.2.2	IDR1n : DIL Input Data Registers 1n (n = 0 to 3).....	985
39.2.3	ODRn : DIL Output Data Registers n (n = 0 to 3)	985
39.2.4	DILCR : DIL Control Register	986
39.3	Operation.....	987
39.3.1	Basic Operation	987
39.4	Usage Notes.....	988
39.4.1	Settings for the Module-Stop Function	988
40.	MIP LCD Controller (MLCD).....	989
40.1	Overview.....	989

40.2	Register Descriptions	991
40.2.1	MLCDCR : MLCD Control Register	991
40.2.2	MLCDSR : MLCD Status Register	993
40.2.3	MLCDADDR : MLCD Address Setting Register	994
40.2.4	MLCDWRCR : MLCD Transmit Data Size Setting Register	995
40.2.5	MLCDSEND : MLCD Same Image Transmit Register	996
40.2.6	MLCDBKCR : MLCD Block Transfer Setting Register	996
40.2.7	MLCDVCOMCTL : MLCD VCOM Control Register	999
40.2.8	MLCDENBCR : MLCD Enable Signal Control Register	1001
40.2.9	DATAm/DATAn/DATaWk : MLCD Drawing Data Setting Register (m = 0 to 31, n = 0 to 15, k = 0 to 7)	1001
40.3	Operation	1002
40.3.1	Timing waveform	1002
40.3.2	Setting and Timing of MLCD_VCOM	1004
40.3.3	Basic Transmission	1006
40.3.4	Same Image Transmission	1008
40.3.5	Block Transmission	1010
40.4	Interrupt Sources and Event Links	1013
40.4.1	Interrupt Requests	1013
40.4.2	Event Link Operation	1013
40.5	Usage Notes	1013
40.5.1	Setting for the Module-Stop Function	1013
40.5.2	Setting an Area for Drawing Data	1013
40.5.3	MLCD_VCOM Output Settings	1013
41.	2D Graphics Data Conversion Circuit (GDT)	1014
41.1	Overview	1014
41.2	Register Descriptions	1015
41.2.1	GDTCR : GDT Control Register	1015
41.2.2	GDTSCR : Reduction Function Control Register	1018
41.2.3	GDTFDCS : Glyph Data Image Conversion Setting Register	1019
41.2.4	GDTPIER : Image Data Processing Interrupt Enable Register	1021
41.2.5	GDTIBUFn : GDT Image Data Input Register (n = 0 to 47)	1021
41.2.6	GDTOBUFn : GDT Image Data Output Register (n = 0 to 31)	1022
41.3	Operation	1022
41.3.1	GDT Processing Flow	1022
41.3.2	Rotation	1027
41.3.3	Scaling Down	1033
41.3.4	Inversion	1043
41.3.5	Monochrome Compositing	1046
41.3.6	Color Compositing	1051

41.3.7	Scrolling	1057
41.3.8	Conversion of Glyph Data into Image Data	1062
41.3.9	Colorization	1070
41.3.10	Color Data Sorting	1074
41.3.11	Endian Conversion Function	1078
41.3.12	Time Required for Processing a Single Image	1079
41.3.13	Interrupt Sources	1080
41.4	Transferring Image Data to a Color LCD	1080
41.4.1	Overview	1080
41.4.2	Peripheral Functions Used for Color Display	1081
41.5	Usage Notes	1087
41.5.1	Handling of Color Images	1087
41.5.2	Settings for the Module-Stop Function	1087
42.	Boundary Scan	1088
42.1	Overview	1088
42.2	Register Descriptions	1089
42.2.1	JTIR : Instruction Register	1089
42.2.2	JTIDR : ID Code Register	1090
42.2.3	JTBPR : Bypass Register	1090
42.2.4	JTBSR : Boundary Scan Register	1090
42.3	Operation	1090
42.3.1	TAP Controller	1090
42.3.2	Commands	1091
42.4	Usage Notes	1092
43.	Trusted Secure IP Lite (TSIP-Lite)	1094
43.1	Overview	1094
43.2	Operation	1095
43.2.1	Operating Modes and State Transition	1095
43.2.2	Encryption Engine	1096
43.2.3	Key Data Installation	1097
43.2.4	Encryption and Decryption	1098
43.2.5	Creation of Key Generation Information (Using a Random Number)	1101
43.2.6	Random Number Generation	1102
43.3	Interrupts	1102
43.4	Usage Notes	1103
43.4.1	Software Standby Mode	1103
43.4.2	Module-Stop Function Setting	1103
43.4.3	TSIP-Lite Library	1103
44.	14-Bit A/D Converter (ADC14)	1104

44.1	Overview.....	1104
44.2	Register Descriptions	1108
44.2.1	ADDRn : A/D Data Registers n	1108
44.2.2	ADDBLDR : A/D Data Duplexing Register	1109
44.2.3	ADTSDR : A/D Temperature Sensor Data Register	1110
44.2.4	ADVSCDR : A/D VSC_VCC Voltage Voltage Data Register	1111
44.2.5	ADRD : A/D Self-Diagnosis Data Register	1113
44.2.6	ADCSR : A/D Control Register	1114
44.2.7	ADANSA0 : A/D Channel Select Register A0	1117
44.2.8	ADANSA1 : A/D Channel Select Register A1	1118
44.2.9	ADANSB0 : A/D Channel Select Register B0	1119
44.2.10	ADANSB1 : A/D Channel Select Register B1	1119
44.2.11	ADANSC0 : A/D Channel Select Register C0	1120
44.2.12	ADANSC1 : A/D Channel Select Register C1	1120
44.2.13	ADSCSn : A/D Channel Conversion Order Setting Register n (n = 0 to 7, 16, 17, 20, 21, 31)	1121
44.2.14	ADADS0 : A/D-Converted Value Addition/Average Channel Select Register 0.....	1122
44.2.15	ADADS1 : A/D-Converted Value Addition/Average Channel Select Register 1.....	1124
44.2.16	ADADC : A/D-Converted Value Addition/Average Count Select Register	1124
44.2.17	ADCER : A/D Control Extended Register	1125
44.2.18	ADSTRGR : A/D Conversion Start Trigger Select Register.....	1127
44.2.19	ADEXICR : A/D Conversion Extended Input Control Registers.....	1128
44.2.20	ADGCEXCR : A/D Group C Extended Input Control Register.....	1129
44.2.21	ADGCTRGR : A/D Group C Trigger Select Register	1129
44.2.22	ADSSTRn/ADSSTRL/ADSSTRT : A/D Sampling State Register	1130
44.2.23	ADDISCR : A/D Disconnection Detection Control Register	1131
44.2.24	ADELCCR : A/D Event Link Control Register	1132
44.2.25	ADGSPCR : A/D Group Scan Priority Control Register.....	1132
44.2.26	ADCMPCR : A/D Compare Function Control Register	1134
44.2.27	ADCMPANSR0 : A/D Compare Function Window A Channel Select Register 0.....	1135
44.2.28	ADCMPANSR1 : A/D Compare Function Window A Channel Select Register 1.....	1136
44.2.29	ADCMPANSER : A/D Compare Function Window A Extended Input Select Register	1136
44.2.30	ADCMPLR0 : A/D Compare Function Window A Comparison Condition Setting Register 0	1137
44.2.31	ADCMPLR1 : A/D Compare Function Window A Comparison Condition Setting Register 1	1138
44.2.32	ADCMPLER : A/D Compare Function Window A Extended Input Comparison Condition Setting Register	1139
44.2.33	ADCMPPDRn : A/D Compare Function Window A Lower-Side/Upper-Side Level Setting Register (n = 0, 1)	1140
44.2.34	ADWINnLB : A/D Compare Function Window B Lower-Side/Upper-Side Level Setting Register (n = L, U)	1141
44.2.35	ADCMPSR0 : A/D Compare Function Window A Channel Status Register 0	1142

44.2.36	ADCMPSR1 : A/D Compare Function Window A Channel Status Register1	1143
44.2.37	ADCMPSER : A/D Compare Function Window A Extended Input Channel Status Register	1144
44.2.38	ADCMPBNSR : A/D Compare Function Window B Channel Select Register	1144
44.2.39	ADCMPBSR : A/D Compare Function Window B Status Register	1146
44.2.40	ADWINMON : A/D Compare Function Window A/B Status Monitor Register.....	1147
44.2.41	ADHVREFCNT : A/D High-Potential/Low-Potential Reference Voltage Control Register	1148
44.2.42	ADEDCRm : A/D Emulator Debug Function Control Register m (m = 0, 1, 4, 5)	1149
44.2.43	ADEDEXCR : A/D Emulator Debug Function Extension Control Register	1150
44.2.44	ADEDDMY0 : A/D Emulator Debug Function Dummy Data Setting Register 0	1151
44.2.45	ADEDDMY1 : A/D Emulator Debug Function Dummy Data Setting Register 1	1151
44.2.46	ADSCLKCR : A/D Sub-Clock Mode Control Register.....	1152
44.2.47	ADCALC : A/D Calibration Control Register	1152
44.3	Operation.....	1153
44.3.1	Scanning Operation	1153
44.3.2	Single Scan Mode.....	1153
44.3.3	Continuous Scan Mode	1157
44.3.4	Group Scan Mode.....	1160
44.3.5	Compare Function for Windows A and B.....	1179
44.3.6	Analog Input Sampling and Scan Conversion Time	1183
44.3.7	Usage Example of A/D Data Register Automatic Clearing Function	1187
44.3.8	A/D-Converted Value Addition/Average Mode.....	1187
44.3.9	Disconnection Detection Assist Function.....	1187
44.3.10	Starting A/D Conversion with an Asynchronous Trigger.....	1189
44.3.11	Starting A/D Conversion with a Synchronous Trigger from a Peripheral Module	1190
44.3.12	Arbitrary Channel Order Change Function	1190
44.3.13	Offset Calibration Function	1190
44.4	Interrupt Sources and DTC, DMAC Transfer Requests	1192
44.4.1	Interrupt Requests	1192
44.5	Event Link Function	1193
44.5.1	Event Output to the ELC.....	1193
44.5.2	ADC14 Operation through an Event from the ELC.....	1194
44.6	Selecting Reference Voltage	1194
44.7	Usage Notes.....	1194
44.7.1	Constraints on Setting the Registers	1194
44.7.2	Constraints on Reading the Data Registers	1194
44.7.3	Constraints on Stopping A/D Conversion	1194
44.7.4	A/D Conversion Restart and Termination Timing.....	1196
44.7.5	Constraints on Scan End Interrupt Handling	1196
44.7.6	Settings for the Module-Stop Function	1196
44.7.7	Notes on Entering the Low-Power States.....	1196

44.7.8	Error in Absolute Accuracy When Disconnection Detection Assistance Is in Use.....	1196
44.7.9	Notes on Canceling Software Standby Mode	1196
44.7.10	Notes on Changing the Reference Voltage	1197
45.	Reference Voltage Generation Circuit (VREF)	1198
45.1	Overview.....	1198
45.2	Register Descriptions	1198
45.2.1	AVCR : Reference Voltage Output Control Register.....	1198
45.3	Operation.....	1199
45.3.1	Operating States and Register Settings	1199
45.3.2	Usage	1200
45.4	Usage Notes.....	1200
45.4.1	Settings for the Module-Stop Function	1200
45.4.2	Notes on AVCC0 Potential and AVTRO Pin Output.....	1200
45.4.3	Notes on 14-bit A/D Converter Usage	1201
46.	Temperature Sensor (TSN)	1202
46.1	Overview.....	1202
46.2	Register Descriptions	1203
46.2.1	TSCR : Temperature Sensor Control Register	1203
46.2.2	TSCDR : Temperature Sensor Calibration Data Register.....	1203
46.3	Using the Temperature Sensor.....	1203
46.3.1	Preparation for Using the Temperature Sensor	1203
46.3.2	Procedures for Using the Temperature Sensor	1205
46.4	Usage Notes.....	1207
46.4.1	Settings for the Module-Stop Function	1207
47.	Data Operation Circuit (DOC).....	1208
47.1	Overview.....	1208
47.2	DOC Register Descriptions	1208
47.2.1	DOCR : DOC Control Register	1208
47.2.2	DODIR : DOC Data Input Register	1209
47.2.3	DODSR : DOC Data Setting Register.....	1210
47.3	Operation.....	1210
47.3.1	Data Comparison Mode	1210
47.3.2	Data Addition Mode	1210
47.3.3	Data Subtraction Mode	1211
47.4	Interrupt Source.....	1211
47.5	Output of an Event Signal to the Event Link Controller (ELC).....	1212
47.6	Usage Notes.....	1212
47.6.1	Settings for the Module-Stop State.....	1212
48.	Memory Mirror Function (MMF).....	1213

48.1	Overview.....	1213
48.2	Register Descriptions	1213
48.2.1	MMSFR : MemMirror Special Function Register	1213
48.2.2	MMEN : MemMirror Enable Register	1214
48.3	Operation.....	1214
48.3.1	Memory Mirror Function.....	1214
48.3.2	Setting Example.....	1217
49.	SRAM	1219
49.1	Overview.....	1219
49.2	Register Descriptions	1219
49.2.1	Trace Control (for the MTB).....	1219
49.2.2	CoreSight (for MTB).....	1219
49.3	Operation.....	1220
49.3.1	Access Cycle	1220
50.	Flash Memory	1221
50.1	Overview.....	1221
50.1.1	Serial Programming Mode	1221
50.1.2	On-chip Debug Mode (OCD Mode)	1221
50.2	Memory Structure	1222
50.3	Register Descriptions	1222
50.3.1	FLWT : Flash Wait Cycle Register	1223
50.3.2	UIDRn : Unique ID Registers n (n = 0 to 3)	1223
50.3.3	FWEPROR : Flash Write Erase Protect Register	1224
50.3.4	FASTAT : Flash Access Status Register.....	1224
50.3.5	FAEINT : Flash Access Error Interrupt Enable Register.....	1225
50.3.6	FRDYIE : Flash Ready Interrupt Enable Register	1226
50.3.7	FSADDR : FACL Command Start Address Register	1226
50.3.8	FSTATR : Flash Status Register	1227
50.3.9	FENTRYR : Flash P/E Mode Entry Register.....	1231
50.3.10	FSUINTR : Flash Sequencer Setup Initialization Register	1232
50.3.11	FCMDR : FACL Command Register	1232
50.3.12	FAWMON : Flash Access Window Monitor Register	1233
50.3.13	FPCKAR : Flash Sequencer Processing Clock Notification Register	1234
50.3.14	FSUACR : Flash Startup Area Control Register	1234
50.4	Operating Modes Associated with the Flash Memory	1235
50.4.1	Normal Startup Mode.....	1236
50.5	Serial Programming Mode.....	1236
50.5.1	SCI Boot Mode	1236
50.5.2	On-chip Debug Mode (OCD mode)	1237

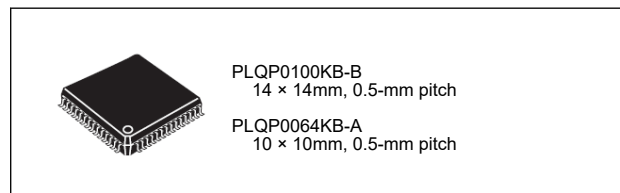
50.6	Functions	1237
50.6.1	Basic Functions	1237
50.6.2	Programming Methods	1239
50.6.3	Programming Commands	1241
50.6.4	Usage of FACL Commands	1244
50.6.5	Security Function (Preventing Tampering)	1257
50.7	Protection (Preventing Unintended Writing)	1259
50.7.1	Software Protection	1260
50.7.2	Error Protection	1260
50.7.3	Boot Program Protection	1261
50.8	Usage Notes	1262
50.8.1	Suspension of Programming/Erase	1263
50.8.2	Reading Areas Where Programming or Erase was Interrupted	1263
50.8.3	Allocation of Vectors for Interrupts and Other Exceptions during Programming and Erase	1263
50.8.4	Notes on Additional Writes	1263
50.8.5	Notes on Programming/Erase	1263
51.	Electrical Characteristics	1264
51.1	Absolute Maximum Ratings	1264
51.2	DC Characteristics	1265
51.2.1	I/O input characteristics (VIH, VIL)	1265
51.2.2	I/O output characteristics (VOH, VOL) (1)	1265
51.2.3	I/O output characteristics (VOL) (2)	1266
51.2.4	Pull-up Resistors	1266
51.2.5	Pin Capacitance	1267
51.2.6	Operating and Standby Current	1268
51.2.7	VCC Rise and Fall Gradient	1274
51.2.8	Internal Linear Regulator Characteristics	1275
51.3	AC Characteristics	1276
51.3.1	Operating Frequency	1276
51.3.2	Clock Timing	1276
51.3.3	Reset Timing	1279
51.3.4	Wakeup Timing	1280
51.3.5	Transition Time Between Operation Modes	1289
51.3.6	Interrupt Input Timing	1292
51.3.7	Trigger Timing of I/O port, POE, GPT, AGT, and ADC14	1293
51.3.8	CAC Timing	1295
51.3.9	SCI Timing	1296
51.3.10	SPI Timing	1301
51.3.11	QSPI Timing	1306
51.3.12	IIC Timing	1307

51.3.13	MLCD Timing	1308
51.3.14	CLKOUT Timing.....	1309
51.3.15	TMR Timing	1310
51.4	A/D Conversion Characteristics.....	1311
51.5	Temperature Sensor Characteristics	1315
51.6	VREF Characteristics	1315
51.7	Oscillation Stop Detection Circuit Characteristics	1316
51.8	Power-on Reset Circuit and Low-voltage Detection Circuit Characteristics	1316
51.9	EHC Characteristics	1320
51.10	Back Bias Voltage Control (VBBC) Circuit Characteristics.....	1323
51.11	Flash Memory Characteristics	1324
51.11.1	Code Flash Memory Characteristics.....	1324
51.12	Boundary Scan Characteristics	1325
51.13	Serial Wire Debug (SWD) Characteristics.....	1327
Appendix 1.	Connecting the Capacitors to the Power Supply Pins	1329
1.1	Example of Connections for Normal Startup Mode	1329
1.2	Example of Connections in Energy Harvesting Startup Mode (1)	1330
1.3	Example of Connections in Energy Harvesting Startup Mode (2)	1332
Appendix 2.	Package Dimensions	1336
Revision History		1341

64 MHz, 32-bit Arm® Cortex®-M0+, 256-KB flash memory, 128-KB SRAM, energy harvesting control circuit, MIP LCD controller, 2D graphic engine, 14-bit ultra-low power consumption A/D converter, VREF circuit, RTC, sub-clock correction circuit (theoretical regulation), security function (optional), SPI, quad SPI

Features

- Arm Cortex-M0+ core incorporated
 - Maximum operating frequency: 64 MHz
 - Arm® Memory Protection Unit (Arm MPU) with 8 regions
 - CoreSight™ Debug Port: SW-DP
- Power-aving functions
 - Back-bias control function based on silicon-on-thin-buried-oxide (SOTB™) process technology
 - Operation at ultra-low power-supply voltages (from 1.62 V to 3.6 V)
 - Three power control modes based on the operating frequency
 - Four low power consumption modes
 - Three power supply modes
- On-chip Code flash memory
 - 256-Kbyte code flash memory
 - No cycles of waiting for access in operation at or below 32 MHz; one cycle of waiting at frequencies above 32 MHz
 - Function for area protection prevents erroneous overwriting or tampering
- On-chip SRAM
 - 128-Kbyte SRAM with no access wait cycles
- Data transfer
 - Four DMA controllers
 - Single data transfer controller (DTC)
- Reset and supply management
 - Power-on reset (POR)
 - Low voltage detection (LVD) can be set.
- Multiple clock sources
 - External crystal oscillator (main clock): 8 to 32 MHz
 - External crystal oscillator (sub-clock): 32.768 kHz
 - High-speed on-chip oscillator (HOCO): 24, 32, 48, or 64 MHz
 - Middle-speed on-chip oscillator (MOCO): 2 MHz
 - Low-speed on-chip oscillator (LOCO): 32 kHz
 - Independent watchdog timer on-chip oscillator: 16 kHz
- Energy harvesting control
 - A power generation element is directly connectable.
 - High-speed startup is possible without having to wait for the charging of a secondary battery.
 - Function to prevent a secondary battery from overcharging
- Independent watchdog timer
 - 14-bit counter, 16-kHz (1/2 LOCO clock frequency) operation
- Sub-clock correction circuit (CCC)
 - The CCC corrects the accuracy of oscillation every 16 seconds (theoretical regulation).
 - Events can be generated per second in deep software standby mode.
- Communication functions
 - Two serial peripheral interfaces
 - Single 128-bit buffer for which up to eight commands can be specified
 - Single 32-bit buffer for which one command can be specified
 - Single quad serial peripheral interface connectable to an external flash memory
 - Two I²C bus interfaces
 - Five serial communications interfaces (SCIg)
 - Asynchronous, clock-synchronous, simple I²C, simple SPI, and smart card interfaces, and IrDA interface version 1.0 (the latter is only applicable to SCI0)
 - Two serial communication interfaces (SCIf) each having a 16-byte FIFO



- Various analog circuits
 - Single 14-bit successive approximation A/D converter
 - High precision: 8 channels, standard precision: 4 channels
 - Single temperature sensor for measuring the internal temperature of the chip
 - VREF circuit for the 14-bit A/D converter reference voltage
- Various timer circuits
 - Six general PWM timers (GPT)
 - Two 32-bit counters
 - Four 16-bit counters
 - Four asynchronous general-purpose timers (AGT) that can be used in standby mode
 - Two 32-bit counters
 - Two 16-bit counters
 - Two 8-bit timers (TMR)
 - Single realtime clock (RTC)
 - Single watchdog timer (WDT)
 - Single low-speed timer (LST) that operates at 1 kHz
 - A circuit for converting hexadecimal numbers to decimal numbers for use as a stopwatch
- Human machine interfaces
 - Single memory-in-pixel (MIP) LCD controller (MLCD) Parallel interface is supported.
 - Single 2D graphics data conversion circuit (GDT)
- Security functions (optional)
 - Single Trusted Secure IP Lite (TSIP)
 - AES (128- or 256-bit key length, supporting ECB, CBC, CMAC, GCM, and others)
 - Key wrapping protects against the leakage of the encryption keys of users.
 - An access management circuit disables illicit access to the encryption engine.
 - Using the other security functions together with area protection enables secure booting and secure over-the-air (OTA) software updates.
- Operating voltage and temperature range
 - VCC = IOVCC = IOVCCn = AVCC0 = 1.62 V to 3.6 V
 - IOVCCn and AVCC0 can each be independently set to a voltage within the range between 1.62 V and 3.6 V.
 - Ta: -40 to +85°C

1. Overview

1.1 Function Outline

Table 1.1 to Table 1.11 show the outline of maximum specifications. The number of peripheral channels differs depending on the number of pins of the package. For details, see Table 1.13.

Table 1.1 Arm core

Feature	Functional description
Arm® Cortex®-M0+ core	- Maximum operating frequency: up to 64 MHz - Arm Cortex-M0+ core: - Revision: r0p1-00rel0 - Armv6-M architecture profile - Single-cycle integer multiplier - Arm Memory Protection Unit (MPU): - Armv6 Protected Memory System Architecture - Eight protect regions - SysTick timer: - Driven by SYSTICCLK (LOCO or ICLK)

Table 1.2 Memory

Feature	Functional description
Code flash memory	- Maximum 256 KB of code flash memory. - No cycles of waiting for access in operation at or below 32 MHz; one cycle of waiting at frequencies above 32 MHz - Prefetch function - On-board programming (three types): - Programming in serial programming mode (SCI boot mode) - Programming in on-chip debug mode - Programming by a routine for code flash memory rewriting within a user program
SRAM	- Maximum 128 KB of SRAM SRAM0: 0x2000_0000 to 0x2000_7FFF SRAM1: 0x2000_8000 to 0x2001_FFFF Both areas are available during low leakage current mode. 64 MHz, No cycles of waiting for access

Table 1.3 System (1 of 2)

Feature	Functional description
Startup modes	Three startup modes: - Normal startup mode - Energy harvesting startup mode - SCI boot mode
Resets	The MCU provides 13 resets. The resets are classified into two types: System resets that initialize the MCU and power shutdown reset that does not initialize the MCU. See section 6, Resets.
Low Voltage Detection (LVD)	The Low Voltage Detection (LVD) module monitors the voltage level input to the VCC pin and VBAT_EHC pin. The detection level can be selected by register settings. The LVD module consists of three separate voltage level detectors (LVD0, LVD1, LVDBAT). LVD0 and LVD1 measure the voltage level input to the VCC pin, and LVDBAT measures the voltage level input to the VBAT_EHC pin. LVD registers allow your application to configure detection of VCC and VBAT_EHC changes at various voltage thresholds. See section 8, Low Voltage Detection (LVD).
Clocks	- The MCU has the following clock generation circuits. - 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 (IWDTLOCO) - Clock output support - CLKOUT pin (capable of outputting all types of clock signals) - CLKOUT32K pin (capable of outputting SOSC clock signals)

Table 1.3 System (2 of 2)

Feature	Functional description
Clock Frequency Accuracy Measurement Circuit (CAC)	The Clock Frequency Accuracy Measurement Circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock selected as the measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range. When measurement is complete or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated. See section 11, Clock Frequency Accuracy Measurement Circuit (CAC) .
Interrupt Controller Unit (ICU)	The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC), the DMA Controller (DMAC), and the Data Transfer Controller (DTC) modules. The ICU also controls non-maskable interrupts. See section 16, Interrupt Controller Unit (ICU) .
Power-saving functions	The MCU has several functions for power saving, such as setting clock dividers, stopping modules, selecting power control mode in operating mode, transitioning to low power consumption mode, and power supply mode per domain. - Three power control modes based on the operating frequency - Boost mode (up to 64 MHz) - Normal mode - High-speed mode (up to 32 MHz) - Low-speed mode (up to 2 MHz) - Low leakage current mode (32.768 kHz) - Five low-power consumption modes - Operating mode - Sleep mode - Software standby mode - Snooze mode - Deep software standby mode - Three power supply modes - All-power supply mode (ALLPWON) - Flash-excluded power supply mode (EXFPWON) - Minimum power supply mode (MINPWON) See section 13, Power-Saving Functions .
Back-bias voltage control ^{*1} (VBBC) function	Program control of the back bias voltage enables low leakage current operation in the low leakage current mode.
Energy harvesting control circuit (EHC)	Starting up of the MCU in the power-saving mode is possible by controlling the power generating element, storage capacitor, and secondary battery.
Register write protection (RWP)	The register write protection function protects important registers from being overwritten due to software errors. The registers to be protected are set with the Protect Register (PRCR). See section 15, Register Write Protection .
Memory Protection Unit (MPU)	The MCU has four Memory Protection Units (MPUs) and a CPU stack pointer monitor function are provided. See section 18, Memory Protection Unit (MPU) .
Key Interrupt Function (KINT)	The key interrupt function (KINT) is generated a key interrupt by detecting a valid edge on the key interrupt input pin. See section 12, Key Interrupt Function (KINT) .

Note 1. Voltage for charging the VBP and VBN pins

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 interconnect (link) modules, allowing direct link between modules without CPU intervention. Event signals can be output regardless of the setting of the associated interrupt request enable bit. See section 21, Event Link Controller (ELC) .

Table 1.5 Direct memory access

Feature	Functional description
DMA Controller (DMAC)	This MCU incorporates an 4-channel direct memory access controller (DMAC). The DMAC is a module to transfer data without the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address. See section 19, DMA Controller (DMAC) .
Data Transfer Controller (DTC)	A Data Transfer Controller (DTC) module is provided for transferring data when activated by an interrupt request. See section 20, 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 GPT32 × 2 channels and a 16-bit timer with GPT16 × 4 channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. See section 24, General PWM Timer (GPT) .
Port Output Enable for GPT (POE)	The Port Output Enable (POE) function can place the General PWM Timer (GPT) output pins in the output disable state See section 23, Port Output Enable for GPT (POE) .
Low power Asynchronous General Purpose Timer (AGT, AGTW)	The low power Asynchronous General Purpose Timer (AGT, AGTW) is a 16-bit, 32-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events. This timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and can be accessed with the AGT register. See section 25, Low Power Asynchronous General Purpose Timer (AGT, AGTW) .
8-bit timers (TMR)	8-bit timer (TMR) can count external events and provide multiple functions such as clearing counters, and outputting interrupt requests and pulses of required duty cycles, using the compare match signals with two registers.
Wake Up Timer (WUPT)	The wake up timer based on 32-bit counter provides multiple functions such as resetting count, and outputting interrupt requests and pulses to external pins when an overflow occurs.
Realtime Clock (RTC)	The realtime clock (RTC) has three counting modes, calendar count mode, binary count mode, and 32-kHz 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. See section 28, Realtime Clock (RTC) .
Clock Correction Circuit (CCC)	The CCC corrects the oscillation accuracy every 16 seconds for the 32.768-kHz subclock. • Clock output after correction: 2.048 kHz/512 Hz • Signal output (CCCOUT): Selectable from 512 Hz/1 Hz, or RTC output (1 Hz/64 Hz) • Support of function for event linking by the ELC
Watchdog Timer (WDT)	The Watchdog Timer (WDT) is a 14-bit down counter that can be used to reset the MCU when the counter underflows because the system has run out of control and is unable to refresh the WDT. In addition, the WDT can be used to generate a non-maskable interrupt or an underflow interrupt. See section 30, Watchdog Timer (WDT) .
Independent Watchdog Timer (IWDT)	The Independent Watchdog Timer (IWDT) consists of a 14-bit down counter that must be serviced periodically to prevent counter underflow. The IWDT provides functionality to reset the MCU or to generate a non-maskable interrupt or an underflow interrupt. Because the timer operates with an independent, dedicated clock source, it is particularly useful in returning the MCU to a known state as a fail-safe mechanism when the system runs out of control. The IWDT can be triggered automatically by a reset, underflow, refresh error, or a refresh of the count value in the registers. See section 31, Independent Watchdog Timer (IWDT) .
Low-Speed Clock Timer (LST)	The low-speed clock timer (LST) contains a 1-kHz timer-counter and a circuit for converting hexadecimal numbers to decimal numbers. This is a 13-bit timer that can be used to indicate a count that needs to be displayed in decimal. • Capable of counting from 0.000 to 1.999 seconds (in units of 0.001 seconds) • A value in decimal notation can be directly stored in a register

Table 1.7 Communication interfaces

Feature	Functional description
Serial Communications Interface (SCI)	The Serial Communications Interface (SCI) × 7 channels have asynchronous and synchronous serial interfaces: - Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA)) 8-bit clock synchronous interface - Simple IIC (master-only) - Simple SPI - Smart card interface The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. SCIn (n = 0, 1) 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 32, Serial Communications Interface (SCI).
IrDA Interface (IrDA)	The IrDA (Infrared Data Association) interface sends and receives IrDA data communication waveforms in association with SCI1 based on the IrDA standard 1.0.
I ² C bus interface (IIC)	The I²C bus interface (IIC) has 2 channels. The IIC module conforms with and provides a subset of the NXP I²C (Inter-Integrated Circuit) bus interface functions. See section 34, I²C Bus Interface (IIC).
Serial Peripheral Interface (SPI)	The SPI provides high-speed full-duplex and transmit-only synchronous serial communications with multiple processors and peripheral devices. See section 35, Serial Peripheral Interface (SPI).
Quad Serial Peripheral Interface (QSPI)	The QSPI is connectable to a serial ROM that has an SPI-compatible interface. 1 channel - Support for extended SPI, dual SPI, and quad SPI protocols - Configurable to SPI mode 0 and SPI mode 3 - Address width selectable from 8, 16, 24, or 32 bits
External bus	QSPI area: Connectable to the QSPI (external device interface)

Table 1.8 Analog

Feature	Functional description
14-bit A/D Converter (ADC14)	A 14-bit successive approximation A/D converter incorporated Up to 12 analog input channels are selectable. The analog input channels and the temperature sensor output are selectable for conversion. The A/D conversion accuracy is selectable between 12-bit and 14-bit conversion making it possible to optimize the tradeoff between speed and resolution in generating a digital value. 14 bits × 12 channels (maximum value) (high accuracy: 8 channels, standard accuracy: 4 channels) - Resolution: 14 bits (14-bit or 12-bit conversion selectable) - Operating mode: - Scan mode (single-scan mode, continuous-scan mode, or group-scan mode) - Group A priority control (only for group-scan mode) - Variable sampling state count - A/D-converted value addition mode or average mode selectable - Disconnection detection assist function - Double-trigger mode (duplication of A/D conversion data) - Support of function for event linking by the ELC - Automatic clear function of A/D data registers - Compare function for window A and window B - Digital compare function - Comparison of values in the comparison register and the data register, and comparison between values in the data registers See section 44, 14-Bit A/D Converter (ADC14).
Temperature Sensor (TSN)	The on-chip Temperature Sensor (TSN) determines and monitors the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is fairly linear. The output voltage is provided to the ADC14 for conversion and can be further used by the end application. See section 46, Temperature Sensor (TSN).
Reference voltage generation circuit (VREF)	The circuit generates two types (1.25 V/2.5 V) of reference voltage. The generated voltage can be used as the reference voltage for the ADC.

Table 1.9 Human machine interfaces

Feature	Functional description
MIP LCD controller (MLCD) ^{*1}	MIP-method liquid crystal panel driver circuit incorporated
2D graphics data conversion circuit (GDT)	A graphic accelerator circuit that handles 2D image processing incorporated • Handling of up to 32-byte image data. Up to 63 × 64 bits for conversion of glyph data into image data. • Rotations of 90-degree clockwise, 90-degree counterclockwise, vertical flip, and horizontal flip • Scaling down to 1/8, 2/8, 3/8, 4/8, 5/8, 6/8, or 7/8 by pixel averaging and to 1/2 by pixel skipping • Inversion allows bit-wise inversion of images; 1 is inverted to 0, and vice versa. • Monochrome compositing of a foreground image, background image, and trimming image • Color compositing of a foreground image and background image, and setting of priority color and transparent color • Scrolling of an image in 1-bit units • Conversion of glyph data into image data • Colorization of monochrome images by RGB values • Color data sorting allows separate R, G, and B images in memory to be sorted into a single area in order of R, G, and B • Endian conversion

Note 1. General three-wire MIP can be supported by combining SPI0 and GDT.

Table 1.10 Data processing

Feature	Functional description
Cyclic Redundancy Check (CRC) calculator	The Cyclic Redundancy Check (CRC) calculator generates CRC codes to detect errors in the data. The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication. Additionally, various CRC-generation polynomials are available. The snoop function allows monitoring 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. See section 37, Cyclic Redundancy Check (CRC) Calculator.
Data Operation Circuit (DOC)	The Data Operation Circuit (DOC) compares, adds, and subtracts 16-bit data. When a selected condition applies, 16-bit data is compared and an interrupt can be generated. See section 47, Data Operation Circuit (DOC).
Divider (DIV)	A circuit for handling high-speed division for signed 32-bit fixed point data • Dividend: Signed 32-bit data • Divisor: Signed 32-bit data
Data inversion and Logical operation (DIL)	• Data inversion – The bit inversion value of input data is output • AND, OR, and XOR operations of two input data – Data inversion enables NAND, NOR, and XNOR operations • Conversion of data alignment per byte width (byte swap) • Bit order inversion of MSB and LSB every 8 bits

Table 1.11 Security

Feature	Functional description
Trusted Secure IP Lite (TSIP-Lite)	• Access management circuit available • Security algorithms: – Common key cryptosystem (symmetrical cryptography): AES key length: 128 bits/256 bits – Encryption usage modes: GCM, ECB, CBC, CMAC, XTS, CTR, GCTR, CCM • Other support features: See section 43, Trusted Secure IP Lite (TSIP-Lite).

1.2 Block Diagram

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

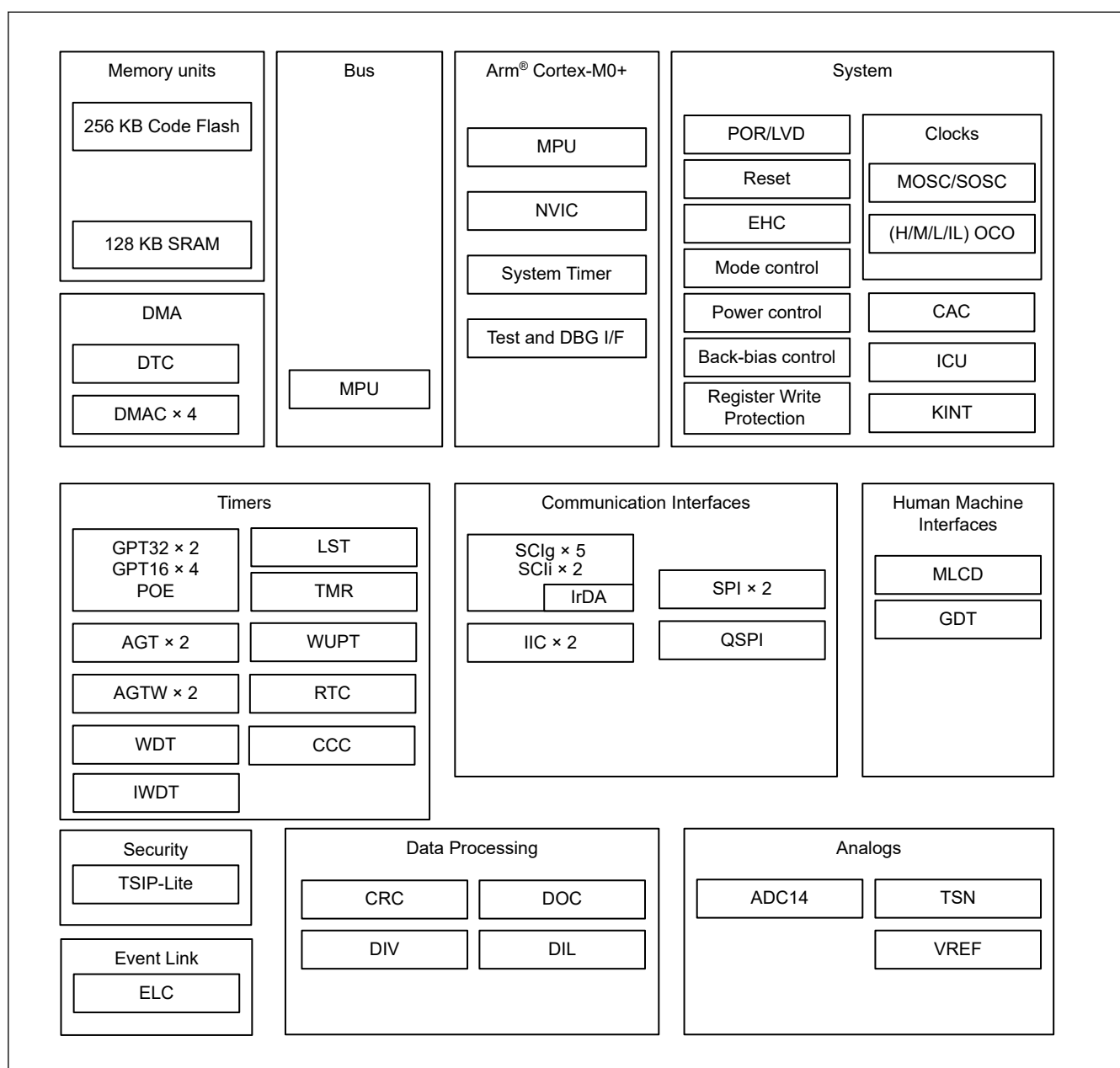


Figure 1.1 Block diagram

1.3 Part Numbering

Table 1.12 shows a list of products.

Table 1.12 Product list

Product part number	Package code	Code flash	SRAM	TSIP-Lite
R7F0E01182CFP	PLQP0100KB-B	256 KB	128 KB	Supported
R7F0E01082CFP	PLQP0100KB-B			Not supported
R7F0E01182CFM	PLQP0064KB-A			Supported
R7F0E01082CFM	PLQP0064KB-A			Not supported
R7F0E01182DBH	TBD (BGA100pin)			Supported
R7F0E01082DBH	TBD (BGA100pin)			Not supported
R7F0E01182DBR	SXBG0072MA-A			Supported
R7F0E01082DBR	SXBG0072MA-A			Not supported
R7F0E01182DNG	PVQN0056LA-A			Supported
R7F0E01082DNG	PVQN0056LA-A			Not supported

1.4 Function Comparison

Table 1.13 Function Comparison (1 of 4)

Part Number		R7F0E01182CFP	R7F0E01082CFP	R7F0E01182CFM	R7F0E01082CFM	R7F0E01182DBH	R7F0E01082DBH
Pin count		100		64		100	
GPIO	I/O pins	73		37		73	
	Input pins	1		1		1	
Package		LQFP				BGA	
Code flash memory		256 KB					
SRAM		128 KB					
CPU operating frequency		32 MHz maximum (Normal mode) 64 MHz maximum (Boost mode) 32.768 kHz maximum (Low leakage current mode)					
Interrupt control	ICU	Yes					
	IRQ	Channels 0 to 9		Channels 0 to 5, 7, and 8		Channels 0 to 9	
Key interrupt	KINT	8 channels					
DMA	DTC	Yes					
	DMAC	Channels 0 to 3					
Event control	ELC	Yes					
Energy harvesting	EHC	Yes					
Back-bias voltage control	VBBC	Yes					

Table 1.13 Function Comparison (2 of 4)

Part Number			R7F0E01182CFP	R7F0E01082CFP	R7F0E01182CFM	R7F0E01082CFM	R7F0E01182DBH	R7F0E01082DBH
Timers	GPT32		Channels 0 and 1					
	GPT16		Channels 2 to 5					
		POE	Yes					
	AGT		Channels 0 and 1					
	AGTW		Channels 0 and 1					
	TMR		Channels 0 and 1					
	WUPT		Yes					
	RTC		Yes					
	CCC		Yes					
	WDT		Yes					
	IWDT		Yes					
	LST		Yes					
Communications	SCIg	w/o FIFO	Channels 2 to 5, and 9					
	SCli	w/ FIFO	Channels 0 and 1					
		IrDA	Yes					
	IIC		Channels 0 and 1		Channel 1		Channels 0 and 1	
	SPI	128 bit buffer	Channel 0					
		32 bit buffer	Channel 1					
	QSPI		Yes					
Analog	ADC14	High precision	8 channels					
		Standard precision	4 channels		No		4 channels	
	TSN		Yes					
	VREF		Yes					
HMI graphics	MLCD		Yes					
	GDT		Yes					
Data Processing	CRC		Yes					
	DOC		Yes					
	DIV		Yes					
	DIL		Yes					
Security	TSIP-Lite		Yes	No	Yes	No	Yes	No

Table 1.13 Function Comparison (3 of 4)

Function		R7F0E01182DBR	R7F0E01082DBR	R7F0E01182DNG	R7F0E01082DNG
Pin count		72		56	
GPIO	I/O pins	43		33	
	Input pins	1		1	
Package		WLBGA		QFN	
Code flash memory		256 KB			
SRAM		128 KB			
CPU operating frequency		32 MHz maximum (Normal mode) 64 MHz maximum (Boost mode) 32.768 kHz maximum (Low leakage current mode)			

Table 1.13 Function Comparison (4 of 4)

Function			R7F0E01182DBR	R7F0E01082DBR	R7F0E01182DNG	R7F0E01082DNG
Interrupt control		ICU		Yes		
		IRQ		Channels 0 to 9		Channels 0 to 5, Channels 7 to 8
Key interrupt		KINT		8 channels		
DMA		DTC		Yes		
		DMAC		Channels 0 to 3		
Event control		ELC		Yes		
Energy harvesting		EHC		Yes		
Back-bias voltage control		VBBC		Yes		
Timers	GPT32		Channels 0 and 1			
	GPT16		Channels 2 to 5			
			POE		Yes	
	AGT		Channels 0 and 1			
	AGTW		Channels 0 and 1			
	TMR		Channels 0 and 1			
	WUPT		Yes			
	RTC		Yes			
	CCC		Yes			
	WDT		Yes			
	IWDT		Yes			
	LST		Yes			
Communications	SCIg	w/o FIFO	Channels 2 to 5, and 9			
	SCli	w/ FIFO	Channels 0 and 1			
		IrDA	Yes			
	IIC		Channels 0 and 1		Channel 1	
	SPI	128 bit buffer	Channel 0			
		32 bit buffer	Channel 1			
QSPI			Yes			
Analog	ADC14	High precision	8 channels			
		Standard precision	4 channels		No	
	TSN		Yes			
	VREF		Yes			
HMI graphics	MLCD		Yes			
	GDT		Yes			
Data Processing	CRC		Yes			
	DOC		Yes			
	DIV		Yes			
	DIL		Yes			
Security	TSIP-Lite		Yes	No	Yes	No

1.5 Pin Functions

Table 1.14 shows a list of pin functions. See the Appendix for examples of how to connect the smoothing capacitor.

Table 1.14 Pin functions (1 of 6)

Function	Signal	I/O	Description
Power supply	VCC/ IOVCC	Input	Normal startup mode Power supply pin. Connect it to the system power supply. Connect to VSS through a 0.1-μF smoothing capacitor. Place the smoothing capacitor close to the pin.*2 Apply voltage prior to the IOVCCn pins. Energy harvesting startup mode Power supply pin. Connect it to the system power supply. Connect to VSS through 0.1-μF smoothing capacitor (1). Place the smoothing capacitor close to the pin. In addition, connect to VSS through smoothing capacitor (2) having capacity of 1/10 of capacity of a storage capacitor connected to the VCC_SU pin to improve robustness against external noise and obtain stable operation of the circuit. For instance, connect a 4.7-μF smoothing capacitor in the case where a 47-μF storage capacitor is connected to the VCC_SU pin. If placing the smoothing capacitor (2) close to this pin is possible, the smoothing capacitor (1) is not required. For more details, see .
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
	VCL	Input	Internal power supply stabilization pin. Connect the pin to VSS through a 4.7-μF smoothing capacitor. Place the smoothing capacitor close to the pin.
	VCLH	Input	Internal power supply stabilization pin. Separately from the VCL pin, connect the VCLH pin to VSS through a 4.7-μF smoothing capacitor. Place the smoothing capacitor close to the pin.
	VBN	—	Back-bias voltage stabilization pin. Connect the pin to VSS through a 0.56-μF smoothing capacitor. Place the smoothing capacitor close to the pin.
	VBP	—	
	VSC_VCC	Input	Normal startup mode Power supply pin supplied from a power generation element. Connect it to the system power supply (0 V) in normal startup mode. Energy harvesting startup mode Power supply pin supplied from a power generation element. Connect this pin to VSC_GND through a smoothing capacitor in parallel with the power generation element. Place the smoothing capacitor close to the pin. While a 4.7-nF to 47-nF smoothing capacitor is recommended, select a capacity value suitably in accordance with stability of a power generating element or the like.

Table 1.14 Pin functions (2 of 6)

Function	Signal	I/O	Description
Power supply	VCC_SU	I/O	Normal startup mode Power supply pin supplied from a storage capacitor. Short it to VCC/IOVCC in normal startup mode. Energy harvesting startup mode Power supply pin supplied from a storage capacitor. When using a photovoltaic cell as a power generating element, connect a storage capacitor with a capacitance value in accordance with an operating temperature, and with a value of at least 10 times VCC. A capacitance value of 47 μ F is required at 25°C. As a temperature becomes higher, a larger capacitance value is required. See section 51.9. EHC Characteristics . Connect this pin to a 100- μ F storage capacitor in the case where other power generating elements are used.
	VSC_GND	Input	VSC_VCC ground pin. Connect it to the system power supply (0 V).
	VBAT_EHC	Input	Normal startup mode Power supply pin supplied from a secondary battery. Connected it to VCC/IOVCC in normal startup mode. Energy harvesting startup mode Power supply pin supplied from a secondary battery. Connect a 2.4-, 2.5-, 2.6-, 2.7-, 2.8-, 2.9-, 3.0-, or 3.1-V secondary battery or a super capacitor in energy harvesting startup mode.
	IOVCC0, IOVCC1	Input	Power supply pin for input/output. Connect the pin to VSS through a 0.1- μ F smoothing capacitor. Place the smoothing capacitor close to the pin. ^{*2 *3} This pin can be left open-circuit when not in use. When the pin is to be used, set the corresponding bit in section 13.2.22. VOCCR : Power Supply Open Control Register .
Clock	XTAL	Input	MOSC resonator connect pin. EXTAL is an external clock input pin.
	EXTAL	Output	
	XCIN	Input	SOSC resonator connect pin
	XCOUT	Output	
	CLKOUT	Output	Clock output pin
	CLKOUT32K	Output	SOSC clock output pin
Clock frequency accuracy measurement	CACREF	Input	Reference clock input pin for the clock frequency accuracy measurement circuit
Startup mode control	MD	Input	Pin for setting the startup mode. The signal level on this pin must not be changed during startup mode transition on release from the reset state.
	EHMD	Input	Pin for setting the energy harvesting mode
System control	RES#	Input	Reset signal input pin. The MCU enters the reset state when this signal goes low.
	BSCANP	Input	IOVCCn pin power supply forced input pin When the boundary scan function is in use, setting this pin to the high level while IOVCCn power is being supplied enables the supply of power to all I/O ports.
Interrupts	NMI	Input	Non-maskable interrupt request pin
	IRQ0 to IRQ9, IRQ0_A_DS to IRQ3_A_DS	Input	Maskable interrupt request pins Pins that have "_DS" appended to their names can be used as triggers for release from deep software standby mode.
KINT	KRM00 to KRM07	Input	A key interrupt can be generated by inputting a falling edge to the key interrupt input pins.
On-chip debugger	SWDIO	I/O	SWD data input/output pin
	SWCLK	Input	SWD clock input pin

Table 1.14 Pin functions (3 of 6)

Function	Signal	I/O	Description
Boundary scan	TMS	Input	Boundary scan pins
	TDI	Input	
	TCK	Input	
	TDO	Output	
GPT, POE	GTIOC0A to GTIOC5A, GTIOC0B to GTIOC5B	I/O	Input capture, output compare, or PWM output pin
	GTETRG, GTETRGB	Input	External trigger input 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	AGTIO0, AGTIO1	I/O	External event input and pulse output pins
	AGTEE0, AGTEE1	Input	External event input enable signals
	AGTO0, AGTO1	Output	Pulse output pins
	AGTOA0, AGTOA1	Output	Compare match A output pins
	AGTOB0, AGTOB1	Output	Compare match B output pins
AGTW	AGTWIO0, AGTWIO1	I/O	External event input and pulse output pins
	AGTWEE0, AGTWEE1	Input	External event input enable signals
	AGTWO0, AGTWO1	Output	Pulse output pins
	AGTWOA0, AGTWOA1	Output	Compare match A output pins
	AGTWOB0, AGTWOB1	Output	Compare match B output pins
TMR	TMCIO, TMCIO1	Input	Input pins for external clocks to be input to the counter
	TMRI0, TMRI1	Input	Input pins for the counter reset
	TMO0, TMO1	Output	Compare match output pins
WUPT	TMWO	Output	Pulse output pin
RTC	RTCIC0 to RTCIC2	Input	Time capture event input pins
	RTCCOUT	Output	Output pin for 1-Hz or 64-Hz clock
CCC	CCCOUT	Output	CCC clock output pin

Table 1.14 Pin functions (4 of 6)

Function	Signal	I/O	Description
SCi	[Asynchronous mode/clock synchronous mode]		
	SCK0, SCK1	I/O	Input/output pins for the clock (clock synchronous mode)
	RXD0, RXD1	Input	Input pins for received data (asynchronous mode/clock synchronous mode)
	TXD0, TXD1	Output	Output pins for transmitted data (asynchronous mode/clock synchronous mode)
	CTS0, CTS1	Input	Input pins for controlling the start of transmission and reception (asynchronous mode/clock synchronous mode)
	RTS0, RTS1	Output	Output pins for controlling the start of transmission and reception (asynchronous mode/clock synchronous mode)
	[Simple I ² C mode]* ¹		
	SSCL0, SSCL1	I/O	Input/output pins for the I ² C clock (simple I ² C mode)
	SSDA0, SSDA1	I/O	Input/output pins for the I ² C data (simple I ² C mode)
	[Simple SPI mode]* ¹		
	SCK0, SCK1	I/O	Input/output pins for the clock (simple SPI mode)
	MISO0, MISO1	I/O	Input/output pins for slave transmission of data (simple SPI mode)
	MOSI0, MOSI1	I/O	Input/output pins for master transmission of data (simple SPI mode)
	SS0, SS1	Input	Chip-select input pins (simple SPI mode)
SCi _g	[Asynchronous mode/clock synchronous mode]		
	SCK2 to SCK5, SCK9	I/O	Input/output pins for the clock (clock synchronous mode)
	RXD2 to RXD5, RXD9	Input	Input pins for received data (asynchronous mode/clock synchronous mode)
	TXD2 to TXD5, TXD9	Output	Output pins for transmitted data (asynchronous mode/clock synchronous mode)
	CTS2 to CTS5, CTS9	Input	Input pins for controlling the start of transmission and reception (asynchronous mode/clock synchronous mode)
	RTS2 to RTS5, RTS9	Output	Output pins for controlling the start of transmission and reception (asynchronous mode/clock synchronous mode)
	[Simple I ² C mode]* ¹		
	SSCL2 to SSCL5, SSCL9	I/O	Input/output pins for the I ² C clock (simple I ² C mode)
	SSDA2 to SSDA5, SSDA9	I/O	Input/output pins for the I ² C data (simple I ² C mode)
	[Simple SPI mode]* ¹		
	SCK2 to SCK5, SCK9	I/O	Input/output pins for the clock (simple SPI mode)
	MISO2 to MISO5, MISO9	I/O	Input/output pins for slave transmission of data (simple SPI mode)
	MOSI2 to MOSI5, MOSI9	I/O	Input/output pins for master transmission of data (simple SPI mode)
	SS2 to SS5, SS9	Input	Chip-select input pins (simple SPI mode)
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 pins
	MOSIA, MOSIB	I/O	Input/output pins for data output from the master
	MISOA, MISOB	I/O	Input/output pins for data output from the slave
	SSLA0, SSLB0	I/O	Input/output pins for slave selection
	SSLA1 to SSLA3, SSLB1 to SSLB3	Output	Output pins for slave selection

Table 1.14 Pin functions (5 of 6)

Function	Signal	I/O	Description
QSPI	QSPCLK	Output	QSPI clock output pin
	QSSL	Output	QSPI slave output pin
	QIO0 to QIO3	I/O	Data 0 to data 3
Analog power supply	AVCC0	Input	Analog power supply pin for a 14-bit A/D converter, a reference voltage generation circuit, and a temperature sensor. Connect the pin to AVSS0 through a 1.0-μF smoothing capacitor. Place the smoothing capacitor close to the pin.*4 This pin can be left open-circuit when not in use. When the pin is to be used, set the corresponding bit in the power supply open control register (VOCR).
	AVSS0	Input	Analog ground pin for the 14-bit A/D converter, reference voltage generation circuit, and temperature sensor. This pin can be left open-circuit when not in use. When the pin is to be used, set the corresponding bit in the power supply open control register (VOCR).
	VREFH0	Input	Analog reference voltage pin for the 14-bit A/D converter. Connect the pin to VREFL0 through a 1.0-μF smoothing capacitor. Place the smoothing capacitor close to the pin.*5 Connect this pin to AVCC0 when not using the A/D converter. Leave this pin open-circuit if AVCC0 is not to be supplied.
	AVTRO	Output	Reference voltage output terminal of reference voltage generation circuit (VREF). Connect the pin to VREFL0 through a 10-μF smoothing capacitor.
	VREFL0	Input	Analog reference ground pin for the 14-bit A/D converter. Connect this pin to AVSS0 when not using the A/D converter. Leave open if AVCC0 is not supplied.
ADC14	AN000 to AN007, AN016, AN017, AN020, AN021	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
MLCD	MLCD_VCOM	Output	Polar signal pin for common electrode
	MLCD_XRST	Output	Output pin for LCD control
	MLCD_SCLK	Output	Communication serial output clock pin
	MLCD_DEN	Output	Data identification signal pin
	MLCD_ENBS	Output	Horizontal directional data enable pin
	MLCD_ENBG	Output	Vertical directional data enable pin
	MLCD_SI0 to MLCD_SI7	Output	Graphics data signal pin

Table 1.14 Pin functions (6 of 6)

Function	Signal	I/O	Description
I/O ports	P000 to P007, P010 to P015	I/O	14-bit input/output pins
	P100 to P113	I/O	14-bit input/output pins
	P200	Input	1-bit input dedicated pin. Multiplexed with the NMI pin function.
	P201 to P205, P207 to P210	I/O	8-bit input/output pins
	P300 to P302, P314 to P315	I/O	5-bit input/output pins
	P409 to P411	I/O	3-bit input/output pins
	P412, P413	I/O	2-bit input/output pin. Multiplexed with the EXTAL and XTAL pin functions.
	P500, P501, P508 to P511	I/O	6-bit input/output pins
	P600 to P604	I/O	5-bit input/output pins
	P700 to P704	I/O	5-bit input/output pins
	P806 to P815	I/O	10-bit input/output pins

Note: Use a laminated ceramic capacitor as a smoothing capacitor.

Note 1. For the SCLi and SCLg interfaces, each communications pin has multiple functions that work differently depending on the mode as follows: RXDn/SCLn/MISO_n, TXDn/SDAn/MOS_n, CTS_n/RTS_n/SS_n

Note 2. In an environment where there is much external noise, optionally connect these pins to VSS through a 10-μF smoothing capacitor close to the respective current sources to improve robustness against external noise and obtain stable operation of the circuit.

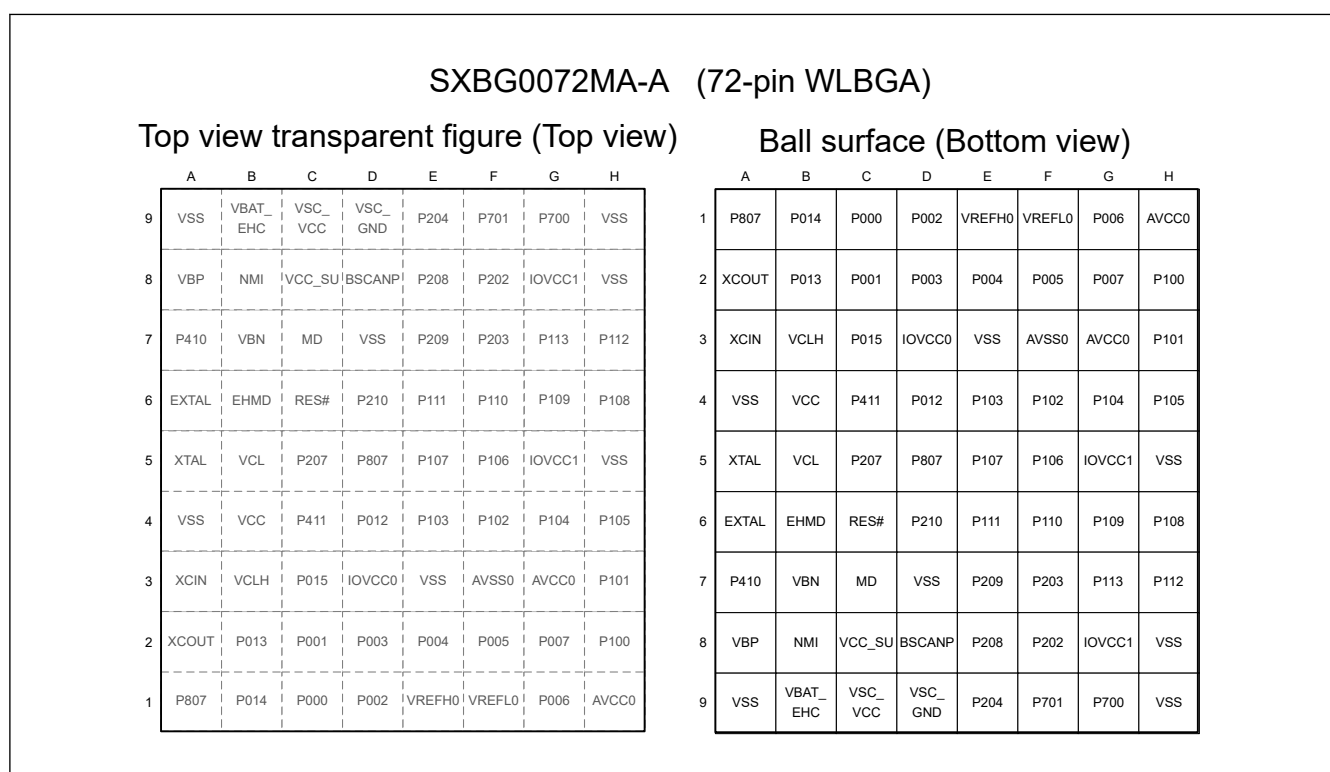
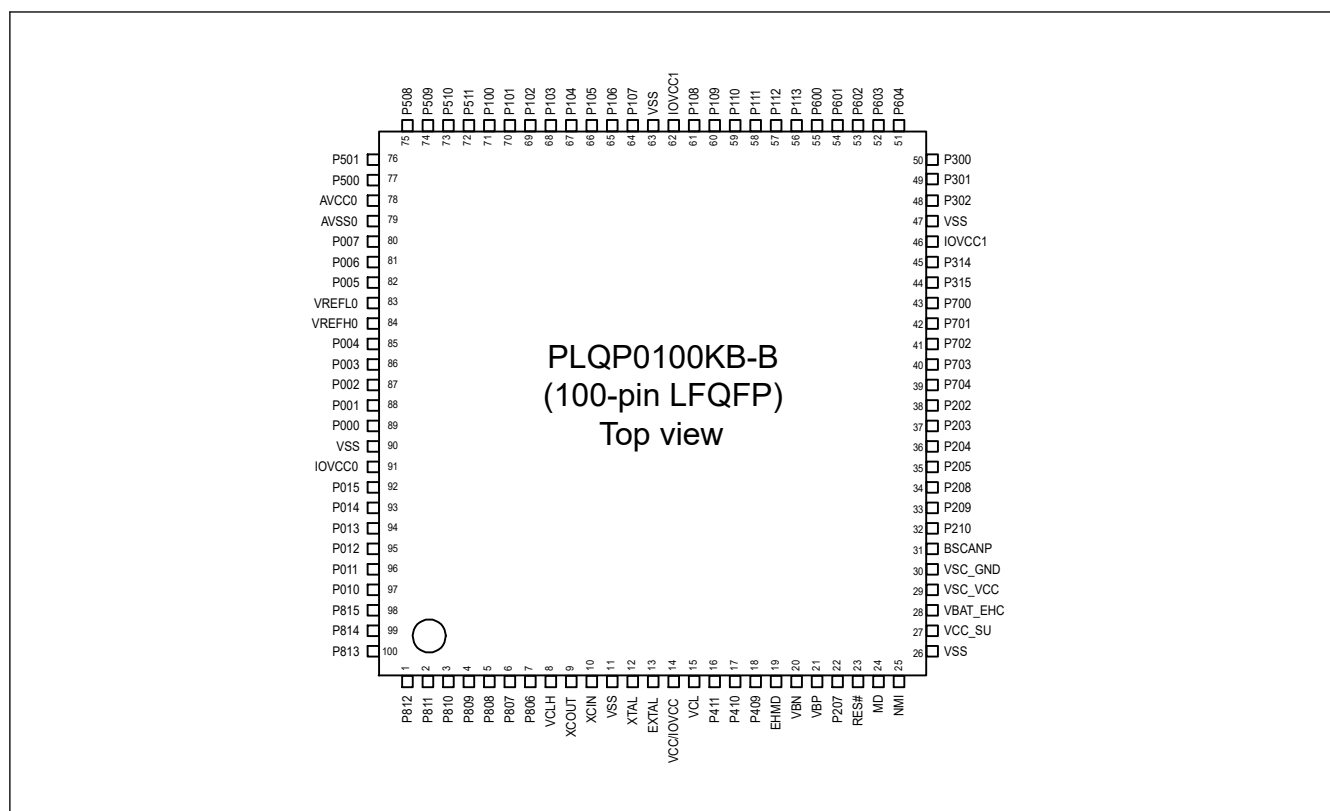
Note 3. When some of the IOVCC0 and IOVCC1 pins are connected at the same voltage, a 10-μF smoothing capacitor can be shared. In the case where the pin is connected to VCC/IOVCC, a 10-μF smoothing capacitor is not necessary.

Note 4. In an environment where there is much external noise, optionally connect this pin to AVSS0 through a 10-μF smoothing capacitor close to the current source to improve robustness against external noise and obtain stable operation of the circuit.

Note 5. In an environment where there is much external noise, optionally connect this pin to VREFL0 through a 10-μF smoothing capacitor close to the current source to improve robustness against external noise and obtain stable operation of the circuit.

1.6 Pin Assignments

Figure 1.2, Figure 1.3 and Figure 1.4 show the pin assignments from the top view. The pin arrangement diagram indicates the positions of power supply pins and I/O ports.



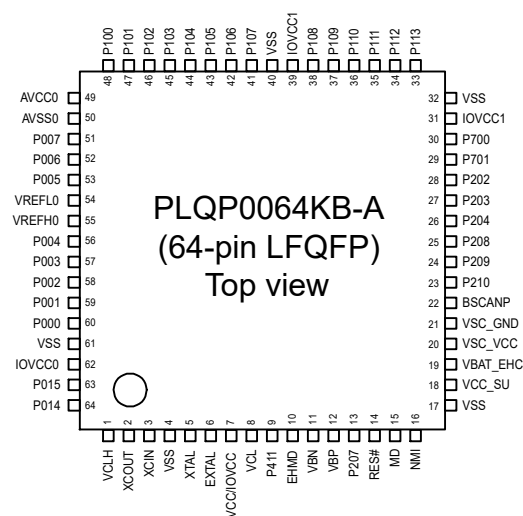


Figure 1.4 Pin assignment for LFQFP 64-pin

TBD
(100-pin BGA)
Ball surface (bottom view)

	1	2	3	4	5	6	7	8	9	10	11	12
M	P812	P814	P811	XCOUT	XCIN	VSS	VCC	P411	EHMD	RES#	P201	VSS
L	IOVCC0	P813	P810	VLCH	XTAL	EXTAL	VCL	VBP	VBN	P207	P200	VCC_SU
K	P010	P011	P809	P807	P806			P409	P410	P209	VBAT_EHC	VSS
J	IOVSS0	P013	P808							P210	BSCANP	VSC_VCC
H	P815	P012	P014							P208	P205	P204
G	P002	P015									P203	P703
F	VREFL	P001									P704	P701
E	VREFH	P007	P003							P202	P702	P314
D	AVSS0	P004	P006							P109	P700	IOVCC1
C	AVCC0	P000	P005	P501	P104			P110	P111	P302	P315	IOVSS1
B	P500	P100	P101	P102	P103	P106	P107	P108	P112	P600	P300	P301
A	P509	P508	P510	P511	P105	IOVSS1	IOVCC1	P113	P602	P601	P603	P604

Figure 1.5 Pin assignment for BGA 100-pin (Bottom view)

TBD
(100-pin BGA)
Top view transparent figure (Top view)

	1	2	3	4	5	6	7	8	9	10	11	12
A	P509	P508	P510	P511	P105	IOVSS1	IOVCC1	P113	P602	P601	P603	P604
B	P500	P100	P101	P102	P103	P106	P107	P108	P112	P600	P300	P301
C	AVCC0	P000	P005	P501	P104			P110	P111	P302	P315	IOVSS1
D	AVSS0	P004	P006							P109	P700	IOVCC1
E	VREFH	P007	P003							P202	P702	P314
F	VREFL	P001									P704	P701
G	P002	P015									P203	P703
H	P815	P012	P014							P208	P205	P204
J	IOVSS0	P013	P808							P210	BSCANP	VSC_VCC
K	P010	P011	P809	P807	P806			P409	P410	P209	VBAT_EHC	VSS
L	IOVCC0	P813	P810	VCLH	XTAL	EXTAL	VCL	VBP	VBN	P207	P200	VCC_SU
M	P812	P814	P811	XCOU	XCIN	VSS	VCC	P411	EHMD	RES#	P201	VSS

Figure 1.6 Pin assignment for BGA 100-pin (Top view)

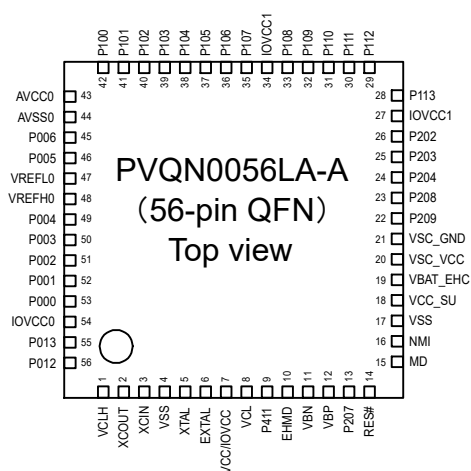


Figure 1.7 Pin assignment for QFN 56-pin

1.7 Pin Lists

Table 1.15 Pin list (1 of 3)

100-pin LQFP	100-pin BGA	72-pin WLGA	64-pin LQFP	56-pin QFN	Power, System, Clock	I/O ports	Timers (CAC, CCC, GPT, AGT, AGTW, TMR, WUPT, RTC)	Communications (SCI, SPI, IIC, QSPI)	Display (MLCD)	External Int. (IRQn, KINT)	Analog (ADC14)	Power
1	M1	—	—	—		P812	AGTWEE1_B	TXD4_C/QSPCLK_A				IOVCC0
2	M3	—	—	—		P811	AGTWIO1_B	QIO0_A				IOVCC0
3	L3	—	—	—		P810	AGTIO1_B/GTIOC3A_B	QIO1_A		IRQ5_B		IOVCC0
4	K3	—	—	—		P809	AGTEE1_B/GTIOC3B_B	QIO2_A		IRQ6_B		IOVCC0
5	J3	F5	—	—		P808	AGTO1_B	RXD3_B/QIO3_A		IRQ2_B		IOVCC0
6	K4	E4	—	—		P807	AGTOA1_B	CTS3_B/QSSL_A		IRQ6_A		IOVCC0
7	K5	—	—	—		P806	AGTOB1_B					IOVCC0
8	L4	G2	1	1	VCLH							
9	M4	H1	2	2	XCOUT							IOVCC
10	M5	G1	3	3	XCIN							IOVCC
11	M6	F1	4	4	VSS							
12	L5	E1	5	5	XTAL	P413	GTIOC0A_A	TXD3_A				IOVCC
13	L6	D1	6	6	EXTAL	P412	GTIOC0B_A	RXD3_A				IOVCC
14	M7	F2	7	7	VCC/IOVCC							
15	L7	E2	8	8	VCL							
16	M8	F3	9	9	CLKOUT32K_A	P411	AGTWEE1_A/GTIOC0B_B	TXD9_A/SCK3_A		IRQ0_A_DS		IOVCC
17	K9	C1	—	—		P410				IRQ9_A		IOVCC
18	K8	—	—	—	CLKOUT32K_B	P409				IRQ9_B		IOVCC
19	M9	D2	10	10	EHMD							IOVCC
20	L9	C2	11	11	VBN							
21	L8	B1	12	12	VBP							
22	L10	E3	13	13		P207	AGTWO1_A/GTIOC0A_B	RXD9_A/CTS3_A		IRQ1_A_DS		IOVCC
23	M10	D3	14	14	RES#							IOVCC
24	M11	C3	15	15	MD	P201						IOVCC
25	L11	B2	16	16		P200				NMI		IOVCC
26	M12	A1	17	17	VSS							
27	L12	B3	18	18	VCC_SU							
28	K11	A2	19	19	VBAT_EHC							
29	J12	A3	20	20	VSC_VCC							
30	K12	A4	21	21	VSC_GND							
31 *1	J11	B4	22 *1	—	BSCANP							IOVCC
32	J10	D4	23	—		P210	AGTWOA1_A					IOVCC
33	K10	C4	24	22		P209	AGTWOB1_A					IOVCC
34	H10	B5	25	23		P208	AGTWIO1_A/TMWO					IOVCC
35	H11	E5	—	—		P205	AGTWO0_B	CTS4_B		IRQ8_C		IOVCC1
36	H12	A5	26	24		P204	ADTRG0_A/AGTO0_A/ GTIU_A/TMCIO_A/RTICIC0_A	SCK4_B		IRQ7_B		IOVCC1
37	G11	D5	27	25		P203	AGTOA0_A/GTIV_A/ TMRI0_A/RTICIC1_A	RXD4_B				IOVCC1
38	E10	C5	28	26		P202	CACREF_A/AGTOB0_A/ GTIW_A/TMO0_A/ CCCOUT_A/RTICOUT_A	TXD4_B		IRQ4_A		IOVCC1
39	F11	—	—	—		P704	AGTWOA0_B	CTS0_C				IOVCC1
40	G12	—	—	—		P703	AGTWOB0_B	TXD0_C				IOVCC1
41	E11	—	—	—		P702	AGTWEE0_B	RXD0_C				IOVCC1
42	F12	A6	29	—		P701	TMRI1/RTICIC2_A	SCL1				IOVCC1
43	D11	A7	30	—		P700	TMO1	SCK0_C/SDA1				IOVCC1

Table 1.15 Pin list (2 of 3)

100-pin LQFP	100-pin BGA	72-pin WLGA	64-pin LQFP	56-pin QFN	Power, System, Clock	I/O ports	Timers (CAC, CCC, GPT, AGT, AGTW, TMR, WUPT, RTC)	Communications (SCI, SPI, IIC, QSPI)	Display (MLCD)	External Int. (IRQn, KINT)	Analog (ADC14)	Power
44	C11	—	—	—		P315	AGTWIO0_B/GTIOC4A_B	TXD5_B				IOVCC1
45	E12	—	—	—		P314	GTIOC4B_B	RXD5_B				IOVCC1
46	D12	B7	31	27	IOVCC1							
47	C12	A8	32	—	VSS							
48	C10	—	—	—		P302	GTIU_B/GTIOC2A_B/ TMCIO_B	CTS5_B				IOVCC1
49	B12	—	—	—		P301	GTIV_B/GTIOC2B_B/ TMRIO_B/CCCOU_B/ RTCOU_B	SCK5_B				IOVCC1
50	B11	—	—	—		P300	GTIW_B/TMO0_B					IOVCC1
51	A12	—	—	—		P604	GTOWLO_B/GTIOC5B_B/ RTCIC0_B	TXD9_B		IRQ3_C		IOVCC1
52	A11	—	—	—		P603	GTETRGB_B/GTIOC5A_B/ RTCIC1_B	RXD9_B				IOVCC1
53	A9	—	—	—		P602	GTOUUP_B/RTCIC2_B	SCK9_B				IOVCC1
54	A10	—	—	—		P601	GTOULO_B	CTS9_B				IOVCC1
55	B10	—	—	—		P600	GTETRGA_B					IOVCC1
56	A8	B8	33	28		P113	AGTEE0_A/GTOWUP_A/ TMC1	TXD4_A/SSLB2_A/QIO0_B	MLCD_VCOM	IRQ3_A_DS		IOVCC1
57	B9	B6	34	29		P112	AGTEE0_B/AGTWEE0_A/ GTOWLO_A	RXD4_A/SSLB3_A/QIO1_B	MLCD_XRST	IRQ8_B		IOVCC1
58	C9	C8	35	30		P111	AGTO0_B/AGTWO0_A/ GTOUUP_A/GTIOC2A_A	CTS4_A/RXD5_A/SSLB1_A/ QIO2_B	MLCD_SCLK			IOVCC1
59	C8	C7	36	31		P110	AGTOA0_B/AGTWOA0_A/ GTOULO_A/GTIOC2B_A	SCK9_A/SCK5_A/MOSIB_A/ QIO3_B	MLCD_DEN			IOVCC1
60	D10	C6	37	32		P109	AGTOB0_B/AGTWOB0_A/ GTOVUP_A	CTS9_A/CTS5_A/MISOB_A/ QSPCLK_B	MLCD_ENBS			IOVCC1
61	B8	D8	38	33		P108	AGTIO0_B/AGTWIO0_A/ GTOVLO_A	SCK4_A/TXD5_A/RSPCKB_A/ QSSL_B	MLCD_ENBG			IOVCC1
62	A7	E7	39	34	IOVCC1							
63	A6	E8	40	—	VSS							
64	B7	D7	41	35	TMS	P107	AGTOB1_A/GTETRGA_A/ GTIOC1A_A	CTS0_A/RSPCKA_A	MLCD_SI0	IRQ7_A/ KRM07_A		IOVCC1
65	B6	D6	42	36	TDO	P106	AGTOA1_A/GTETRGB_A/ GTIOC1B_A	TXD0_A/SSLB0_A	MLCD_SI1	IRQ3_B/ KRM06_A		IOVCC1
66	A5	E6	43	37	TDI	P105	AGTO1_A/GTIOC4A_A	RXD0_A/MISOA_A	MLCD_SI2	IRQ8_A/ KRM05_A		IOVCC1
67	C5	F8	44	38	TCK	P104	AGTIO1_A/GTIOC4B_A	SCK0_A/MOSIA_A	MLCD_SI3	IRQ4_B/ KRM04_A		IOVCC1
68	B5	F7	45	39		P103	AGTEE1_A/GTIOC5A_A	CTS2_A/CTS1_A/SSLA0_A	MLCD_SI4	KRM03_A		IOVCC1
69	B4	F6	46	40		P102	AGTIO0_A/GTIOC5B_A	TXD2_A/TXD1_A/IRTXD1_A/ SSLA1_A	MLCD_SI5	KRM02_A		IOVCC1
70	B3	G8	47	41		P101	ADTRG0_B/GTIOC0A_C	RXD2_A/RXD1_A/IRRXD1_A/ SSLA2_A	MLCD_SI6	KRM01_A		IOVCC1
71	B2	H8	48	42		P100	CACREF_B/GTIOC0B_C	SCK2_A/SCK1_A/SSLA3_A	MLCD_SI7	KRM00_A		IOVCC1
72	A4	—	—	—		P511	GTOVUP_B/GTIOC1B_B	SCK0_B		KRM03_B		IOVCC1
73	A3	—	—	—		P510	GTOVLO_B/GTIOC1A_B	RXD0_B		KRM02_B	AN021	IOVCC1
74	A1	—	—	—		P509		TXD0_B		KRM01_B	AN020	IOVCC1
75	A2	—	—	—		P508				IRQ4_C	AN017	IOVCC1
76	C4	—	—	—		P501					AN016	IOVCC1
77	B1	—	—	—		P500	GTOWUP_B	CTS0_B				IOVCC1
78	C1	G7	49	43	AVCC0							
79	D1	G6	50	44	AVSS0							
80	E2	H7	51	—		P007					AN007	AVCC0
81	D3	J7	52	45		P006					AN006	AVCC0

Table 1.15 Pin list (3 of 3)

100-pin LQFP	100-pin BGA	72-pin WLGA	64-pin LQFP	56-pin QFN	Power, System, Clock	I/O ports	Timers (CAC, CCC, GPT, AGT, AGTW, TMR, WUPT, RTC)	Communications (SCI, SPI, IIC, QSPI)	Display (MLCD)	External Int. (IRQn, KINT)	Analog (ADC14)	Power
82	C3	H6	53	46		P005					AN005	AVCC0
83	F1	J6	54	47	VREFL0							
84	E1	J5	55	48	VREFH0/AVTRO							
85	D2	H5	56	49		P004					AN004	AVCC0
86	E3	H4	57	50		P003					AN003	AVCC0
87	G1	J4	58	51		P002					AN002	AVCC0
88	F2	H3	59	52		P001					AN001	AVCC0
89	C2	J3	60	53		P000					AN000	AVCC0
90	J1	G5	61	—	VSS							
91	L1	G4	62	54	IOVCC0							
92	G2	G3	63	—	CLKOUT	P015	GTIOC3A_A	SSLA1_B		IRQ5_A		IOVCC0
93	H3	F4	64	—		P014	GTIOC3B_A	SSLA0_B		IRQ2_A_DS		IOVCC0
94	J2	J2	—	55		P013		SCK3_B/SCL0				IOVCC0
95	H2	H2	—	56		P012		TXD3_B/SDA0				IOVCC0
96	K2	—	—	—		P011		RSPCKA_B				IOVCC0
97	K1	—	—	—		P010		MOSIA_B				IOVCC0
98	H1	—	—	—		P815	AGTWOB1_B	CTS4_C/MISOA_B				IOVCC0
99	M2	—	—	—		P814	AGTWOA1_B	SCK4_C/SSLA2_B				IOVCC0
100	L2	—	—	—		P813	AGTWO1_B	RXD4_C/SSLA3_B				IOVCC0

Note: Note the following regarding pin names:

- For the SCli and SCIg interfaces, each communication pin has multiple functions that work differently depending on the mode as follows: RXDn/SCLn/MISO_n, TXDn/SDAn/MOSIn, CTSn/RTSn/SSn
- Renesas recommends using the sets of pins that have the same letter ("_A", "_B", "_C" to indicate group membership) appended to their names. For the SPI, QSPI, and SCI interfaces, the AC portion of the electrical characteristics is measured per group.
- Pins that have "_DS" appended to their names can be used as triggers for release from deep software standby.

Note 1. LQFP package does not have BSCANP function, so connect to GND.

2. CPU

The MCU is based on the Arm[®] Cortex[®]-M0+ core.

2.1 Overview

2.1.1 CPU

- Arm Cortex-M0+
 - Revision: r0p1-00rel0
 - Armv6-M architecture profile
 - Single-cycle integer multiplier
- Memory Protection Unit (MPU)
 - Armv6 Protected Memory System Architecture
 - 8 protected regions
- SysTick timer
 - Driven by LOCO clock or ICLK

See reference 1. and reference 2. in [section 2.8. References](#) for details.

2.1.2 Debug

- Arm[®] CoreSight[™] MTB-M0+
 - Revision: r0p1-00rel0
 - Buffer size: 32 KB MTB SRAM
- Data Watchpoint Unit (DWT)
 - Two comparators for watchpoints
- Breakpoint Unit (BPU)
 - Four instruction comparators
- CoreSight Debug Access Port (DAP)
 - Serial Wire-Debug Port (SW-DP)
- Debug Register Module (DBGREG)
 - Reset control
 - Halt control

See reference 1. and reference 2. in [section 2.8. References](#) for details.

2.1.3 Operating Frequency

The operating frequencies for the MCU are as follows:

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

2.1.4 Block Diagram

[Figure 2.1](#) shows a block diagram of the Cortex-M0+ core.

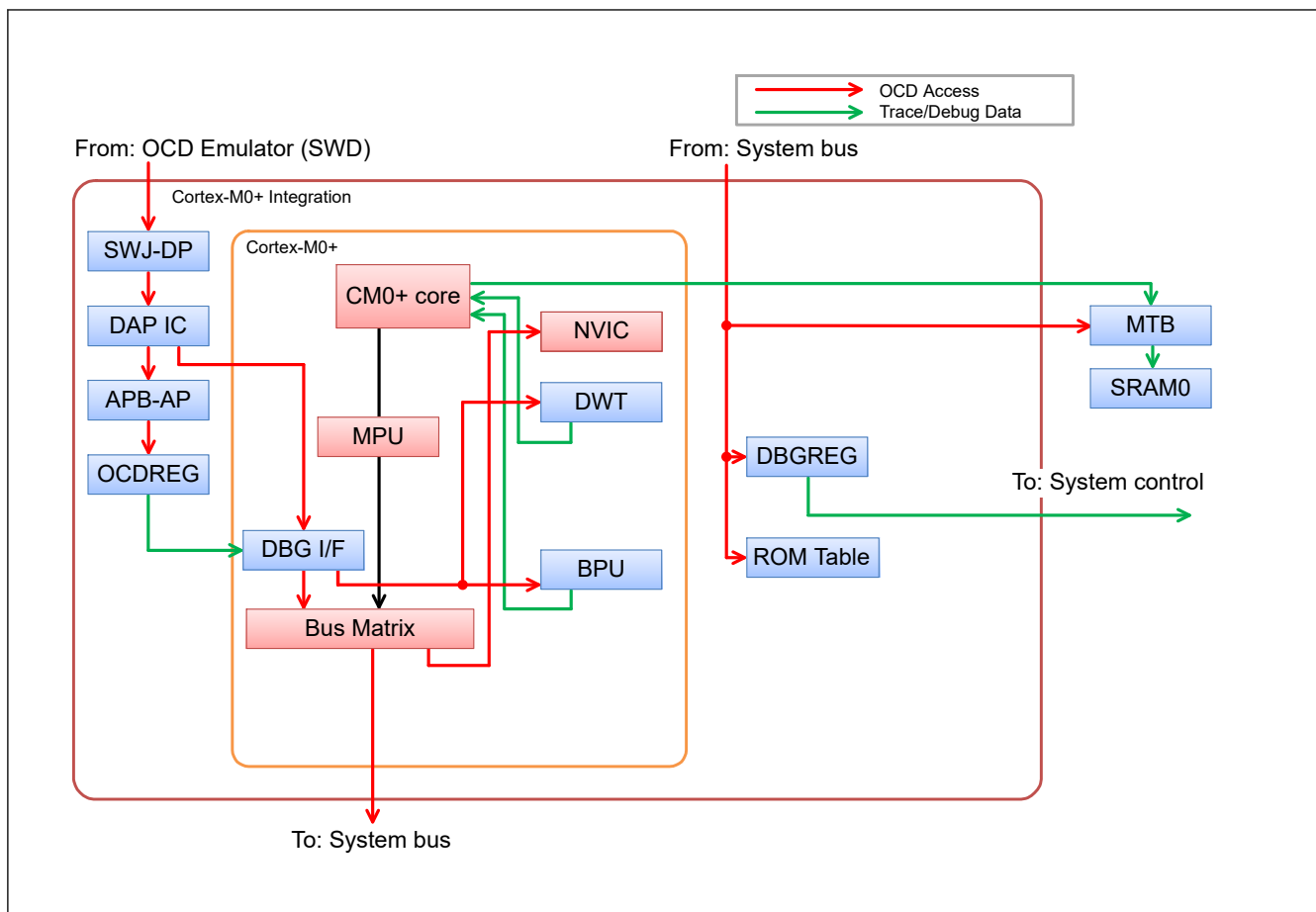


Figure 2.1 Cortex-M0+ block diagram

2.2 Implementation Options

Table 2.1 shows the implementation options of the MCU.

Table 2.1 Implementation options

Option	Implementation
MPU	Included, 8 protect regions
Single-cycle multiplier	Included
Number of interrupts	32
Sleep mode power saving	Sleep mode and other low power modes are supported. For more details, see section 13, Power-Saving Functions . Note: SCB.SCR.SLEEPDEEP is ignored.
Endianness	Little-endian
SysTick	See reference 3. in section 2.8. References .
System reset request output	The SYSRESETREQ bit in Application Interrupt and Reset Control Register causes a CPU reset
Vector table offset register	Included

For details, see reference 3. in [section 2.8. References](#).

2.3 SWD Interface

Table 2.2 shows the SWD pins.

Table 2.2 SWD pins

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

2.4 Debug Function

2.4.1 Debug Mode Definition

[Table 2.3](#) shows the CPU debug modes and usage conditions.

Table 2.3 CPU debug mode and conditions

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

Note 1. OCD connect is determined by the CDBGWUPREQ 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.CDBGWUPREQ bit.

Note 2. Debug authentication is defined by the Armv6-M Architecture. Enabled means that both invasive and non-invasive CPU debugging are permitted. Disabled means that both are not permitted.

2.4.2 Debug Mode Effects

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

2.4.2.1 Low power mode

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

2.4.2.2 Reset

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

Table 2.4 Reset or interrupt and mode setting (1 of 2)

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

Table 2.4 Reset or interrupt and mode setting (2 of 2)

Reset or interrupt name	Control in On-Chip Debug (OCD) mode	
	OCD break mode	OCD run mode
Deep software standby reset	Same as user mode	
MINPWON mode reset	Same as user mode	
Software reset	Same as user mode	

Note: In OCD break mode, the CPU is halted. In OCD run mode, the CPU is in OCD mode and the CPU is not halted.

Note 1. The IWDT and WDT always stop in this mode.

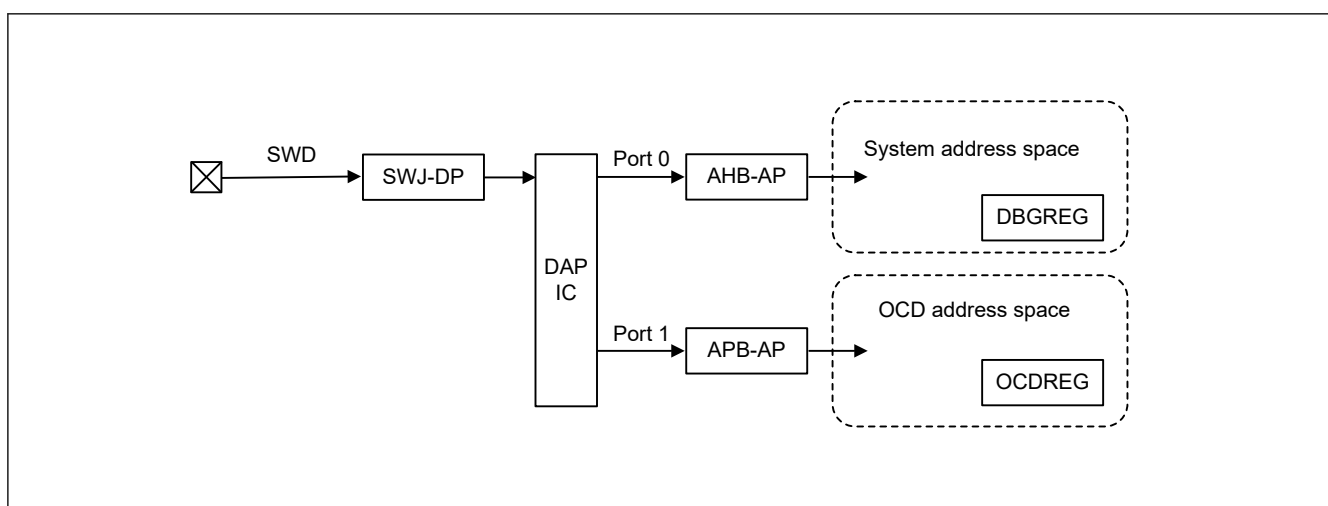
2.5 Programmers Model

2.5.1 Address Spaces

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

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

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

**Figure 2.2** SWD authentication block diagram

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

2.5.2 Cortex-M0+ Peripheral Address Map

In the system address space, the Cortex-M0+ core has a Private Peripheral Bus (PPB) that can only be accessed from the CPU and OCD emulator. [section 2.5.2. Cortex-M0+ Peripheral Address Map](#) shows the address map of the MCU.

Table 2.5 Cortex-M0+ peripheral address map

Component name	Start address	End address	Note
DWT	0xE000_1000	0xE000_1FFF	See reference 2. in section 2.8. References .
BPU	0xE000_2000	0xE000_2FFF	See reference 2. in section 2.8. References .
SCS	0xE000_E000	0xE000_EFFF	See reference 2. in section 2.8. References .
ROM Table	0xE00F_F000	0xE00F_FFFF	See section 2.5.4. CoreSight ROM Table and reference 5. in section 2.8. References .

2.5.3 External Debug Address Map

In the system address space, the Cortex-M0+ 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-M0+ external debug components.

Table 2.6 External debug address map

Component name	Start address	End address	Note
MTB (SRAM area)	0x2000_0000	0x2000_7FFF	MTB uses up to 32 KB as trace buffer See reference 6. in section 2.8. References .
MTB (SFR area)	0x4001_9000	0x4001_9FFF	See reference 6. in section 2.8. References .
ROM Table	0x4001_A000	0x4001_AFFF	See reference 6. in section 2.8. References .

2.5.4 CoreSight ROM Table

The MCU contains two CoreSight ROM Tables. One ROM Table is the root that contains a list of external debug components and a pointer to Arm components. The other ROM Table contains a list of Arm components.

2.5.4.1 ROM entries

[Table 2.7](#) shows the first ROM Table that contains a pointer to the Arm system area and the user area component information. [Table 2.8](#) shows the second ROM Table that contains Arm system area component information. See reference 5. and reference 6. for details.

Table 2.7 ROM Table 1

#	Address	Access size	R/W	Value	Target module pointer
0	0x4001_A000	32 bits	R	0xA00E5003	Arm Cortex-M0+ processor
1	0x4001_A004	32 bits	R	0xFFFFF003	MTB
2	0x4001_A008	32 bits	R	0x00000000	End of entries

Table 2.8 ROM Table 2

#	Address	Access size	R/W	Value	Target module pointer
0	0xE00F_F000	32 bits	R	0xFFF0F003	SCS
1	0xE00F_F004	32 bits	R	0xFFF02003	DWT
2	0xE00F_F008	32 bits	R	0xFFF03003	BPU
3	0xE00F_F00C	32 bits	R	0x00000000	End of entries

2.5.4.2 CoreSight component registers

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

[Table 2.10](#) shows the registers. [Table 2.10](#) shows the registers. See reference 5. in [section 2.8. References](#) for details of each register.

Table 2.9 CoreSight component registers in the CoreSight ROM Table (Renesas Unique ID) (1 of 2)

Name	Address	Access size	R/W	Initial value
Arm CM0+	0x4001_A000	32 bits	R	0xA00E5003
MTB	0x4001_A004	32 bits	R	0xFFFFF003
PID4	0x4001_AFD0	32 bits	R	0x00000004
PID5	0x4001_AFD4	32 bits	R	0x00000000
PID6	0x4001_AFD8	32 bits	R	0x00000000
PID7	0x4001_AFDC	32 bits	R	0x00000000
PID0	0x4001_AFE0	32 bits	R	0x0000002A

Table 2.9 CoreSight component registers in the CoreSight ROM Table (Renesas Unique ID) (2 of 2)

Name	Address	Access size	R/W	Initial value
PID1	0x4001_AFE4	32 bits	R	0x00000030
PID2	0x4001_AFE8	32 bits	R	0x0000000A
PID3	0x4001_AFEC	32 bits	R	0x00000000
CID0	0x4001_AFF0	32 bits	R	0x0000000D
CID1	0x4001_AFF4	32 bits	R	0x00000010
CID2	0x4001_AFF8	32 bits	R	0x00000005
CID3	0x4001_AFFC	32 bits	R	0x000000B1

Table 2.10 CoreSight component registers in the CoreSight ROM Table (CoreSight-ID)

Name	Address	Access size	R/W	Initial value
SCS	0xE00F_F000	32 bits	R	0xFFFF0F003
DWT	0xE00F_F004	32 bits	R	0xFFFF02003
BPU	0xE00F_F008	32 bits	R	0xFFFF03003
PID4	0xE00F_FFD0	32 bits	R	0x00000004
PID5	0xE00F_FFD4	32 bits	R	0x00000000
PID6	0xE00F_FFD8	32 bits	R	0x00000000
PID7	0xE00F_FFDC	32 bits	R	0x00000000
PID0	0xE00F_FFE0	32 bits	R	0x000000C0
PID1	0xE00F_FFE4	32 bits	R	0x000000B4
PID2	0xE00F_FFE8	32 bits	R	0x0000000B
PID3	0xE00F_FFEC	32 bits	R	0x00000000
CID0	0xE00F_FFF0	32 bits	R	0x0000000D
CID1	0xE00F_FFF4	32 bits	R	0x00000010
CID2	0xE00F_FFF8	32 bits	R	0x00000005
CID3	0xE00F_FFFC	32 bits	R	0x000000B1

2.5.5 DBGREG Module

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

[Table 2.11](#) shows the DBGREG registers other than the CoreSight component registers.

Table 2.11 Non-CoreSight DBGREG registers

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

2.5.5.1 DBGSTR : Debug Status Register

Base address: DBG = 0x4001_B000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	CDBG PWRU PACK	CDBG PWRU PREQ	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

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

2.5.5.2 DBGSTOPCR : Debug Stop Control Register

Base address: DBG = 0x4001_B000

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	DBGS TOP_L VDBA T	DBGS TOP_L VD1	DBGS TOP_L VD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DBGS TOP_I WDT	DBGS TOP_I WDT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	DBGSTOP_IWDT	Mask bit for IWDT reset/interrupt 0: Enable IWDT reset/interrupt 1: Mask IWDT reset/interrupt and stop IWDT count when CPU is in OCD break mode	R/W
1	DBGSTOP_WDT	Mask bit for WDT reset/interrupt 0: Enable WDT reset/interrupt 1: Mask WDT reset/interrupt and stop WDT count when CPU is in OCD break mode	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	DBGSTOP_LVD0	Mask bit for LVD0 reset 0: Enable LVD0 reset 1: Mask LVD0 reset	R/W
17	DBGSTOP_LVD1	Mask bit for LVD1 reset/interrupt 0: Enable LVD1 reset/interrupt 1: Mask LVD1 reset/interrupt	R/W
18	DBGSTOP_LVDBAT	Mask bit for LVDBAT reset/interrupt 0: Enable LVDBAT reset/interrupt 1: Mask LVDBAT reset/interrupt	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

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

2.5.5.3 DBGREG CoreSight component registers

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

Table 2.12 shows the registers. See reference 4. in [section 2.8. References](#) for details of each register.

Table 2.12 DBGREG CoreSight component registers

Name	Address	Access size	R/W	Initial value
PID4	0x4001_BFD0	32 bits	R	0x00000004
PID5	0x4001_BFD4	32 bits	R	0x00000000
PID6	0x4001_BFD8	32 bits	R	0x00000000
PID7	0x4001_BFDC	32 bits	R	0x00000000
PID0	0x4001_BFE0	32 bits	R	0x00000005
PID1	0x4001_BFE4	32 bits	R	0x00000030
PID2	0x4001_BFE8	32 bits	R	0x0000001A
PID3	0x4001_BFEC	32 bits	R	0x00000000
CID0	0x4001_BFF0	32 bits	R	0x0000000D
CID1	0x4001_BFF4	32 bits	R	0x000000F0
CID2	0x4001_BFF8	32 bits	R	0x00000005
CID3	0x4001_BFFC	32 bits	R	0x000000B1

2.5.6 OCDREG Module

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

Table 2.13 lists the OCDREG registers.

Table 2.13 OCDREG registers list

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

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

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

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

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

Base address: CPU_OCD = 0x8000_0000

Offset address: 0x000 + 0x100 × n

Bit position: 31

0

Bit field:

IAUTHn: AID (32+32×n -1) to (32×n) bits

Value after reset: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

2.5.6.2 MCUSTAT : MCU Status Register

Base address: CPU_OCD = 0x8000_0000

Offset address: 0x400

Bit position:

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

Bit field:

—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit position:

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

Bit field:

—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUS TOPC LK	CPUS LEEP	AUTH

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1/0*1 1/0*1 0

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

Note 1. Depends on the MCU status.

2.5.6.3 MCUCTRL : MCU Control Register

Base address: CPU_OCD = 0x8000_0000

Offset address: 0x410

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

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

Bit	Symbol	Function	R/W
0	EDBGRQ	External Debug Request Writing 1 to the bit causes a CPU halt. When the EDBGRQ bit is set to 0 or the CPU is halted, the EDBGRQ bit is cleared. 0: Debug event not requested 1: Debug event requested	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	DBIRQ	Debug Interrupt Request Writing 1 to the bit wakes up the MCU from low power mode. The condition can be cleared by writing 0 to the DBIRQ bit. 0: Debug interrupt not requested 1: Debug interrupt requested	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set DBIRQ and EDBGRQ to the same value.

2.5.6.4 OCDREG CoreSight registers

OCDREG has the CoreSight registers defined in the Arm CoreSight architecture.

Table 2.14 shows the registers. See reference 4. in [section 2.8. References](#) for details of each register.

Table 2.14 OCDREG CoreSight registers

Name	Address	Access size	R/W	Initial value
PID4	0x8000_0FD0	32 bits	R	0x00000004
PID5	0x8000_0FD4	32 bits	R	0x00000000
PID6	0x8000_0FD8	32 bits	R	0x00000000
PID7	0x8000_0FDC	32 bits	R	0x00000000
PID0	0x8000_0FE0	32 bits	R	0x00000004
PID1	0x8000_0FE4	32 bits	R	0x00000030
PID2	0x8000_0FE8	32 bits	R	0x0000000A
PID3	0x8000_0FEC	32 bits	R	0x00000000
CID0	0x8000_0FF0	32 bits	R	0x0000000D
CID1	0x8000_0FF4	32 bits	R	0x000000F0
CID2	0x8000_0FF8	32 bits	R	0x00000005
CID3	0x8000_0FFC	32 bits	R	0x000000B1

2.6 SysTick Timer

The SysTick timer provides a simple 24-bit down counter. The reference clock for the timer can be selected as the CPU clock (ICLK) or SysTick timer clock (SYSTICCLK). See [section 9, Clock Generation Circuit](#) and reference 1. in [section 2.8. References](#) for details.

2.7 OCD Emulator Connection

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

[Figure 2.3](#) shows a block diagram of the authentication mechanism.

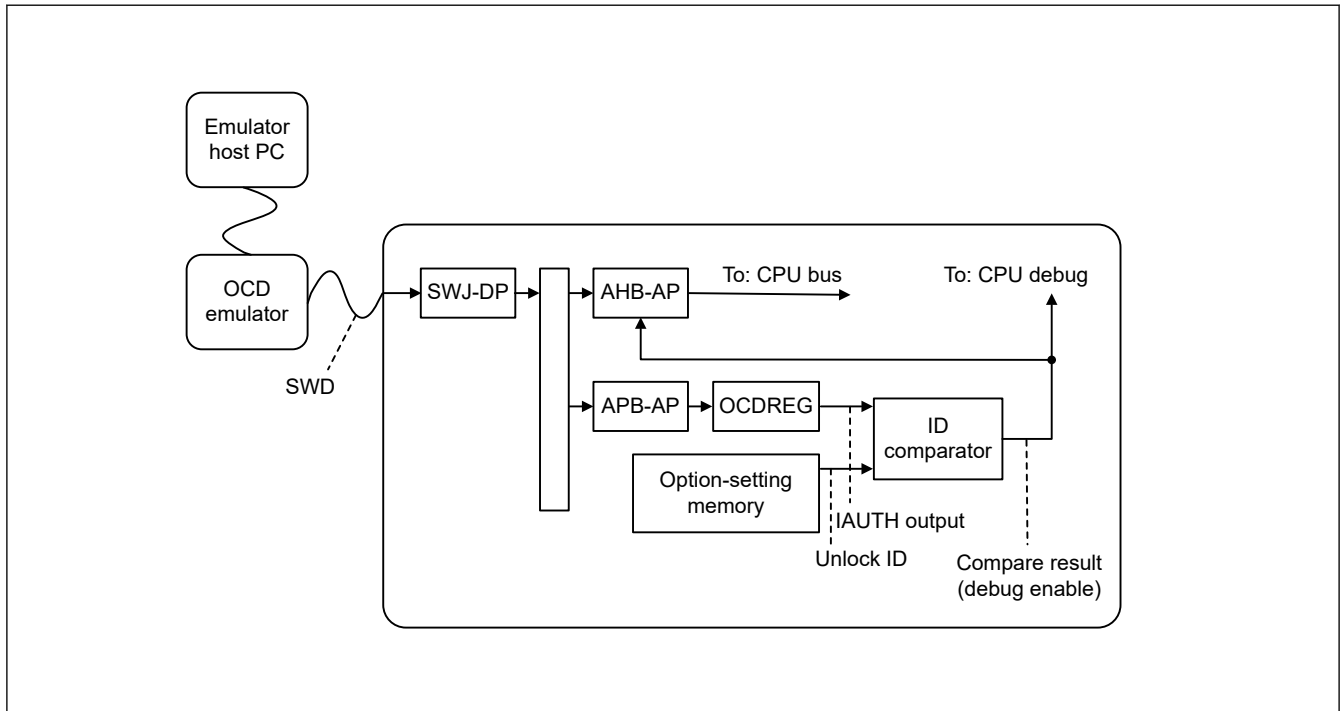


Figure 2.3 SWD Authentication mechanism block diagram

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

After the OCD emulator gets access permission, the OCD emulator must set the DBGEN bit in the System Control OCD Control Register (SYOCDCCR). In addition, the OCD emulator must clear the DBGEN bit before disconnecting. See [section 13.2.21. SYOCDCCR : System Control OCD Control Register](#).

2.7.1 Unlock ID Code

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

2.7.2 Restrictions on Connecting an OCD emulator

To start a SWD connection from an emulator, the MCU must be able to enter OCD mode. To do so, however, there are some restrictions depending on the current MCU status. [Table 2.15](#) lists in which mode and power consumption status the MCU can transition to OCD mode.

Since the MCU cannot transition to OCD mode while in EXFPWON and normal modes, MINPWON and normal modes, or VBB mode, change the MCU to OCD mode while in ALLPWON and normal modes, and then change it to EXFPWON and

normal modes, MINPWON and normal modes, or VBB mode in order to carry out debugging in EXFPWON and normal modes, MINPWON and normal modes, or VBB mode. For details, see [section 13, Power-Saving Functions](#).

Table 2.15 Status of the MCU That can Transition to OCD Mode

Current Operating Mode before Transitioning to OCD Mode			Transition to OCD Mode
Power Control Mode	Power Supply Mode	Low Power Consumption Modes	
Boost mode	—	Operating mode	Possible* ¹
		Sleep mode	Possible* ¹
Normal mode	ALLPWON	Operating mode	Possible* ¹
		Sleep mode	Possible* ¹
		Snooze mode	Impossible
		Software standby mode	Impossible
		Deep software standby mode	Impossible
	EXFPWON mode MINPWON mode	All modes	Impossible
VBB mode	ALLPWON mode EXFPWON mode MINPWON mode	All modes	Impossible
Mode transition period			Impossible

Note 1. After transition to the OCD mode, set 1 to the SYOCDCCR.DBGEN bit (on-chip debugger enable). For details on the SYOCDCCR.DBGEN bit, see [section 13.2.21. SYOCDCCR : System Control OCD Control Register](#).

2.7.2.1 Mode Transitions while in OCD Mode

Some restrictions apply to mode transitions while in OCD mode. [Table 2.16](#) lists availability of mode transitions between the power control modes.

Table 2.16 Availability of Mode Transitions between the Power Control Modes while in OCD Mode

Current Power Control Mode	Power Control Mode to Transition to	Availability of Mode Transitions between the Power Control Modes
Boost mode	Normal mode	Impossible
Normal mode	Boost mode	Impossible
	VBB mode	Possible* ¹⁺²
VBB mode	Normal mode	Possible* ¹

Note 1. Although power control mode transition between normal and VBB modes is possible, the status of the power in normal mode is maintained in order to continue debugging. Functions such as status flagging can be emulated.

Note 2. After transition to the OCD mode while in ALLPWON and normal modes, set the SYOCDCCR.DBGEN bit to 1 (on-chip debugger enable) before transition to the VBB mode. For details on the SYOCDCCR.DBGEN bit, see [section 13.2.21. SYOCDCCR : System Control OCD Control Register](#).

After transition to the OCD mode while in ALLPWON and normal modes, set the SYOCDCCR.DBGEN bit to 1 (on-chip debugger enable) before transition to EXFPWON or MINPWON mode. For details on the SYOCDCCR.DBGEN bit, see [section 13.2.21. SYOCDCCR : System Control OCD Control Register](#).

2.7.2.2 Entering Low Power Consumption Mode while in OCD Mode

The MCU can enter low power consumption mode even while it is in OCD mode.

After transition to the OCD mode, set the SYOCDCCR.DBGEN bit to 1 (on-chip debugger enable) before transition to low power consumption mode. For details on the SYOCDCCR.DBGEN bit, see [section 13.2.21. SYOCDCCR : System Control OCD Control Register](#).

If system bus access is required and the MCU is in software standby, snooze, or deep software standby mode, set the MCUCTRL.DBIRQ bit in OCDREG to 1 to wake the MCU up from the low power consumption mode. Simultaneously,

using the MCUCTRL.EDBGRQ bit in OCDREG, the emulator can wake up the MCU without starting CPU execution. [Table 2.17](#) lists availability of access to system bus while in OCD mode.

Table 2.17 Availability of Access to System Bus while in OCD Mode

Current mode	Current Low power mode	Access to System Bus
Boost mode	Operating mode	Possible
	Sleep mode	Possible
Normal mode	Operating mode	Possible
	Sleep mode	Possible
	Snooze mode	Impossible
	Software standby mode	Impossible
	Deep software standby mode	Impossible
VBB mode	Operating mode	Possible
	Sleep mode	Possible
	Snooze mode	Impossible
	Software standby mode	Impossible
	Deep software standby mode	Impossible
Mode transition period		Impossible

2.7.2.3 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.

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

2.7.2.4 Connecting sequence and SWD authentication

Because the OCD emulator is protected by the SWD authentication mechanism, the OCD might be required to input the ID code to the SWD authentication registers. The OSIS value in the option-setting memory determines whether the code is required. After negation of the RES# pin, a wait time is required before comparing the OSIS value at cold start. See [section 51.3.3. Reset Timing](#). The SWD authentication process is described in detail below.

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

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

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

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

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

(3) When the value in the OSIS register becomes “ALeRASE” in ASCII code

Data in the flash memory are deleted. For details, see [section 50, Flash Memory](#).

1. Set the ASCII code “ALeRASE” (0x414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFF) in the IAUTH0 to IAUTH3

registers.

2. Place the MCU in the reset state.
3. Wait until MCUSTAT.CPUSTOPCLK = 1 (deletion completed).
4. Reset the MCU then release it from the reset state so that it enters the OCD mode.
5. Confirm that all bits of the unlock ID code are 1 (the code is 0xFFFF_FFFF_FFFF_FFFF_FFFF_FFFF_FFFF_FFFF).

(4) When bits in the OSIS register are not all 1s

ID authentication is required and the emulator must write the 128-bit unlock ID code to the IAUTH0 to IAUTH3 registers in OCDREG before using AHB-AP.

1. Connect the emulator to the MCU through the SWD interface.
2. Set up the SWJ-DP to access the DAP bus. In the setup, the emulator must assert CDBGPWRUPREQ in the SWJ-DP control status register, then wait until CSDBGPWRUPACK in the same register is asserted.
3. Set up APB-AP to access OCDREG. APB-AP is connected to the DAP bus port 1.
4. Write the 128-bit unlock ID code to the IAUTH0 to IAUTH3 registers in OCDREG using APB-AP.
5. If the 128-bit unlock ID code matches the OSIS register value, AHB-AP is authorized to issue an AHB transaction. The authentication result can be confirmed in the AUTH bit in the MCUSTAT register or the DbgStatus bit in the AHB-AP control status word register.
 - When the DbgStatus bit is 1, the 128-bit ID code is a match with the OSIS value. AHB transfers are permitted.
 - When the DbgStatus bit is 0, the 128-bit ID code is not a match with the OSIS value. AHB transfers are not permitted.
6. Set up AHB-AP to access the system address space. AHB-AP is connected to the DAP bus port 0.
7. Start accessing the CPU debug resources using AHB-AP.

2.8 References

1. *ARM®v6-M Architecture Reference Manual* (ARM DDI 0419E)
2. *Cortex™-M0+ Technical Reference Manual* (ARM DDI 0484C)
3. *Cortex™-M0+ Device Generic User Guide* (ARM DUI 0662B)
4. *ARM® CoreSight™ SoC-400 Technical Reference Manual* (ARM DDI 0480G)
5. *ARM® CoreSight™ Architecture Specification* (ARM IHI 0029E)
6. *CoreSight™ MTB-M0+ Technical Reference Manual* (ARM DDI 0486B)

3. Startup Modes

3.1 Types and Selection of Startup Mode

Table 3.1 shows the startup modes selected by the levels on the mode setting pins (MD and EHMD). For details on each of the startup modes, see [section 3.2. Details of Startup Modes](#).

Table 3.1 Types of Startup Mode Selected by the Levels on the Startup Mode Setting Pin and Energy Harvesting Mode Setting Pin

Mode-setting pin		Startup mode
MD	EHMD	
1	1	Energy harvesting startup
	0	Normal startup
0	—	SCI boot mode

3.2 Details of Startup Modes

3.2.1 Normal Startup and Energy Harvesting Startup Modes

In normal startup and energy harvesting startup modes, all input and output pins are available for use as input or output ports, inputs or outputs for peripheral functions, or as interrupt inputs. When a reset is released while the MD pin is high, the chip starts in singlechip mode and the on-chip flash memory is enabled. The EHMD pin can be used to select normal startup or energy harvesting startup. For details, see [section 3.3.2. Power-up Sequence](#).

3.2.2 Serial Programming Mode

3.2.2.1 SCI Boot Mode

In this mode, the on-chip flash memory programming routine (SCI boot program), stored in the boot area within the MCU, is used. The code flash memory can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (UART) SCI. For details, see [section 50, Flash Memory](#). The MCU starts in SCI boot mode if the MD pin is held low on release from the reset state.

3.2.3 On-chip Debug Mode

In this mode, the MCU can be externally controlled by connecting an external emulator or flash memory programmer through the SWD interface.

3.3 Startup Modes Transitions

3.3.1 Startup Mode Determined by the Mode Setting Pins

Figure 3.1 shows startup mode transitions determined by the settings of the MD pin and the EHMD pin.

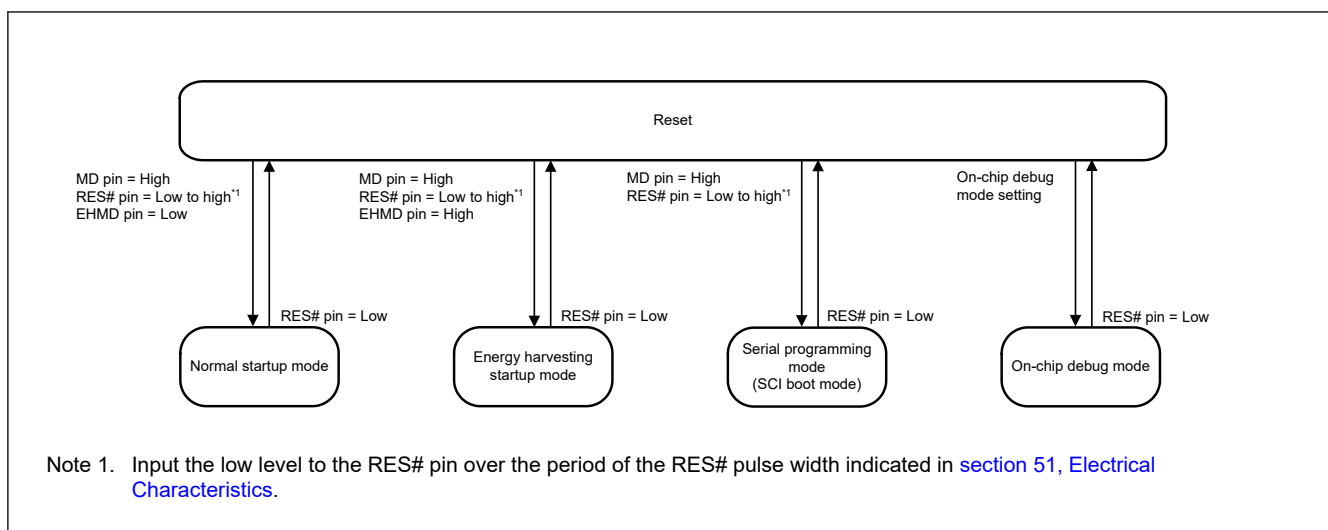


Figure 3.1 Startup Mode Determined by Levels on the Startup Mode Setting Pin and Energy Harvesting Mode Setting Pin

3.3.2 Power-up Sequence

The normal startup or energy harvesting startup mode is selected by the state of the EHMD pin on release from the reset state as shown in [Table 3.2](#).

Table 3.2 Types of startup mode selected by the state of the EHMD pin

EHMD pin state	Startup mode
Low	Normal startup mode
High	Energy harvesting startup mode

The procedure for using the low leakage current mode as one of power control modes depends on the selected startup mode.

In the normal startup mode, the MCU starts with the back bias voltage control (VBBC) circuit disabled. Using the low leakage current mode after normal startup requires waiting for completion of the startup setting and initial setup of the VBBC circuit after release from the internal reset state. The initial setup of the VBBC circuit is the operation of charging an external capacitor connected between VBP and VBN. Setting the back bias voltage control (VBBC) enable bit (VBBCR.VBBEN) to 1 starts this initial setup. When the initial setup is completed, the back bias voltage control (VBBC) initial setup completion flag (VBBST.VBBSTUP) is set to 1. Transition to the low leakage current mode becomes possible when VBBST.VBBSTUP flag is 1.

In the energy harvesting startup mode, the initial setup of the VBBC circuit starts and is completed during the internal reset period. Consequently, the MCU can enter the low leakage current mode immediately, since the VBBCR.VBBEN bit and the VBBST.VBBSTUP flag will be 1 at the time of release from the internal reset state. Although the internal reset period for the energy harvesting startup mode is longer than that for the normal startup mode, the amount of current drawn is reduced during the initial setup of the VBBC circuit.

For details on the low leakage current mode, see [section 13, Power-Saving Functions](#).

4. Address Space

4.1 Address Space

The MCU supports a 4-GB linear address space ranging from 0x0000_0000 to 0xFFFF_FFFF that can contain both program and data. [Figure 4.1](#) shows the memory map.

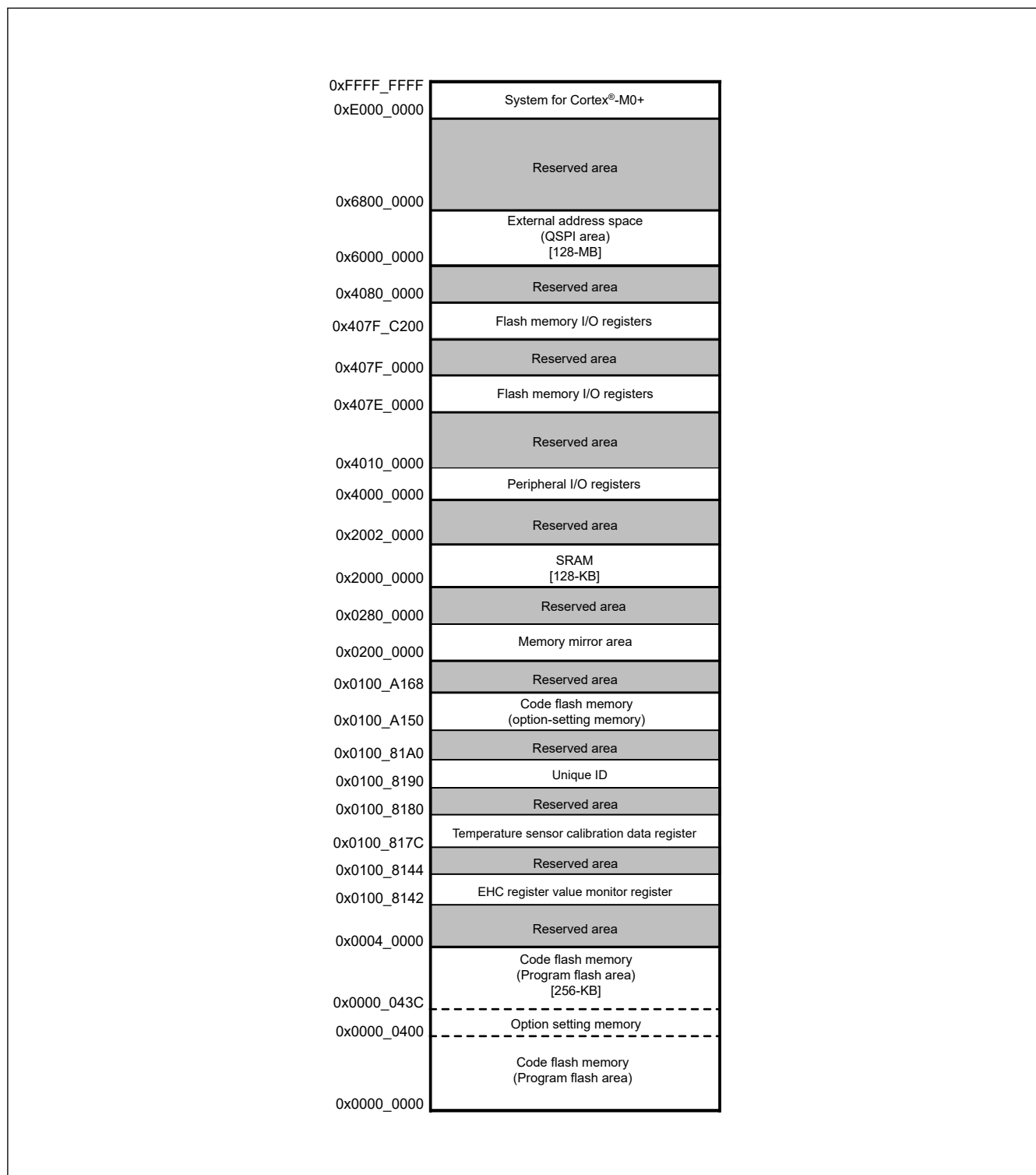


Figure 4.1 Memory map

4.2 External Address Space

The QSPI area is provided as external address space. The QSPI area is divided into two areas, the QSPI I/O registers, and external SPI device space.

Figure 4.2 shows the address ranges associated with the QSPI area.

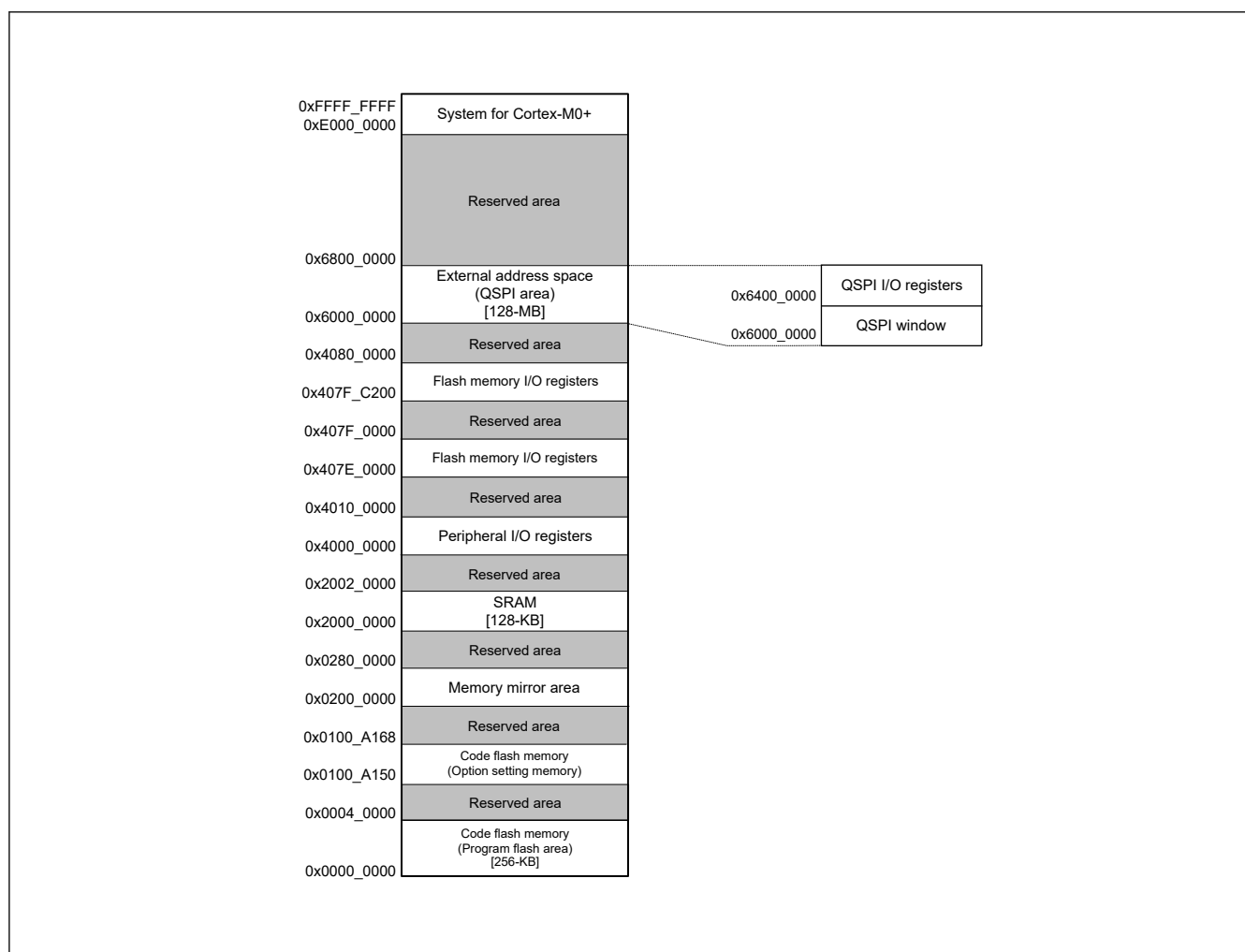


Figure 4.2 External address space

5. I/O Registers

This section describes I/O register addresses and access cycles by function.

5.1 Address Information

Table 5.1 lists the address information for I/O registers in this product.

Table 5.1 I/O Register Address (1 of 2)

Start Address	End Address	Base address symbol	Description
0x4000_0000	0x4000_4FFF	RMPU, MMF, BUS	Renesas memory protection unit, Memory mirror function, Bus control
0x4000_5000	0x4000_5FFF	DMAC0-3, DMA, DTC	DMA controller, Data transfer controller
0x4000_6000	0x4000_6FFF	ICU	Interrupt controller
0x4001_B000	0x4001_BFFF	DBG	Debug function
0x4001_E000	0x4001_E17F	SYSC	System control
0x4001_E180	0x4001_E3FF	EHC	Energy harvesting controller
0x4001_E400	0x4001_EFFF	SYSC	System control
0x4004_0000	0x4004_011F	PORT0-8	Port control register
0x4004_0800	0x4004_0FFF	PFS	Port mn pin function select register
0x4004_1000	0x4004_10FF	ELC	Event link controller
0x4004_1240	0x4004_133F	SCI2-5, SCI9	Serial communication interface
0x4004_2000	0x4004_21FF	POE	Port Output Enable Module
0x4004_4000	0x4004_40FF	RTC	Realtime clock
0x4004_4200	0x4004_42FF	WDT	Watchdog timer
0x4004_4400	0x4004_44FF	IWDT	Independent watchdog timer
0x4004_4600	0x4004_46FF	CAC	Clock frequency accuracy measurement circuit
0x4004_7000	0x4004_70FF	MSTP	Module stop control registers B, C, D
0x4005_2000	0x4005_207F	TMR0, TMR1	8-bit timer
0x4005_3000	0x4005_31FF	IIC0, IIC1	I ² C bus interface
0x4005_4100	0x4005_41FF	DOC	Data operation circuit
0x4005_5000	0x4005_55FF	GPT320, GPT321, GPT162-5	General PWM timer (32 bits, 16 bits)
0x4005_5FF0	0x4005_5FFF	GPT_OPS	Output phase switch control
0x4005_C000	0x4005_C1FF	ADC140	14-bit A/D converter
0x4005_D000	0x4005_D0FF	TSN	Temperature sensor
0x4007_0000	0x4007_003F	SCI0, SCI1	Serial communication interface
0x4007_0200	0x4007_03FF	MLCD	MIP LCD controller
0x4007_0800	0x4007_0DFF	GDT	2D graphics data conversion circuit
0x4007_0F00	0x4007_0FFF	IRDA	Infrared communication
0x4007_2000	0x4007_21FF	SPI0, SPI1	Serial peripheral interface (128 bits, 32 bits)
0x4007_4000	0x4007_40FF	CRC	Cyclic Redundancy Check Calculator
0x4008_0000	0x4008_00FF	KINT	Key interrupt function
0x4008_0400	0x4008_04FF	CCC	Clock correction circuit
0x4008_4000	0x4008_41FF	AGT0, AGT1	Low power asynchronous general-purpose 16-bit timer

Table 5.1 I/O Register Address (2 of 2)

Start Address	End Address	Base address symbol	Description
0x4008_4200	0x4008_43FF	AGTW0, AGTW1	Low power asynchronous general-purpose 32-bit timer
0x4008_4400	0x4008_47FF	LST	Low-speed clock timer
0x4008_4480	0x4008_44FF	WUPT	Wakeup timer
0x4008_4500	0x4008_457F	DIL	Data Inversion and Logic operation
0x4008_4600	0x4008_46FF	DIV	Divider
0x4008_6A00	0x4008_6AFF	VREF	Reference voltage generation circuit
0x407F_E000	0x407F_E0FF	FACI	Flash Application Command Interface
0x6400_0000	0x67FF_FFFF	QSPI	Quad-serial peripheral interface

5.2 Access Cycle

Table 5.2 lists the access cycle information of the I/O registers in the MCU. The following statements apply to Table 5.2:

- Registers are grouped by corresponding modules.
- The number of access cycles indicates the number of cycles based on the specified reference clock.
- In the I/O register area, reserved addresses that are not allocated to registers must not be accessed. If access is attempted, further operation cannot be guaranteed.
- The number of access cycles for I/O registers 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 between ICLK and PCLK. “PCLK” refers to both PCLKA and PCLKB. For the internal peripheral bus, see [section 17, Buses](#).
- 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, 1 cycle of PCLK is added to the divided clock synchronization cycles.
- The numbers of cycles are applicable when access by the CPU does not conflict with bus access by another bus master (DMAC or DTC).

Table 5.2 I/O Register Access Cycle (1 of 2)

Base Address Symbol	Start Address	End Address	ICLK = PCLK		ICLK > PCLK ^{*1}		Cycle Unit	Related Function
			Read	Write	Read	Write		
RMPU, MMF, BUS, DMAC0-3, DMA, DTC, ICU, DBG	0x4000_0000	0x4000_6FFF	3	3	—	—	ICLK	CPU, Renesas memory protection unit, Memory mirror function, Bus control, DMA controller, Data transfer controller, Interrupt controller, Debug function
	0x4001_A000	0x4001_BFFF			—	—		
SYSC ^{*4} , EHC	0x4001_E000	0x4001_E3FF	3	3	—	—	ICLK	Power-saving function, Resets, Clock generation function, Register write protection function, Low voltage detection, Energy harvesting control circuit
	0x4001_E400	0x4001_E412	5	5	—	—		
	0x4001_E413	0x4001_E413	3	3	—	—		
	0x4001_E414	0x4001_E420	5	5	—	—		
	0x4001_E421	0x4001_E421	3	3	—	—		
	0x4001_E422	0x4001_E4E0	5	5	—	—		
	0x4001_E4E1	0x4001_E4E1	3	3	—	—		
	0x4001_E4E2	0x4001_E4FF	5	5	—	—		
	0x4001_E500	0x4001_EFFF	3	3	—	—		

Table 5.2 I/O Register Access Cycle (2 of 2)

Base Address Symbol	Start Address	End Address	ICLK = PCLK		ICLK > PCLK ^{*1}		Cycle Unit	Related Function
			Read	Write	Read	Write		
PORT0 to PORT8, PFS, ELC	0x4004_0000	0x4004_10FF	3	3	2-3	2-3	PCLKB	I/O ports, Event link controller
SCI2 to SCI5, SCI9	0x4004_1240	0x4004_133F	3	3	2-3 ^{*2}	2-3 ^{*2}	PCLKB	Serial communications interface
POE	0x4004_2000	0x4004_21FF	3	3	2-3	2-3	PCLKB	Port output enable for GPT
RTC, WDT, IWD, CAC, MSTP	0x4004_4000	0x4004_70FF	3	3	2-3	2-3	PCLKB	Realtime clock, Watchdog timer, Independent watchdog timer, Clock frequency accuracy measurement circuit, Module stop control
TMR0, TMR1	0x4005_2000	0x4005_207F	3	3	2-3	2-3	PCLKB	8-bit timer
IIC0, IIC1	0x4005_3000	0x4005_31FF	3	3	2-3	2-3	PCLKB	I ² C bus interface
DOC	0x4005_4100	0x4005_41FF	3	3	2-3	2-3	PCLKB	Data operation circuit
GPT0 to GPT5 ^{*5} , GPT_OPS	0x4005_5000	0x4005_5FFF	6	4	5-6	3-4	PCLKB	General PWM timer
ADC140	0x4005_C000	0x4005_C1FF	3	3	2-3	2-3	PCLKB	14-bit A/D converter
TSN	0x4005_D000	0x4005_D0FF	3	3	2-3	2-3	PCLKB	Temperature sensor
SCI0, SCI1	0x4007_0000	0x4007_003F	3	3	—	—	PCLKA	Serial communications interface
MLCD, GDT	0x4007_0200	0x4007_0DFF	3	3	—	—	PCLKA	MIP LCD controller, 2D graphics data conversion circuit
IRDA	0x4007_0F00	0x4007_0FFF	3	3	—	—	PCLKA	Infrared communication
SPI0, SPI1	0x4007_2000	0x4007_21FF	3	3	—	—	PCLKA	Serial peripheral interface
CRC	0x4007_4000	0x4007_40FF	3	3	—	—	PCLKA	CRC calculator
KINT	0x4008_0000	0x4008_00FF	3	3	2-3	2-3	PCLKB	Key interrupt function
CCC	0x4008_0400	0x4008_04FF	4	4	3-4	3-4	PCLKB	Clock correction circuit
AGT0, AGT1, AGTW0, AGTW1	0x4008_4000	0x4008_41FF	4	4	3-4	3-4	PCLKB	Low power asynchronous general-purpose timer
LST	0x4008_4400	0x4008_447F	4	4	3-4	3-4	PCLKB	Low-speed clock timer
WUPT	0x4008_4480	0x4008_44FF	3	3	2-3	2-3	PCLKB	Wakeup timer
DIL	0x4008_4500	0x4008_457F	4	4	3-4	3-4	PCLKB	Data inversion and logic operation
DIV	0x4008_4680	0x4008_46FF	4	4	3-4	3-4	PCLKB	Divider
VREF	0x4008_6A80	0x4008_6AFF	4	4	3-4	3-4	PCLKB	Reference voltage generation circuit
QSPI	0x6400_0000	0x67FF_FFFF	4 ^{*3}	14 ^{*3}	—	—	PCLKA	Quad-serial peripheral interface

Note 1. If the number of PCLK cycles is a non-integer (for example 1.5), the minimum value is rounded down to an integer, and the maximum value is rounded off to an integer. 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 5.2. When accessing an 8-bit register (FTDRH, FTDRH, FRDRH, and FRDRH), the access cycles are as shown in Table 5.2.

Note 3. The access cycles depend on the QSPI bus cycles.

Note 4. These values indicate the minimum numbers of cycles for access by the CPU. They do not include the cycles required for changes in the source of the ICLK clock and frequency after changes to the SCKSCR and SCKDIVCR registers.

Note 5. GPT0 to GPT5 indicates GPT320, GPT321, GPT162, GPT163, GPT164, and GPT165.

6. Resets

6.1 Overview

The MCU provides 13 resets. The resets are classified into two types: System resets that initialize the MCU and power shutdown reset that does not initialize the MCU.

[Table 6.1](#) and [Table 6.2](#) list the reset names and sources.

Table 6.1 System Resets

Reset name	Source
RES# pin reset	Voltage input to the RES# pin is driven low
Power-on reset	Rising of the voltage on the VCC pin (monitored voltage: V_{POR}) ^{*1} , or falling of the voltage on the VCC pin to V_{POR}
Independent watchdog timer reset	The independent watchdog timer underflows, or a refresh error occurs.
Watchdog timer reset	The watchdog timer underflows, or a refresh error occurs.
Voltage monitor 0 reset	Falling of the voltage on the VCC pin (monitored voltage: V_{det0}) ^{*1}
Voltage monitor 1 reset	Falling of the voltage on the VCC pin (monitored voltage: V_{det1}) ^{*1}
Voltage monitor BAT reset	Falling of the voltage on the VBAT_EHC pin (monitored voltage: V_{detBAT}) ^{*1}
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)
Deep software standby reset ^{*2}	Canceling of Deep Software Standby mode by an interrupt

Note 1. For details on the voltages to be monitored (V_{POR} , V_{det0} , V_{det1} , and V_{detBAT}), see [section 8, Low Voltage Detection \(LVD\)](#) and [section 51, Electrical Characteristics](#).

Note 2. For details on each mode, see [section 13, Power-Saving Functions](#).

Table 6.2 Power Shutdown Reset

Reset name	Source
MINPWON mode reset ^{*1}	Transition from minimum power supply mode (MINPWON) to flash-excluded power supply mode (EXFPWON)

Note 1. For details on each mode, see [section 13, Power-Saving Functions](#).

The internal state and pins are initialized by a reset. [Table 6.3](#) and [Table 6.4](#) list the targets initialized by resets.

Table 6.3 Reset detect flags initialized by each reset source

Target to be initialized	RES# pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	Voltage monitor 1 reset, voltage monitor BAT reset	Software reset	Bus master MPU error reset	Bus slave MPU error reset	Stack pointer error reset	MINPWON mode reset	Deep software standby reset
Power-On Reset Detect Flag (RSTSR0.PORF)	✓	—	—	—	—	—	—	—	—	—	—	—
Voltage Monitor 0 Reset Detect Flag (RSTSR0.LVD0RF)	✓	✓	—	—	—	—	—	—	—	—	—	—
Independent Watchdog Timer Reset Detect Flag (RSTSR1.IWDTRF)	✓	✓	✓	—	—	—	—	—	—	—	—	✓
Watchdog Timer Reset Detect Flag (RSTSR1.WDTRF)	✓	✓	✓	—	—	—	—	—	—	—	—	✓
Voltage Monitor 1 Reset Detect Flag (RSTSR0.LVD1RF)	✓	✓	✓	—	—	—	—	—	—	—	—	—
Voltage Monitor BAT Reset Detect Flag (RSTSR0.LVDBATRF)	✓	✓	✓	—	—	—	—	—	—	—	—	—
Bus Slave MPU Error Reset Detect Flag (RSTSR1.BUSSRF)	✓	✓	✓	—	—	—	—	—	—	—	—	✓
Bus Master MPU Error Reset Detect Flag (RSTSR1.BUSMRF)	✓	✓	✓	—	—	—	—	—	—	—	—	✓
Stack Pointer Error Reset Detect Flag (RSTSR1.SPERF)	✓	✓	✓	—	—	—	—	—	—	—	—	✓
Software Reset Detect Flag (RSTSR1.SWRF)	✓	✓	✓	—	—	—	—	—	—	—	—	✓
Deep Software Standby Reset Detect Flag (RSTSR0.DPSRSTF)	✓	✓	✓	—	—	—	—	—	—	—	—	—
Cold/Warm Start Determination Flag (RSTSR2.CWSF)	—	✓	—	—	—	—	—	—	—	—	—	—

Note: ✓ : Initialized to 0
 — : Not initialized

Table 6.4 Module-related registers initialized by each reset source (1 of 2)

Target to be initialized		RES# pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	Voltage monitor 1 reset, Voltage monitor BAT reset	Software reset	Bus master MPU error reset	Bus slave MPU error reset	Stack pointer error reset	MINPWON mode reset	Deep software standby reset
Registers related to the SOSC	SOSCCR	—	✓	—	—	—	—	—	—	—	—	—	—
	SOMCR	—	✓	—	—	—	—	—	—	—	—	—	—
Register related to the RTC*1		—	✓	—	—	—	—	—	—	—	—	—	—
Registers related to the CCC	ADJUSTR.ADJUST[8:0]	—	✓	—	—	—	—	—	—	—	—	—	—
	R128CNT	—	✓	—	—	—	—	—	—	—	—	—	—
	R128CTRL.CADJUSCEN, R128CTRL.ADJUSTEN, R128CTRL.COPSEL, R128CTRL.CRTCOS, R128CTRL.CPIE, R128CTRL.CPES_EX, R128CTRL.CPES[1:0], R128CTRL.CRTCOS, R128CTRL.PF128HZ, R128CTRL.PF64HZ, R128CTRL.PF16HZ, R128CTRL.PF4HZ, R128CTRL.PF2HZ, R128CTRL.PF1HZ, R128CTRL.PFEN,	—	✓	—	—	—	—	—	—	—	—	—	—
	R128CTRL.CRESET, R128CTRL.CCIE, R128CTRL.CEIE, R128CTRL.PFWR0ST	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	✓
Register related to the WUPT		✓	✓	✓	—	—	—	—	—	—	—	—	—
Registers related to the voltage monitor function 1	LVD1CR0, LVCMPCR.LVD1E, LVDLVL.LVD1LVL	✓	✓	✓	✓	✓	—	—	—	—	—	—	—
	LVD1CR1, LVD1SR	✓	✓	✓	✓	✓	—	—	—	—	—	—	✓
Registers related to the voltage monitor function BAT	LVDBATCR0, LVCMPCR.LVDBATE, LVDLVL.LVDBATLVL	✓	✓	✓	✓	✓	—	—	—	—	—	—	—
	LVDBATCR1, LVDBATSR	✓	✓	✓	✓	✓	—	—	—	—	—	—	✓
Registers related to the bus	BUSnERRADD, BUSnERRSTAT	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	✓
Registers related to the MPU	MSPMPUCTL.ERROR, PSPMPUCTL.ERROR	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	✓
Pin state		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	*2
Registers related to reducing power consumption	PWSTCR, VOCCR, DPSBYCR, DPSIER0, DPSIER1, DPSIFR0, DPSIFR1, DPSIEGR0, DPSIEGR1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
	SYOCDRCR	—	✓	—	—	—	—	—	—	—	—	—	—
	LDOCR	✓	✓	✓	—	—	—	—	—	—	—	—	—
	VBBCR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	✓

Table 6.4 Module-related registers initialized by each reset source (2 of 2)

Target to be initialized		RES# pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	Voltage monitor 1 reset, Voltage monitor BAT reset	Software reset	Bus master MPU error reset	Bus slave MPU error reset	Stack pointer error reset	MINPWON mode reset	Deep software standby reset
Registers related to the flash memory	FWEPROR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Arm debug module	DBGSTR, DBGSTOPCR, IAUTHn, MCUSTAT, MCUCTRL	—	✓	—	—	—	—	—	—	—	—	—	—
CPU, internal state, and registers other than the above	ISO1 area functions*3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	✓
	ISO2 area functions*3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Registers related to the EHC*4	EHCCR0	✓	✓	—	—	—	—	—	—	—	—	—	—
	EHCCR1	—	✓	—	—	—	—	—	—	—	—	—	—

Note: ✓ : Initialized
 — : Not initialized

Note 1. The RTC has a software reset. For details on the target bits, see [section 28, Realtime Clock \(RTC\)](#).

Note 2. Depends on the setting of DPSBYCR.IOKEEP.

Note 3. For details on the power source area for each function, see [section 13, Power-Saving Functions](#).

Note 4. The EHC has a software reset. See [section 51.9. EHC Characteristics](#) for details on the target bits.

The occurrence of the RES# pin reset or the voltage monitor 0, voltage monitor 1, or voltage monitor BAT reset during writing to the registers listed below may destroy the values set in the SOSCCR, SOMCR, RSTSR0, RSTSR2, SYOCDCR, LDOCR, LVCMPER, LVDLVL, LVD1CR0, and LVDBATCR0 registers. However, values set in the registers other than RSTSR0 and RSTSR2 will not be destroyed as long as the write protection function is enabled in the PRCR register.

DPSBYCR, DPSWCR, DPSIER0, DPSIER1, DPSIFR0, DPSIFR1, DPSIEGR0, DPSIEGR1, SYOCDCR, RSTSR0, RSTSR2, FWEPROR, LVCMPER, LVDLVL, LVDTR, LVD1CR0, LVDBAT, CR0, PWSTCR, LDOCR, VOICR, SOSCCR, SOMCR, VBBCR

When a reset is canceled, reset exception handling starts. [Table 6.5](#) lists the pin related to the reset function.

Table 6.5 Pin related to reset

Pin name	I/O	Function
RES#	Input	Reset pin

6.2 Register Descriptions

6.2.1 RSTSR0 : Reset Status Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x410

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DPSR STF	—	—	—	LVDB ATRF	LVD1R F	LVD0R F	PORF

Value after reset: x¹ 0 0 0 x¹ x¹ x¹ x¹

Bit	Symbol	Function	R/W
0	PORF	Power-On Reset Detect Flag 0: Power-on reset not detected 1: Power-on reset detected	R/W ²
1	LVD0RF	Voltage Monitor 0 Reset Detect Flag 0: Voltage monitor 0 reset not detected 1: Voltage monitor 0 reset detected	R/W ²
2	LVD1RF	Voltage Monitor 1 Reset Detect Flag 0: Voltage monitor 1 reset not detected 1: Voltage monitor 1 reset detected	R/W ²
3	LVDBATRF	Voltage Monitor BAT Reset Detect Flag 0: Voltage monitor BAT reset undetected 1: Voltage monitor BAT reset detected	R/W ²
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	DPSRSTF	Deep Software Standby Reset Flag 0: Deep software standby mode cancellation not requested by an interrupt. 1: Deep software standby mode cancellation requested by an interrupt.	R/W ²

Note 1. The value after reset depends on the reset source.

Note 2. The register is cleared when a reset source listed in [Table 6.3](#) occurs or when 0 is written to clear a flag. Bits other than the flag that is cleared should be set to 1.

PORF flag (Power-On Reset Detect Flag)

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

[Setting condition]

- When a power-on reset occurs.

[Clearing conditions]

- When a reset listed in [Table 6.3](#) occurs
- When 0 is written to the flag.

LVD0RF flag (Voltage Monitor 0 Reset Detect Flag)

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

[Setting condition]

- When a voltage monitor 0 reset occurs.

[Clearing conditions]

- When a reset listed in [Table 6.3](#) occurs
- When 0 is written to the flag.

LVD1RF flag (Voltage Monitor 1 Reset Detect Flag)

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

[Setting condition]

- When a voltage monitor 1 reset occurs.

[Clearing conditions]

- When a reset listed in [Table 6.3](#) occurs
- When 0 is written to the flag.

LVDBATRF flag (Voltage Monitor BAT Reset Detect Flag)

The LVDBATRF flag indicates that VBAT_EHC voltage becomes V_{detBAT} or less.

[Setting condition]

- When a voltage monitor BAT reset occurs.

[Clearing conditions]

- When a reset listed in [Table 6.3](#) occurs
- When 0 is written to the flag.

DPSRSTF flag (Deep Software Standby Reset Flag)

The DPSRSTF flag indicates that deep software standby mode has been canceled by an external or internal interrupt and that an internal reset (deep software standby reset) occurred when the exception from Deep Software Standby Mode occur.

[Setting condition]

- When deep software standby mode is cancelled by an external or an internal interrupt. For details, see [section 13, Power-Saving Functions](#).

[Clearing conditions]

- When a reset listed in [Table 6.3](#) occurs.
- When 0 is written to the flag.

6.2.2 RSTSR1 : Reset Status Register 1

Base address: SYSC = 0x4001_E000

Offset address: 0x0C0

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

Bit	Symbol	Function	R/W
0	IWDTRF	Independent Watchdog Timer Reset Detect Flag 0: Independent watchdog timer reset not detected 1: Independent watchdog timer reset detected	R/(W) ^{*2}
1	WDTRF	Watchdog Timer Reset Detect Flag 0: Watchdog timer reset not detected 1: Watchdog timer reset detected	R/(W) ^{*2}
2	SWRF	Software Reset Detect Flag 0: Software reset not detected 1: Software reset detected	R/(W) ^{*2}
9:3	—	These bits are read as 0. The write value should be 0.	R/W
10	BUSSRF	Bus Slave MPU Error Reset Detect Flag 0: Bus slave MPU error reset not detected 1: Bus slave MPU error reset detected	R/(W) ^{*2}
11	BUSMRF	Bus Master MPU Error Reset Detect Flag 0: Bus master MPU error reset not detected 1: Bus master MPU error reset detected	R/(W) ^{*2}
12	SPERF	CPU Stack Pointer Error Reset Detect Flag 0: CPU stack pointer error reset not detected 1: CPU stack pointer error reset detected	R/(W) ^{*2}
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source.

Note 2. The register is cleared when a reset source listed in [Table 6.3](#) occurs or when 0 is written to clear a flag. Bits other than the flag that is cleared should be set to 1.

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 6.3](#) occurs
- When 0 is written to the flag.

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 6.3](#) occurs
- When 0 is written to the flag.

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 6.3](#) occurs
- When 0 is written to the flag.

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 6.3](#) occurs
- When 0 is written to the flag.

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 6.3](#) occurs
- When 0 is written to the flag.

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 6.3](#) occurs
- When 0 is written to the flag.

6.2.3 RSTSR2 : Reset Status Register 2

Base address: SYSC = 0x4001_E000

Offset address: 0x411

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

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

Note 1. The value after reset depends on the reset source.

Note 2. Only 1 can be written to set the flag.

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

CWSF flag (Cold/Warm Start Determination Flag)

The CWSF flag indicates the type of reset processing, either cold start or warm start. 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 to this flag.

[Clearing condition]

- When a reset listed in [Table 6.3](#) 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 51, Electrical Characteristics](#).

6.3.2 Power-On Reset

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

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

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

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

Figure 6.1 shows example of operations during a power-on reset.

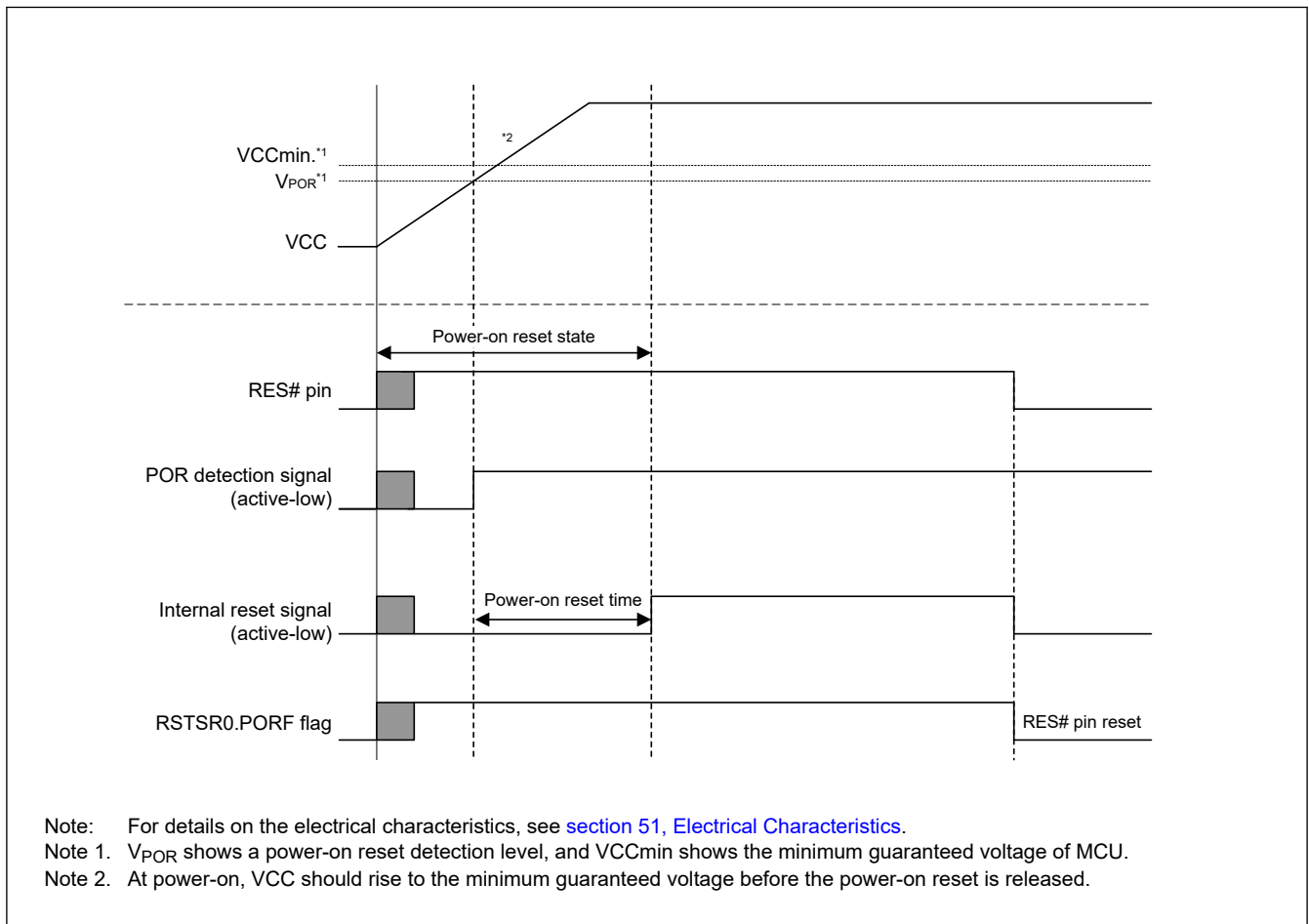


Figure 6.1 Example of operations during a power-on reset

6.3.3 Voltage Monitor Reset

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

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

Similarly, when the Voltage Monitor BAT Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or an interrupt by the voltage detection circuit) and the Voltage Monitor BAT 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 BAT Circuit Control Register 0 (LVDBATCR0), the RSTSR0.LVDBATRF flag is set to 1 and the voltage detection circuit generates a voltage monitor BAT reset if VBAT_EHC falls to or below V_{detBAT} .

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

above V_{det1} . When the LVD1CR0.RN bit is 1 and VCC falls to or below V_{det1} , the CPU is released from the internal reset state and starts reset exception handling when the LVD1 reset time (t_{LVD1}) elapses.

The timing for release from the voltage monitor BAT reset state is similarly selectable with the Voltage Monitor BAT Reset Negate Select bit (RN) in LVDBATCR0. Voltage detection levels V_{det1} and V_{detBAT} can be changed by making new settings in the Voltage Detection Level Select Register (LVDLVLR).

Figure 6.2 and Figure 6.3 show an example of operation during a voltage monitor 0/1 reset and a voltage monitor BAT reset. For details on the voltage monitor 0/1 reset and voltage monitor BAT reset, see section 8, Low Voltage Detection (LVD).

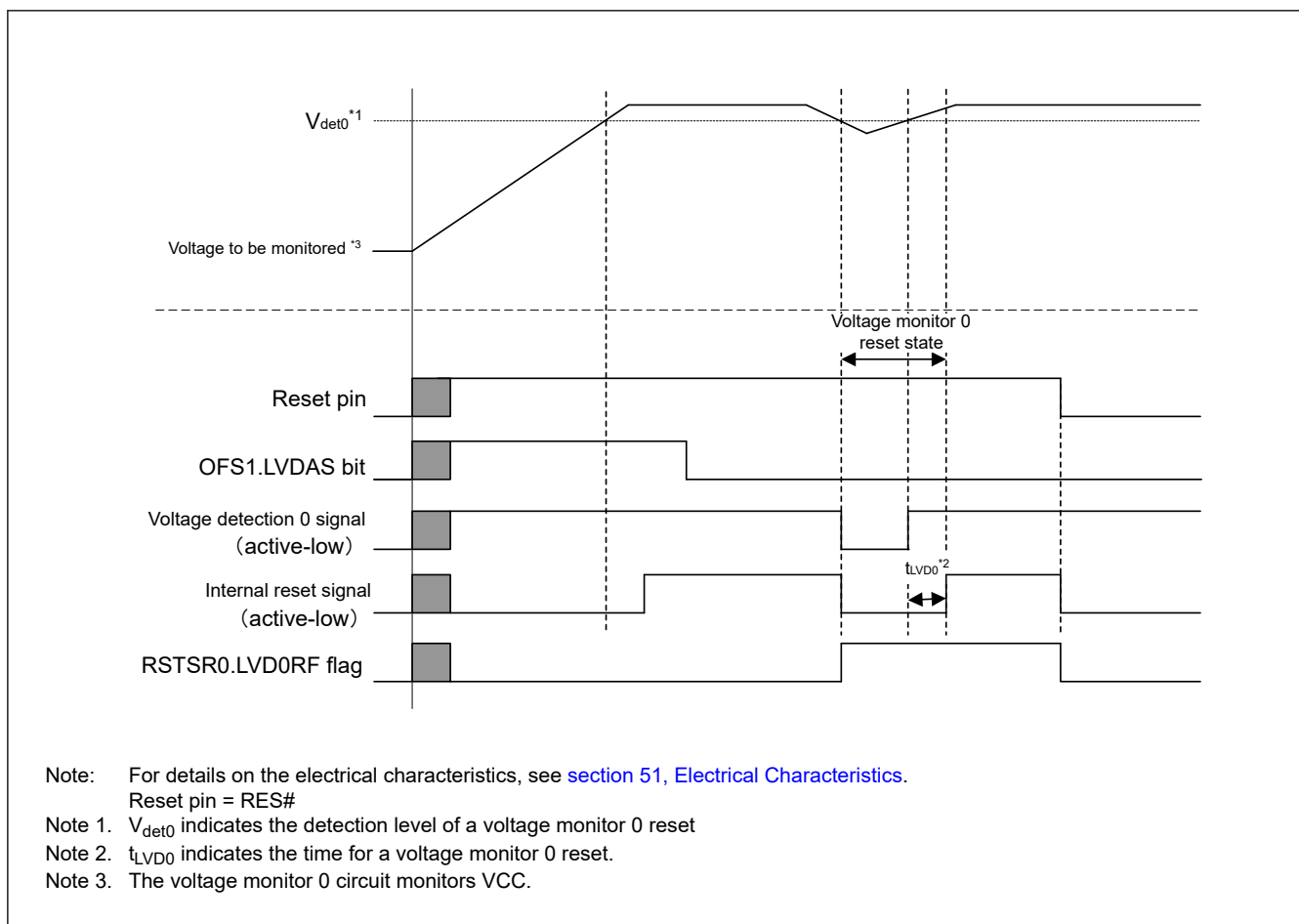


Figure 6.2 Example of operations during voltage monitor 0 resets

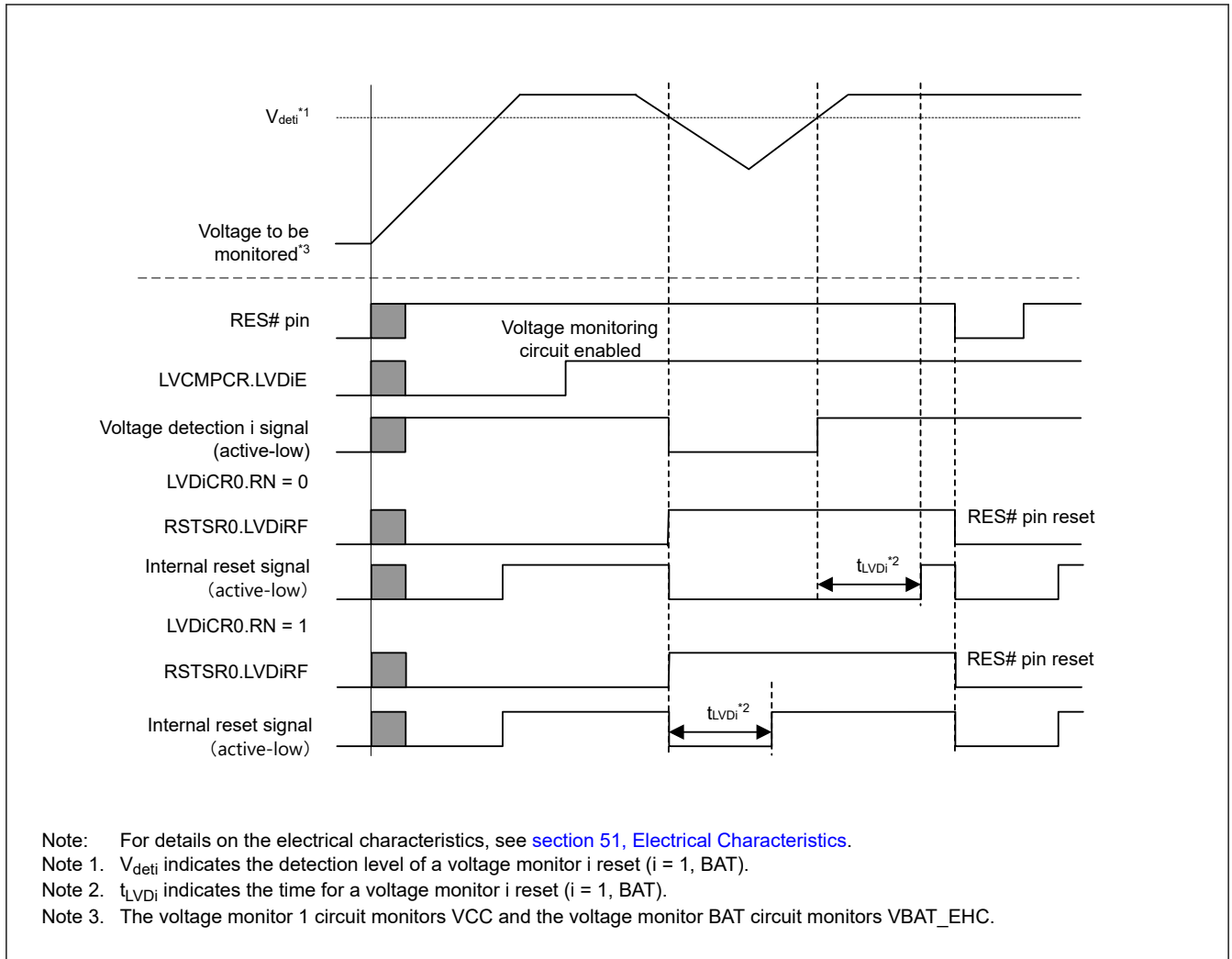


Figure 6.3 Example of operations during voltage monitor 1 and BAT resets

6.3.4 Deep Software Standby Reset

This is an internal reset generated when deep software standby mode is canceled by an interrupt.

When a deep software standby mode cancellation source is generated, a deep software standby reset is generated. The deep software standby reset is canceled after t_{DSBY} (return time after deep software standby mode cancellation) has elapsed. At the same time, deep software standby mode is also canceled.

When t_{DSBYWT} (wait time after deep software standby mode cancellation) has elapsed after deep software standby mode has been canceled, the internal reset is canceled and the CPU starts the reset exception handling.

For details of the deep software standby reset, see [section 13, Power-Saving Functions](#).

6.3.5 Independent Watchdog Timer Reset

The independent watchdog timer reset is an internal reset generated from the Independent Watchdog Timer (IWDG). Output of the reset from the IWDG 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 IWDG 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 31, Independent Watchdog Timer \(IWDG\)](#).

6.3.6 Watchdog Timer Reset

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

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

For details on the watchdog timer reset, see [section 30, Watchdog Timer \(WDT\)](#).

6.3.7 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 *Cortex®-M0+ Technical Reference Manual*.

6.3.8 MINPWON Mode Reset

The following events cause a MINPWON mode reset.

- When power supply mode changes from minimum power supply mode (MINPWON) to flash-excluded power supply mode (EXFPWON)
- When power supply mode changes from all power supply mode (ALLPWON) to the Software Standby mode in minimum power supply mode (MINPWON) and the Software Standby mode is released
- When power supply mode changes from flash-excluded power supply mode (EXFPWON) to the Software Standby mode in minimum power supply mode (MINPWON) and the Software Standby mode is released

The target of this reset is the ISO2 domain functions, not the ISO1 domain functions (including the CPU) or AWO domain functions. For details on transitions to flash-excluded power supply mode (EXFPWON) and minimum power supply mode (MINPWON), and the power supply domains for the functions, see [section 13, Power-Saving Functions](#).

6.3.9 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 6.4](#) shows an example of cold/warm start determination operation.

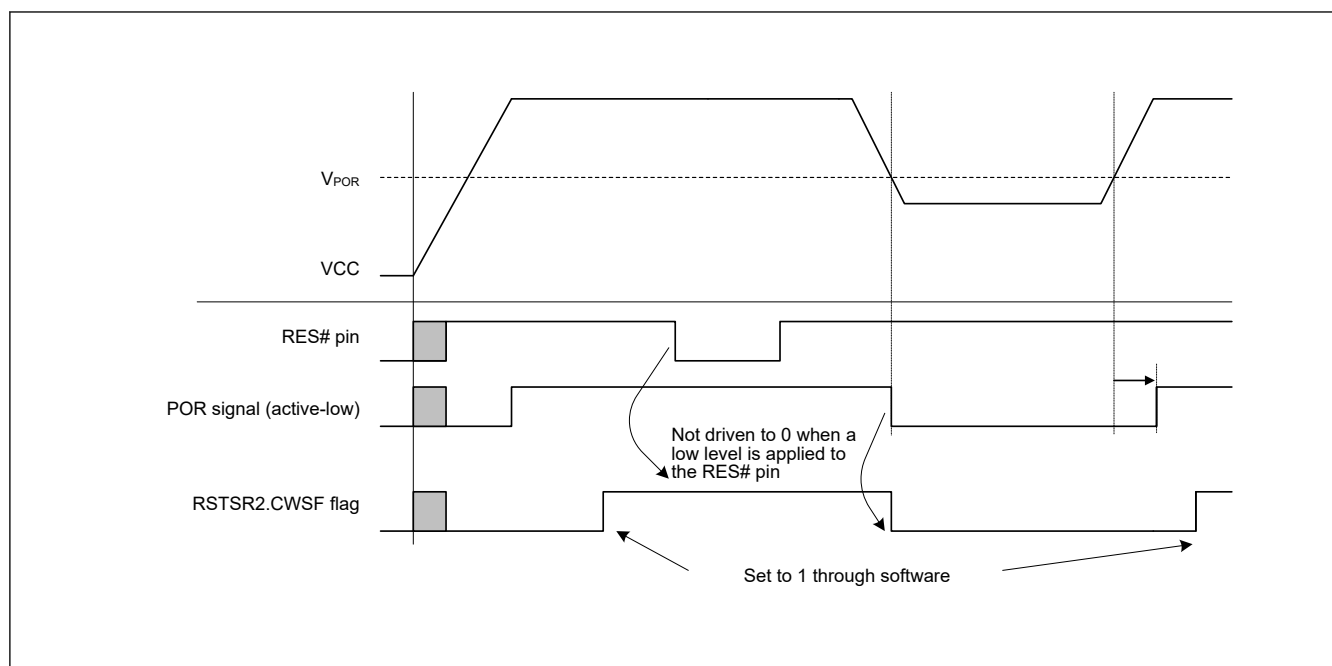


Figure 6.4 Example of cold/warm start determination operation

6.3.10 Determination of Reset Generation Source

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

Figure 6.5 shows an example of the flow to identify a reset generation source. The reset flag must be written with 0 after it is read as 1.

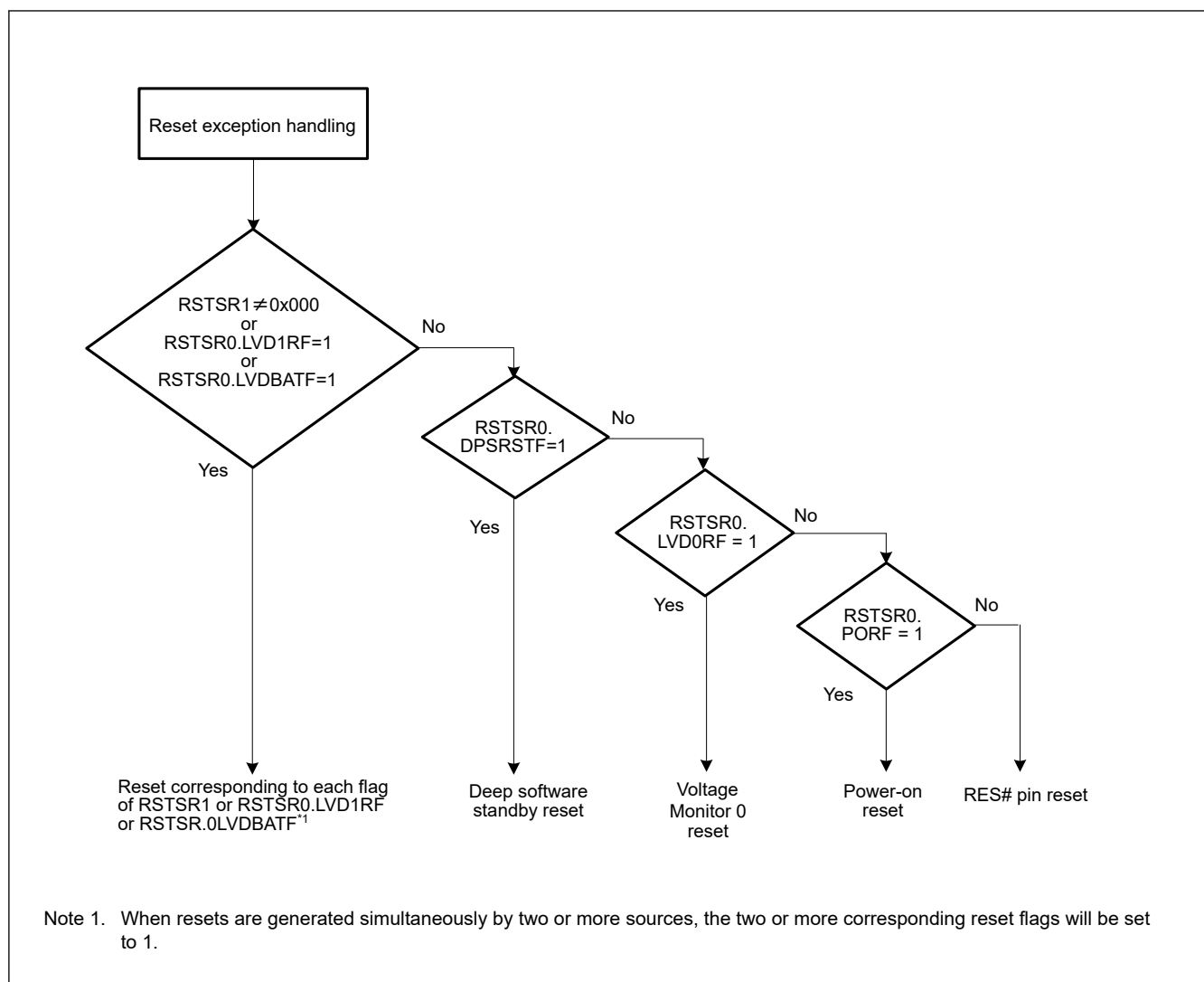


Figure 6.5 Example of reset generation source determination flow

7. Option-Setting Memory

7.1 Overview

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

Figure 7.1 shows the option-setting memory area.

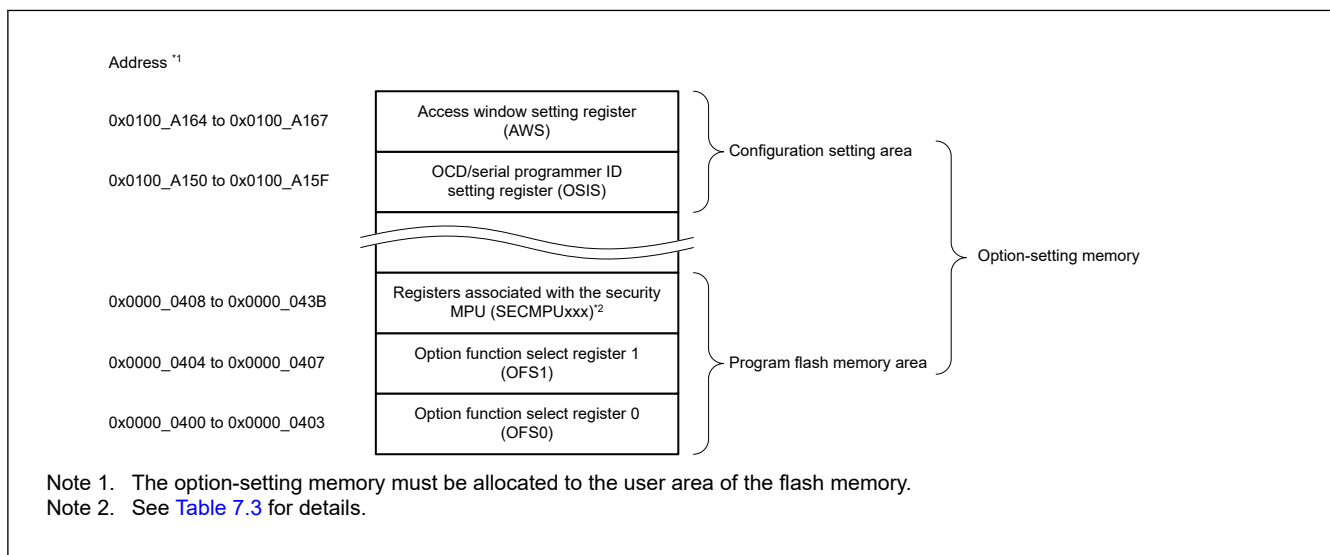


Figure 7.1 Option-setting memory area

7.2 Register Descriptions

7.2.1 OFS0 : Option Function Select Register 0

Address: 0x0000_0400

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	WDTS TPCTL	—	WDTR STIRQ S	WDTRPSS[1:0]	WDTRPES[1:0]	WDTCCKS[3:0]				WDTTOPS[1:0]	WDTS TRT	WDTC LKSEL			

Value after reset:

User setting*1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	IWDT STPC TL	—	IWDT RSTIR QS	IWDRPSS[1:0]		IWDRPES[1:0]		IWDTCKS[3:0]				IWDTTOPS[1:0]		IWDT STRT	—

Value after reset:

User setting*1

Bit	Symbol	Function	R/W
0	—	When read, this bit returns the written value. The write value should be 1.	R
1	IWDTSTRT	IWDT Start Mode Select 0: Automatically activate IWDT after a reset (auto start mode) 1: Disable IWDT after a reset	R
3:2	IWDTTOPS[1:0]	IWDT Timeout Period Select 0 0: 128 cycles (0x007F) 0 1: 512 cycles (0x01FF) 1 0: 1024 cycles (0x03FF) 1 1: 2048 cycles (0x07FF)	R

Bit	Symbol	Function	R/W
7:4	IWDTCKS[3:0]	IWDT-Dedicated Clock Frequency Division Ratio Select 0x0: × 1 0x2: × 1/16 0x3: × 1/32 0x4: × 1/64 0xF: × 1/128 0x5: × 1/256 Others: Setting prohibited	R
9:8	IWDRPES[1:0]	IWDT Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (no window end position setting)	R
11:10	IWDRPSS[1:0]	IWDT Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (no window start position setting)	R
12	IWDRSTIRQS	IWDT Reset Interrupt Request Select 0: Enable non-maskable interrupt request or interrupt request 1: Enable reset	R
13	—	When read, this bit returns the written value. The write value should be 1.	R
14	IWDTSTPCTL	IWDT Stop Control 0: Continue counting 1: Stop counting when in Sleep, Snooze, or Software Standby mode	R
15	—	When read, this bit return the written value. The write value should be 1.	R
16	WDTCLKSEL	WDT Clock Source (WDTCLK) Select 0: CCC_2K clock 1: PCLKB	R
17	WDTSTRT	WDT Start Mode Select 0: Automatically activate WDT after a reset (auto start mode) 1: Stop WDT after a reset (register start mode)	R
19:18	WDTTOPS[1:0]	WDT Timeout Period Select 0 0: 1024 cycles (0x03FF) 0 1: 4096 cycles (0x0FFF) 1 0: 8192 cycles (0x1FFF) 1 1: 16384 cycles (0x3FFF)	R
23:20	WDTCKS[3:0]	WDT Clock Frequency Division Ratio Select 0x1: WDTCLK divided by 4 0x4: WDTCLK divided by 64 0xF: WDTCLK divided by 128 0x6: WDTCLK divided by 512 0x7: WDTCLK divided by 2048 0x8: WDTCLK divided by 8192 Others: Setting prohibited	R
25:24	WDRPES[1:0]	WDT Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (no window end position setting)	R
27:26	WDRPSS[1:0]	WDT Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (no window start position setting)	R
28	WDRSTIRQS	WDT Reset Interrupt Request Select 0: Enable interrupt request or non-maskable interrupt request 1: Enable reset	R

Bit	Symbol	Function	R/W
29	—	When read, these bits return the written value. The write value should be 1.	R
30	WDTSTPCTL	WDT Stop Control See Table 7.2 .	R
31	—	When read, these bits return the written value. The write value should be 1.	R

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

IWDTSTRT bit (IWDT Start Mode Select)

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

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

The IWDTTOPS[1:0] bits specify the timeout period, that is, the time it takes for the down counter to underflow, as 128, 512, 1024, or 2048 cycles of the frequency-divided clock set in the IWDTCKS[3:0] bits. The 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 31, Independent Watchdog Timer \(IWDT\)](#).

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

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

For details, see [section 31, Independent Watchdog Timer \(IWDT\)](#).

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

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

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

For details, see [section 31, Independent Watchdog Timer \(IWDT\)](#).

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

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

For details, see [section 31, Independent Watchdog Timer \(IWDT\)](#).

IWDRSTIRQS bit (IWDT Reset Interrupt Request Select)

The IWDRSTIRQS 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 31, Independent Watchdog Timer \(IWDT\)](#).

IWDTSTPCTL bit (IWDT Stop Control)

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

[Table 7.1](#) shows the count stop control by the IWDTSTPCTL bit.

Table 7.1 Count Stop Control by the IWDTSTPCTL Bit (1 of 2)

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

Table 7.1 Count Stop Control by the IWDTSTPCTL Bit (2 of 2)

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

For details, see [section 31, Independent Watchdog Timer \(IWDT\)](#).

WDTCLKSEL bit (WDT Clock Source (WDTCLK) Select)

The WDTCLKSEL bit specifies the clock source for WDT (WDTCLK). To select PCLKB as WDTCLK, set this bit to 1. To select CCC_2K as WDTCLK, set this bit to 0. Note the following when selecting CCC_2K as WDTCLK.

- When selecting the auto start mode at the same time, wait until oscillation of the sub-clock oscillator (SOSC) is stable, and enable operation of the clock correction circuit (CCC). For details, see [section 10, Clock Correction Circuit \(CCC\)](#).
- Selecting CCC_2K for WDTCLK may lengthen the time required to detect a WDT timeout relative to when PCLKB is selected, which may lead to delays in the detection of runaway conditions. Accordingly, thoroughly consider the timing of refreshing of the WDT down-counter.

WDTSTRT bit (WDT Start Mode Select)

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

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

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

For details, see [section 30, Watchdog Timer \(WDT\)](#).

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

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

For details, see [section 30, Watchdog Timer \(WDT\)](#).

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

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

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

For details, see [section 30, 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 30, Watchdog Timer \(WDT\)](#).

WDRSTIRQS bit (WDT Reset Interrupt Request Select)

The WDRSTIRQS 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 30, Watchdog Timer \(WDT\)](#).

WDTSTPCTL bit (WDT Stop Control)

The WDTSTPCTL bit specifies whether to stop counting when entering Sleep mode, Snooze mode, or Software Stand-by mode.

Table 7.2 shows the count stop control by the WDTSTPCTL bit.

Table 7.2 Count Stop Control by the WDTSTPCTL Bit

WDTSTPCTL	Mode	Counting of WDT when WDTCLK = PCLKB	Counting of WDT when WDTCLK = CCC_2K
0	Sleep mode	Continue counting	Continue counting
	Snooze / software standby mode	Stop counting	
1	Sleep mode	Stop counting	Stop counting
	Snooze / software standby mode		

For details, see section 30, Watchdog Timer (WDT).

7.2.2 OFS1 : Option Function Select Register 1

Address: 0x0000_0404

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SLPW MN	—	—

Value after reset: The value set by the user*1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	VBATSEL[2:0]			—	—	HOCO FRQ	HOCO EN	—	—	VDSEL[2:0]			LVDA S	—	—

Value after reset: The value set by the user*1

Bit	Symbol	Function	R/W
1:0	—	When read, these bits return the written value. The write value should be 1.	R
2	LVDA S	Voltage Detection 0 Circuit Start 0: Enable voltage monitor 0 reset after a reset 1: Disable voltage monitor 0 reset after a reset	R
5:3	VDSEL[2:0]	Voltage Detection 0 Level Select*2 0 0 0: V _{det0_0} 0 0 1: V _{det0_1} 0 1 0: V _{det0_2} 0 1 1: V _{det0_3} 1 0 0: Setting prohibited Others: Setting prohibited	R
7:6	—	When read, these bits return the written value. The write value should be 1.	R
8	HOCOEN	HOCO Oscillation Enable 0: Enable HOCO oscillation after a reset 1: Disable HOCO oscillation after a reset	R
9	HOCOFREQ	HOCO Frequency Setting 0: 24 MHz 1: 32 MHz	R
11:10	—	When read, these bits return the written value. The write value should be 1.	R

Bit	Symbol	Function	R/W
14:12	VBATSEL[2:0]	Secondary Battery (VBAT) Charging Voltage Select 0 0 0: V _{batc_0} (2.4 V) 0 0 1: V _{batc_1} (2.5 V) 0 1 0: V _{batc_2} (2.6 V) 0 1 1: V _{batc_3} (2.7 V) 1 0 0: V _{batc_4} (2.8 V) 1 0 1: V _{batc_5} (2.9 V) 1 1 0: V _{batc_6} (3.0 V) 1 1 1: V _{batc_7} (3.1 V)	R
15:17	—	When read, these bits return the written value. The write value should be 1.	R
18	SLPWMN	Power Reduction Function of Flash Memory 0: Enabled current consumption reduction function of flash memory in low leakage current mode. 1: Disabled current consumption reduction function of flash memory in low leakage current mode.	R
31:19	—	When read, these bits return the written value. The write value should be 1.	R

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

Note 2. See [section 51, Electrical Characteristics](#) for the voltage levels to be detected.

LVDAS bit (Voltage Detection 0 Circuit Start)

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

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

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

HOCOEN bit (HOCO Oscillation Enable)

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

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

VBATSEL[2:0] bits (Secondary Battery (VBAT) Charging Voltage Select)

The VBATSEL bit selects the charging voltage for the secondary battery (VBAT) that is coupled to the VBAT pin.

SLPWMN bit (Power Reduction Function of Flash Memory)

The SLPWMN bit selects whether to enable or disable the power reduction feature of flash memory in low leakage current mode. When this function is enabled, the operating current is reduced during low leakage current mode operation in all power supply mode (ALLPWON), however the effect of the sub-clock oscillator noise filter is reduced.

7.2.3 MPU Registers

[Table 7.3](#) shows the registers related to the MPU function. For details, see [section 18, Memory Protection Unit \(MPU\)](#).

The security MPU is disabled on erasure of the flash memory. If incorrect data is written to an MPU register, the MCU might fail to operate. See [section 18, Memory Protection Unit \(MPU\)](#) to set the correct data.

Table 7.3 MPU registers (1 of 2)

Register name	Symbol	Function	Address	Size (byte)
Security MPU Program Counter Start Address Register 0	SECMPUPCS0	Specifies the security fetch region of code flash or SRAM.	0x0000_0408	4
Security MPU Program Counter End Address Register 0	SECMPUPCE0	Specifies the security fetch region of code flash or SRAM.	0x0000_040C	4

Table 7.3 MPU registers (2 of 2)

Register name	Symbol	Function	Address	Size (byte)
Security MPU Program Counter Start Address Register 1	SECMPUPCS1	Specifies the security fetch region of code flash or SRAM	0x0000_0410	4
Security MPU Program Counter End Address Register 1	SECMPUPCE1	Specifies the security fetch region of code flash or SRAM.	0x0000_0414	4
Security MPU Region 0 Start Address Register	SECMPUS0	Specifies the secure program and data of code flash	0x0000_0418	4
Security MPU Region 0 End Address Register	SECMPUE0	Specifies the secure program and data of code flash.	0x0000_041C	4
Security MPU Region 1 Start Address Register	SECMPUS1	Specifies the secure program and data of SRAM.	0x0000_0420	4
Security MPU Region 1 End Address Register	SECMPUE1	Specifies the secure program and data of SRAM.	0x0000_0424	4
Security MPU Region 2 Start Address Register	SECMPUS2	Specifies the secure data of security function.	0x0000_0428	4
Security MPU Region 2 End Address Register	SECMPUE2	Specifies the secure data of security function.	0x0000_042C	4
Security MPU Access Control Register	SECMPUAC	Specifies the security enabled/disabled region.	0x0000_0438	2

7.2.4 AWS : Access Window Setting Register

Address: 0x0100_A164

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BTFLG	—	—	—	—	—	FAWE[9:0]									

Value after reset: User setting

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

Value after reset: User setting

Bit	Symbol	Function	R/W
9:0	FAWS[9:0]	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 [21:12]. When the size of the code flash memory is 256 KB, the start address of the block is specifiable within the range from 0 to 63.	R
14:10	—	When read, these bits return the written value. The write value should be 1.	R
15	FSPR	Protection of Access Window and Startup Area Select Function This bit controls the programming of the write/erase protection for the access window, the Startup Area Select Flag (BTFLG), and the temporary boot swap control. When this bit is set to 0, it cannot be changed to 1. 0: Executing the configuration setting command for programming the access window (FAWE[9:0], FAWS[9:0]) and the Startup Area Select Flag (BTFLG) is invalid 1: Executing the configuration setting command for programming the access window (FAWE[9:0], FAWS[9:0]) and the Startup Area Select Flag (BTFLG) is valid	R

Bit	Symbol	Function	R/W
25:16	FAWE[9:0]	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 [21:12].	R
30:26	—	When read, these bits return the written value. The write value should be 1.	R
31	BTFLG	Startup Area Select Flag This bit specifies whether the address of the startup area is exchanged for the boot swap function. 0: First 32-Kbyte area (0x0000_0000 to 0x0000_7FFF) and second 32-Kbyte area (0x0000_8000 to 0x0000_FFFF). are exchanged 1: First 32-Kbyte area (0x0000_0000 to 0x0000_7FFF) and second 32-Kbyte area (0x0000_8000 to 0x0000_FFFF). are not exchanged	R

The value of the AWS register is actually stored in the code flash memory. The power is not supplied to the flash memory when the MCU is not in the ALLPWON mode, in which case, the value of the AWS register cannot be read. In cases where the value of the AWS register is required when the MCU is not in the ALLPWON mode, read the value of the FAWMON register described in [section 50, Flash Memory](#). The FAWMON register holds a copy of the value of the AWS register and can be read in any of the power supply modes.

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[9:0] bits and the FAWE[9:0] bits. The settings for the FAWS[9:0] and FAWE[9:0] bits are as follows:

FAWE[9:0] = FAWS[9:0]: The P/E command is allowed to execute in the full program flash area.

FAWE[9:0] > FAWS[9:0]: The P/E command is only allowed to execute in the window from the block pointed to by the FAWS[9:0] bits to the block one lower than the block pointed to by the FAWE[9:0] bits.

FAWE[9:0] < FAWS[9:0]: The P/E command is not allowed to execute in the program flash area.

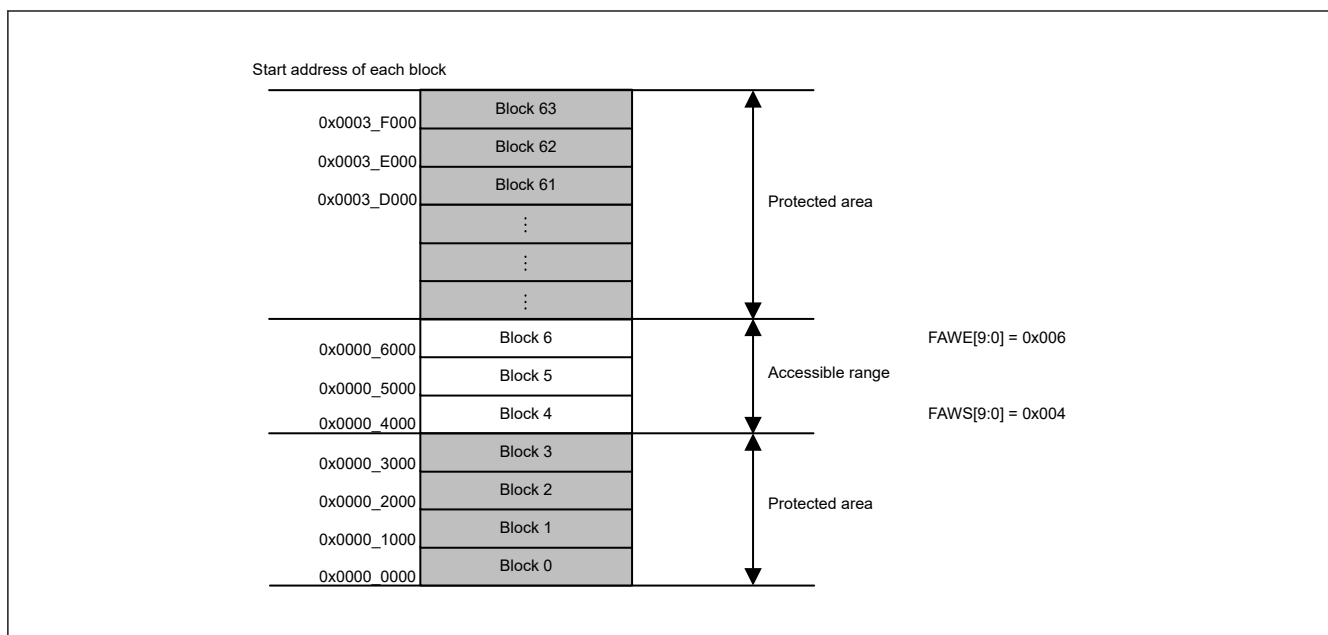


Figure 7.2 Access window overview

7.2.5 OSIS : OCD/Serial Programmer ID Setting Register

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

Address: 0x0100_A150, 0x0100_A154, 0x0100_A158h, 0x0100_A15C

Bit position: 31

0

Bit field:

Value after reset:

User setting

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

ID code bits [127] and [126] determine whether the ID code protection is enabled, and the authentication method to use with the host. [Table 7.4](#) shows how the ID code determines the authentication method.

Table 7.4 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) On-chip debug mode (SWD boot mode)	0xFF, ..., 0xFF (all bytes are 0xFF)	Protection disabled	The ID code is not checked, the ID code always matches, and the connection to the serial programmer or on-chip debugger is permitted.
	Bit [127] = 1, bit [126] = 1, and at least one of the 16 bytes is not 0xFF	Protection enabled	Matching ID code indicates that authentication is complete and connection to the serial programmer or the on-chip debugger is permitted. Mismatching ID code indicates transition to the ID code protection wait state. When the ID code sent from the serial programmer or the on-chip debugger is ALeRASE in ASCII code (0x414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFF), the content of the user flash area is erased and all bits in the OSIS register are 1. However, when the AWS.FSPR bit is 0 or security MPU is enabled, the content of the user flash area is not erased.
	Bit [127] = 1 and bit [126] = 0	Protection enabled	Matching ID code indicates that authentication is complete and connection to the serial programmer or the on-chip debugger is permitted. Mismatching ID code indicates transition to the ID code protection wait state.
	Bit [127] = 0	Protection enabled	The ID code is not checked, the ID code is always mismatching, the connection to the serial programmer or the on-chip debugger is prohibited, and Renesas cannot access the test mode.

7.3 Setting Option-Setting Memory

7.3.1 Allocation of Data in Option-Setting Memory

Programming data is allocated to the addresses in the option-setting memory shown in [Figure 7.1](#). The allocated data is used by tools such as a flash programming software or an on-chip debugger.

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

7.3.2 Setting Data for Programming Option-Setting Memory

Allocating data according to the procedure described in [section 7.3.1. Allocation of Data in Option-Setting Memory](#), alone does not actually write the data to the option-setting memory. You must also follow one of the actions described in this section.

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

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

For details of the programming command, the configuration setting command, and the startup area select function, see [section 50, Flash Memory](#).

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

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

The MCU provides two setting procedures:

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

7.4 Usage 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 within the scope of programming, write 1 to all bits of reserved areas and all reserved bits. If 0 is written to these bits, normal operation cannot be guaranteed.

7.4.2 Note on FSPR Bit

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

7.4.3 Note on the ID code protection

Once bit 127 of the OSIS register is set to 0, the on-chip debugger (OCD) or serial programmer cannot be connected to this MCU any more. Use the OSIS register with the utmost care.

8. Low Voltage Detection (LVD)

8.1 Overview

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

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

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

Table 8.1 LVD specifications

Parameter		Voltage monitor 0	Voltage monitor 1	Voltage monitor BAT
Means for setting up operation		OFS1 register	Registers	Registers
Target for monitoring		VCC pin input voltage	VCC pin input voltage	VBAT_EHC pin input voltage
Monitored voltage		V_{det0}	V_{det1}	V_{detBAT}
Detected event		Voltage falls past V_{det0}	Voltage rises or falls past V_{det1}	Voltage rises or falls past V_{detBAT}
Detection voltage		Selectable from 6 different levels in the OFS1.VDSEL[2:0] bits	Selectable from 8 different levels in the LVDLVL.R.LVD1LVL[2:0] bits	Selectable from 5 different levels in the LVDLVL.R.LVDBATLVL[2:0] bits
Monitoring flag		None	LVD1SR.MON flag: Monitors whether voltage is higher or lower than V_{det1}	LVDBATSR.MON flag: Monitors whether voltage is higher or lower than V_{detBAT}
			LVD1SR.DET flag: V_{det1} passage detection	LVDBATSR.DET flag: V_{detBAT} passage detection
Process on voltage detection	Reset	Voltage monitor 0 reset	Voltage monitor 1 reset	Voltage monitor BAT reset
		Reset when $V_{\text{det0}} > \text{VCC}$ CPU restart after specified time with $\text{VCC} > V_{\text{det0}}$	Reset when $V_{\text{det1}} > \text{VCC}$ CPU restart timing selectable: after specified time with $\text{VCC} > V_{\text{det1}}$ or $V_{\text{det1}} > \text{VCC}$	Reset when $V_{\text{detBAT}} > \text{VBAT\_EHC}$ CPU restart timing selectable: after specified time with either $\text{VBAT\_EHC} > V_{\text{detBAT}}$ or $V_{\text{detBAT}} > \text{VBAT\_EHC}$
	Interrupt	No interrupt	Voltage monitor 1 interrupt	Voltage monitor BAT interrupt
			Non-maskable or maskable interrupt selectable	Non-maskable or maskable interrupt selectable
			Interrupt request issued when $V_{\text{det1}} > \text{VCC}$ and $\text{VCC} > V_{\text{det1}}$ or either	Interrupt request issued when $V_{\text{detBAT}} > \text{VBAT\_EHC}$ and $\text{VBAT\_EHC} > V_{\text{detBAT}}$ or either
Digital filter	Switching between enable and disable	No digital filter function	Available	Available
	Sampling time	—	1/n LOCO frequency $\times 2$ (n: 2, 4, 8, 16)	1/n LOCO frequency $\times 2$ (n: 2, 4, 8, 16)
Event link function		None	Available Output of event signals on detection of V_{det1} crossings	None

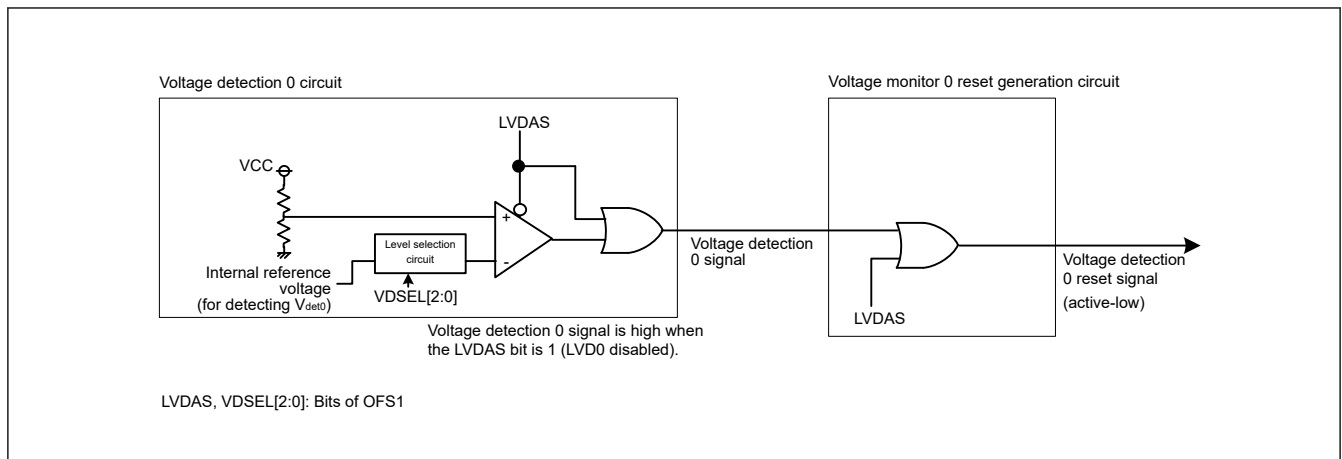


Figure 8.1 Block diagram of voltage monitor 0 reset generation circuit

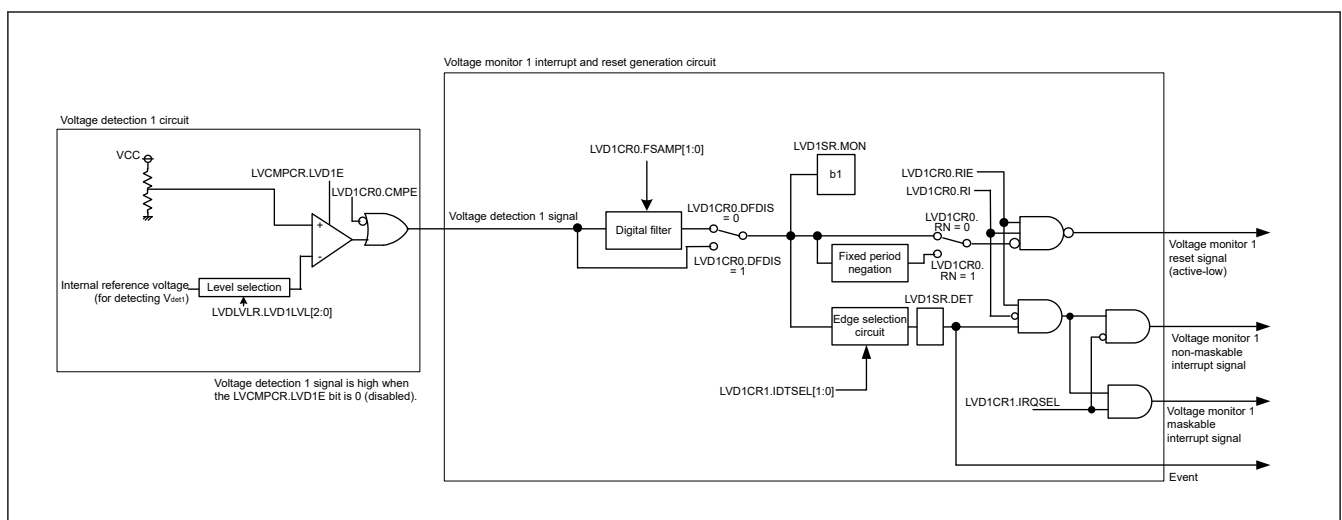


Figure 8.2 Block diagram of voltage monitor 1 interrupt and reset circuit

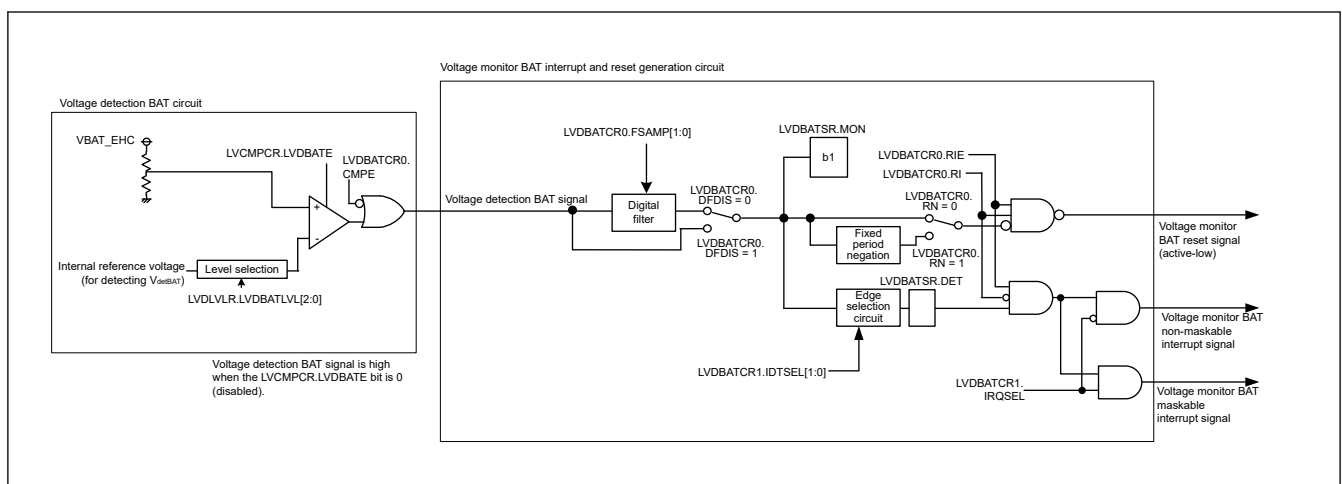


Figure 8.3 Block diagram of voltage monitor BAT interrupt and reset circuit

8.2 Register Descriptions

8.2.1 LVCMPCR : Voltage Monitor Circuit Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x417

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

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

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

LVD1E bit (Voltage Detection 1 Enable)

When using voltage detection 1 interrupt/reset or the LVD1SR.MON flag, set the LVD1E bit to 1. The voltage detection 1 circuit starts when LVD1 operation stabilization time ($t_{d(E-A)}$) elapses after the LVD1E bit value is changed from 0 to 1. For details on $t_{d(E-A)}$, see [section 51, Electrical Characteristics](#).

LVDBATE bit (Voltage Detection BAT Enable)

When using voltage detection BAT interrupt/reset or the LVDBATSR.MON flag, set the LVDBATE bit to 1. The voltage detection BAT circuit starts when $t_{d(E-A)}$ elapses after the LVDBATE bit value is changed from 0 to 1. For details on $t_{d(E-A)}$, see [section 51, Electrical Characteristics](#).

8.2.2 LVDLVLRL : Voltage Detection Level Select Register

Base address: SYSC = 0x4001_E000

Offset address: 0x418

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

Bit	Symbol	Function	R/W
2:0	LVD1LVL[2:0]	Voltage Detection 1 Level Select*1 0 0 0: V_{det1_0} 0 0 1: V_{det1_1} 0 1 0: V_{det1_3} 0 1 1: V_{det1_5} 1 0 0: V_{det1_7} 1 0 1: V_{det1_9} 1 1 0: V_{det1_B} 1 1 1: V_{det1_D}	R/W
3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6:4	LVDBATLVL[2:0]	Voltage Detection BAT Level Select*1 0 1 1: V_{detBAT_5} 1 0 0: V_{detBAT_7} 1 0 1: V_{detBAT_9} 1 1 0: V_{detBAT_B} 1 1 1: V_{detBAT_D} Others: Setting prohibited	R/W
7	—	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. See [section 51, Electrical Characteristics](#) for the voltage levels to be detected.

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

8.2.3 LVD1CR0 : Voltage Monitor 1 Circuit Control Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x41A

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

Bit	Symbol	Function	R/W
0	RIE	Voltage Monitor 1 Interrupt/Reset Enable 0: Disable 1: Enable	R/W
1	DFDIS	Voltage monitor 1 Digital Filter Disabled Mode Select 0: Enable the digital filter 1: Disable the digital filter	R/W
2	CMPE	Voltage Monitor 1 Circuit Comparison Result Output Enable 0: Disable voltage monitor 1 circuit comparison result output 1: Enable voltage monitor 1 circuit comparison result output	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	FSAMP[1:0]	Sampling Clock Select 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency	R/W
6	RI	Voltage Monitor 1 Circuit Mode Select 0: Generate voltage monitor 1 interrupt on V_{det1} crossing 1: Enable voltage monitor 1 reset when the voltage falls to and below V_{det1}	R/W
7	RN	Voltage Monitor 1 Reset Negate Select 0: Negate after a stabilization time (t_{LVD1}) when $VCC > V_{det1}$ is detected 1: Negate after a stabilization time (t_{LVD1}) on assertion of the LVD1 reset	R/W

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

RIE bit (Voltage Monitor 1 Interrupt/Reset Enable)

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

DFDIS bit (Voltage monitor 1 Digital Filter Disabled Mode Select)

The DFDIS bit disables the digital filter circuit. Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) when this bit is 0 (enabled). Set this bit to 1 (disabled) when using the voltage monitor 1 circuit in Software Standby mode or in Deep Software Standby mode.

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

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

FSAMP[1:0] bits (Sampling Clock Select)

The FSAMP[1:0] bits can be rewritten only when the LVD1CR0.DFDIS bit is 1 (digital filter circuit disabled). Do not rewrite these bits if the LVD1CR0.DFDIS bit is 0 (digital filter circuit enabled).

RI bit (Voltage Monitor 1 Circuit Mode Select)

When the RI bit is 1 (voltage monitor 1 reset selected), transition to Deep Software Standby mode cannot be made. In this case, transition to Software Standby mode is made. To enter Deep Software Standby mode, set the RI bit to 0 (voltage monitor 1 interrupt selected).

RN bit (Voltage Monitor 1 Reset Negate Select)

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

8.2.4 LVDBATCR0 : Voltage Monitor BAT Circuit Control Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x41B

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

Value after reset: 1 0 0 0 0 0 1 0

Bit	Symbol	Function	R/W
0	RIE	Voltage Monitor BAT Interrupt/Reset Enable 0: Disable 1: Enable	R/W
1	DFDIS	Voltage monitor BAT Digital Filter Disabled Mode Select 0: Enable the digital filter 1: Disable the digital filter	R/W
2	CMPE	Voltage Monitor BAT Circuit Comparison Result Output Enable 0: Disable voltage monitor BAT circuit comparison result output 1: Enable voltage monitor BAT circuit comparison result output	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	FSAMP[1:0]	Sampling Clock Select 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency	R/W
6	RI	Voltage Monitor BAT Circuit Mode Select 0: Generate voltage monitor BAT interrupt on V_{detBAT} crossing 1: Enable voltage monitor BAT reset when the voltage falls to and below V_{detBAT}	R/W

Bit	Symbol	Function	R/W
7	RN	Voltage Monitor BAT Reset Negate Select 0: Negate after a stabilization time (t_{LVDBAT}) when $V_{BAT\_EHC} > V_{detBAT}$ is detected 1: Negate after a stabilization time (t_{LVDBAT}) on assertion of the LVDBAT reset	R/W

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

RIE bit (Voltage Monitor BAT Interrupt/Reset Enable)

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

DFDIS bit (Voltage monitor BAT Digital Filter Disabled Mode Select)

The DFDIS bit disables the digital filter circuit. Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) when this bit is 0 (digital filter enabled). Set this bit to 1 (digital filter disabled) when using the voltage monitor BAT circuit in Software Standby mode or in Deep Software Standby mode.

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

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

FSAMP[1:0] bits (Sampling Clock Select)

The FSAMP[1:0] bits can be rewritten only when the LVDBATCR0.DFDIS bit is 1 (digital filter circuit disabled). Do not rewrite these bits if the LVDBATCR0.DFDIS bit is 0 (digital filter circuit enabled).

RI bit (Voltage Monitor BAT Circuit Mode Select)

When the RI bit is 1 (voltage monitor BAT reset selected), transition to Deep Software Standby mode cannot be made. In this case, transition to Software Standby mode is made. To enter Deep Software Standby mode, set the RI bit to 0 (voltage monitor BAT interrupt selected).

RN bit (Voltage Monitor BAT Reset Negate Select)

If the RN bit is set to 1 (negating LVDBAT reset in a specified time after its assertion), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Additionally, for a transition to Software Standby or Deep Software Standby mode, the only possible value for the RN bit is 0 (negation follows a stabilization time when $V_{BAT\_EHC} > V_{detBAT}$ is detected). Do not set the RN bit to 1 (negation follows a stabilization time after assertion of the LVDBAT reset signal) when this is the case.

8.2.5 LVD1CR1 : Voltage Monitor 1 Circuit Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0E0

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

Bit	Symbol	Function	R/W
1:0	IDTSEL[1:0]	Voltage Monitor 1 Interrupt Generation Condition Select 0 0: When $VCC \geq V_{det1}$ (rise) is detected 0 1: When $VCC < V_{det1}$ (fall) is detected 1 0: When fall and rise are detected 1 1: Settings prohibited	R/W
2	IRQSEL	Voltage Monitor 1 Interrupt Type Select 0: Non-maskable interrupt 1: Maskable interrupt ^{*1}	R/W

Bit	Symbol	Function	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

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

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

8.2.6 LVD1SR : Voltage Monitor 1 Circuit Status Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0E1

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

Value after reset: 0 0 0 0 0 0 0 1 0

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

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

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

DET flag (Voltage Monitor 1 Voltage Variation Detection Flag)

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

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

MON flag (Voltage Monitor 1 Signal Monitor Flag)

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

8.2.7 LVDBATCR1 : Voltage Monitor BAT Circuit Control Register 1

Base address: SYSC = 0x4001_E000

Offset address: 0x0E2

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	IRQSE L	IDTSEL[1:0]	

Value after reset: 0 0 0 0 0 0 0 0 1

Bit	Symbol	Function	R/W
1:0	IDTSEL[1:0]	Voltage Monitor BAT Interrupt Generation Condition Select 0 0: When $V_{BAT\_EHC} \geq V_{detBAT}$ (rise) is detected 0 1: When $V_{BAT\_EHC} < V_{detBAT}$ (fall) is detected 1 0: When fall and rise are detected 1 1: Settings prohibited	R/W

Bit	Symbol	Function	R/W
2	IRQSEL	Voltage Monitor BAT Interrupt Type Select 0: Non-maskable interrupt 1: Maskable interrupt ^{*1}	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

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

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

8.2.8 LVDBATSR : Voltage Monitor BAT Circuit Status Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0E3

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

Value after reset: 0 0 0 0 0 0 0 1 0

Bit	Symbol	Function	R/W
0	DET	Voltage Monitor BAT Voltage Variation Detection Flag 0: Not detected 1: V_{detBAT} crossing is detected	R/W ^{*1}
1	MON	Voltage Monitor BAT Signal Monitor Flag 0: $V_{\text{BAT\_EHC}} < V_{\text{detBAT}}$ 1: $V_{\text{BAT\_EHC}} \geq V_{\text{detBAT}}$ or MON is disabled	R
7:2	—	These bits are read as 0. The write value should be 0.	R/W

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

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

DET flag (Voltage Monitor BAT Voltage Variation Detection Flag)

The DET flag is enabled when the LVCMPCR.LVDBATE bit is 1 (voltage detection BAT circuit enabled) and the LVDBATCR0.CMPE bit is 1 (voltage monitor BAT circuit comparison result output enabled).

Set the DET flag to 0 after setting LVDBATCR0.RIE is 0 (disabled). When setting LVDBATCR0.RIE bit to 1 (enabled) after setting it to 0, wait for 2 or more PCLKB cycles which have elapsed.

MON flag (Voltage Monitor BAT Signal Monitor Flag)

The MON flag is enabled when the LVCMPCR.LVDBATE bit is 1 (voltage detection BAT circuit enabled) and the LVDBATCR0.CMPE bit is 1 (voltage monitor BAT circuit comparison result output enabled).

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

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

Step	Monitoring the comparison results from voltage monitor 1	
Setting up the voltage detection 1 circuit	1	Set LVCMPCR.LVD1E = 0 to disable voltage detection 1 before writing to the LVDLVLR.LVD1LVL[2:0] bits.
	2	Select the detection voltage in the LVDLVLR.LVD1LVL[2:0] bits.
	3	Set LVCMPCR.LVD1E = 1 to enable the voltage detection 1 circuit.
	4	Wait for at least $t_{d(E-A)}$ for the LVD1 operation stabilization time after LVD1 is enabled.*1
Setting the digital filter*2	5	Select the sampling clock for the digital filter in the LVD1CR0.FSAMP[1:0] bits.
	6	Set LVD1CR0.DFDIS = 0 to enable the digital filter.
	7	Wait for at least $2n + 3$ cycles of the LOCO, where $n = 2, 4, 8,$ or 16 , and the sampling clock for the digital filter is the LOCO frequency-divided by n .
Enabling output	8	Set LVD1CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 1.

Note 1. Steps 5 to 7 can be performed during the wait time of step 4. For details of $t_{d(E-A)}$, see [section 51, Electrical Characteristics](#).

Note 2. Steps 5 to 7 are not required if the digital filter is not in use.

8.3.3 Monitoring V_{detBAT}

[Table 8.3](#) shows the procedures to set up monitoring against V_{detBAT} . After the settings are complete, the comparison results from voltage monitor BAT can be monitored in the LVDBATSR.MON flag.

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

Step	Monitoring the results of comparison by voltage monitor BAT	
Setting up the voltage detection BAT circuit	1	Set LVCMPCR.LVDBATE = 0 to disable voltage detection BAT before writing to the LVDLVLR.LVDBATLVL[2:0] bits.
	2	Select the detection voltage in the LVDLVLR.LVDBATLVL[2:0] bits.
	3	Set LVCMPCR.LVDBATE = 1 to enable the voltage detection BAT circuit.
	4	Wait for at least $t_{d(E-A)}$ for the LVDBAT operation stabilization time after LVDBAT is enabled.*1
Setting the digital filter*2	5	Select the sampling clock for the digital filter in the LVDBATCR0.FSAMP[1:0] bits.
	6	Set LVDBATCR0.DFDIS = 0 to enable the digital filter.
	7	Wait for at least $2n + 3$ cycles of the LOCO, where $n = 2, 4, 8,$ or 16 , and the sampling clock for the digital filter is the LOCO frequency-divided by n .
Enabling output	8	Set LVDBATCR0.CMPE = 1 to enable output of the comparison results from voltage monitor BAT.

Note 1. Steps 5 to 7 can be performed during the wait time of step 4. For details of $t_{d(E-A)}$, see [section 51, Electrical Characteristics](#).

Note 2. Steps 5 to 7 are not required if the digital filter is not in use.

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.

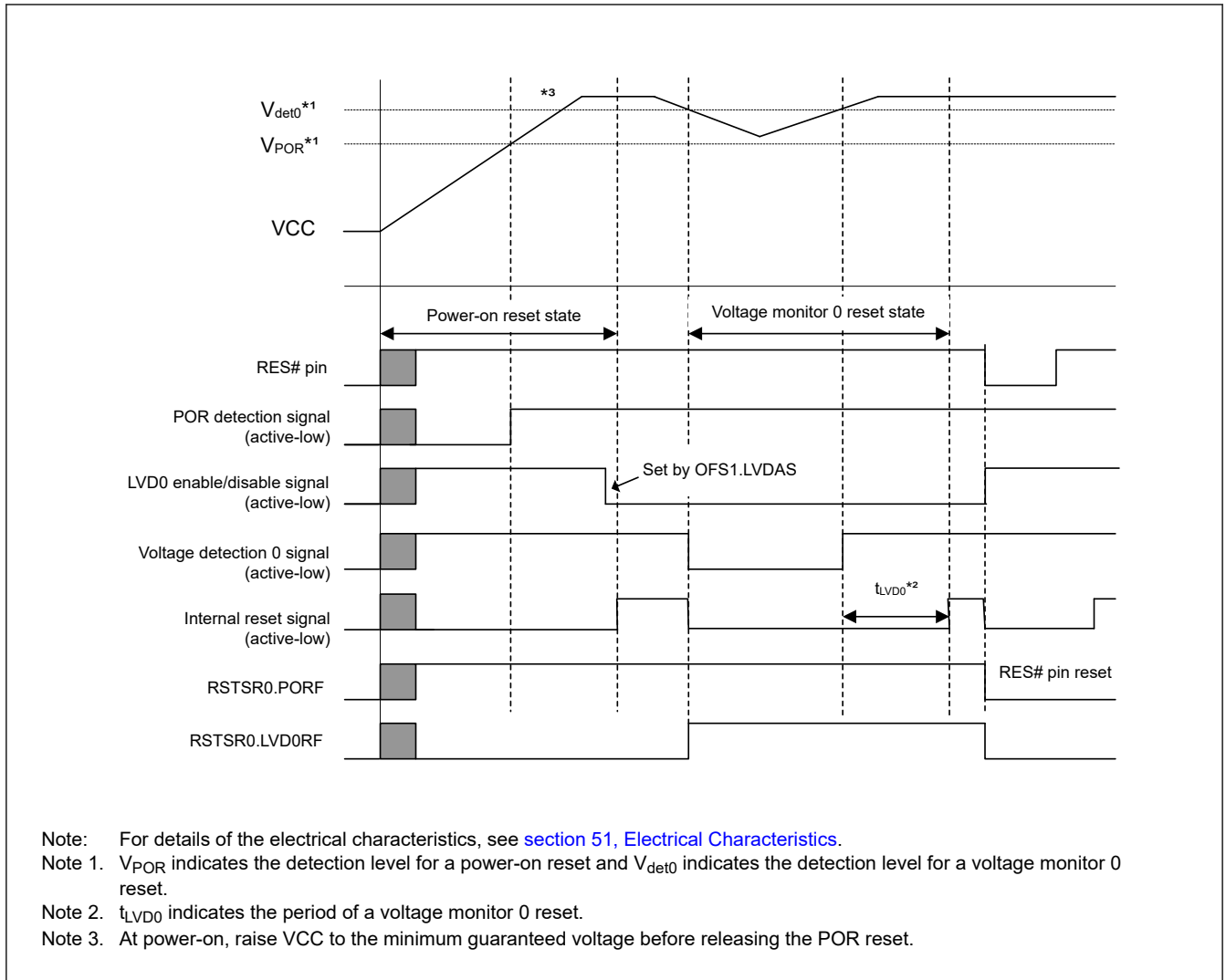


Figure 8.4 Example of voltage monitor 0 reset operation

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 procedures for setting bits related to the voltage monitor 1 interrupt/reset so that voltage monitoring occurs. [Table 8.5](#) shows the procedures for setting bits related to the voltage monitor 1 interrupt/reset so that voltage monitoring stops. [Figure 8.5](#) shows an example of operations for a voltage monitor 1 interrupt. For the operation of the voltage monitor 1 reset, see [section 6.3.3. Voltage Monitor Reset](#) in [section 6, Resets](#).

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

(1) Setting in Software Standby mode

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

(2) Settings in Deep Software Standby mode

- Disable the digital filter ($LVD1CR0.DFDIS = 1$).
- Enable the voltage monitor 1 interrupt ($LVD1CR0.RI = 0$). If the voltage monitor 1 reset is enabled ($LVD1CR0.RI = 1$), a transition to Deep Software Standby mode is not possible, and the operation transitions to Software Standby mode instead.

Table 8.4 Procedures for setting bits related to voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring occurs

Step	Voltage monitor 1 interrupt (voltage monitor 1 ELC event output)		Voltage monitor 1 reset
Setting up the voltage detection 1 circuit	1	Set LVCMPCR.LVD1E = 0 to disable voltage detection 1 before writing to the LVDLVLRL register.	
	2	Select the detection voltage in the LVDLVLRL.LVD1LVL[2:0] bits.	
	3	Set LVCMPCR.LVD1E = 1 to enable the voltage detection 1 circuit.	
	4	Wait for at least $t_{d(E-A)}$ for the LVD1 operation stabilization time after LVD1 is enabled.* ¹	
Setting the digital filter* ³	5	Select the sampling clock for the digital filter in the LVD1CR0.FSAMP[1:0] bits.	
	6	Set LVD1CR0.DFDIS = 0 to enable the digital filter.	
	7	Wait for at least $2n + 3$ LOCO cycles, where $n = 2, 4, 8,$ or 16 , and the sampling clock for the digital filter is the LOCO frequency-divided by n .* ⁴	
Setting up the voltage monitor 1 interrupt or reset	8	Set LVD1CR0.RI = 0 to select the voltage monitor 1 interrupt.	- Set LVD1CR0.RI = 1 to select the voltage monitor 1 reset. - Select the type of reset negation in the LVD1CR0.RN bit.
	9	- Select the interrupt request condition in the LVD1CR1.IDTSEL[1:0] bits. - Select the interrupt type in the LVD1CR1.IRQSEL bit.	—
Enabling output	10	Set LVD1SR.DET = 0.	
	11	Set LVD1CR0.RIE = 1 to enable the voltage monitor 1 interrupt or reset.* ²	
	12	Set LVD1CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 1.	

Note 1. Steps 5 to 11 can be performed during the wait time in step 4. For details on $t_{d(E-A)}$, see [section 51, Electrical Characteristics](#).

Note 2. Step 11 is not required if only the ELC event signal is to be output.

Note 3. Steps 5 to 7 are not required if the digital filter is not in use.

Note 4. Steps 8 to 11 can be performed during the wait time of step 7.

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

Step	Voltage monitor 1 interrupt (voltage monitor 1 ELC event output), voltage monitor 1 reset	
Stopping the enabling output	1	Set LVD1CR0.CMPE = 0 to disable output of the comparison results from voltage monitor 1.
	2	Wait for at least $2n + 3$ cycles of the LOCO, where $n = 2, 4, 8,$ or 16 , and the sampling clock for the digital filter is the LOCO frequency-divided by n .* ²
	3	Set LVD1CR0.RIE = 0 to disable the voltage monitor 1 interrupt or reset.* ¹
Stopping the digital filter	4	Set LVD1CR0.DFDIS = 1 to disable the digital filter.* ^{2, 3}
Stopping the voltage detection 1 circuit	5	Set LVCMPCR.LVD1E = 0 to disable the voltage detection 1 circuit.

Note 1. Step 3 is not required if only the ELC event signal is to be output.

Note 2. Steps 2 and 4 are not required if the digital filter is not in use.

Note 3. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least 2 LOCO clock cycles before re-enabling it.

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

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

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

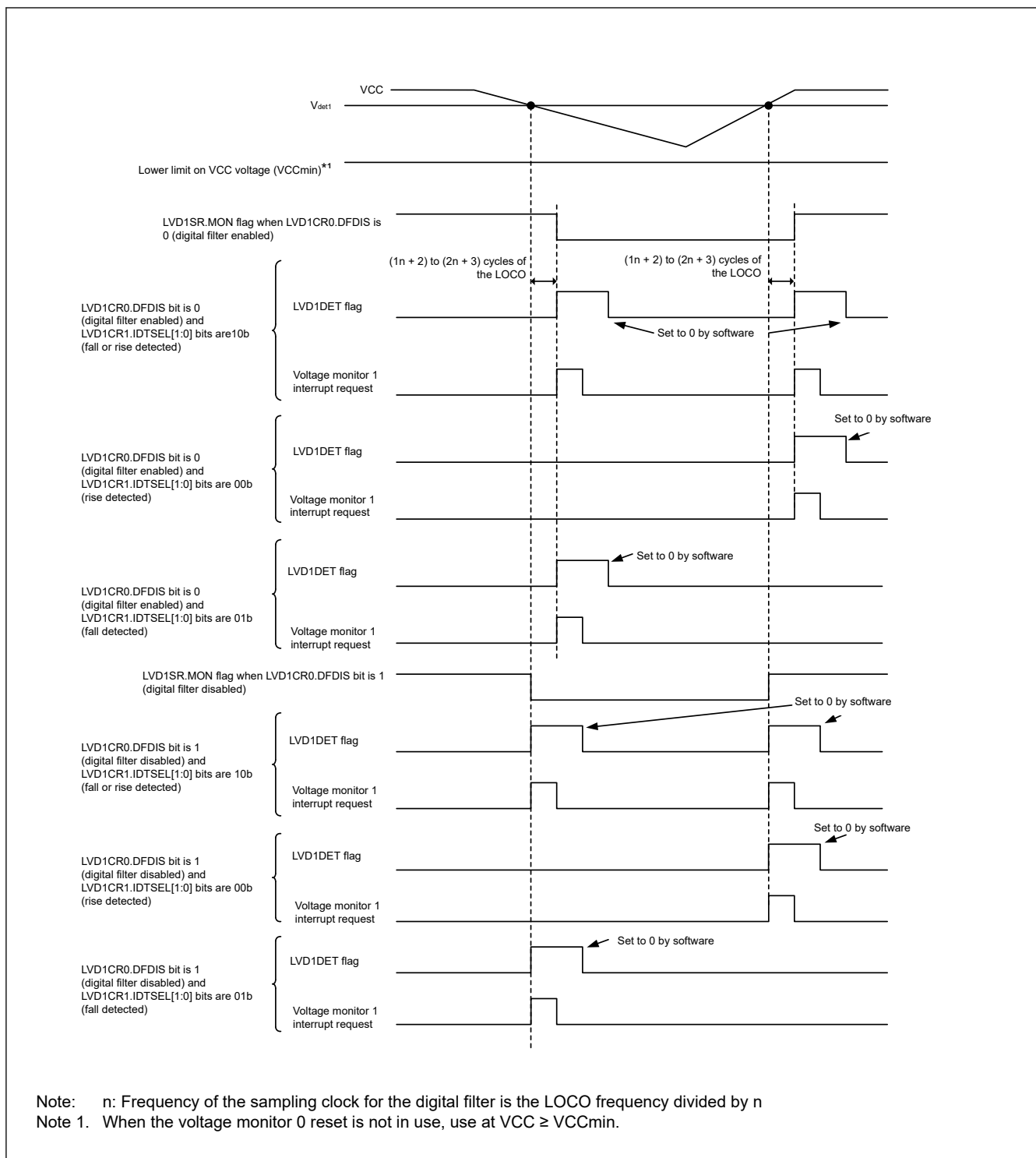


Figure 8.5 Example of voltage monitor 1 interrupt operation

8.6 Interrupt and Reset from Voltage Monitor BAT

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

Table 8.6 shows the procedures for setting bits related to the voltage monitor BAT interrupt/reset so that voltage monitoring occurs. Table 8.7 shows the procedures for setting bits related to the voltage monitor BAT interrupt/reset so that voltage monitoring stops. Figure 8.6 shows an example of operations for a voltage monitor BAT interrupt. For the operation of the voltage monitor BAT reset, see section 6.3.3. Voltage Monitor Reset in section 6, Resets.

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

(1) Setting in Software Standby mode

- Disable the digital filter (LVDBATCR0.DFDIS = 1)
- When $V_{BAT\_EHC} > V_{detBAT}$ is detected, negate the voltage monitor BAT reset signal (LVDBATCR0.RN = 0) following a stabilization time.

(2) Settings in Deep Software Standby mode

- Disable the digital filter (LVDBATCR0.DFDIS = 1).
- Enable the voltage monitor BAT interrupt (LVDBATCR0.RI = 0). If the voltage monitor BAT reset is enabled (LVDBATCR0.RI = 1), a transition to Deep Software Standby mode is not possible, and the operation transitions to Software Standby mode instead.

Table 8.6 Procedures for setting bits related to voltage monitor BAT interrupt and voltage monitor BAT reset so that voltage monitoring occurs

Step	Voltage monitor BAT interrupt		Voltage monitor BAT reset
BAT circuit	1	Set LVCMPCR.LVDBATE = 0 to disable voltage detection BAT before writing to the LVDLVLR register.	
	2	Select the detection voltage in the LVDLVLR.LVDBATLVL[2:0] bits.	
	3	Set LVCMPCR.LVDBATE = 1 to enable the voltage detection BAT circuit.	
	4	Wait for at least $t_{d(E-A)}$ for the LVDBAT operation stabilization time after LVDBAT is enabled.*1	
Setting the digital filter*2	5	Select the sampling clock for the digital filter in the LVDBATCR0.FSAMP[1:0] bits.	
	6	Set LVDBATCR0.DFDIS = 0 to enable the digital filter.	
	7	Wait for at least $2n + 3$ LOCO cycles, where $n = 2, 4, 8,$ or 16 , and the sampling clock for the digital filter is the LOCO frequency-divided by n .*3	
Setting up the voltage monitor BAT interrupt or reset	8	Set LVDBATCR0.RI = 0 to select the voltage monitor BAT interrupt.	• Set LVDBATCR0.RI = 1 to select the voltage monitor BAT reset. • Select the type of reset negation in the LVDBATCR0.RN bit.
	9	• Select the interrupt request condition in the LVDBATCR1.IDTSEL[1:0] bits. • Select the interrupt type in the LVDBATCR1.IRQSEL bit.	—
Enabling output	10	Set LVDBATSR.DET = 0.	
	11	Set LVDBATCR0.RIE = 1 to enable the voltage monitor BAT interrupt or reset.	
	12	Set LVDBATCR0.CMPE = 1 to enable output of the comparison results from voltage monitor BAT.	

Note 1. Steps 5 to 11 can be performed during the wait time in step 4. For details on $t_{d(E-A)}$, see [section 51, Electrical Characteristics](#).

Note 2. Steps 5 to 7 are not required if the digital filter is not in use.

Note 3. Steps 8 to 11 can be performed during the wait time of step 7.

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

Step	Voltage monitor BAT interrupt, voltage monitor BAT reset	
Settings to stop enabling output	1	Set LVDBATCR0.CMPE = 0 to disable output of the comparison results from voltage monitor BAT.
	2	Wait for at least $2n + 3$ cycles of the LOCO, where $n = 2, 4, 8,$ or 16 , and the sampling clock for the digital filter is the LOCO frequency-divided by n .*1
	3	Set LVDBATCR0.RIE = 0 to disable the voltage monitor BAT interrupt or reset.
Stopping the digital filter	4	Set LVDBATCR0.DFDIS = 1 to disable the digital filter. *1, *2
Stopping the voltage detection BAT circuit	5	Set LVCMPCR.LVDBATE = 0 to disable the voltage detection BAT circuit.

Note 1. Steps 2 and 4 are not required if the digital filter is not in use.

Note 2. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least 2 LOCO clock cycles before re-enabling it.

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

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

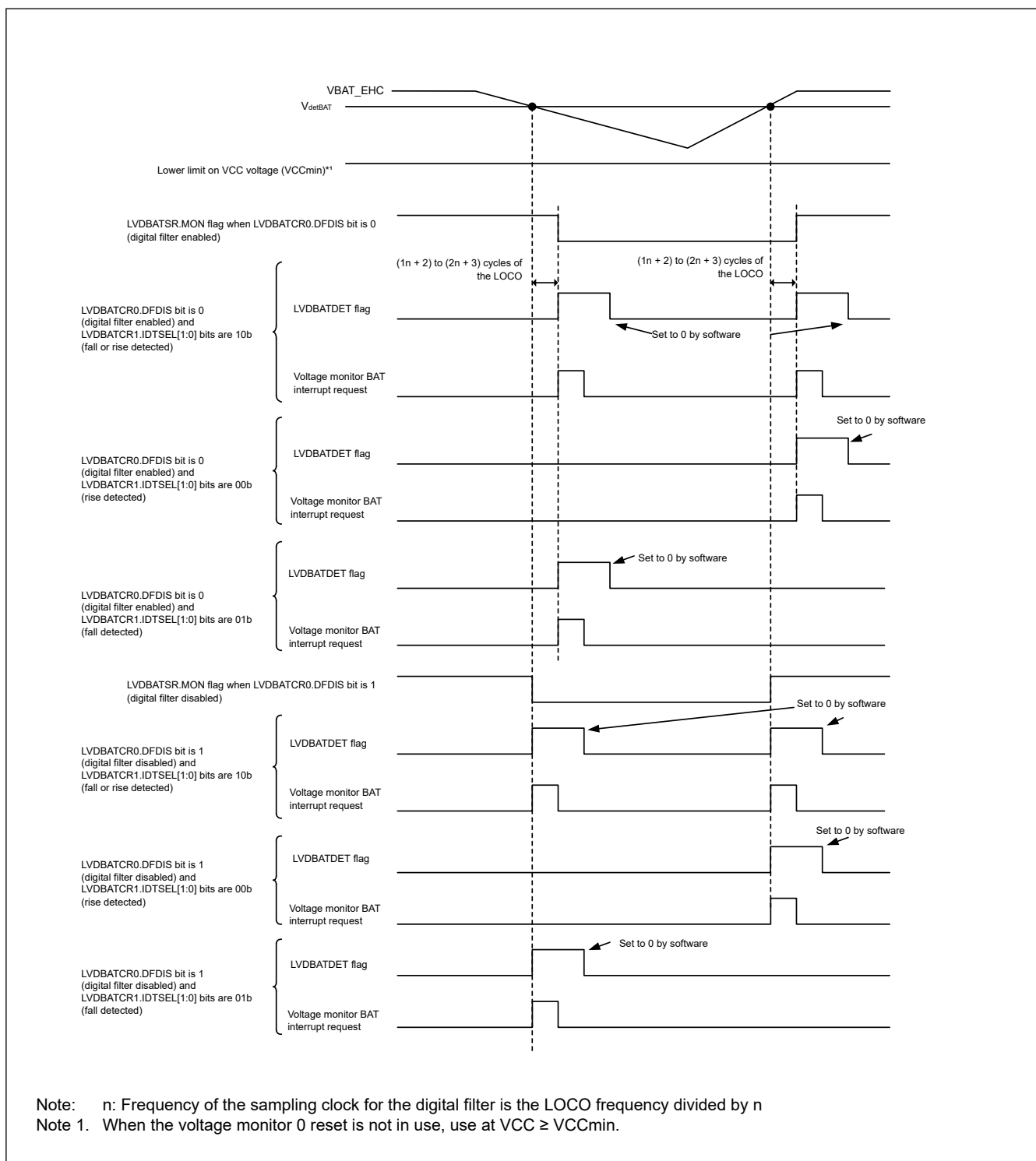


Figure 8.6 Example of voltage monitor BAT interrupt operation

8.7 Event Link Controller (ELC) Output

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

(1) V_{det1} Crossing Detection Event

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

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

8.7.1 Interrupt Handling and Event Linking

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

In contrast, as soon as an interrupt source is generated, an event link signal is output as the event signal to the other module through the ELC, regardless of the state of the interrupt enable bit. (LVD1 only)

It is possible to output voltage monitor 1 interrupt in Software Standby and Deep Software Standby modes.

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

8.8 Usage Notes

8.8.1 Register Write Protection

The registers described in this chapter are protected by the register write protection function. To access the registers, cancel the protection. For details, see [section 15, Register Write Protection](#).

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](#) show a block diagram, and [Table 9.3](#) lists the I/O pins.

Table 9.1 Clock generation circuit specifications for the clock sources

Clock source	Description	Specification
Main clock oscillator (MOSC)	Resonator frequency	8 MHz to 32 MHz
	External clock input frequency	Up to 32 MHz
	External resonator or additional circuit	ceramic resonator, crystal
	Connection pins	EXTAL, XTAL
	Drive capability switching	Available
	Oscillation stop detection function	Available
	Oscillation stabilization wait circuit	Available
	Low consumption oscillation function	Available
Sub-clock oscillator (SOSC)	Resonator frequency	32.768 kHz
	External resonator or additional circuit	crystal resonator
	Connection pins	XCIN, XCOUT
	Drive capability switching	Available
	Noise filter	Available
High-speed on-chip oscillator (HOCO)	Oscillation frequency	24/32/48/64 MHz
	FLL function	Available
	Oscillation stabilization wait circuit	Available
Middle-speed on-chip oscillator (MOCO)	Oscillation frequency	2 MHz
Low-speed on-chip oscillator (LOCO)	Oscillation frequency	32.768 kHz
IWDT-dedicated on-chip oscillator (IWDTLOCO)	Oscillation frequency	16 kHz
Clock correction circuit (CCC) output clock (CCC_2K)	Input frequency	2.048 kHz
External clock input for JTAG (TCK)	Input frequency	Up to 10 MHz
External clock input for SWD (SWCLK)	Input frequency	Up to 12.5 MHz

Table 9.2 Clock generation circuit specifications for the internal clocks (1 of 2)

Item	Clock source	Clock supply destination	Specification
System clock (ICLK)	MOSC/SOSC/HOCO/MOCO/LOCO	CPU, DTC, DMAC, Flash, SRAM	Up to 32 MHz (Normal mode) Up to 64 MHz (Boost mode) 1 MHz to 32 MHz (on Flash P/E) Division ratios: 1/2/4/8/16/32/64
Peripheral module clock A (PCLKA)	MOSC/SOSC/HOCO/MOCO/LOCO	Peripheral module (SCI0, SCI1, SPI0, SPI1, CRC, IrDA, TSIP-Lite, MLCD, GDT, and QSPI)	Same as ICLK
Peripheral module clock B (PCLKB)	MOSC/SOSC/HOCO/MOCO/LOCO	Peripheral module (other than supply destination of PCLKA)	Up to 32 MHz Division ratios: 1/2/4/8/16/32/64
AGT clock (AGTSCLK/AGTLCLK)	SOSC/LOCO/PCLKB	AGT	Up to 32 MHz
CAC Main clock (CACMCLK)	MOSC	CAC	Up to 32 MHz

Table 9.2 Clock generation circuit specifications for the internal clocks (2 of 2)

Item	Clock source	Clock supply destination	Specification
CAC Sub clock (CACSCCLK)	SOSC	CAC	32.768 kHz
CAC LOCO clock (CACLCLK)	LOCO	CAC	32.768 kHz
CAC MOCO clock (CACMOCLK)	MOCO	CAC	2 MHz
CAC HOCO clock (CACHCLK)	HOCO	CAC	24/32/48/64 MHz
CACCCC2K	CCC_2K	CAC	2.048 kHz
CAC IWDTLOCO clock (CACILCLK)	IWDTLOCO	CAC	16 kHz
LOCO clock	LOCO	VBBC	32.768 kHz
SOSC clock	SOSC	RTC, VBBC, WUPT	32.768 kHz
CCC2K clock (CCC_2K)	CCC	WDT	2.048 kHz
CCC clock (CCC32K)	SOSC	CCC	32.768 kHz
IWDT clock (IWDTCLK)	IWDTLOCO	IWDT	16 kHz
WDT clock (WDTCLK)	PCLKB/CCC_2K	WDT	Up to 32 MHz
SysTick timer clock (SYSTICCLK)	LOCO	SysTick timer	32.768 kHz
JTAG clock (JTAGTCK)	TCK pin	TAP controller	Up to 10 MHz
Serial wire clock (SWCLK)	SWCLK pin	OCD	Up to 12.5 MHz
External pin output clock (CLKOUT)	MOSC/SOSC/LOCO/MOCO/HOCO/CCC_2K	CLKOUT pin	Up to 32 MHz
External pin output sub-clock (CLKOUT32K)	SOSC	CLKOUT32 pin	32.768 kHz

Note: Restrictions on the clock frequency settings: $ICLK/PCLKA \geq PCLKB$
 Restrictions on the clock frequency ratio (N: integer, and up to 64): $ICLK/PCLKA:PCLKB = N:1$
 PCLKA and ICLK are at the same speed.

Note: The minimum ICLK frequency for the flash memory is 1 MHz in Programming/Erase mode.

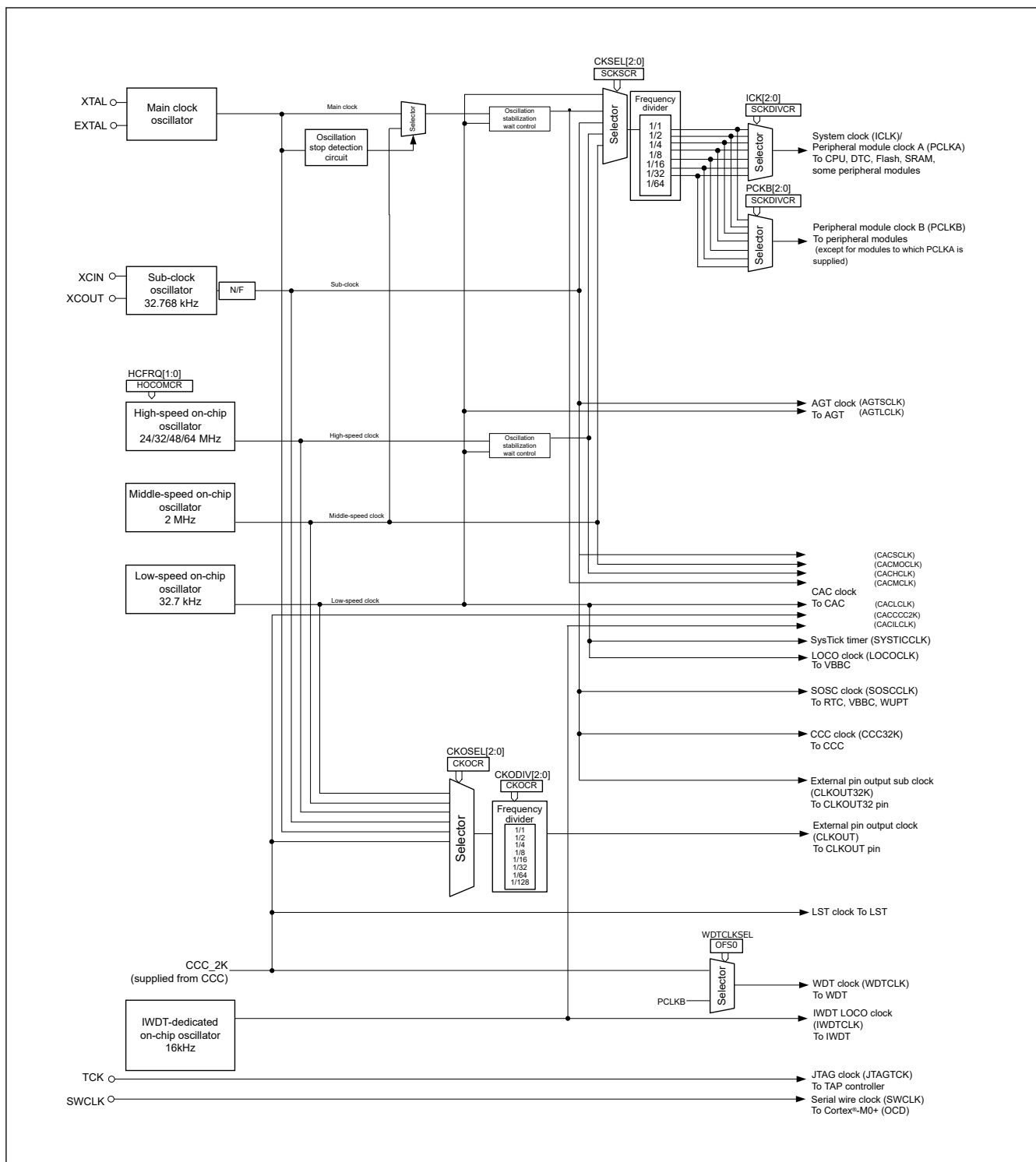


Figure 9.1 Clock generation circuit block diagram

Table 9.3 Clock generation circuit input/output pins (1 of 2)

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

Table 9.3 Clock generation circuit input/output pins (2 of 2)

Pin name	I/O	Description
SWCLK	Input	This pin is used to input the clock from the SWD
CLKOUT	Output	This pin is used to output clocks
CLKOUT32K	Output	This pin is used to output one cycle of the sub-clock

9.2 Register Descriptions

9.2.1 SCKDIVCR : System Clock Division Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	ICK[2:0]		—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	PCKB[2:0]		—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

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

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

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

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

The PCKB[2:0] bits select the frequency of peripheral module clock B (PCLKB). Note that the voltage conditions may limit the range of workable frequencies if the 14-bit A/D converter is to be used. For details, see [section 51.4. A/D Conversion Characteristics](#).

ICK[2:0] bits (System Clock (ICLK) Select/Peripheral Module Clock (PCLKA) Select)

The ICK[2:0] bits select the frequency of the system clock for the CPU, DMAC, DTC, flash memory, and SRAM, and of peripheral module clock A (PCLKA).

9.2.2 SCKSCR : System Clock Source Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x026

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

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

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

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 (PCLKA and PCLKB)

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

Check the operation state of clock sources in each product operating mode, and do not select the clock source to be stopped in SCKSCR. The clock sources should be switched when there are no occurring internal asynchronous interrupt. For details, see [section 13, Power-Saving Functions](#).

9.2.3 MOSCCR : Main Clock Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x032

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

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

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

Note 1. MOMCR register must be set before setting MOSTP to 0.

The MOSCCR register controls the main clock oscillator.

MOSTP bit (Main Clock Oscillator Stop)

The MOSTP bit starts or stops the main clock oscillator.

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. After setting the MOSTP bit to 0, confirm that the OSCSF.MOSCSF bit is set to 1 before using the main clock oscillator.

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

The following restrictions apply when starting and stopping operation:

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

9.2.4 SOSCCR : Sub-Clock Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x480

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

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

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

Note 1. 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.

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
- After setting the SOSTP bit to 0, use the sub-clock only after the sub-clock oscillation stabilization time (t_{SUBOSCWT}) has elapsed.
- 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 or Deep Software Standby mode
- When a transition to Software Standby or Deep 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).
- VBBR.VBBEN = 1 (VBBC operation enable) and VBBR.CLKSEL = 1 (SOSC is selected as VBBC operating clock).

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

Base address: SYSC = 0x4001_E000

Offset address: 0x490

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

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

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

The LOCOCR register controls the LOCO clock.

LCSTP bit (LOCO Stop)

The LCSTP bit starts or stops the LOCO clock.

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

The following restrictions apply when starting and stopping operation:

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

Writing 1 to LCSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 010b (system clock source = LOCO).
- VBBCCR.VBBEN = 1 (VBBC operation enable) and VBBCCR.CLKSEL = 0 (LOCO is selected as VBBC operating clock).

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

9.2.6 HOCOCCR : High-Speed On-Chip Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x036

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

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

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

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 you are using the HOCO (HCSTP = 0), set the OFS1.HOCOFREQ bit to an optimum value, and set the HOCOMCR.HCFREQ[1:0] bits.

The HOCOCCR register controls the HOCO clock.

HCSTP bit (HOCO Stop)

The HCSTP bit starts or stops the HOCO clock.

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

Placing the MCU in the operating mode (OPE) or software standby mode (SSTBY) in the minimum power supply mode(MINPWON) while the HOCO is operating (HOCOCCR.HCSTP = 0) is prohibited. Stop the HOCO (HOCOCCR.HCSTP = 1) before placing the MCU in OPE or SSTBY mode in MINPWON. For details, see [section 13, Power-Saving Functions](#).

The following limitations apply when starting and stopping operation:

- After stopping the HOCO clock, confirm that the OSCSF.HOCOSF is 0 before restarting the HOCO clock.
- Confirm that the HOCO clock operates and that the OSCSF.HOCOSF is 1 before stopping the HOCO clock.

- Regardless of whether the HOCO clock is selected as the system clock, confirm that the OSCSF.HOCOSF is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode.
- When a transition to Software Standby or Deep Software Standby mode is to follow the setting of the HOCO clock to stop, confirm that the OSCSF.HOCOSF is set to 0 after setting the HOCO clock and before executing the WFI instruction.

Writing 1 to HCSTP is prohibited under the following condition:

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

9.2.7 HOCOMCR : High-Speed On-Chip Oscillator Mode Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x037

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

Bit	Symbol	Function	R/W
1:0	HCFRQ[1:0]	HOCO Oscillation Frequency Setting 0 0: 24 MHz 0 1: 32 MHz 1 0: 48 MHz 1 1: 64 MHz	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

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

Note 1. Depends on the setting of OFS1.HOCOFRQ.

The High-Speed On-Chip Oscillator Mode Control Register controls the oscillation frequency of the HOCO. Writing to the HOCOMCR register is prohibited when the HOCOCR.HCSTP bit is 0 (operate the HOCO clock).

HCFRQ[1:0] bits (HOCO Oscillation Frequency Setting)

The HCFRQ[1:0] bits select the oscillation frequency of the HOCO.

The frequency of the HOCO can also be selected by OFS1.HOCOFRQ, and the value of OFS1.HOCOFRQ is applied after a reset release. 48 MHz and 64 MHz cannot be set by OFS1.HOCOFRQ.

To make the HOCO oscillate at 48 MHz or 64 MHz, set HOCOMCR.HCFRQ[1:0] to 10b (48 MHz) or 11b (64 MHz) in boost mode. Setting HOCOMCR.HCFRQ[1:0] to 10b (48 MHz) or 11b (64 MHz) in a mode other than boost mode is prohibited.

Additionally, writing to the PWSTCR register is prohibited while the HOCO oscillates at 48 MHz or 64 MHz.

Table 9.4 shows the relationship between operating modes and HOCO oscillation frequency select.

Table 9.4 Relationship between operating modes and HOCO oscillation frequency select

HOCO oscillation frequency setting		Normal mode	Boost mode
24 MHz	Selectable by OFS1.HOCOFRQ or HOCOMCR.HCFRQ[1:0]	Possible	Possible
32 MHz		Possible	Possible
48 MHz	Selectable by HOCOMCR.HCFRQ[1:0]	Not possible	Possible
64 MHz		Not possible	Possible

9.2.8 MOCOCCR : Middle-Speed On-Chip Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x038

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

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

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

The MOCOCCR register controls the MOCO clock.

MCSTP bit (MOCO Stop)

The MCSTP bit starts or stops the MOCO clock.

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

The following restrictions apply when starting and stopping the oscillator:

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

9.2.9 FLLCR1 : FLL Control Register1

Base address: SYSC = 0x4001_E000

Offset address: 0x039

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

Bit	Symbol	Function	R/W
0	FLLCN	FLL Enable 0: FLL function is disabled 1: FLL function is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: HOCO must be stopped (HOCOCR.HCSTP = 1) before FLLCR1.FLLEN is modified.
 Note: SOSC must be operating with stabilization while FLL is enabled (FLLCR1.FLLEN = 1).
 Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The FLLCR1 register controls the FLL function of the HOCO.

FLLEN bit (FLL Enable)

This bit enables or disables the FLL function of the HOCO.

If FLL is enabled, the frequency accuracy is guaranteed after FLL is stabilized. The FLL stabilization can be checked by the CAC frequency measurement, but it must be executed after HOCO stabilization.

In addition, you must disable FLL by setting the FLLEN bit to 0 before transitioning to Software Standby mode.

Table 9.5 show an example flow of the FLL setting for each case.

Table 9.5 FLL setting flow

Step	Operation
1	Start (After reset release)
2	Operate the SOSC (SOSCCR.SOSTP = 0)
3	Wait for the SOSC oscillation stabilization
4	Enable FLL function (FLLCR1.FLLEN = 1)
5	Operate the HOCO clock (HOCOCR.HCSTP = 0)
6	Wait for the HOCO frequency correction stabilization
7	Wait for the HOCO oscillation stabilization (OSCSF.HOCOSF = 1)
8	End (possible to use HOCO)

9.2.10 OSCSF : Oscillation Stabilization Flag Register

Base address: SYSC = 0x4001_E000

Offset address: 0x03C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MOSC SF	—	—	HOCO SF

Value after reset: 0 0 0 0 0 0 0 0/1^{*1}

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

Note 1. The value after reset depends on the OFS1.HOCOEN setting.

When OFS1.HOCOEN = 1 (disable HOCO), the value after reset of HOCOSF is 0.

When OFS1.HOCOEN = 0 (enable HOCO), the HOCOSF value is set to 0 immediately after reset is released, and the HOCOSF value is set to 1 after the HOCO oscillation stabilization wait time elapses.

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

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

HOCOSF flag (HOCO Clock Oscillation Stabilization Flag)

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

[Setting condition]

- When the HOCO clock is stopped and the HOCOCR.HCSTP bit is set to 0, and then the HOCO oscillation stabilization time is counted by the LOCO clock and supply of the HOCO clock within the MCU is started. For the HOCO oscillation stabilization time, see [section 51, Electrical Characteristics](#).

[Clearing condition]

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

MOSCSF flag (Main Clock Oscillation Stabilization Flag)

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

[Setting condition]

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

[Clearing condition]

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

9.2.11 OSTDCR : Oscillation Stop Detection Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x040

Bit position:	7	6	5	4	3	2	1	0
Bit field:	OSTD E	—	—	OSTD DIVSE L	—	—	—	OSTDI E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OSTDIE	Oscillation Stop Detection Interrupt Enable 0: Disable oscillation stop detection interrupt (do not notify the POE) 1: Enable oscillation stop detection interrupt (notify the POE)	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	OSTDDIVSEL	Oscillation Stop Detection Main Clock Frequency Select 0: 8 MHz ≤ frequency of the main clock oscillator ≤ 16 MHz 1: 16 MHz < frequency of the main clock oscillator	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	OSTDE	Oscillation Stop Detection Function Enable 0: Disable oscillation stop detection function 1: Enable oscillation stop detection function	R/W

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

The OSTDCR register controls the oscillation stop detection function.

OSTDIE bit (Oscillation Stop Detection Interrupt Enable)

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

If the Oscillation Stop Detection flag in the Oscillation Stop Detection Status Register (OSTDSR.OSTDF) requires clearing, set the OSTDIE bit to 0 before clearing OSTDF. Wait for at least 2 PCLKB cycles before setting the OSTDIE bit to 1. By

reading the I/O register whose access cycle number is defined by PCLKB, it is possible to secure waiting time of 2 or more cycles of PCLKB.

OSTDDIVSEL bit (Oscillation Stop Detection Main Clock Frequency Select)

Set according to the main clock oscillator frequency. Change this setting while the OSTDE bit is 0. Set this bit to 0 when the frequency of the main clock oscillator is between 8 and 16 MHz, inclusive. Set this bit to 1 when the frequency of the main clock oscillator is above 16 MHz.

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 (MOCOCCR.MCSTP) is set to 0 and the MOCO operation starts. The MOCO clock cannot be stopped while the oscillation stop detection function is enabled. Writing 1 to the MOCOCCR.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 or Deep Software Standby mode. To transition to Software Standby or Deep Software Standby mode, first set the OSTDE bit to 0, then execute the WFI instruction.

9.2.12 OSTDSR : Oscillation Stop Detection Status Register

Base address: SYSC = 0x4001_E000

Offset address: 0x041

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

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

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

Note 1. This bit can only be set to 0. To clear this bit, write 0 two consecutive times.

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

OSTDF flag (Oscillation Stop Detection Flag)

The OSTDF flag indicates the main clock oscillator status. When this flag is 1, it indicates that the main clock oscillation stop was detected. After this stop is detected, the OSTDF flag is not set to 0 even when the main clock oscillation is restarted. To set the OSTDF flag to 0, write 0 two consecutive times.

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

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

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

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

[Setting condition]

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

[Clearing condition]

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

9.2.13 MOSCWTCR : Main Clock Oscillator Wait Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0A2

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

Bit	Symbol	Function	R/W
3:0	MSTS[3:0]	Main Clock Oscillator Wait Time Setting 0x0: Setting prohibited 0x1: Wait time = 2 cycles 0x2: Wait time = 5 cycles 0x3: Wait time = 13 cycles 0x4: Wait time = 29 cycles 0x5: Wait time = 61 cycles 0x6: Wait time = 125 cycles 0x7: Wait time = 253 cycles 0x8: Wait time = 509 cycles 0x9: Wait time = 1021 cycles Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

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

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

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

Set the main clock oscillation stabilization time to a period longer than or equal to the stabilization time recommended by the oscillator manufacturer. When the main clock is input externally, set these bits to 0x1 because the oscillation stabilization time is not required.

The wait time set in these bits is counted using: 1 cycle (μ s) = 1/(fLOCO_max [MHz]) (min.) (fLOCO_max: maximum frequency for LOCO) The LOCO clock automatically oscillates when necessary, regardless of the value of the LOCO.LCSTP 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.

9.2.14 MOMCR : Main Clock Oscillator Mode Oscillation Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x413

Bit position:	7	6	5	4	3	2	1	0
Bit field:	OSCL PEN	MOSE L	MODRV[2:0]			—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
5:3	MODRV[2:0] ^{*1}	Main Clock Oscillator Drive Capability Switching 0 0 0: Setting prohibited 0 0 1: Oscillation current is small ⋮ 1 1 1: Oscillation current is large	R/W
6	MOSEL ^{*2}	Main Clock Oscillator Switching 0: Resonator 1: External clock input	R/W
7	OSCLPEN	Main Clock Oscillator Low Consumption Oscillation Function Enable 0: Disable 1: Enable	R/W

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

Note: This register can be written while MOSCCR.MOSTP is 1 (MOSC stopped).

Note 1. An optimum drive capacity must be set according to the resonator to be used and the load capacitance of the system.

Note 2. The EXTAL and XTAL pins are also used as general ports. In the initial state, the port function is selected.

MODRV[2:0] bit (Main Clock Oscillator Drive Capability Switching)

The MODRV[2:0] bits switch the drive capability of the main clock oscillator.

As the setting value of this bit is increased from 001b to 111b, the oscillation current increases. Adjust as appropriate for your system.

MOSEL bit (Main Clock Oscillator Switching)

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

OSCLPEN bit (Main Clock Oscillator Low Consumption Oscillation Function Enable)

The OSCLPEN bit controls the low consumption oscillation function of the main clock oscillator.

When OSCLPEN = 1, oscillation is performed at a lower current compared to OSCLPEN = 0.

9.2.15 SOMCR : Sub-Clock Oscillator Mode Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x481

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SONF STP	—	—	SODR V	—

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	SODRV	Sub-Clock Oscillator Drive Capability Switching 0: Standard CL 1: Low CL	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	SONFSTP	Sub-Clock Oscillator Noise Filter Disable 0: Noise filter is enabled 1: Noise filter is disabled	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

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

When changing the bits other than SONFSTP, do so while SOSCCR.SOSTP is 1 (SOSC stopped).

SODRV bit (Sub-Clock Oscillator Drive Capability Switching)

The SODRVbit switches the drive capability of the sub-clock oscillator.

SONFSTP bit (Sub-Clock Oscillator Noise Filter Disable)

The SONFSTP bit enables or disables the noise filter for the sub-clock oscillator.

When changing this bit while SOSCCR.SOSTP is 0 (SOSC is enabled), do so by executing the instructions from SRAM.

Do not execute the instructions from the flash memory.

The current consumption can be reduced by setting this bit to 1 to disable the noise filter. However, the sub-clock becomes easily affected by noise, only use the sub-clock after thorough evaluation.

9.2.16 CKOCR : Clock Out Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x03E

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

Bit	Symbol	Function	R/W
2:0	CKOSEL[2:0]	Clock Out Source Select 0 0 0: HOCO 0 0 1: MOCO 0 1 0: LOCO 0 1 1: MOSC 1 0 0: SOSC 1 0 1: CCC_2K Others: Setting prohibited	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	CKODIV[2:0]	Clock Output Frequency Division Ratio 0 0 0: × 1/1 0 0 1: × 1/2 0 1 0: × 1/4 0 1 1: × 1/8 1 0 0: × 1/16 1 0 1: × 1/32 1 1 0: × 1/64 1 1 1: × 1/128	R/W
7	CKOEN	Clock Out Enable 0: Disable clock out 1: Enable clock out	R/W

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

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

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

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

The CKODIV[2:0] bits specify the clock division ratio. Set the CKOEN bit to 0 when changing the division ratio. The division ratio of the output clock frequency must be set to a value no higher than the characteristics of the CLKOUT pin output frequency. For details on the characteristics of the CLKOUT pin, see [section 51, 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 or Deep Software Standby mode if the selecting clock out source clock is stopped in that mode.

9.2.16.1 Setting Procedure for Clock Output

Set the clock output by the following procedure.

(1) When selecting other than CCC_2K as clock output source

1. Set the CKOEN bit to 0 (clock output is prohibited).
2. Enable the oscillation of clock output source. When it has already oscillated, skip this step.
3. Set the associated value of the source clock to the CKOSEL[2:0] bits and set division ratio to the CKODIV[2:0] bits.
4. Check the stability of the source clock.
5. Set the CKOEN bit to 1 and enable clock output.

(2) When selecting CCC_2K as clock output source

1. Set the CKOEN bit to 0 (clock output is prohibited).
2. Enable the oscillation of sub-clock oscillator. When it has already oscillated, skip this step.
3. Wait the oscillation stabilization of the sub-clock oscillator.
4. Set the R128CTRL.CADJUSCEN bit to 1 and enable CCC.
5. Set the associated value of CCC_2K to the CKOSEL[2:0] bits and set division ratio to the CKODIV[2:0] bits.
6. Set the CKOEN bit to 1 and enable clock output.

9.2.17 CKO32CR : Clock Output 32-kHz Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x03D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CKO32EN	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	CKO32EN	Clock Output 32-kHz Enable 0: Disable clock output for 32 kHz 1: Enable clock output for 32 kHz.	R/W

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

The CKO32CR register controls output from the CLKOUT32K pin.

CKO32EN bit (Clock Output 32-kHz Enable)

The CKO32EN bit enables output from the CLKOUT32K pin.

When this bit is set to 1, the clock of the sub-clock oscillator is output. When this bit is set to 0, low is output.

When changing this bit, confirm that the clock of the sub-clock oscillator is stable. Otherwise, a glitch might be generated in the output.

The CKO32EN bit must be cleared before entering Software Standby or Deep Software Standby mode if the clock output source of the selected sub-clock oscillator is stopped in that mode.

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

Because the resistor values vary according to the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If the manufacturer recommends using an external feedback resistor (R_f), insert an R_f 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.

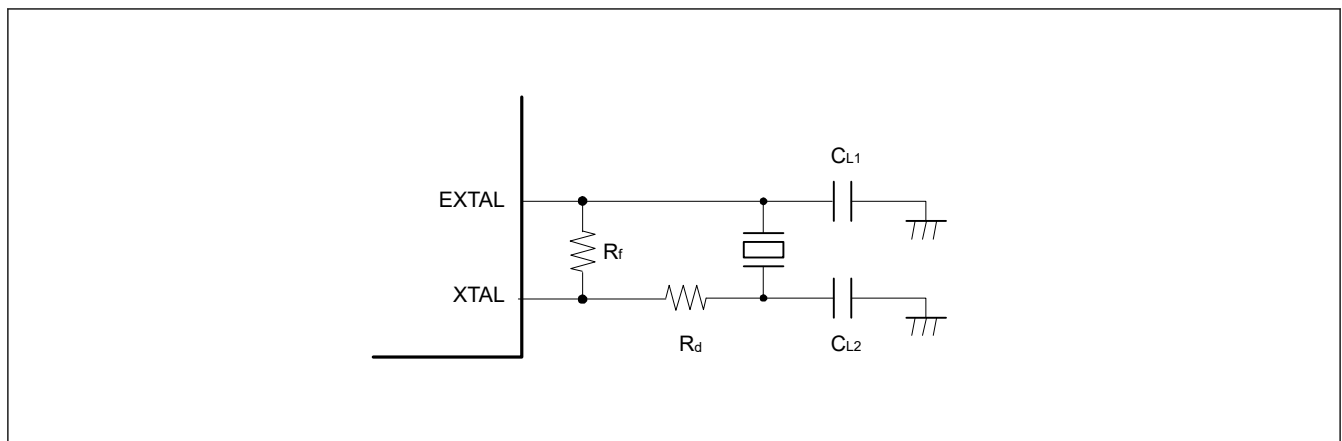


Figure 9.2 Example of crystal resonator connection

Figure 9.3 shows an equivalent circuit of the crystal resonator.

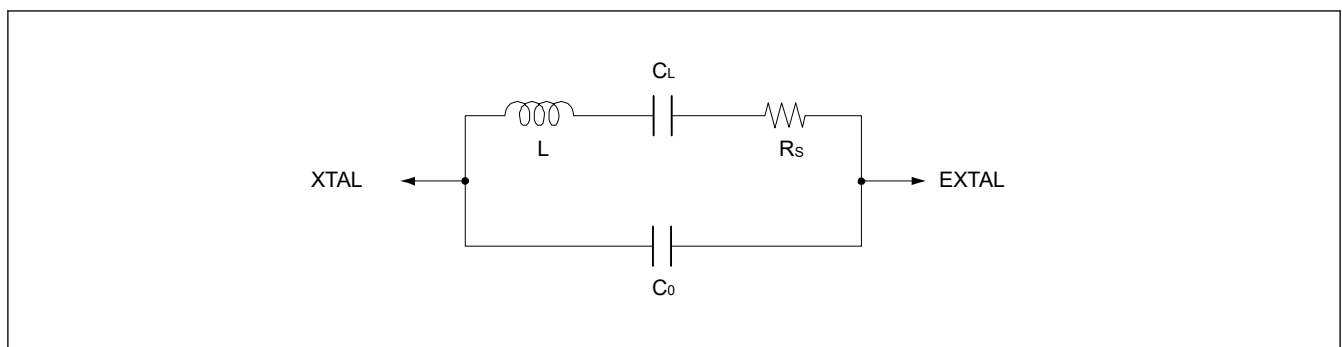


Figure 9.3 Equivalent circuit of the crystal resonator

9.3.2 External Clock Input

Figure 9.4 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.

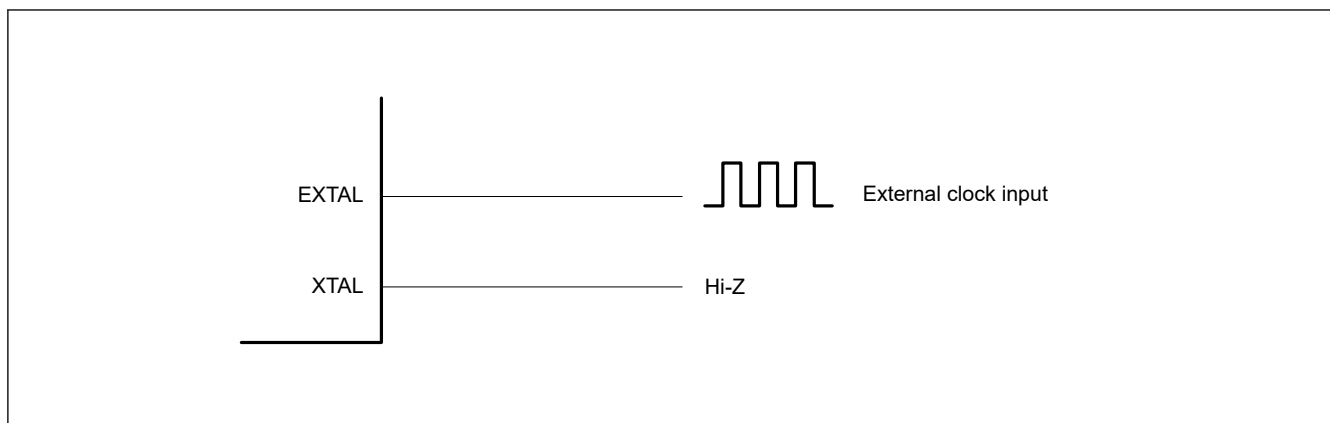


Figure 9.4 Equivalent circuit for external clock

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 9.5](#). A damping resistor (R_d) 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 (R_f), insert an R_f 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](#).

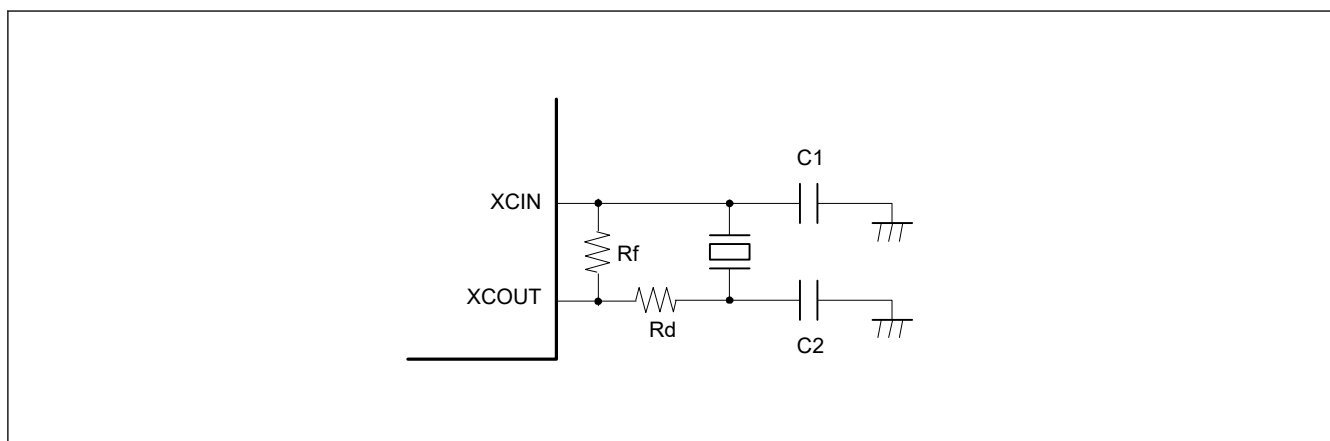


Figure 9.5 Connection example of 32.768-kHz crystal resonator

[Figure 9.6](#) shows an equivalent circuit for the 32.768-kHz crystal resonator.

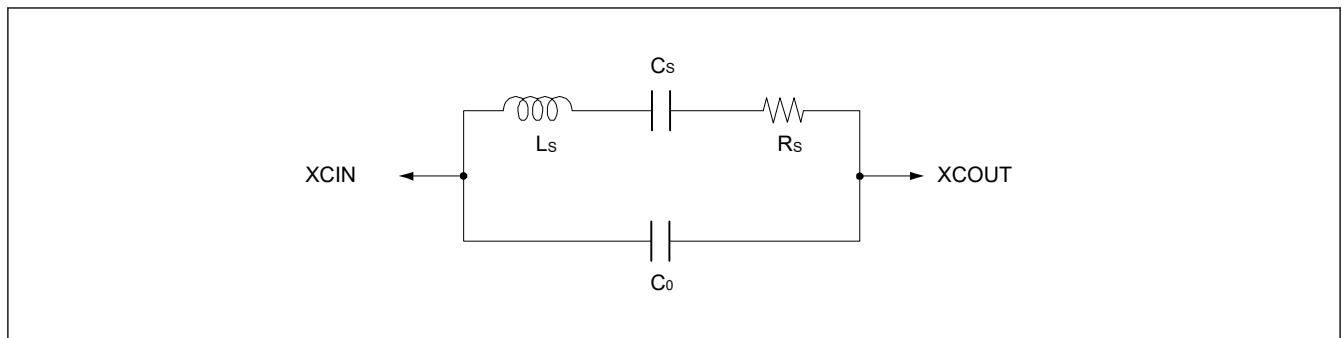


Figure 9.6 Equivalent circuit for the 32.768-kHz crystal resonator

9.4.2 Pin Handling When the Sub-Clock Oscillator Is Not Used

When the sub-clock oscillator is not in use, connect the XCIN pin to VSS through a resistor (to pull VSS down) and leave the XCOUT pin open as shown in [Figure 9.7](#). In addition, if an oscillator is not connected, set the Sub-Clock Oscillator Stop bit (SOSCCR.SOSTP) to 1 to stop the oscillator.

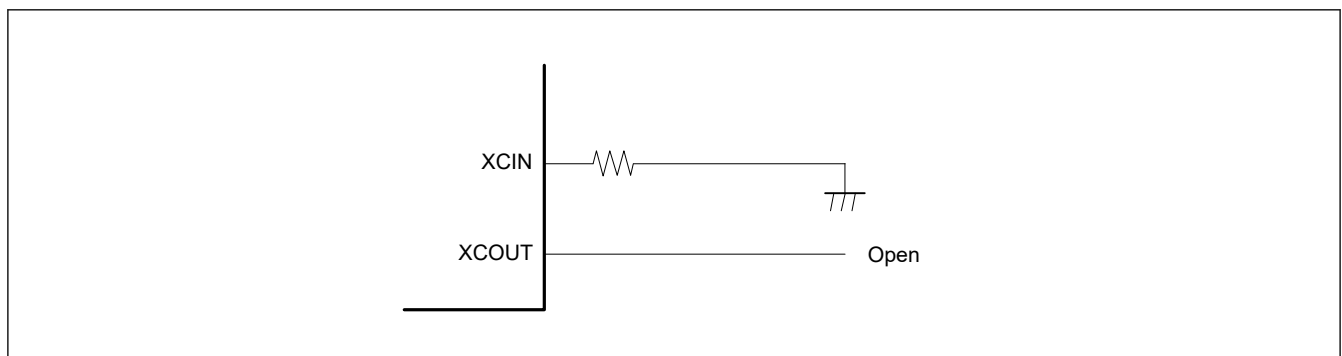


Figure 9.7 Pin handling when the sub-clock oscillator is not used

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 oscillation stop detection flag (OSTDSR.OSTDF) changes from 0 to 1. When the main clock is selected, the system clock and the CAC main clock are switched to the MOCO clock. [Figure 9.8](#) shows the flow for returning the system clock from the MOCO clock to the main clock after detecting oscillation stop.

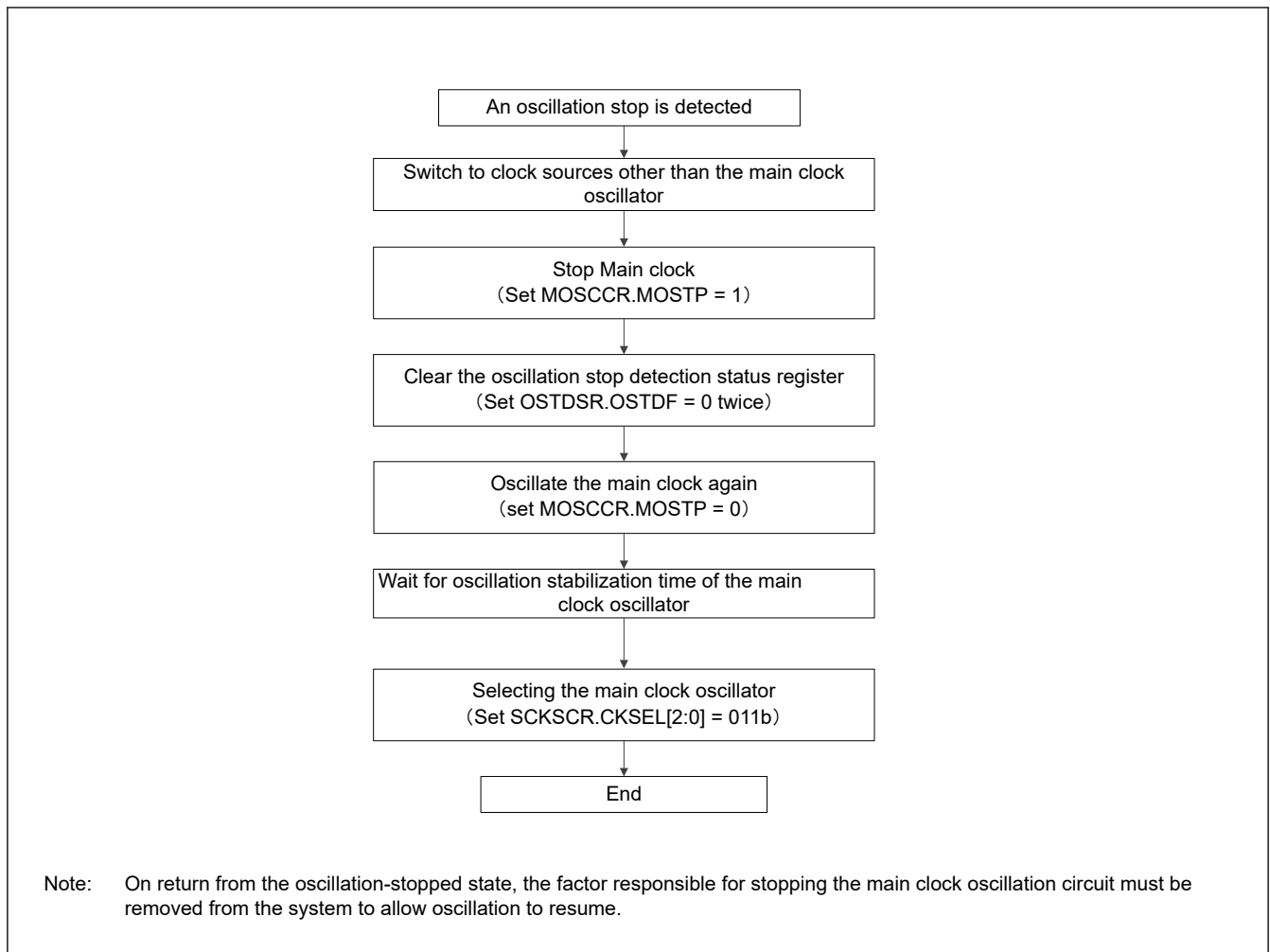


Figure 9.8 Flow of recovery on detection of oscillator stop

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 (POE) is notified of the main clock oscillator stop. On receiving the notification, the POE sets the Oscillation Stop Detection Flag in the POE Group n Setting Register (POEGGn.OSTPF) to 1 (n = A, B).

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 16, Interrupt Controller Unit \(ICU\)](#).

9.6 Internal Clock

Clock sources for the internal clock signals include:

- Main clock
- SOSC clock
- HOCO clock
- MOCO clock
- LOCO clock
- IWDTLOCO clock
- CCC_2K clock

The following internal clocks are produced from these sources.

- Operating clock for the CPU, DTC, flash memory, SRAM, and peripheral modules: system clock (ICLK)/peripheral module clock (PCLKA)
- Operating clock for the peripheral modules: Peripheral module clock (PCLKB)
- Operating clock for the CAC: CAC clock (CACCLK)
- Operating clock for the VBBC: LOCO clock (LOCOCLK)
- Operating clock for the RTC/VBBC/WUPT: SOSC clock (SOSCCLK)
- Operating clock for the AGT: AGT-dedicated LOCO clock (AGTLCLK)
- Operating clock for the AGT: AGT-dedicated sub-clock (AGTSCLK)
- Operating clock for the SysTick Timer: SysTick Timer-dedicated clock (SYSTICCLK)
- Operating clock for the IWDG: IWDG clock (IWDGCLK)
- Operating clock for the WDT: WDT clock (WDTCLK)
- Clock for the CLKOUT pin output: External pin output clock (CLKOUT)
- Clock for the CLKOUT32K pin output: External pin output sub-clock (CLKOUT32K)
- CCC count clock: CCC-dedicated sub-clock (CCC32K)
- Clock for boundary scan: JTAG clock (JTAGTCK)
- Serial wire debug clock: External input serial wire debug clock (SWCLK)

For details of the registers used to set the frequencies of the internal clocks, see [section 9.2. Register Descriptions](#). If the value of any of these bits is changed, subsequent operation is at the frequency determined by the new value.

9.6.1 System Clock (ICLK)/Peripheral Module Clock (PCLKA)

The system clock (ICLK) and peripheral module clock (PCLKA) are the operating clocks for the modules shown in [Table 9.2](#). Specify the frequency in the following bits:

- SCKDIVCR.ICK[2:0] bits
- SCKSCR.CKSEL[2:0] bits
- OFS1.HOCOFREQ bit and HOCOMCR.HCFREQ[1:0] bits

When the ICLK clock source is switched, the duration of the ICLK clock cycle becomes longer during the clock source transition period. See [Figure 9.9](#) and [Figure 9.10](#).

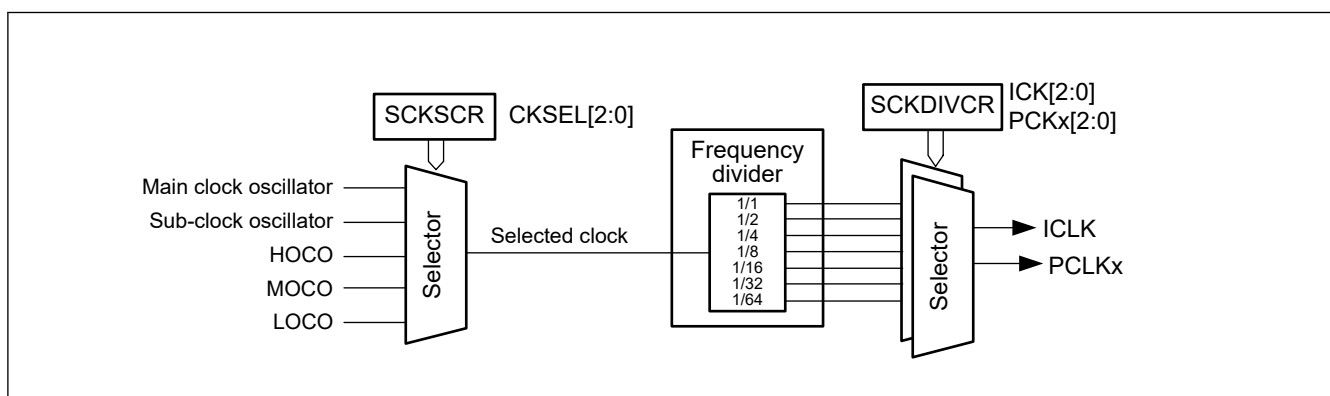


Figure 9.9 Block diagram of clock source selector

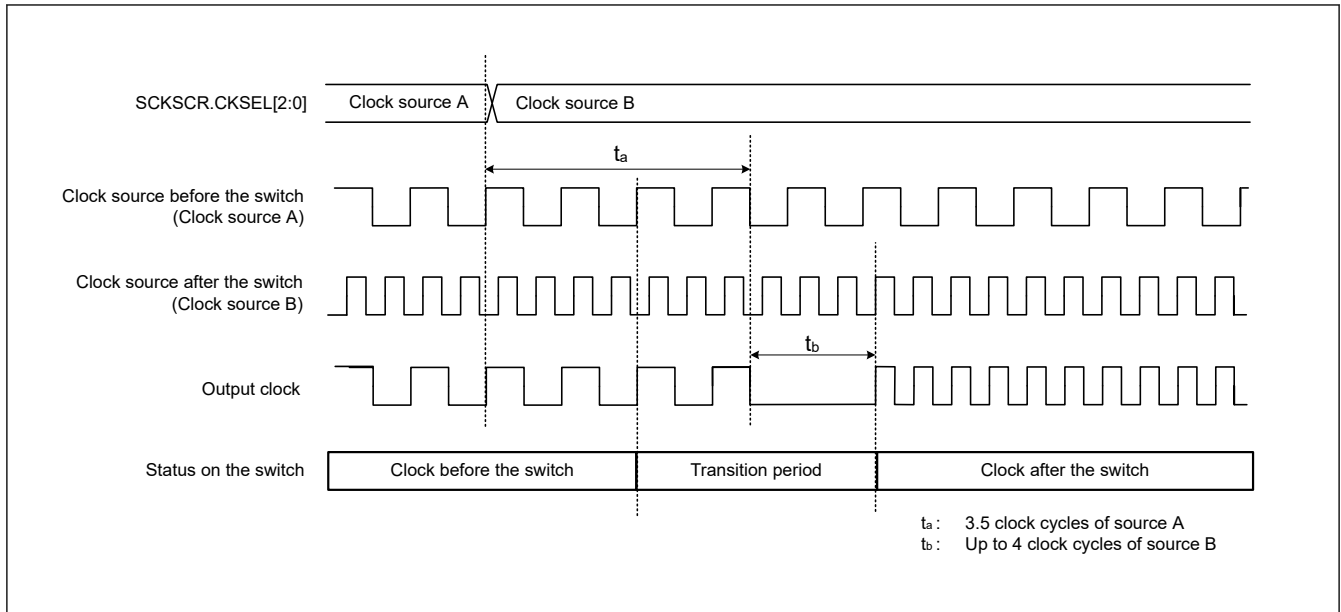


Figure 9.10 Timing of clock source switching

9.6.2 Peripheral Module Clock (PCLKB)

The peripheral module clock, PCLKB, is the operating clock for the peripheral modules other than peripheral modules to which PCLKA is supplied. Specify the frequency in the following bits:

Do not set PCLKB to a frequency higher than ICLK/PCLKA.

- SCKDIVCR.PCKB[2:0] bits
- SCKSCR.CKSEL[2:0] bits
- OFS1.HOCOFREQ bit and HOCOMCR.HCFREQ[1:0] bits

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 9.9](#) and [Figure 9.10](#).

9.6.3 CAC Clock (CACCLK)

The CAC clock, CACCLK, is an 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)
- IWDG-dedicated on-chip oscillator. (IWDGLOCO)
- CCC_2K

9.6.4 SOSC Clock (SOSC), LOCO Clock (LOCOCLK)

The SOSC and LOCO clocks are the operating clocks for RTC/VBBC. (RTC use only SOSC)

SOSCCLK is generated by the sub-clock oscillator, and LOCOCLK is generated by the LOCO clock.

9.6.5 IWDG-Dedicated Clock (IWDGCLK)

The IWDG-dedicated clock, IWDGCLK, is the operating clock for the IWDG. IWDGCLK is internally generated by the IWDG-dedicated on-chip oscillator.

9.6.6 WDT Clock (WDTCLK)

The WDT clock, WDTCLK, is the operating clock for the WDT. WDTCLK is selected from PCLKB or CCC_2K. The OFS0 register is used to select a clock. For details, see [section 7, Option-Setting Memory](#).

9.6.7 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.

9.6.8 SysTick Timer-Dedicated Clock (SYSTICCLK)

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

9.6.9 External Pin 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
- OFS1.HOCOFREQ bit

9.6.10 External Pin Output Sub-Clock (CLKOUT32K)

The sub-clock output, CLKOUT32K, externally outputs the clock generated by the sub-clock oscillator. When CKO32CR.CKO32EN is set to 1, CLKOUT32K is output to the CLKOUT32K pin.

9.6.11 CCC-Dedicated Sub-Clock (CCC32K)

The CCC-dedicated sub-clock, CCC32K, is the count clock of the CCC. CCC32K is generated by the sub-clock oscillator.

9.6.12 JTAG Clock (JTAGTCK)

The JTAG clock (JTAGTCK) is the clock for the JTAG.

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

9.6.13 Serial Wire Debug Clock (SWCLK)

The serial wire debug clock, SWCLK, is the clock for serial wire debugging.

The serial wire debug clock is generated by the external input serial wire debug clock (SWCLK).

9.7 Usage Notes

9.7.1 Notes on register access

The registers described in this chapter are protected from register access by the register write protection function. Remove the protection before accessing. For details, see [section 15, Register Write Protection](#).

9.7.2 Notes on Clock Generation Circuit

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

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

Each frequency must meet the following conditions:

- Select each frequency that is within the operation-guaranteed range of the Operating frequency (f) specified in the AC characteristics. See [section 51, Electrical Characteristics](#).
- Set the frequencies of ICLK/PCLKA and PCLKB so that the following relations are satisfied.
 - Restriction on the clock frequency settings: $\text{ICLK/PCLKA} \geq \text{PCLKB}$
 - Restriction on the clock frequency ratio (N: integer, and up to 64): $\text{ICLK/PCLKA:PCLKB} = \text{N:1}$

In order to certainly execute the subsequent processing after changing the clock frequency by rewriting the SCKDIVCR and SCKSCR registers, the values after the change of these registers must be checked. Confirm that the value read from these registers is the specified value before executing the subsequent processing.

9.7.3 Notes on Resonator and Oscillator

Because various resonator characteristics relate closely to your board design, adequate evaluation is required before use. See the resonator connection example in [Figure 9.5](#). Resonator circuit constants, oscillator drive capability and oscillation stabilization wait time vary depending on the resonator used and the load capacitance of the mounted circuit. Perform matching evaluation with the manufacturer of the resonator to be used, and set an appropriate value. [Table 9.6](#) lists the registers for which appropriate setting values should be determined by matching evaluation.

Table 9.6 Registers that affect matching evaluation

Applicable register	Control contents
MOMCR.MODRV[2:0] bits	The MODRV [2: 0] bits can control the drive capability of the main clock oscillator. Set the drive capacity according to the system.
SOMCR.SODRV bit	The SODRV bit can control the driving capability of the subclock oscillator. Set the drive capacity according to the system.

9.7.4 Notes on Board Design

When using a crystal resonator, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in [Figure 9.11](#) to prevent electromagnetic induction from interfering with correct oscillation.

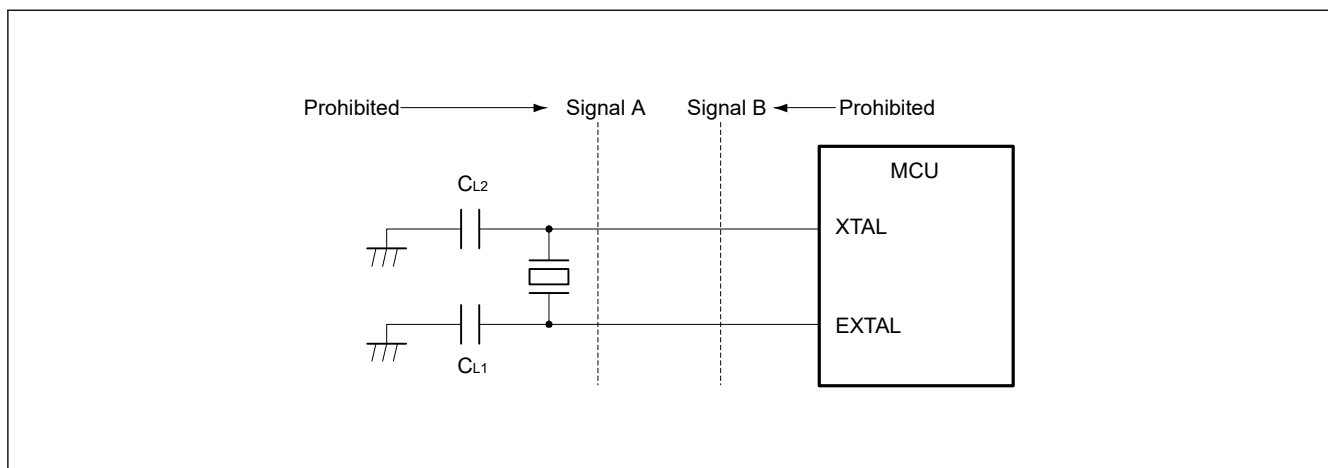


Figure 9.11 Signal routing in board design for oscillation circuit (applies to the sub-clock oscillator for the main clock oscillator)

9.7.5 Notes on Resonator Connect Pin

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

9.7.6 Notes on using the debugger

When the debugger is connected, the LOCO clock and HOCO clock may be supplied regardless of the setting values of the LOCOCR.LCSTP bit and HOCOCR.HCSTP bit. If the operation differs between debug mode and user mode, check the setting value of the SCKSCR.CKSEL[2:0] bits.

10. Clock Correction Circuit (CCC)

10.1 Overview

Clock Correction Circuit (CCC) provides a function to correct the frequency accuracy of the sub-clock oscillator by the period of 16 seconds. Periodic interrupts and overflow interrupts based on the corrected clock are available.

Table 10.1 Clock correction circuit specifications

Parameter	Specifications
Clock input (before correction)	Sub-clock 32.768 kHz
Clock output (after correction)	Output of 2.048 kHz and 512 Hz are possible
Signal output pin (CCCOUT)*1	Output of 512 Hz or 1 Hz is possible
Interrupts*2	- Periodic interrupts (CCC_PRD) Interrupt periods are selectable from 1, 1/2, 1/4, 1/16, 1/64, and 1/128 seconds. The flags can be set in response to each interrupt. - Overflow interrupt (CCC_CUP) On occurrence of a 128-Hz counter overflow - Read error interrupt (CCC_ERR) When the timings of counting up and reading from the 128-Hz counter overlapped.
Event link function	- Periodic event output (ELC_CPRD) - Overflow event output (ELC_CCUP)
Sub-clock correction	- Correction period: 16 seconds - Correction accuracy: 1.91 ppm - Correction range: -487 to +487 ppm - Correction operation: correct at specified periods
Binary display of sub-second range	States of 1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, 64 Hz, and 128 Hz can be confirmed by reading the 128-Hz counter.

Note 1. Output is possible even in software standby or deep software standby mode.

Note 2. Issuance is possible even in software standby or deep software standby mode.

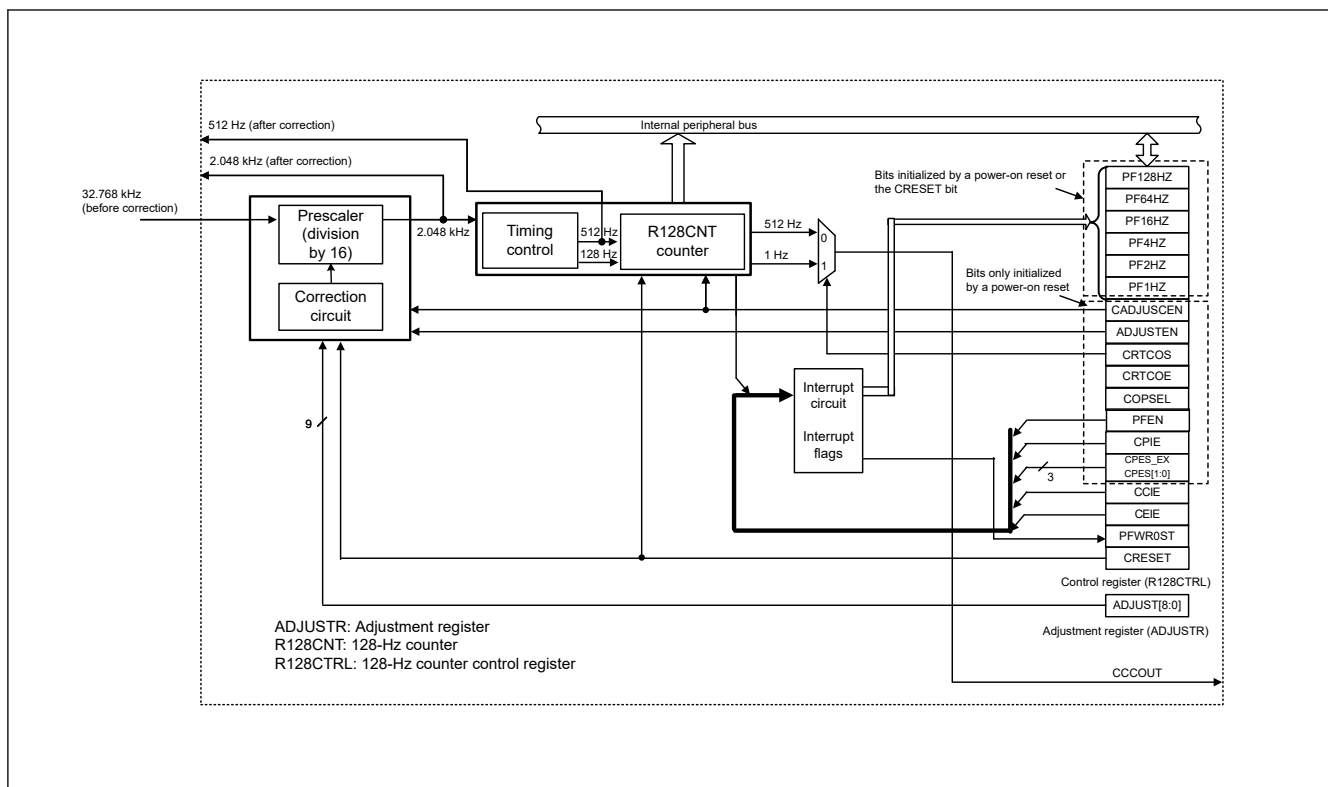


Figure 10.1 CCC block diagram

Table 10.2 CCC I/O pin

Pin name	I/O	Description
CCCOUT	Output	Outputs 1-Hz and 512-Hz waveforms.

10.2 Register Descriptions

10.2.1 ADJUSTR : Adjustment Register

Base address: CCC = 0x4008_0400

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADJUST[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	ADJUST[8:0]	Adjustment These bits specify the value to adjust for corrections. When adjusting the negative side correction, set in the range of 0x001 to 0x0FF. When adjusting the positive side correction, set in the range of 0x1FE to 0x100.	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: See [section 10.4.4. Notes on Reading and Writing Registers](#).

The Adjustment Register is only initialized by a power-on reset.

ADJUST[8:0] bits (Adjustment)

The ADJUST[8:0] bits specify the value and method to use in adjusting. Adjustment for errors is on average made within every 16 seconds.

10.2.2 R128CNT : 128-Hz Counter

Base address: CCC = 0x4008_0400

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CF1H Z	CF2H Z	CF4H Z	CF8H Z	CF16H Z	CF32H Z	CF64H Z	CF128 HZ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CF128HZ	Toggles every 128 Hz	R
1	CF64HZ	Toggles every 64 Hz	R
2	CF32HZ	Toggles every 32 Hz	R

Bit	Symbol	Function	R/W
3	CF16HZ	Toggles every 16 Hz	R
4	CF8HZ	Toggles every 8 Hz	R
5	CF4HZ	Toggles every 4 Hz	R
6	CF2HZ	Toggles every 2 Hz	R
7	CF1HZ	Toggles every 1 Hz	R
31:8	—	These bits are read as 0.	R

The 128-Hz Counter counts upward based on the 128-Hz clock. The state in the sub-second range can be confirmed by reading this register. Setting the R128CTRL.CRESET bit to 1 causes a software reset, that clears the counter to 0x0000. This register is initialized by a power-on reset, but not by a RES# pin reset. Read this counter following the steps described in section [section 10.3.3. Procedure for Reading from 128-Hz Counter](#).

10.2.3 R128CTRL : 128-Hz Counter Control Register

Base address: CCC = 0x4008_0400

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PFEN	—	—	—	—	—	—	—	—	PFWR0ST	PF1HZ	PF2HZ	PF4HZ	PF16HZ	PF64HZ	PF128HZ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CPES_EX	CEIE	CRTC OE	CPES[1:0]	CPIE	CCIE	CRTC OS	COPSEL	CRESET	ADJUSTEN	CADJUSCEN	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CADJUSCEN	CCC Enable 0: Stop* ¹ 1: Operate	R/W
1	ADJUSTEN	Frequency Correction Enable 0: Disable CCC* ¹ 1: Enable CCC	R/W
2	CRESET	Software Reset 0: No effect 1: Initialize the CCC and the 128-Hz counter	R/W
3	COPSEL	CCCOUT Output Port Select 0: Output is from CCCOUT_A* ¹ 1: Output is from CCCOUT_B	R/W
4	CRTCOS	CCCOUT Output Select 0: 1 Hz* ¹ 1: 512 Hz	R/W
5	CCIE	Overflow Interrupt Enable 0: Disable overflow interrupt request 1: Enable overflow interrupt request	R/W
6	CPIE	Periodic Interrupt Enable 0: Disable periodic interrupt request* ¹ 1: Enable periodic interrupt request	R/W
8:7	CPES[1:0]	Periodic Interrupt Select* ¹ For details, see Table 10.3 .	R/W

Bit	Symbol	Function	R/W
9	CRTCOE	CCCOUT Output Enable 0: Disable CCCOUT output* ¹ 1: Enable CCCOUT output	R/W
10	CEIE	Read Error Interrupt Enable 0: Disable read error interrupt request 1: Enable read error interrupt request	R/W
11	CPES_EX	Periodic Interrupt Select* ¹ For details, see Table 10.3 .	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	PF128HZ	1/128-Second Periodic Interrupt Flag 0: A 1/128-second periodic interrupt has not been generated* ² 1: A 1/128-second periodic interrupt has been generated	R/W
17	PF64HZ	1/64-Second Periodic Interrupt Flag 0: A 1/64-second periodic interrupt has not been generated* ² 1: A 1/64-second periodic interrupt has been generated	R/W
18	PF16HZ	1/16-Second Periodic Interrupt Flag 0: A 1/16-second periodic interrupt has not been generated* ² 1: A 1/16-second periodic interrupt has been generated	R/W
19	PF4HZ	1/4-Second Periodic Interrupt Flag 0: A 1/4-second periodic interrupt has not been generated* ² 1: A 1/4-second periodic interrupt has been generated	R/W
20	PF2HZ	1/2-Second Periodic Interrupt Flag 0: A 1/2-second periodic interrupt has not been generated* ² 1: A 1/2-second periodic interrupt has been generated	R/W
21	PF1HZ	1-Second Periodic Interrupt Flag 0: A 1-second periodic interrupt has not been generated* ² 1: A 1-second periodic interrupt has been generated	R/W
22	PFWR0ST	Periodic Interrupt Status Flag 0: The flag has been cleared 1: The flag is being cleared	R
30:23	—	These bits are read as 0. The write value should be 0.	R/W
31	PFEN	Periodic Interrupt Enable 0: Periodic interrupt is disabled* ¹ 1: Periodic interrupt is enabled	R/W

Note: See section [section 10.4.4. Notes on Reading and Writing Registers](#).

Note 1. This bit is only initialized by a power-on reset.

Note 2. This bit is initialized by a power-on reset or the CRESET bit.

CADJUSCEN bit (CCC Enable)

The CADJUSCEN bit starts or stops the dividing of correction circuit. Make settings for the ADJUSTEN bit and the ADJUSTR register before setting this bit to 1. This bit is initialized only by a power-on reset.

ADJUSTEN bit (Frequency Correction Enable)

The ADJUSTEN bit enables or disables the correction circuit. This bit is initialized only by a power-on reset.

CRESET bit (Software Reset)

Setting the CRESET bit to 1 initializes the 128-Hz counter (R128CNT), correction circuit, internal counters, and PF128HZ to PF1HZ flags to 0. The bits other than the PF128HZ to PF1HZ flags in the 128-Hz Counter Control Register (R128CTRL), and the Adjustment Register (ADJUSTR) are not initialized by this bit.

COPSEL bit (CCCOUT Output Port Select)

The COPSEL bit selects the CCCOUT output port.*¹

CRTCOS bit (CCCOUT Output Select)

The CRTCOS bit selects the period for CCCOUT output.*¹

CCIE bit (Overflow Interrupt Enable)

The CCIE bit enables or disables overflow interrupts. Modify this bit while CCCOUT output is disabled (CRTCOE = 0).

CPIE bit (Periodic Interrupt Enable)

The CPIE bit enables or disables periodic interrupts. Modify this bit while CCCOUT output is disabled (CRTCOE = 0). This bit is initialized only by a power-on reset.

CPES_EX, CPES[1:0] bits (Periodic Interrupt Select)

These bits select the period for generating periodic interrupts. This bit is initialized only by a power-on reset. [Table 10.3](#) shows the period for generating periodic interrupts.

Table 10.3 Period for generating periodic interrupts

Setting value			Period for generating periodic interrupts
CPES_EX	CPES[1]	CPES[0]	
0	0	0	Every 1/64 seconds
0	0	1	Every 1/16 seconds
0	1	0	Every 1/4 seconds
0	1	1	Every 1 seconds
1	0	0	Every 1/2 seconds
1	0	1	Every 1/128 seconds
1	1	0	Setting prohibited (interrupts are not generated)
1	1	1	Setting prohibited (interrupts are not generated)

CRTCOE bit (CCCOUT Output Enable)

The CRTCOE bit enables CCCOUT output.*²

For output of CCCOUT to an external pin, enable this bit and make settings of the associated pin as well.

CEIE bit (Read Error Interrupt Enable)

The CEIE bit enables or disables read error interrupts.

PF128HZ to PF1HZ flags (1/128-to-1 Second Periodic Interrupt Flags)

The PF128HZ to PF1HZ flags indicate that 1/128-to-1 second periodic interrupts have been generated. These flags are set to 1 when a periodic interrupt is generated.

[Clearing conditions]

The flag is cleared to 0 by writing 0 to the associated bit or by a software reset.

When a periodic interrupt and software reset occur at the same time, the software reset is given priority over the interrupt and this flag is cleared to 0.

[Setting condition]

The flag is set to 1 when an periodic interrupt has occurred.

When a periodic interrupt conflicts with the CPU writing 0 to the flag, the periodic interrupt is given priority and this flag is set to 1.

The flag remains 0 when PFEN = 0 or CADJUSCEN = 0.

PFWR0ST flag (Periodic Interrupt Status Flag)

The PFWR0ST bit indicates clearing state of the PF1HZ to PF128HZ flags.

This bit being 1 indicates that the PF1HZ to PF128HZ flags has been set to 1 and this bit being 0 indicates that the PF1HZ to PF128HZ flags have been cleared to 0.

PFEN bit (Periodic Interrupt Enable)

The PFEN bit enables or disables the periodic interrupt flag.*2

Note 1. This bit should only be re-written while the counter is stopped (CADJUSCEN = 0) and CCCOUT output is disabled (CRTCOE = 0).

Note 2. This bit should only be re-written while the counter is stopped (CADJUSCEN = 0). Writing to this bit and R128CTRL.CADJUSCEN bit at the same time is prohibited.

10.3 Operation**10.3.1 Correction of the Sub-Clock Signal**

The clock correction circuit (CCC) adjusts timing errors (delay or advance) on output of the 2.048 kHz signal caused by jitter of the 32.768-kHz clock (sub-clock) oscillation. This MCU operates on 32768 clock cycles per second, and the seconds signal can be advanced when the frequency of the 32.768-kHz clock is too high and can be delayed when it is low. The adjustment can be enabled or disabled by using the R128CTRL.ADJUSTEN bit. When adjustment is enabled, set the adjustment value in the ADJUSTR register. Table 10.4 lists the adjustment rates specified by the ADJUSTR register.

Table 10.4 Adjustment rates specified by the ADJUSTR register

b8	b7	b6	b5	b4	b3	b2	b1	b0	Hex.	Correction rate (ppm)
0	1	1	1	1	1	1	1	1	0x0FF	−486.3
0	1	1	1	1	1	1	1	0	0x0FE	−484.4
⋮										
0	0	0	0	0	0	0	1	1	0x003	−5.72
0	0	0	0	0	0	0	1	0	0x002	−3.81
0	0	0	0	0	0	0	0	1	0x001	−1.91
0	0	0	0	0	0	0	0	0	0x000	0
1	1	1	1	1	1	1	1	1	0x1FF	0
1	1	1	1	1	1	1	1	0	0x1FE	+1.91
1	1	1	1	1	1	1	0	1	0x1FD	+3.81
1	1	1	1	1	1	1	0	0	0x1FC	+5.72
⋮										
1	0	0	0	0	0	0	0	1	0x101	+484.4
1	0	0	0	0	0	0	0	0	0x100	+486.3

Note: Errors are not corrected when the setting of the ADJUSTR[8:0] bits is 0x000 or 0x1FF.

When the operating frequency is below 32.768 kHz (e.g., 32.766 kHz), setting the ADJUSTR[8] bit to 0 allows corrections to be made by adding the specified value in the prescaler circuit (division by 16).

When the operating frequency is above 32.768 kHz (e.g., 32.769 kHz), setting the ADJUSTR[8] bit to 1 allows correction to be made by subtracting the specified value in the prescaler circuit (division by 16).

Adjustment for errors are made at the times within 16 seconds listed below.

Examples:

Setting 0x001 allows adjustment for errors at 8 seconds.

Setting 0x002 allows adjustment for errors at 4 and 12 seconds.

Setting 0x003 allows adjustment for errors at 4, 8, and 12 seconds.

Setting 0x004 allows adjustment for errors at 2, 6, 10, and 14 seconds.

10.3.2 Interrupt source

The CCC has three interrupt sources: periodic, overflow, and read error. Table 10.5 lists the interrupt sources.

Table 10.5 Interrupt sources

Name	Interrupt source	CPU interrupt	DMAC activation	DTC activation
CCC_PRD	Periodic interrupt	Possible	Possible	Possible
CCC_CUP	Overflow interrupt	Possible	Possible	Possible
CCC_ERR	Read error interrupt	Possible	Possible	Possible

When you transfer data by using the DMAC or DTC, configure and enable the either applicable and then configure the CCC. For details on setting the DMAC and DTC, see [section 19, DMA Controller \(DMAC\)](#) and [section 20, Data Transfer Controller \(DTC\)](#).

10.3.2.1 Periodic Interrupt (CCC_PRD)

A periodic interrupt occurs by the period of 1, 1/2, 1/4, 1/16, 1/64, or 1/128 seconds. The interruption period is selectable by using the Periodic Interrupt Select bits (R128CTRL.CPES_EX, CPES[1:0]). The periodic interrupt output is enabled or disabled by using the Periodic Interrupt Enable bit (R128CTRL.CPIE).

For example, this interrupt is useful when you wish to wake up the CPU from standby every 1 second.

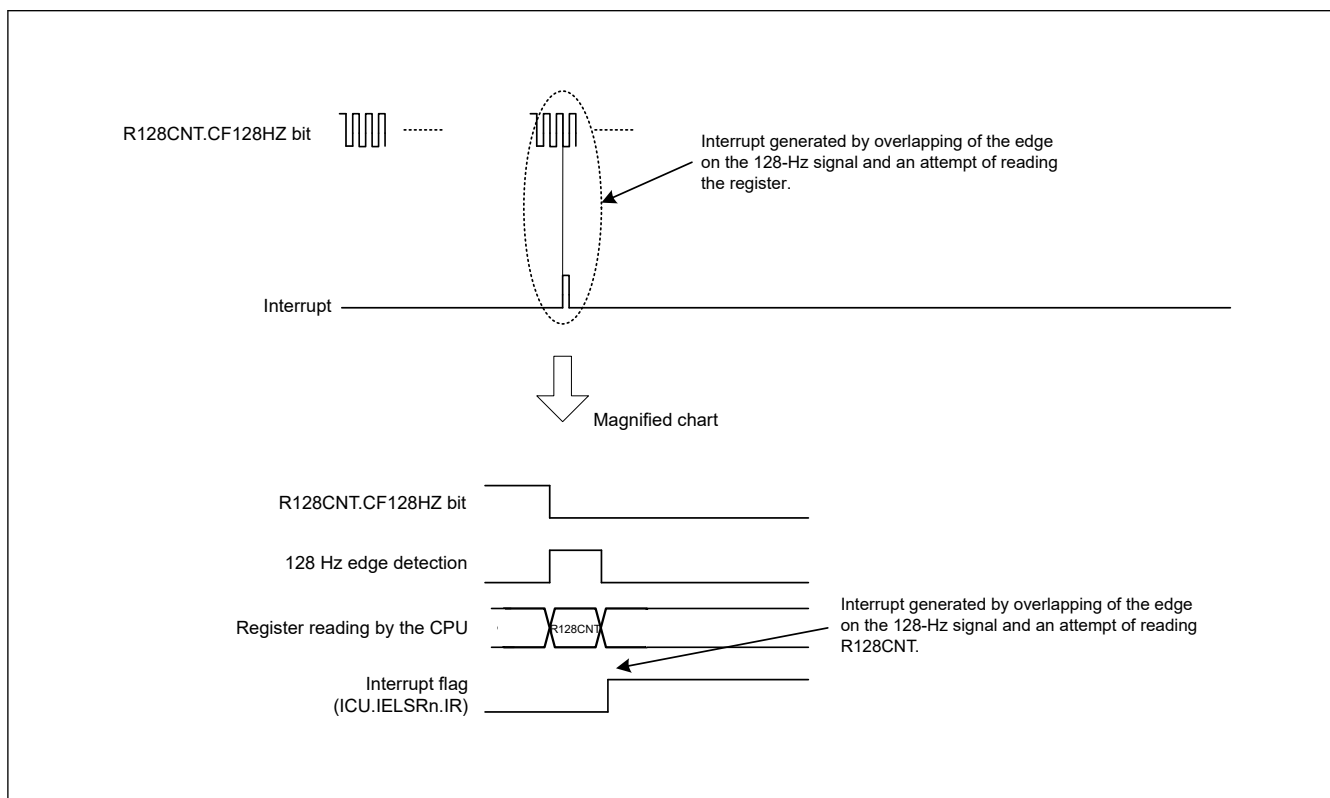
10.3.2.2 Overflow Interrupt (CCC_CUP)

An overflow interrupt occurs when the 128-Hz counter (R128CNT) overflows. The overflow interrupt is enabled or disabled by using the Overflow Interrupt Enable bit (R128CTRL.CCIE).

10.3.2.3 Read Error Interrupt (CCC_ERR)

A read error interrupt occurs when a change on the 128-Hz counter (R128CNT) and an attempt to read from the same happened at the same time. The read error interrupt is enabled or disabled by using the Read Error Interrupt Enable bit (R128CTRL.CEIE).

A timing chart of a read error interrupt is shown in [Figure 10.2](#).

**Figure 10.2** Timing chart of a read error interrupt

10.3.3 Procedure for Reading from 128-Hz Counter

The correct value is not obtained if the 128-Hz counter is incremented while a reading from the counter is attempted. In this case, another read access to the 128-Hz counter is required. Figure 10.3 (a) illustrates the procedure for reading without using interrupts and Figure 10.3 (b) illustrates the procedure for reading using interrupts.

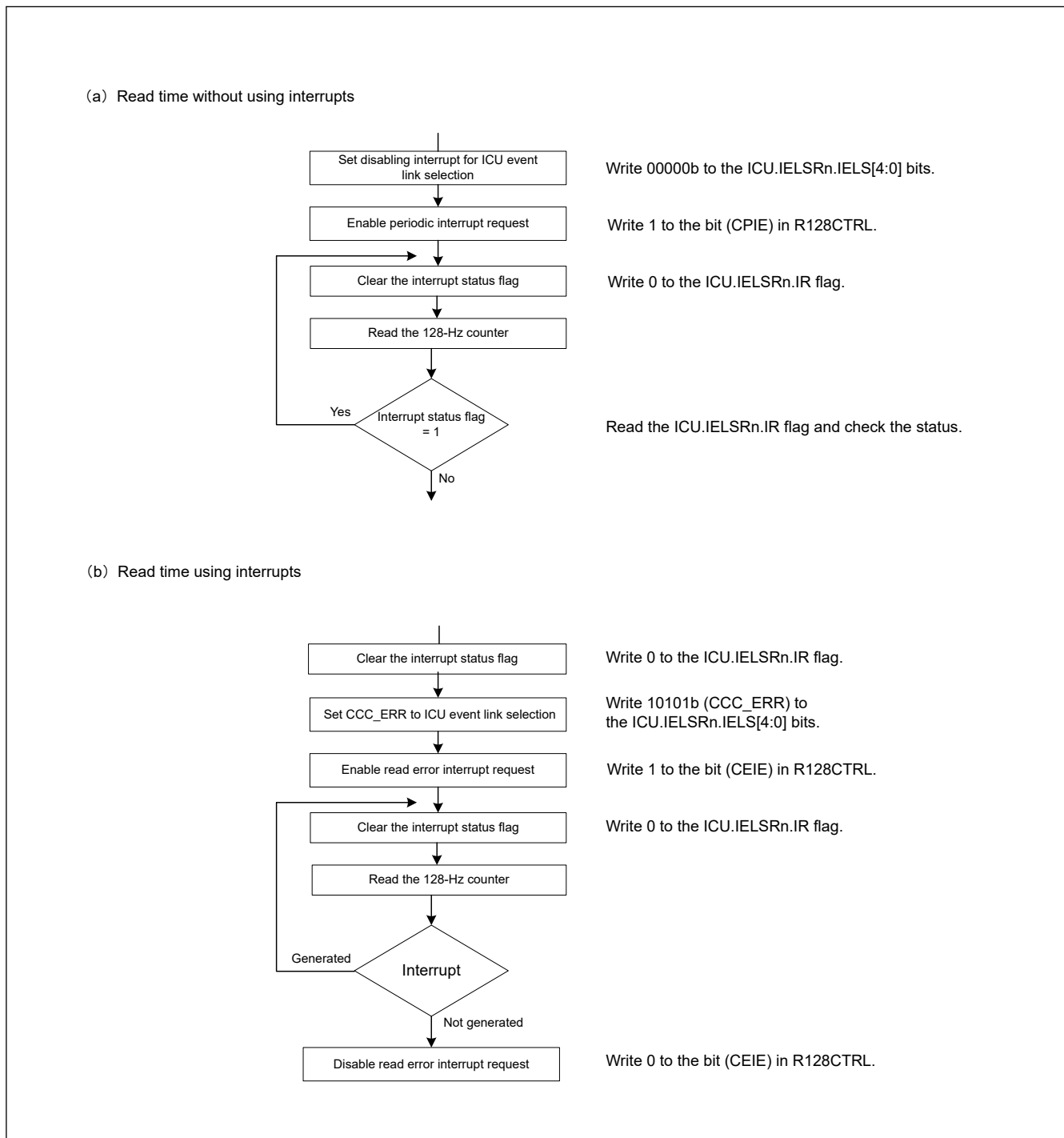


Figure 10.3 Read procedure of 128-Hz counter

10.3.4 Event Link Output

The Clock Correction Circuit (CCC) can operate another module which was preliminary configured by outputting interrupt sources as events to the Event Link Controller (ELC). To prevent unintended outputs of periodic events, configure the CCC registers before setting the ELC.

10.3.4.1 Periodic Event Output

Periodic events are output at the period specified in the CPES_EX and CPES[1:0] bits in R128CTRL, which are 1, 1/2, 1/4, 1/16, 1/64, or 1/128- second interval.

10.3.4.2 Overflow Event Output

An overflow event is output on generation of a 128-Hz counter overflow.

10.3.4.3 Interrupt Request and Event Link

The CCC provides enabling bits that control whether to enable or disable interrupts individually for periodic, overflow, and read error interrupts. When an interrupt source is generated, an interrupt request signal is output if the associated interrupt enable bit is enabled. On the other hand, output of event link signals happen regardless of the settings of the associated interrupt enable bit when an interrupt source is generated. Periodic interrupts can be output while in Software Standby mode but event signals for the ELC are not output.

10.4 Usage Notes

10.4.1 Precautions for Using Periodic Interrupts

Figure 10.4 illustrates notes on periodic interrupts. When the clock correction function is used, the interrupt generation period after correction shifts for the length of correction. This means that the period for generating interrupts immediately after the CPES_EX and CPES[1:0] bits in R128CTRL are set is not guaranteed.

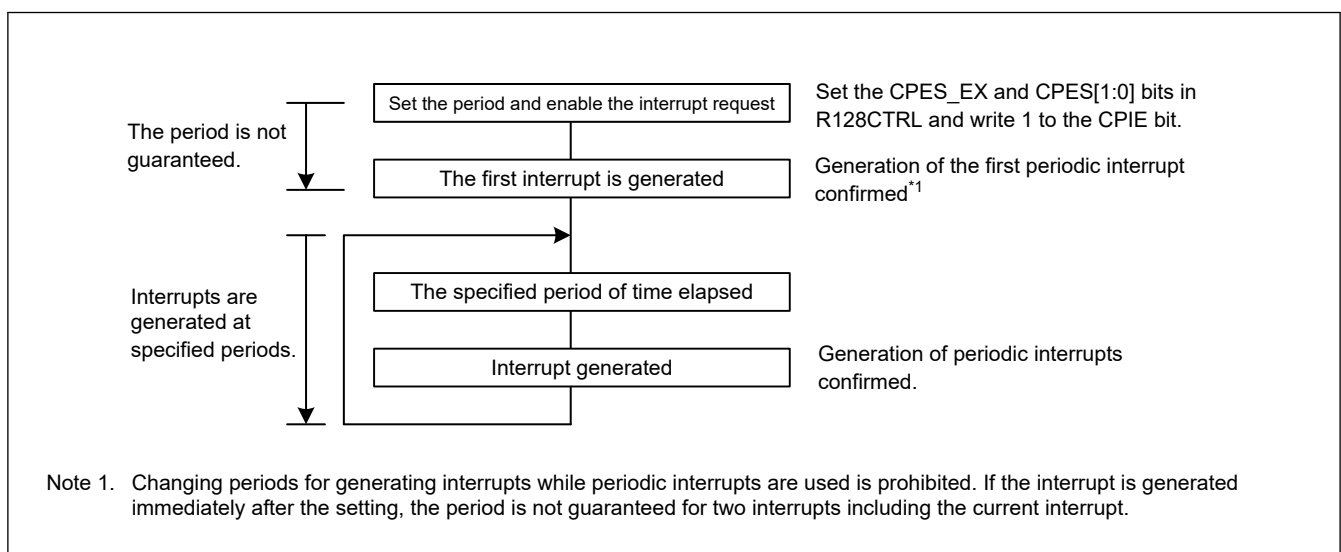


Figure 10.4 Usage notes on the periodic interrupt function

10.4.2 Notes on CCCOUT Output

When the CCCOUT Output Select bit (R128CTRL.CRTCOS) in the 128-Hz Counter Control register (R128CTRL) is changed, the periods for CCCOUT output dependent on the changes made to registers such as for stopping or running the counter or setting of the correction circuit. Additionally, when the clock correction function is in use, the period of CCCOUT output immediately after the correction function is enabled is shifted by the length of correction.

When the output of pulses generated by the CCC from the CCCOUT pin is enabled, output continues even on deep standby. After release from deep software standby, output still continues without requiring the re-setting of the CRTCOS, CRTCOE, and COPSEL bits.

Note that the output of pulses generated by the CCC on deep software standby is regardless of the setting for I/O port retention (IOKEEP bit in section 12.2.14, Deep Software Standby Control Register (DPSBYCR)).

10.4.3 Notes on Transition to Low Power Consumption Modes after Setting Registers

A transition to a low power consumption state (Software Standby or Deep Software Standby mode) while writing or updating a register may lead to a corruption of the value in the register. After setting the register, check that the written values are reflected before entering a low power consumption state.

10.4.4 Notes on Reading and Writing Registers

The ADJUSTR and R128CTRL registers are read in synchronization with the bus clock signal (PCLKB) and written to in synchronization with the input clock before correction (sub-clock). Before proceeding with further processing following writing to these registers, ensure that the values of all bits have actually been updated.

When reading from the 128-Hz counter following return from a reset (excluding initialization by the CRESET bit) or a low power consumption state (Software Standby or Deep Software Standby mode), wait for 1/128 second while the counter is enabled (R128CTRL.CADJUSCEN = 1).

If a system reset is asserted immediately after the register is written, another writing should be done after the time of 6 cycles or more of the input clock before correction (sub-clock) elapsed since the reset is released.

10.4.5 Module-Stop Function Settings

Operation of the CCC can be disabled or enabled using the Module Stop Control Register D (MSTPCRD). For the value after reset, the operation of the CCC is inactive. The registers become accessible when the module-stop state is canceled. The CCC counter (R128CNT in [Figure 10.1](#)) does not depend on the module-stop function. For details, see [section 13, Power-Saving Functions](#).

11. Clock Frequency Accuracy Measurement Circuit (CAC)

11.1 Overview

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

[Table 11.1](#) lists the CAC specifications, [Figure 11.1](#) shows the CAC block diagram, and [Table 11.2](#) lists the CAC I/O pin.

Table 11.1 CAC specifications

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

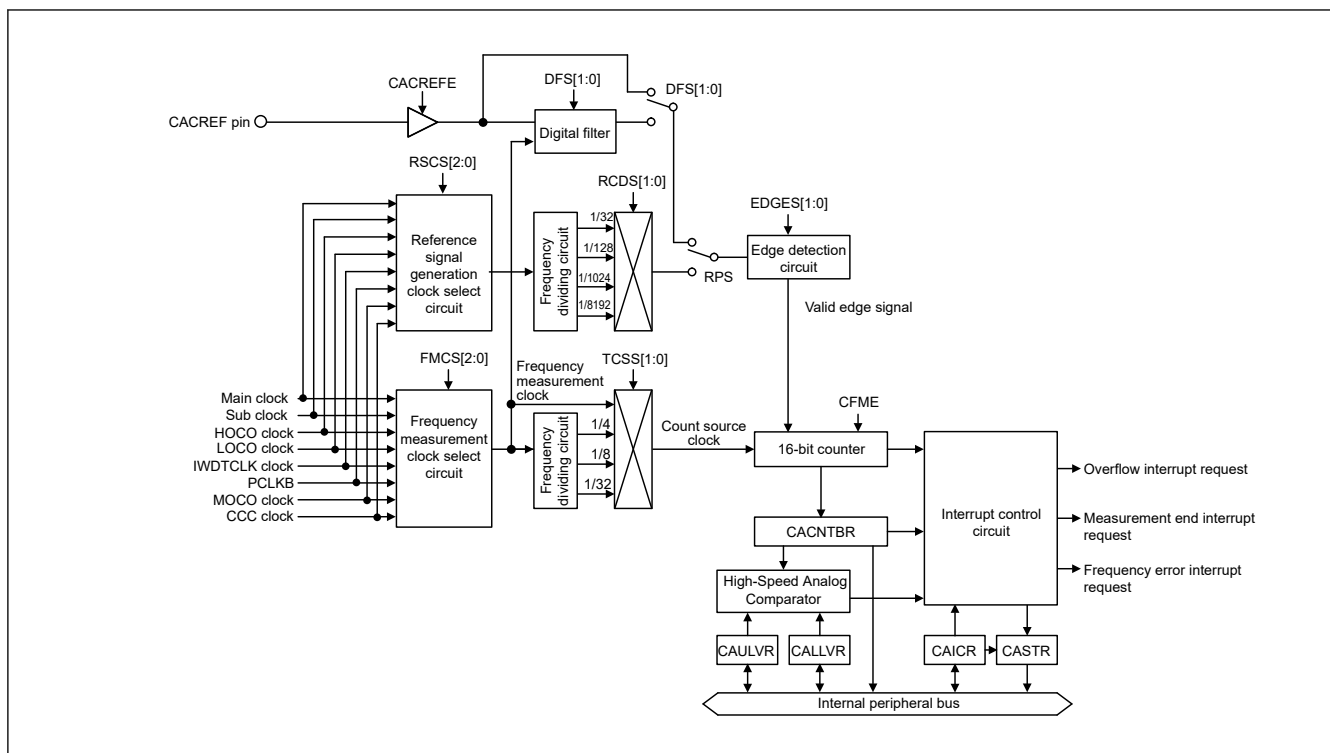


Figure 11.1 CAC block diagram

Table 11.2 CAC I/O pin

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

11.2 Register Descriptions

11.2.1 CACR0 : CAC Control Register 0

Base address: CAC = 0x4004_4600

Offset address: 0x00

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

Value after reset: 0 0 0 0 0 0 0 0 0

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

CFME bit (Clock Frequency Measurement Enable)

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

11.2.2 CACR1 : CAC Control Register 1

Base address: CAC = 0x4004_4600

Offset address: 0x01

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

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

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

CACREFE bit (CACREF Pin Input Enable)

The CACREFE bit enables the CACREF pin input.

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

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

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

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

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

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

11.2.3 CACR2 : CAC Control Register 2

Base address: CAC = 0x4004_4600

Offset address: 0x02

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

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

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

RPS bit (Reference Signal Select)

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

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

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

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

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

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

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

11.2.4 CAICR : CAC Interrupt Control Register

Base address: CAC = 0x4004_4600

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	OVFF CL	MEND FCL	FERR FCL	—	OVFIE	MEND IE	FERR IE
Value after reset:	0	0	0	0	0	0	0	0

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

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

FERRIE bit (Frequency Error Interrupt Request Enable)

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

MENDIE bit (Measurement End Interrupt Request Enable)

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

OVFIE bit (Overflow Interrupt Request Enable)

The OVFIE bit enables or disables the overflow interrupt request.

FERRFCL bit (FERRF Clear)

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

MENDFCL bit (MENDF Clear)

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

OVFFCL bit (OVFF Clear)

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

11.2.5 CASTR : CAC Status Register

Base address: CAC = 0x4004_4600

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	OVFF	MEND F	FERR F

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	FERRF	Frequency Error Flag 0: Clock frequency is within the allowable range 1: Clock frequency has deviated beyond the allowable range (frequency error).	R
1	MENDF	Measurement End Flag 0: Measurement is in progress 1: Measurement ended	R

Bit	Symbol	Function	R/W
2	OVFF	Overflow Flag 0: Counter has not overflowed 1: Counter overflowed	R
7:3	—	These bits are read as 0.	R

FERRF flag (Frequency Error Flag)

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

[Setting condition]

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

[Clearing condition]

- 1 is written to the FERRFCL bit.

MENDF flag (Measurement End Flag)

The MENDF flag indicates the end of measurement.

[Setting condition]

- Measurement ends.

[Clearing condition]

- 1 is written to the MENDFCL bit.

OVFF flag (Overflow Flag)

The OVFF flag indicates that the counter overflowed.

[Setting condition]

- The counter overflows.

[Clearing condition]

- 1 is written to the CAICR.OVFFCL bit.

11.2.6 CAULVR : CAC Upper-Limit Value Setting Register

Base address: CAC = 0x4004_4600

Offset address: 0x06

Bit position: 15

0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

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

11.2.7 CALLVR : CAC Lower-Limit Value Setting Register

Base address: CAC = 0x4004_4600

Offset address: 0x08

Bit position: 15 0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

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

11.2.8 CACNTBR : CAC Counter Buffer Register

Base address: CAC = 0x4004_4600

Offset address: 0x0A

Bit position: 15 0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

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

11.3 Operation

11.3.1 Measuring Clock Frequency

The CAC measures the clock frequency using the CACREF pin input or an internal clock as a reference. [Figure 11.2](#) shows an operating example of the CAC.

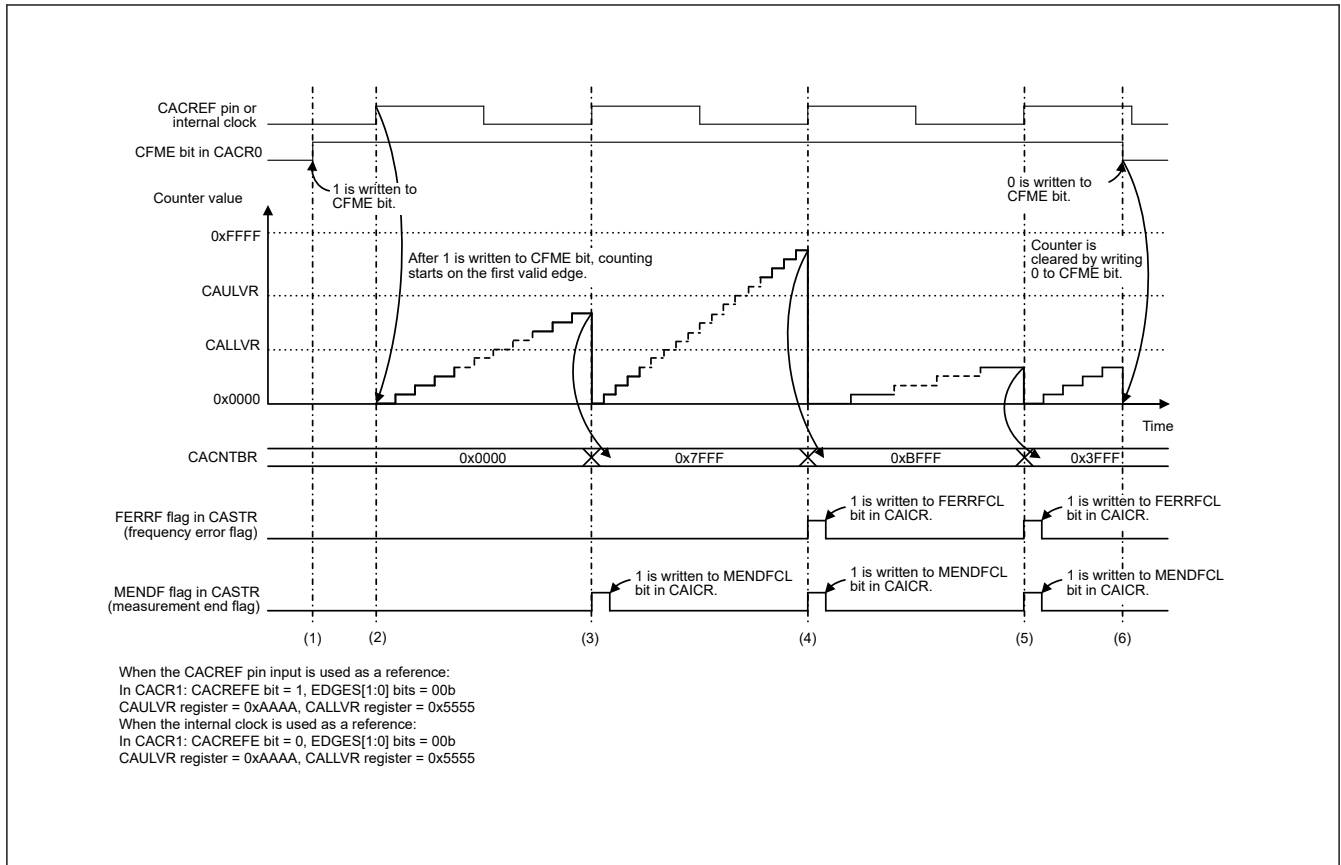


Figure 11.2 CAC operating example

The events in Figure 11.2 are:

1. When the CACREF pin input is used as reference (CACR1.CACREFE = 1), frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is set to 0 and the CACR1.CACREFE bit is set to 1. When the internal clock is used as reference (CACR1.CACREFE = 0), frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is set to 1.
2. When the CACREF pin input is used as reference, after 1 is written to the CFME bit, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits (rising edge (CACR1.EDGES[1:0] = 00b) in Figure 11.2) is input from the CACREF pin. When the internal clock is used as reference, after 1 is written to the CFME bit, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits (rising edge (CACR1.EDGES[1:0] = 00b) in Figure 11.2) is input based on the clock source selected by the CACR2.RSCS[2:0] bits.
3. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If both $CACNTBR \leq CAULVR$ and $CACNTBR \geq CALLVR$ are true, only the MENDF flag in CASTR is set to 1, because the clock frequency is correct. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
4. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If $CACNTBR > CAULVR$, the FERRF flag in CASTR is set to 1, because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. The MENDF flag in CASTR is set to 1 at the end of measurement. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
5. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If $CACNTBR < CALLVR$, the FERRF flag in CASTR is set to 1, because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. The MENDF flag in CASTR is set to 1 at the end of measurement. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
6. When the CFME bit in CACR0 is 1, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR every time a valid edge is input. Writing 0 to the CFME bit in CACR0 clears the counter and stops up-counting.

11.3.2 Digital Filtering of Signals on CACREF Pin

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

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

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

11.4 Interrupt Requests

The CAC generates three types of interrupt requests:

- Frequency error interrupt
- Measurement end interrupt
- Overflow interrupt

When an interrupt source is generated, the associated status flag is set to 1. [Table 11.3](#) provides information on the CAC interrupt requests.

Table 11.3 CAC interrupt requests

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

11.5 Usage Notes

11.5.1 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 13, Power-Saving Functions](#).

12. Key Interrupt Function (KINT)

12.1 Overview

The key interrupt function (KINT) is generated a key interrupt by detecting a valid edge on the key interrupt input pin. Figure 12.1 shows a block diagram and Table 12.1 lists the input pins.

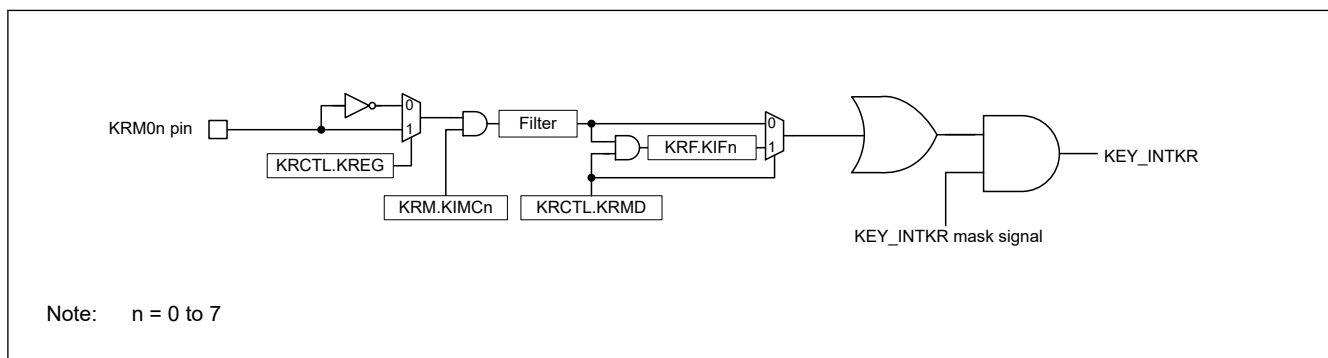


Figure 12.1 KINT block diagram

All key return factors are merged by an OR gate, and the key interrupt signal, KEY_INTKR, is the output of the AND gate to mask the merged key return factor by the KEY_INTKR mask signal. When using KRF.KIFn flag (KRCTL.KRMD = 1), the KEY_INTKR mask signal is used as the output mask that is asserted by clearing KRF.KIFn flag.

Table 12.1 KINT I/O pins

Pin name	I/O	Function
KRM00 to KRM07	Input	Key interrupt input pins

12.2 Register Descriptions

12.2.1 KRCTL : Key Return Control Register

Base address: KINT = 0x4008_0000

Offset address: 0x00

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

Bit	Symbol	Function	R/W
0	KREG	Detection Edge Selection (KRM00 to KRM07 pins) 0: Falling edge 1: Rising edge	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	KRMD	Usage of Key Interrupt Flags (KRF.KIF0 to KRF.KIF7) 0: Do not use key interrupt flags 1: Use key interrupt flags	R/W

The KRCTL register controls the usage of the key interrupt flags, KRF.KIFn (n = 0 to 7), and sets the detection edge.

12.2.2 KRF : Key Return Flag Register

Base address: KINT = 0x4008_0000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	KIF7	KIF6	KIF5	KIF4	KIF3	KIF2	KIF1	KIF0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	KIF0 to KIF7	Key Interrupt Flag n 0: No interrupt detected 1: Interrupt detected	R/W

The KRF register controls the key interrupt flags, KIF0.

When KRCTL.KRMD = 0, setting the KIFn flag to 1 is prohibited. When setting the KIFn flag to 1, the KIFn value does not change.

To clear the KIFn flag, confirm the target flag is 1 before writing 0 to the bit, then write 1 to the other flags.

12.2.3 KRM : Key Return Mode Register

Base address: KINT = 0x4008_0000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	KIMC7	KIMC6	KIMC5	KIMC4	KIMC3	KIMC2	KIMC1	KIMC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	KIMC0 to KIMC7	Key Interrupt Mode Control n 0: Do not detect key interrupt signals 1: Detect key interrupt signals	R/W

The KRM register sets the key interrupt mode.

An interrupt is generated when the target bit in the KRM register is set while a low level (KRCTL.KREG = 0) or a high level (KRCTL.KREG = 1) is being input to the KRM0n pin. To ignore this interrupt, set the KRM register after disabling the interrupt handling.

KINT can be assigned in the PmnPFS.PSEL[4:0] bits. The on-chip pull-up resistors can also be applied by setting the associated key interrupt input pin in the pull-up resistor. For details, see [section 22, I/O Ports](#).

12.3 Operation

12.3.1 Operation When Not Using the Key Interrupt Flags (KRCTL.KRMD = 0)

A key interrupt signal, KEY_INTKR, is generated when the valid edge specified in the KRCTL.KREG bit is input to a KRM0n pin. To identify the channel to which the valid edge is input, read the port register and check the port level after the KEY_INTKR signal is generated.

The KEY_INTKR signal changes based on the input level of the KRM0n pin.

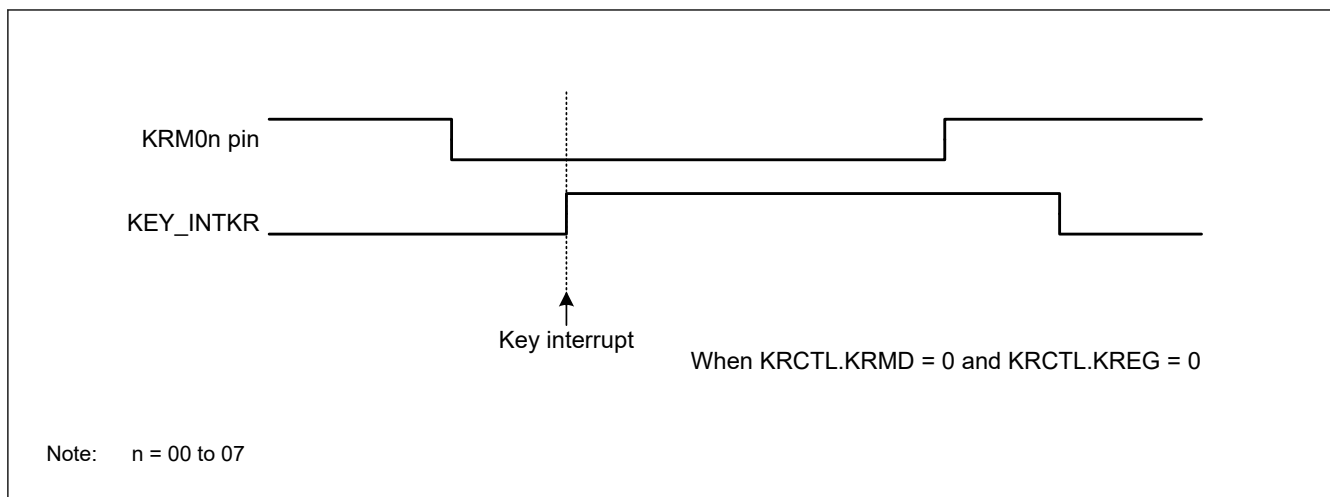


Figure 12.2 Operation of KEY_INTKR signal when a key interrupt is input to a single channel

Figure 12.3 shows the operation when a valid edge is input to multiple KRM0n pins. The KEY_INTKR signal is set while a low level is being input to one pin (when KRCTL.KREG = 0). Therefore, even if a falling edge is input to another pin in this period, the KEY_INTKR signal is not generated again. See [1] in Figure 12.3.

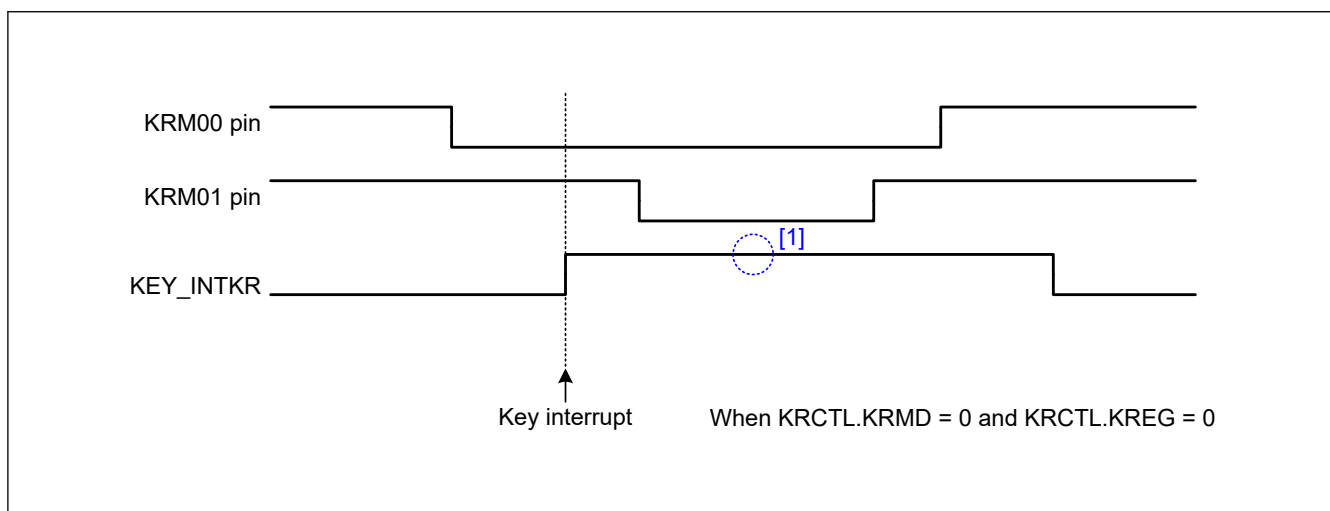


Figure 12.3 Operation of KEY_INTKR signal when key interrupts are input to multiple channels

12.3.2 Operation When Using the Key Interrupt Flags (KRCTL.KRMD = 1)

The KEY_INTKR signal is generated when the valid edge specified in the KRCTL.KREG bit is input to KRM0n pins. To identify the channels to which the valid edge is input, read the KRF register after the KEY_INTKR signal is generated. If the KRCTL.KRMD bit is set to 1, clear the KEY_INTKR signal by clearing the associated bit in the KRF register.

As Figure 12.4 shows, only one interrupt is generated each time a falling edge is input to one channel, (when KRCTL.KREG = 0), regardless of whether the KRF.KIFn flag is cleared before or after a rising edge is input.

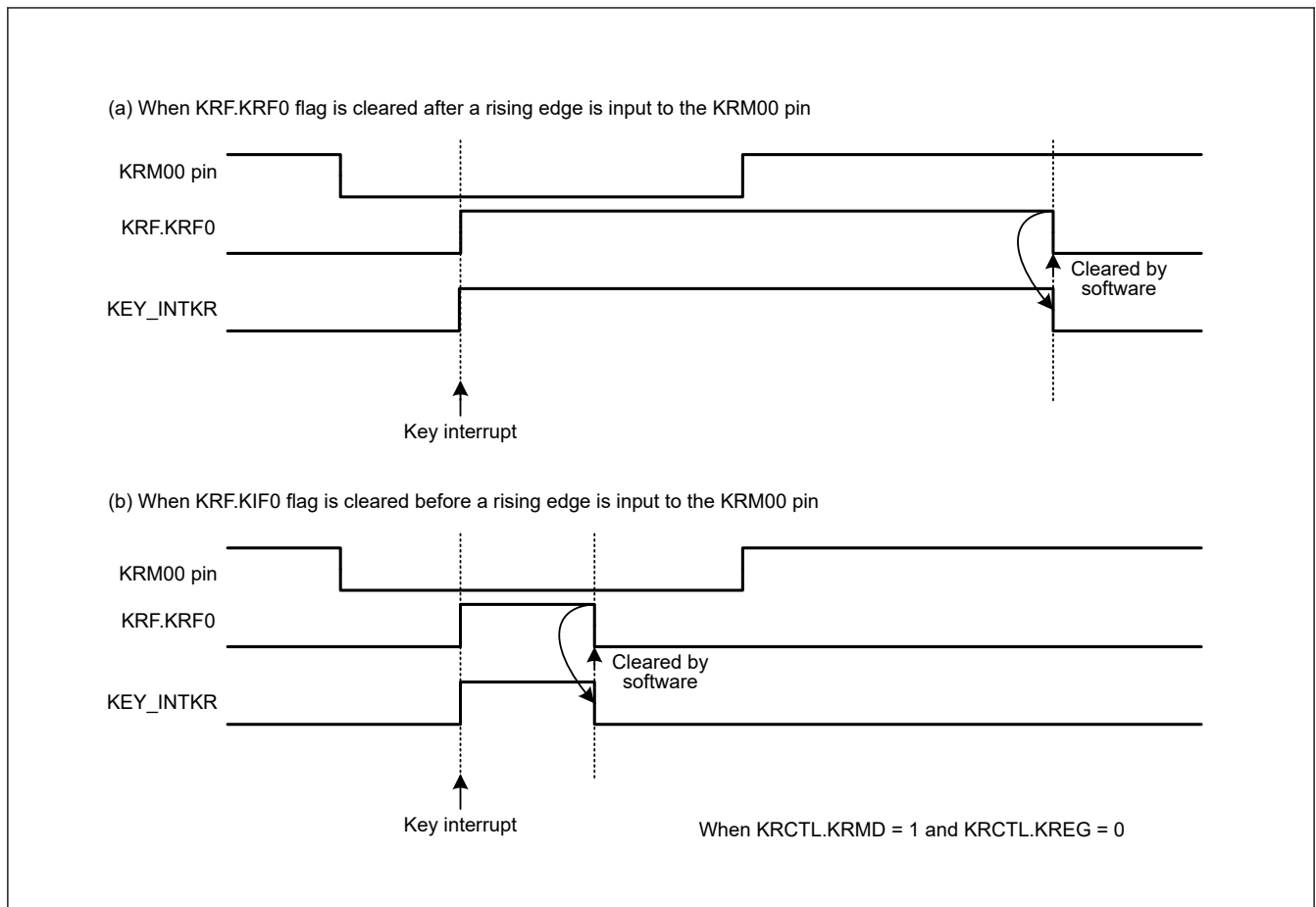


Figure 12.4 Basic operation of KEY_INTKR signal when key interrupt flag is used

Figure 12.5 shows the operation when a valid edge is input to multiple KRM0n pins. A falling edge is also input to the KRM00 and KRM05 pins after a falling edge is input to the KRM00 pin (when KRCTL.KREG = 0). The KRF.KIF1 flag is set when the KRF.KIF0 flag is cleared. The KEY_INTKR signal generates 1 PCLKB clock, after the KRF.KIF0 flag is cleared. See [1] in Figure 12.5.

Also, after a falling edge is input to the KRM05 pin, the KRF.KIF5 flag is set. The KRF.KIF1 flag is cleared at time [2] in the figure. The KEY_INTKR signal generates 1 PCLKB clock, after the KRF.KIF1 flag is cleared. See [3] in the figure. It is therefore possible to generate the KEY_INTKR signal when a valid edge is input to multiple channels.

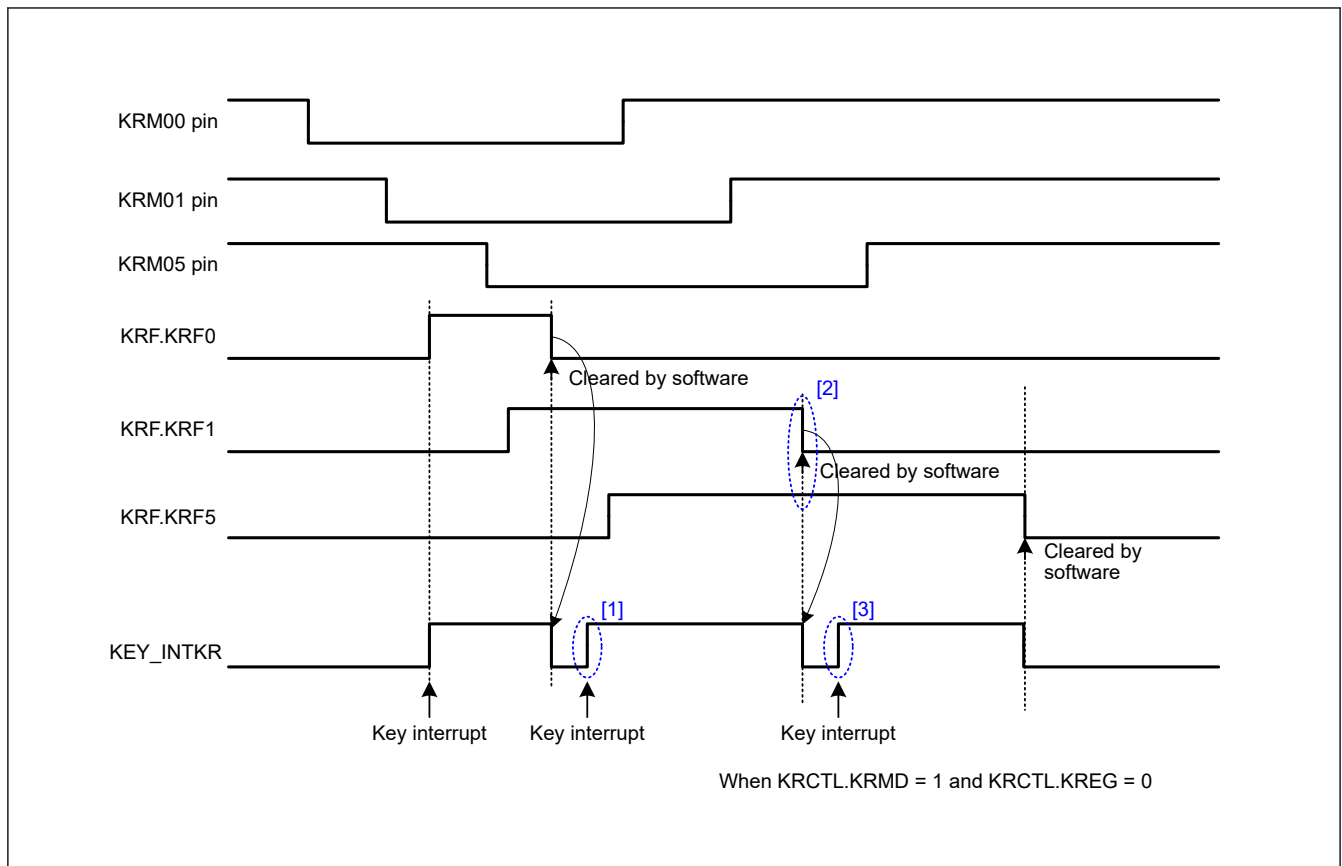


Figure 12.5 Operation of KEY_INTKR signal when key interrupts are input to multiple channels

12.4 Usage Notes

- If the KEY_INTKR signal is used as the snooze request, the KRCTL.KRMD bit should be set to 0.
- If the KEY_INTKR signal is used as the interrupt source for returning to Normal mode from Snooze and Software Standby modes, the KRCTL.KRMD bit should be set to 1.
- When KINT is assigned to a pin, this pin input is always enabled in the Software Standby mode, and if the pin level changes, the associated KRF.KIFn flag can be set. Therefore, a KEY_INTKR signal might be generated on canceling Software Standby mode. To ignore changes to the KRM0n pin during a Software Standby, clear the associated KRM.KIMCn bit before entering Software Standby. After canceling Software Standby mode, the KRF.KIFn flag should be cleared before the associated KRM.KIMCn bit can be set.

13. Power-Saving Functions

13.1 Overview

This MCU has the following power-saving functions:

1. Clock division function
2. Module stop function
3. Power control modes
4. Low power consumption modes
5. Power supply modes

Table 13.1 shows an overview of the Power-Saving Functions.

Table 13.1 Specifications of the Power-Saving Functions

Item	Specification
Clock division function	The frequency division ratio can be selected independently for the system clock/peripheral module clock A (ICLK/PCLKA), and peripheral module clock B (PCLKB)* ¹
Module stop function	Clock supply can be stopped independently for each peripheral module.
Power control modes	Power consumption can be reduced in normal operating (OPE), sleep (SLEEP), software standby (SSTBY) and snooze (SNOOZE) modes by selecting an appropriate power control mode according to the operating frequency. • Boost mode (BOOST): Max 64 MHz • Normal mode (NORMAL): Max 32 MHz – High-speed mode: Max 32 MHz – Low-speed mode: Max 2 MHz • Low leakage current mode (VBB): Max 32.768 kHz
Low power consumption modes	• Operating mode (OPE) • Sleep mode (SLEEP) • Snooze mode (SNOOZE) • Software standby mode (SSTBY) • Deep software standby mode (DSTBY)
Power supply modes	Power supply mode is selectable. Power consumption can be reduced by cutting off power domains that are not to be used. • All power supply mode (ALLPWON) • Exclude-flash power supply mode (EXFPWON) • Minimum power supply mode (MINPWON)

Note 1. ICLK and PCLKA have the same frequency division ratio. For details, see [section 9, Clock Generation Circuit](#)

Figure 13.1 shows an overview of the state transitions of this MCU.

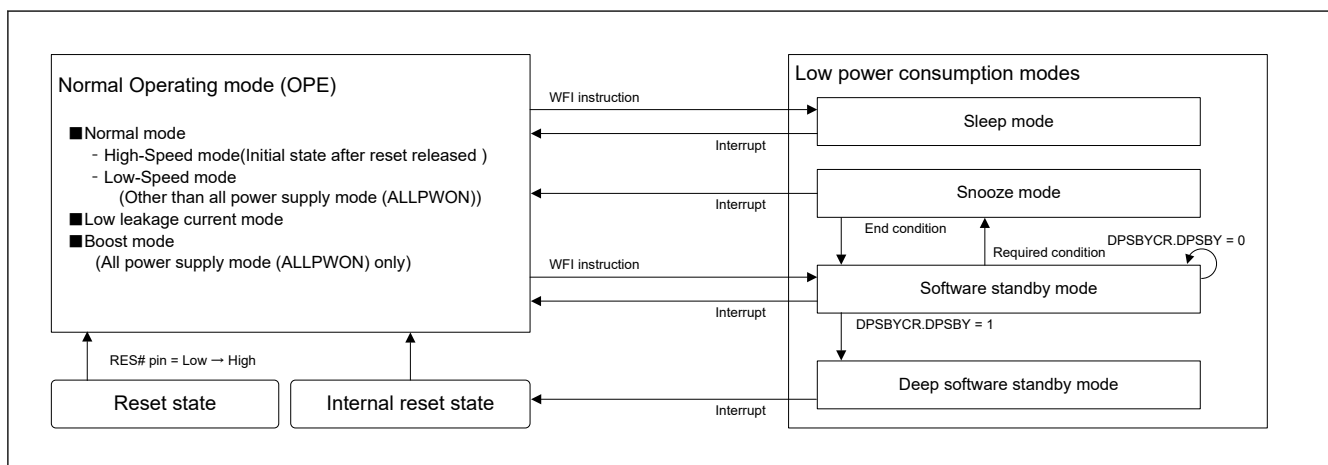


Figure 13.1 Overview of the State Transitions

Normal operating mode (OPE) can be selected from the following three modes according to the operating frequency.

- Normal operating mode (NORMAL OPE)
Operation at up to 32 MHz is allowed. Furthermore, low power consumption is achieved by selection from the following two modes according to the operating frequency.
 - High-speed mode: Maximum operating frequency of 32 MHz. This mode is enabled after a reset release.
 - Low-speed mode: Maximum operating frequency of 2 MHz. This mode is not available in all power supply mode (ALLPWON).
- Boost operating mode (BOOST OPE)
Operation at up to 64 MHz is allowed. This mode is the most suitable for high-speed processing. This mode can only be entered in all power supply mode (ALLPWON).
- Low leakage current operating mode (VBB OPE)
When the MCU is operating at a frequency no higher than 32.768 kHz, the leakage current can be reduced by back bias voltage control (VBBC), and the MCU consumes less power in operation.

Additionally, transition from each operating mode to the following low power consumption modes is possible:

- Sleep mode (SLEEP)
Low power consumption is achieved by stopping the CPU operation.
- Software standby mode (SSTBY)
Lower power consumption is achieved compared to sleep mode by stopping the CPU, flash memory, most peripheral functions, and oscillator. The state of functions whose operation is stopped is retained. Transition from boost operating mode to this mode is not possible.
- Snooze mode (SNOOZE)
In this mode transitioned from software standby mode, some peripheral functions are allowed to operate while the CPU is stopped as in software standby mode. Since the CPU does not operate, the average current of the system can be reduced. This mode can only be entered from software standby mode.
- Deep software standby mode (DSTBY)
Lower power consumption is achieved compared to software standby mode by cutting off the power supply to functions other than the specific functions such as the SOSC and CCC. The state of the CPU, peripheral modules, and SRAM that are to be powered off are undefined.

Furthermore, in this MCU, the power domain is divided into four to achieve low power consumption, and it can control power supply/shutoff of each power domain.

Table 13.2 lists the association between the four power domains and each function.

Table 13.2 Power Domain for each Function (1 of 2)

Function Category	Power Domain Function Category			
	AWO Domain	ISO1 Domain	ISO2 Domain	ISO3 Domain
CPU	—	CPU	—	—
Power supply system	Power-on reset circuit (POR) Low voltage detection circuit 0 (LVD0) Low voltage detection circuit 1 (LVD1) Low voltage detection circuit BAT (LVDBAT) Back bias voltage control (VBBC) Energy harvesting control (EHC)	—	—	—
Memory	—	RAM	—	Flash memory
Clock	Sub-clock oscillator (SOSC) Clock correction circuit (CCC)	Main clock oscillator (MOSC) Middle-speed on-chip oscillator (MOCO) Low-speed on-chip oscillator (LOCO) IWDt-dedicated on-chip oscillator (IWDTLCO) Main clock oscillator stop detection function Clock output function	High-speed on-chip oscillator (HOCO)	—

Table 13.2 Power Domain for each Function (2 of 2)

Function Category	Power Domain Function Category			
	AWO Domain	ISO1 Domain	ISO2 Domain	ISO3 Domain
Peripheral function	Realtime Clock (RTC) Wakeup timer (WUPT)	Data Transfer Controller (DTC) DMA Controller (DMAC) Memory Protection Unit (MPU) Interrupt Controller (ICU) Event Link Controller (ELC) Asynchronous General Purpose Timers n : 16 bit (AGTn (n = 0, 1)) Asynchronous General Purpose Timers n : 32 bit (AGTWn (n = 0, 1)) Independent Watchdog Timer (IWDT) SysTick timer Serial Communications Interface 0 (SCI0) I ² C Bus Interface 0 (IIC0) MIP Liquid Crystal Controller (MLCD) Key Interrupt Function (KINT) Low-Speed Clock Timer (LST)	Other than the left	—
Analog function	—	—	14-Bit A/D Converter (ADC14) Temperature Sensor (TSN) Reference Voltage Generation Circuit (VREF)	—
Pin interrupt	NMI pin IRQ0_DS to IRQ3_DS pins	IRQ0 to IRQ9 pins	—	—

A combination of the power supply and power shutoff of each power domain can be selected from the following three power supply modes. Power consumption can be reduced by cutting off the supply to domains for which power is not currently necessary

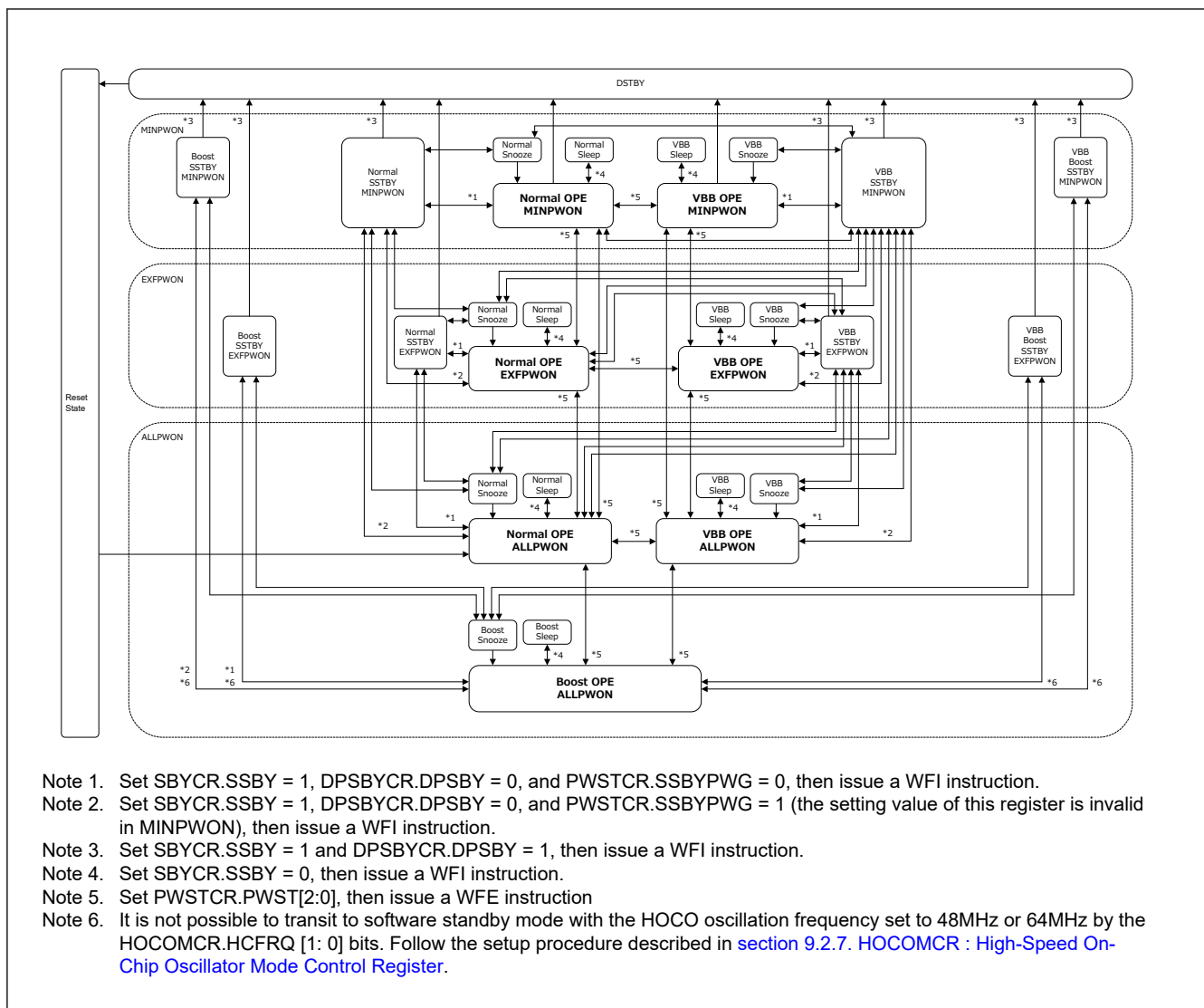
- All power supply mode (ALLPWON)
In this mode, all power domains are supplied with power.
- Exclude-flash power supply mode (EXFPWON)
In this mode, the flash power domain (ISO3) is powered off. The flash memory is disabled due to power shutoff, so instruction execution from the SRAM is necessary.
- Minimum power supply mode (MINPWON)
In this mode, the flash power domain (ISO3) and the power domain (ISO2) of most peripheral functions are powered off, and power is only supplied to minimal functions. The flash memory is disabled due to power shutoff, so instruction execution from the SRAM is necessary. Additionally, the function of the ISO2 domain is in a stop (undefined) state.

Table 13.3 shows the state of each power domain in each power supply mode when deep software standby is enabled.

Table 13.3 State of Power Domains in each Power Supply Mode when Deep Software Standby is Enabled

Power Supply Mode	AWO Domain	ISO1 Domain	ISO2 Domain	ISO3 Domain
ALLPWON	Power supply			
EXFPWON	Power supply			Shutoff
MINPWON	Power supply		Shutoff	
DSTBY	Power supply	Shutoff		

Figure 13.2 shows the product state transitions with power supply modes added to Figure 13.1.

**Figure 13.2 Detailed State Transitions**

[Table 13.4](#) shows the state of this MCU in each operating mode (OPE).

Table 13.4 State of MCU in each Operating Mode (OPE) (1 of 2)

		ALLPWON			EXFPWON			MINPWON		
		BOOST OPE	NORMAL OPE	VBB OPE	NORMAL OPE		VBB OPE	NORMAL OPE		VBB OPE
					High-speed	Low-speed		High-speed	Low-speed	
AWO domain	Power-on reset circuit (POR)	Operating	Operating	Operating	Operating	Operating	Operating	Operating	Operating	Operating
	Back bias voltage control (VBBC)	Selectable	Selectable	Operating	Selectable	Selectable	Operating	Selectable	Selectable	Operating
	Other functions	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable

Table 13.4 State of MCU in each Operating Mode (OPE) (2 of 2)

		ALLPWON			EXFPWON			MINPWON		
		BOOST OPE	NORMAL OPE	VBB OPE	NORMAL OPE		VBB OPE	NORMAL OPE		VBB OPE
					High-speed	Low-speed		High-speed	Low-speed	
ISO1 domain	CPU	Operating	Operating	Operating	Operating	Operating	Operating	Operating	Operating	Operating
	Main clock oscillator (MOSC)	Selectable	Selectable	Operation prohibited	Selectable	Selectable	Operation prohibited	Selectable	Selectable	Operation prohibited
	Main clock oscillator stop detection function	Selectable	Selectable	Operation prohibited	Selectable	Selectable	Operation prohibited	Selectable	Selectable	Operation prohibited
	Middle-speed on-chip oscillator (MOCO)	Selectable	Selectable	Operation prohibited	Selectable	Selectable	Operation prohibited	Selectable	Selectable	Operation prohibited
	Clock output function	Selectable	Selectable	Selectable *2	Selectable	Selectable	Selectable *2	Selectable	Selectable	Selectable *2
	Other functions	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
ISO2 domain	High-speed on-chip oscillator (HOCO)	Selectable	Selectable *1	Operation prohibited	Selectable *1	Selectable *1	Operation prohibited	Operation prohibited	Operation prohibited	Operation prohibited
	Other functions	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)
ISO3 domain	Flash memory	Operating	Operating	Operating	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)

Note: "Selectable" means that operating or not operating can be selected by control registers.

"Operation prohibited" means that the function must be stopped before entering the target mode.

"Stopped (retained)" means that the values of the internal registers are retained, and the operations are suspended.

"Stop (undefined)" means that the values of the internal registers are undefined, and the circuit is powered off.

Note 1. Selecting HOCO 48-MHz/64-MHz oscillation is prohibited in modes other than BOOST OPE.

Note 2. Clock is output only when any of LOCO/SOSC/CCC_2K clocks is selected by the CKOCR.CKOSEL[2:0] bits.

Table 13.5 shows the way to enter and cancel low power consumption modes and operating states in each modes (SLEEP/SNOOZE/SSTBY/DSTBY)

Table 13.5 Entering and Canceling Low Power Consumption Modes and Operating States in each Mode (1 of 12)

Way of Entry/Cancellation and Operating State	Power Supply Mode: ALLPWON						DSTBY
	BOOST SLEEP	NORMAL SLEEP	BOOST SNOOZE	NORMAL SNOOZE	VBB SLEEP	VBB SNOOZE	
Condition for transition	Register setting + WFI instruction	Register setting + WFI instruction	Register setting + SNOOZE request occurrence in SSTBY mode	Register setting + SNOOZE request occurrence in SSTBY mode	Register setting + WFI instruction	Register setting + SNOOZE request occurrence in SSTBY mode	Register setting + WFI instruction
Canceling method	All interrupts. Any reset available in the mode.	All interrupts. Any reset available in the mode.	Interrupts shown in Table 13.6. Any reset available in the mode.	Interrupts shown in Table 13.6. Any reset available in the mode.	All interrupts. Any reset available in the mode.	Interrupts shown in Table 13.6. Any reset available in the mode.	Interrupts shown in Table 13.6. Any reset available in the mode.

Table 13.5 Entering and Canceling Low Power Consumption Modes and Operating States in each Mode (2 of 12)

Way of Entry/Cancellation and Operating State		Power Supply Mode: ALLPWON						DSTBY
		BOOST SLEEP	NORMAL SLEEP	BOOST SNOOZE	NORMAL SNOOZE	VBB SLEEP	VBB SNOOZE	
State after cancellation by an interrupt		Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Reset state
State after cancellation by a reset		Program execution state (reset processing)	Program execution state (reset processing)	Program execution state (reset processing)	Program execution state (reset processing)	Program execution state (reset processing)	Program execution state (reset processing)	Program execution state (reset processing)
AWO domain	Sub-clock oscillator (SOSC)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Clock correction circuit (CCC)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Power-on reset circuit (POR)	Operating	Operating	Operating	Operating	Operating	Operating	Operating
	Low voltage detection circuit 0 (LVD0)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Low voltage detection circuit 1 (LVD1)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Low voltage detection circuit BAT (LVDBAT)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Back bias voltage control (VBBC)	Selectable	Selectable	Selectable	Selectable	Operating	Operating	Stopped
	Energy harvesting control (EHC)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	NMI pin, IRQ0_DS to IRQ3_DS pin interrupt	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Wakeup timer (WUPT)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Realtime Clock (RTC)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable

Table 13.5 Entering and Canceling Low Power Consumption Modes and Operating States in each Mode (3 of 12)

Way of Entry/Cancellation and Operating State		Power Supply Mode: ALLPWON						DSTBY
		BOOST SLEEP	NORMAL SLEEP	BOOST SNOOZE	NORMAL SNOOZE	VBB SLEEP	VBB SNOOZE	
ISO1 domain	CPU	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (undefined)
	SRAM	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Stopped (undefined)
	Main clock oscillator (MOSC)	Selectable	Selectable	Selectable * ¹	Selectable * ¹	Operation prohibited	Operation prohibited	Stopped
	Main clock oscillator stop detection function	Selectable	Selectable	Operation prohibited	Operation prohibited	Operation prohibited	Operation prohibited	Operation prohibited
	Middle-speed on-chip oscillator (MOCO)	Selectable	Selectable	Selectable	Selectable	Operation prohibited	Operation prohibited	Stopped
	Low-speed on-chip oscillator (LOCO)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Stopped
	IWDT-dedicated on-chip oscillator (IWDTLOCO)	Selectable * ²	Selectable * ²	Selectable * ²	Selectable * ²	Selectable * ²	Selectable * ²	Stopped
	Independent Watchdog Timer (IWDT)	Selectable * ²	Selectable * ²	Selectable * ²	Selectable * ²	Selectable * ²	Selectable * ²	Stopped (undefined)
	Clock output function	Selectable	Selectable	Selectable	Selectable	Selectable * ³	Selectable * ³	Stopped (undefined)
	Data Transfer Controller (DTC)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Stopped (undefined)
	DMA Controller (DMAC)	Selectable	Selectable	Operation prohibited	Operation prohibited	Selectable	Operation prohibited	Stopped (undefined)
	Event Link Controller (ELC)	Selectable	Selectable	Selectable * ⁴	Selectable * ⁴	Selectable	Selectable * ⁴	Stopped (undefined)
	Asynchronous General Purpose Timers n : 16 bit (AGTn (n = 0, 1))	Selectable	Selectable	Selectable * ⁵ * ⁷	Selectable * ⁵ * ⁷	Selectable	Selectable * ⁵ * ⁷	Stopped (undefined)
	Asynchronous General Purpose Timers n : 32 bit (AGTWn (n = 0, 1))	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Stopped (undefined)
	SysTick timer	Selectable	Selectable	Stopped (retained)	Stopped (retained)	Selectable	Stopped (retained)	Stopped (undefined)
	Serial Communications Interface 0 (SCI0)	Selectable	Selectable	Selectable * ¹ * ⁶	Selectable * ¹ * ⁶	Selectable	Selectable * ⁶	Stopped (undefined)
	I ² C Bus Interface 0 (IIC0)	Selectable	Selectable	Operation prohibited	Operation prohibited	Selectable	Operation prohibited	Stopped (undefined)
	MIP Liquid Crystal Controller (MLCD)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Stopped (undefined)
	Key interrupt function (KINT)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Stopped (undefined)
	LST	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Stopped (undefined)
	IRQ0 to IRQ9 pin interrupt	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable	Stopped (undefined)

Table 13.5 Entering and Canceling Low Power Consumption Modes and Operating States in each Mode (4 of 12)

Way of Entry/Cancellation and Operating State		Power Supply Mode: ALLPWON						DSTBY
		BOOST SLEEP	NORMAL SLEEP	BOOST SNOOZE	NORMAL SNOOZE	VBB SLEEP	VBB SNOOZE	
ISO2 domain	High-speed on-chip oscillator (HOCO)	Selectable	Selectable	Selectable	Selectable	Operation prohibited	Operation prohibited	Stopped
	WDT	Selectable *8	Selectable *8	Selectable *8 or stop (initial state)*11	Selectable *8 or stop (initial state)*11	Selectable *8	Selectable *8 or stop (initial state)*11	Stopped (undefined)
	14-Bit A/D Converter (ADC14)	Selectable	Selectable	Selectable *8 or stop (initial state)*11	Selectable *8 or stop (initial state)*11	Selectable	Selectable *8 or stop (initial state)*11	Stopped (undefined)
	DOC	Selectable	Selectable	Selectable *8 or stop (initial state)*11	Selectable *8 or stop (initial state)*11	Selectable	Selectable *8 or stop (initial state)*11	Stopped (undefined)
	Other functions	Selectable	Selectable	Operation prohibited	Operation prohibited	Selectable	Operation prohibited	Stopped (undefined)
ISO3 domain	Flash memory	Operating	Operating	Stopped (retained)	Stopped (retained)	Operating	Stopped (retained)	Stopped (retained)

Table 13.5 Entering and Canceling Low Power Consumption Modes and Operating States in each Mode (5 of 12)

Way of Entry/Cancellation and Operating State		Power Supply Mode: EXFPWON				
		NORMAL SLEEP	NORMAL SNOOZE	NORMAL SSTBY	VBB SLEEP	VBB SNOOZE
Condition for transition		Register setting + WFI instruction	Register setting + SNOOZE request occurrence in SSTBY mode	Register setting + WFI instruction	Register setting + WFI instruction	Register setting + SNOOZE request occurrence in SSTBY mode
Canceling method		All interrupts. Any reset available in the mode.	Interrupts shown in Table 12.6. Any reset available in the mode.	Interrupts shown in Table 12.6. Any reset available in the mode.	All interrupts. Any reset available in the mode.	Interrupts shown in Table 12.6. 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)	Program execution state (interrupt processing)	Program execution state (interrupt processing)
State after cancellation by a reset		Program execution state (reset processing)	Program execution state (reset processing)	Program execution state (reset processing)	Program execution state (reset processing)	Program execution state (reset processing)

Table 13.5 Entering and Canceling Low Power Consumption Modes and Operating States in each Mode (6 of 12)

Way of Entry/Cancellation and Operating State		Power Supply Mode: EXFPWON					
		NORMAL SLEEP	NORMAL SNOOZE	NORMAL SSTBY	VBB SLEEP	VBB SNOOZE	VBB SSTBY
AWO domain	Sub-clock oscillator (SOSC)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Clock correction circuit (CCC)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Power-on reset circuit (POR)	Operating	Operating	Operating	Operating	Operating	Operating
	Low voltage detection circuit 0 (LVD0)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Low voltage detection circuit 1 (LVD1)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Low voltage detection circuit BAT (LVDBAT)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Back bias voltage control (VBBC)	Selectable	Selectable	Selectable	Operating	Operating	Operating
	Energy harvesting control (EHC)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	NMI pin, IRQ0_DS to IRQ3_DS pin interrupt	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Wakeup timer (WUPT)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Realtime Clock (RTC)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable

Table 13.5 Entering and Canceling Low Power Consumption Modes and Operating States in each Mode (7 of 12)

Way of Entry/Cancellation and Operating State		Power Supply Mode: EXFPWON					
		NORMAL SLEEP	NORMAL SNOOZE	NORMAL SSTBY	VBB SLEEP	VBB SNOOZE	VBB SSTBY
ISO1 domain	CPU	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)
	SRAM	Selectable	Selectable	Stopped (retained/undefined)*10	Selectable	Selectable	Stopped (retained/undefined)*10
	Main clock oscillator (MOSC)	Selectable	Selectable *1	Stopped	Operation prohibited	Operation prohibited	Operation prohibited
	Main clock oscillator stop detection function	Selectable	Operation prohibited	Operation prohibited	Operation prohibited	Operation prohibited	Operation prohibited
	Middle-speed on-chip oscillator (MOCO)	Selectable	Selectable	Stopped	Operation prohibited	Operation prohibited	Operation prohibited
	Low-speed on-chip oscillator (LOCO)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	IWDT-dedicated on-chip oscillator (IWDTLOCO)	Selectable *2	Selectable *2	Selectable *2	Selectable *2	Selectable *2	Selectable *2
	Independent Watchdog Timer (IWDT)	Selectable *2	Selectable *2	Selectable *2	Selectable *2	Selectable *2	Selectable *2
	Clock output function	Selectable	Selectable	Selectable *3	Selectable *3	Selectable *3	Selectable *3
	Data Transfer Controller (DTC)	Selectable	Selectable	Stopped (retained)	Selectable	Selectable	Stopped (retained)
	DMA Controller (DMAC)	Selectable	Operation prohibited	Stopped (retained)	Selectable	Operation prohibited	Stopped (retained)
	Event Link Controller (ELC)	Selectable	Selectable *4	Stopped (retained)	Selectable	Selectable *4	Stopped (retained)
	Asynchronous General Purpose Timers n : 16 bit (AGTn (n = 0, 1))	Selectable	Selectable *5 *7	Selectable *5 *7	Selectable	Selectable *5 *7	Selectable *5 *7
	Asynchronous General Purpose Timers n : 32 bit (AGTWn (n = 0, 1))	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	SysTick timer	Selectable	Stopped (retained)	Stopped (retained)	Selectable	Stopped (retained)	Stopped (retained)
	Serial Communications Interface 0 (SCI0)	Selectable	Selectable *1 *6	Stopped (retained)	Selectable	Selectable *6	Stopped (retained)
	I ² C Bus Interface 0 (IIC0)	Selectable	Operation prohibited	Stopped (retained)	Selectable	Operation prohibited	Stopped (retained)
	MIP Liquid Crystal Controller (MLCD)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Key interrupt function (KINT)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	LST	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	IRQ0 to IRQ9 pin interrupt	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable

Table 13.5 Entering and Canceling Low Power Consumption Modes and Operating States in each Mode (8 of 12)

Way of Entry/Cancellation and Operating State		Power Supply Mode: EXFPWON					
		NORMAL SLEEP	NORMAL SNOOZE	NORMAL SSTBY	VBB SLEEP	VBB SNOOZE	VBB SSTBY
ISO2 domain	High-speed on-chip oscillator (HOCO)	Selectable	Selectable	Stopped	Operation prohibited	Operation prohibited	Operation prohibited
	WDT	Selectable ^{*8}	Selectable ^{*8} or stop (initial state) ^{*11}	Selectable ^{*8} or stop (undefined) ^{*11}	Selectable ^{*6}	Selectable ^{*8} or stop (initial state) ^{*11}	Selectable ^{*8} or stop (undefined) ^{*11}
	14-Bit A/D Converter (ADC14)	Selectable	Selectable ^{*8} or stop (initial state) ^{*11}	Selectable ^{*8} or stop (undefined) ^{*11}	Selectable	Selectable ^{*8} or stop (initial state) ^{*11}	Selectable ^{*8} or stop (undefined) ^{*11}
	DOC	Selectable	Selectable ^{*8} or stop (initial state) ^{*11}	Selectable ^{*8} or stop (undefined) ^{*11}	Selectable	Selectable ^{*8} or stop (initial state) ^{*11}	Selectable ^{*8} or stop (undefined) ^{*11}
	Other functions	Selectable	Operation prohibited	Selectable ^{*8} or stop (undefined) ^{*11}	Selectable	Operation prohibited	Selectable ^{*8} or stop (undefined) ^{*11}
ISO3 domain	Flash memory	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)

Table 13.5 Entering and Canceling Low Power Consumption Modes and Operating States in each Mode (9 of 12)

Way of Entry/Cancellation and Operating State		Power Supply Mode: MINPWON					
		NORMAL SLEEP	NORMAL SNOOZE	NORMAL SSTBY	VBB SLEEP	VBB SNOOZE	VBB SSTBY
Condition for transition		Register setting + WFI instruction	Register setting + SNOOZE request occurrence in SSTBY mode	Register setting + WFI instruction	Register setting + WFI instruction	Register setting + SNOOZE request occurrence in SSTBY mode	Register setting + WFI instruction
Canceling method		All interrupts. Any reset available in the mode.	Interrupts shown in Table 13.6 . Any reset available in the mode.	Interrupts shown in Table 13.6 . Any reset available in the mode.	All interrupts. Any reset available in the mode.	Interrupts shown in Table 13.6 . Any reset available in the mode.	Interrupts shown in Table 13.6 . 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)	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)
State after cancellation by a reset		Program execution state (reset processing)	Program execution state (reset processing)	Program execution state (reset processing)	Program execution state (reset processing)	Program execution state (reset processing)	Program execution state (reset processing)

Table 13.5 Entering and Canceling Low Power Consumption Modes and Operating States in each Mode (10 of 12)

Way of Entry/Cancellation and Operating State		Power Supply Mode: MINPWON					
		NORMAL SLEEP	NORMAL SNOOZE	NORMAL SSTBY	VBB SLEEP	VBB SNOOZE	VBB SSTBY
AWO domain	Sub-clock oscillator (SOSC)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Clock correction circuit (CCC)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Power-on reset circuit (POR)	Operating	Operating	Operating	Operating	Operating	Operating
	Low voltage detection circuit 0 (LVD0)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Low voltage detection circuit 1 (LVD1)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Low voltage detection circuit BAT (LVDBAT)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Back bias voltage control (VBBC)	Selectable	Selectable	Selectable	Operating	Operating	Operating
	Energy harvesting control (EHC)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	NMI pin, IRQ0_DS to IRQ3_DS pin interrupt	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Wakeup timer (WUPT)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Realtime Clock (RTC)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable

Table 13.5 Entering and Canceling Low Power Consumption Modes and Operating States in each Mode (11 of 12)

Way of Entry/Cancellation and Operating State		Power Supply Mode: MINPWON					
		NORMAL SLEEP	NORMAL SNOOZE	NORMAL SSTBY	VBB SLEEP	VBB SNOOZE	VBB SSTBY
ISO1 domain	CPU	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)
	SRAM	Selectable	Selectable	Stopped (retained/undefined)*10	Selectable	Selectable	Stopped (retained/undefined)*10
	Main clock oscillator (MOSC)	Selectable	Selectable *1	Stopped	Operation prohibited	Operation prohibited	Operation prohibited
	Main clock oscillator stop detection function	Selectable	Operation prohibited	Operation prohibited	Operation prohibited	Operation prohibited	Operation prohibited
	Middle-speed on-chip oscillator (MOCO)	Selectable	Selectable	Stopped	Operation prohibited	Operation prohibited	Operation prohibited
	Low-speed on-chip oscillator (LOCO)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	IWDT-dedicated on-chip oscillator (IWDTLOCO)	Selectable *2	Selectable *2	Selectable *2	Selectable *2	Selectable *2	Selectable *2
	Independent Watchdog Timer (IWDT)	Selectable *2	Selectable *2	Selectable *2	Selectable *2	Selectable *2	Selectable *2
	Clock output function	Selectable	Selectable	Selectable *3	Selectable *3	Selectable *3	Selectable *3
	Data Transfer Controller (DTC)	Selectable	Selectable	Stopped (retained)	Selectable	Selectable	Stopped (retained)
	DMA Controller (DMAC)	Selectable	Operation prohibited	Stopped (retained)	Selectable	Operation prohibited	Stopped (retained)
	Event Link Controller (ELC)	Selectable	Selectable *4	Stopped (retained)	Selectable	Selectable *4	Stopped (retained)
	Asynchronous General Purpose Timers n : 16 bit (AGTn (n = 0, 1))	Selectable	Selectable *5 *7	Selectable *5 *7	Selectable	Selectable *5 *7	Selectable *5 *7
	Asynchronous General Purpose Timers n : 32 bit (AGTWn (n = 0, 1))	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	SysTick timer	Selectable	Stopped (retained)	Stopped (retained)	Selectable	Stopped (retained)	Stopped (retained)
	Serial Communications Interface 0 (SCI0)	Selectable	Selectable *1 *6	Stopped (retained)	Selectable	Selectable *6	Stopped (retained)
	I ² C Bus Interface 0 (IIC0)	Selectable	Operation prohibited	Stopped (retained)	Selectable	Operation prohibited	Stopped (retained)
	MIP Liquid Crystal Controller (MLCD)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	Key interrupt function (KINT)	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	LST	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable
	IRQ0 to IRQ9 pin interrupt	Selectable	Selectable	Selectable	Selectable	Selectable	Selectable

Table 13.5 Entering and Canceling Low Power Consumption Modes and Operating States in each Mode (12 of 12)

Way of Entry/Cancellation and Operating State		Power Supply Mode: MINPWON					
		NORMAL SLEEP	NORMAL SNOOZE	NORMAL SSTBY	VBB SLEEP	VBB SNOOZE	VBB SSTBY
ISO2 domain	High-speed on-chip oscillator (HOCO)	Operation prohibited	Operation prohibited	Operation prohibited	Operation prohibited	Operation prohibited	Operation prohibited
	WDT	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)
	14-Bit A/D Converter (ADC14)	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)
	DOC	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)
	Other functions	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)	Stopped (undefined)
ISO3 domain	Flash memory	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)	Stopped (retained)

Note: "Selectable" means that operating or stopping can be selected by control registers.

"Operation prohibited" means that the function must be stopped before entering the target mode.

"Stopped" means the action that the function is stopped when the mode transitions to the corresponding mode.

"Stopped (retained)" means that the values of the internal registers are retained, and the operations are suspended.

"Stop (undefined)" means that the values of the internal registers are undefined, and the circuit is powered off.

Note 1. When using SCI0 in snooze mode, make sure that the MOSCCR.MOSTP bit is set to 1.

Note 2. For IWDTC-dedicated on-chip oscillator and IWDTC, operating or stopping can be selected by setting the IWDTC Stop Control bit (IWDTCSTPCTL) in the Option Function Select Register 0 (OFS0) in IWDTC auto start mode. For more details, see [Table 13.12](#).

Note 3. Clock is output only when any of LOCO/SOSC/CCC_2K clocks is selected by the CKOCR.CKOSEL[2:0] bits.

Note 4. This only applies to events which are available in snooze mode.

Note 5. AGT1 and AGTW1 can operate when the underflow event signal from SOSC/LOCO/AGT0/AGTW0 is selected as the count source.

Note 6. The falling edge of the RXD0 signal can be detected in the snooze mode, and when the SCI is set in the asynchronous mode.

Note 7. AGT0 and AGTW0 are operable only when either of SOSC/LOCO is selected

Note 8. Operating or stopping can be selected by setting the WDT Stop Control bit (WDTSTPCTL) and WDT Clock Source Select bit (WDTCLKSEL) in the Option Function Select Register 0 (OFS0), and the bit for Count Stop Control in Sleep, Snooze, or Software Standby Mode (SLCSTP) in the WDT Count Stop Control Register (WDTCSSTPR). For more details, see [Table 13.13](#).

Note 9. When using the 14-bit A/D Converter in snooze mode, set the ADCMPCCR.CMPAE and ADCMPCCR.CMPBE bits to 1.

Note 10. Retained or undefined is selectable depending on the setting of RAM Cutoff Control Register (RAMSDCR). For details, see [section 13.2.10. RAMSDCR : RAM Cutoff Control Register](#).

Note 11. When PWSTCR.SSBYPWG is set to 1 and transitions to software standby mode, it will stop (undefined). Re-setting is required to use these functions.

Table 13.6 Interrupt Sources for Canceling Snooze, Software Standby, and Deep Software Standby Modes (1 of 2)

Interrupt Source	Name	Software Standby Mode	Snooze Mode	Deep Software Standby Mode
Non-maskable interrupt		Yes	Yes	Yes
Port	PORT_IRQn (n = 4 to 9)	Yes	Yes	No
	PORT_IRQn_DS (n = 0 to 3)	Yes	Yes	Yes
LVD	LVD_LVD1	Yes	Yes	Yes
	LVD_LVDBAT	Yes	Yes	Yes
EHC	SOL_DH	Yes	Yes	No
	SOL_DL	Yes	Yes	No
AGT0	AGT0_AGTCAI	Yes	Yes	No
AGT1	AGT1_AGTI	Yes	Yes	No
	AGT1_AGTCAI	Yes	Yes	No
AGTW0	AGTW0_AGTCAI	Yes	Yes	No
AGTW1	AGTW1_AGTCAI	Yes	Yes	No

Table 13.6 Interrupt Sources for Canceling Snooze, Software Standby, and Deep Software Standby Modes (2 of 2)

Interrupt Source	Name	Software Standby Mode	Snooze Mode	Deep Software Standby Mode
WUPT	WUPT_OVI	Yes	Yes	Yes
IWDT	IWDT_NMIUNDF	Yes	Yes	No
RTC	RTC_ALM	Yes	Yes	Yes
	RTC_PRD	Yes	Yes	Yes
KINT	KEY_INTKR	Yes	Yes	No
CCC	CCC_PRD	Yes	Yes	Yes
ICU	ICU_SNZCANCEL	No	Yes	No
ADC14	ADC140_WCMPPM	No	Yes with SELSR0 register*1 *3	No
	ADC140_WCMPUM	No	Yes with SELSR0 register*1 *3	No
SCIO	SCIO_AM	No	Yes with SELSR0 register*1 *3	No
	SCIO_RXI_OR_ERI	No	Yes with SELSR0 register*1 *3	No
DTC	DTC_COMPLETE	No	Yes with SELSR0 register*1 *3	No
DOC	DOC_DOPCI	No	Yes with SELSR0 register*1	No

Note: Operation might be disabled depending on the power supply mode. For the state of each function in low power consumption modes, see [Table 13.5](#).

Note 1. To use the interrupt request as a trigger for exiting snooze mode, the interrupt request must be selected by the SELSR0 register. For details, see [section 16, Interrupt Controller Unit \(ICU\)](#). When a trigger selected in the SELSR0 register occurs during the transition from normal operating mode to software standby mode after execution of a WFI instruction, whether the interrupt request can be accepted depends on the timing of the occurrence of the interrupt request.

Note 2. Either SCIO_AM or SCIO_RXI_OR_ERI can be set.

Note 3. When this snooze end request is enabled in the SNZEDCR register, do not select the request in the SELSR0 register because the destination of transition from snooze mode differs between the case where the SNZEDCR is used and the case where the SELSR0 register is used

13.2 Register Descriptions

13.2.1 MSTPCRA : Module Stop Control Register A

Base address: SYSC = 0x4001_E000

Offset address: 0x01C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	MSTP A22	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

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

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

13.2.2 MSTPCRB : Module Stop Control Register B

Base address: MSTP = 0x4004_7000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTP B31	MSTP B30	MSTP B29	MSTP B28	MSTP B27	MSTP B26	—	—	—	MSTP B22	—	—	MSTP B19	MSTP B18	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MSTP B9	MSTP B8	—	MSTP B6	MSTP B5	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 1. The write value should be 1.	R/W
5	MSTPB5	IrDA Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
6	MSTPB6	QSPI Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
7	—	This bit is read as 1. The write value should be 1.	R/W
8	MSTPB8	IIC1 Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
9	MSTPB9	IIC0 Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
17:10	—	These bits are read as 1. The write value should be 1.	R/W
18	MSTPB18	SPI1 Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
19	MSTPB19	SPI0 Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21:20	—	These bits are read as 1. The write value should be 1.	R/W
22	MSTPB22	SCI9 Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
25:23	—	These bits are read as 1. The write value should be 1.	R/W
26	MSTPB26	SCI5 Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
27	MSTPB27	SCI4 Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
28	MSTPB28	SCI3 Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
29	MSTPB29	SCI2 Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
30	MSTPB30	SCI1 Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31	MSTPB31	SCI0 Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

13.2.3 MSTPCRC : Module Stop Control Register C

Base address: MSTP = 0x4004_7000

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTP C31	—	—	—	—	MSTP C26	MSTP C25	—	—	MSTP C22	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MSTP C15	MSTP C14	MSTP C13	—	—	—	—	—	—	—	—	—	—	—	MSTP C1	MSTP C0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MSTPC0	CAC Module Stop ^{*1} 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
1	MSTPC1	CRC Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
12:2	—	These bits are read as 1. The write value should be 1.	R/W
13	MSTPC13	DOC Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14	MSTPC14	ELC Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
15	MSTPC15	DIV Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21:16	—	These bits are read as 1. The write value should be 1.	R/W
22	MSTPC22	DIL Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
24:23	—	These bits are read as 1. The write value should be 1.	R/W
25	MSTPC25	MLCD Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
26	MSTPC26	GDT Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
30:27	—	These bits are read as 1. The write value should be 1.	R/W
31	MSTPC31	TSIP-Lite Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Note 1. The MSTPC0 bit must be written while the oscillation of the clock to be controlled by this bit is stable. To enter Software Standby mode after writing this bit, wait for 2 cycles of the slowest clock from the clocks output by the oscillators, then execute a WFI instruction.

13.2.4 MSTPCRD : Module Stop Control Register D

Base address: MSTP = 0x4004_7000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	MSTP D22	—	—	—	MSTP D18	MSTP D17	MSTP D16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MSTP D14	MSTP D13	MSTP D12	MSTP D11	MSTP D10	MSTP D9	—	MSTP D7	MSTP D6	MSTP D5	—	MSTP D3	MSTP D2	MSTP D1	MSTP D0
Value after reset:	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MSTPD0	LST Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
1	MSTPD1	TMR Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
2	MSTPD2	AGT1 Module Stop ^{*1} 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
3	MSTPD3	AGT0 Module Stop ^{*1} 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
4	—	This bit is read as 1. The write value should be 1.	R/W
5	MSTPD5	GPT320 and GPT321 Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
6	MSTPD6	GPT162 to GPT165 Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
7	MSTPD7	CCC Module Stop ^{*2} 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
8	—	This bit is read as 1. The write value should be 1.	R/W
9	MSTPD9	RTC Module Stop ^{*3} 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
10	MSTPD10	IWDT Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
11	MSTPD11	WDT Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
12	MSTPD12	AGTW0 Module Stop ^{*1} 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
13	MSTPD13	AGTW1 Module Stop*1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14	MSTPD14	POE Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
15	—	This bit is read as 1. The write value should be 1.	R/W
16	MSTPD16	ADC14 Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
17	MSTPD17	VREF Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
18	MSTPD18	WUPT Module Stop*4 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21:19	—	These bits are read as 1. The write value should be 1.	R/W
22	MSTPD22	TSN Module Stop 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31:23	—	These bits are read as 1. The write value should be 1.	R/W

Note 1. When the sub-clock oscillator or LOCO is selected as the count source, AGTn (n = 0, 1) and AGTWn (n = 0, 1) counting does not stop even if the MSTPD2, 3, 12, 13 bit is set to 1.

Note 2. CCC counting does not stop even if the MSTPD7 bit is set to 1.

Note 3. Even if the MSTPD9 bit is set to 1, RTC counting does not stop.

Note 4. Even if the MSTPD18 bit is set to 1, WUPT counting does not stop.

13.2.5 FSTPCR : Function Stop Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x140

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DBGS TP	—	—	—	—	—	—	—

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	DBGSTP	Debug function stop 0: Disable 1: Enable (power-saving functions are enabled)	R/W

Note: When writing to the FSTPCR register, set the PRC.PRC1 bit to 1 before writing.

DBGSTP bit (Debug function stop)

Unnecessary debug clock is stopped in user mode by enabling DBGSTP bit, thereby reducing power consumption during operation. When the debugger is connected, it becomes invalid and the debug clock is always supplied.

13.2.6 PWSTCR : Power Supply State Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x420

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SSBY ACC	SSBY VBB	SSBY PWG	—	PWST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	PWST[2:0]	Power Supply State Select 0 0 0: Transition to one of the following modes - Switching high-speed/low-speed modes in normal mode (NORMAL) - Software standby mode - Deep software standby mode 0 0 1: Transition to all power supply mode (ALLPWON) ^{*1} 0 1 0: Transition to exclude-flash power supply mode (EXFPWON) 0 1 1: Transition to minimum power supply mode (MINPWON) 1 0 0: Transition to normal mode (NORMAL) 1 0 1: Transition to boost mode (BOOST) 1 1 0: Transition to low leakage current mode (VBB) 1 1 1: Setting prohibited	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	SSBYPWG	Select voltage cut-off area in software standby mode 0: In software standby mode, shut off only the flash power domain (ISO3). 1: In software standby mode, shut off the flash power domain (ISO3) and the power domain (ISO2) for most peripheral functions.	R/W
5	SSBYVBB	Back bias voltage control selection in software standby mode 0: Back bias voltage control is not performed in software standby mode. 1: Back bias voltage control is performed in the software standby mode.	R/W
6	SSBYACC	Fast transition / return select in software standby mode 0: Not selects transition to software standby mode and reduction of recovery time 1: Selects transition to software standby mode under specific conditions and reduction of recovery time ^{*2}	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: When writing to the PWSTCR register, set the PRC.PRC1 bit to 1 before writing.

Note: It is not possible to transit to software standby mode with the HOCO oscillation frequency set to 48MHz or 64MHz by the HOCOMCR.HCFRQ [1: 0] bits. Follow the setup instructions on [section 9.2.7. HOCOMCR : High-Speed On-Chip Oscillator Mode Control Register](#).

Note 1. When the chip is to be placed in all power supply mode (ALLPWON), setting ICLK ≤ 4 MHz is required before executing the WFE instruction.

Note 2. Transition to software standby mode and return time are shortened in the following cases.

SSTBY transition / return with ALLPWON

- Normal mode ↔ Software standby mode
- VBB mode ↔ Software standby mode

The following conditions at EXFPWON:

- High-Speed mode ↔ Software standby mode
- Low-Speed mode ↔ Software standby mode (when SSBYPWG = 1, SSBYVBB = 1)
- VBB ↔ Software standby mode (when SSBYPWG = 1)

The following conditions at MINPWON:

- High-Speed mode ↔ Software standby mode
- Low-Speed mode ↔ Software standby mode (when SSBYVBB = 1 is set)

PWST[2:0] bit (Power Supply State Select)

The PWST[2:0] bits select the power supply state.

In each power supply mode (ALLPWON, EXFPWON, or MINPWON), power supply or power shutoff is controlled for each power domain. Current consumption can be reduced by cutting off unnecessary power supply area. See [Table 13.2](#) for the power domain of each function, and see [Table 13.6](#) for power supply/power shutoff of the power domain in each power supply mode.

In each power supply mode, a power control mode (BOOST, NORMAL, or VBB) is selected according to the operating frequency.

In mode transition by the WFE instruction, software standby mode is temporarily entered and then the specified mode is entered. Therefore, the following bits must be set besides these bits, and then, a WFE instruction must be executed at the end.

- SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0 (software standby mode set)
- SNZCR.SNZE = 0 (snooze mode disabled)

To change high-speed/low-speed mode in normal mode in each power supply mode, set these bits to 000b before setting the OPCCR register. At that time, a WFI/WFE instruction does not need to be executed.

To enter SSTBY or DSTBY in each power supply mode, set these bits to 000b and then execute a WFI instruction.

For details on switching of the power supply mode and power control mode, see [section 13.5. Functions for Reducing Power](#) and also [section 13.7.14. Notes on Switching Power Supply, Power Control, and Low Power Consumption Modes](#).

SSBYPWG bit (Select voltage cut-off area in software standby mode)

The SSBYPWG bit is used to select the power shutdown area in software standby mode. When this bit is set to 1, the area where power is cut off during software standby mode increases, and the current consumption during software standby mode can be reduced.

When this bit is set to 1 and a transition is made to software standby mode, it is necessary to reset the functions in the ISO2 and ISO3 areas after returning from software standby mode.

When transitioning from the minimum power supply mode (MINPWON) to the software standby mode, this bit setting is invalid.

SSBYVBB bit (Back bias voltage control selection in software standby mode)

The SSBYVBB bit is used to select the back bias voltage application in software standby mode.

When this bit is set to 1, back bias voltage application is selected and current consumption during software standby mode can be reduced.

When transitioning to software standby mode, set this bit to 1 before completing the back bias voltage control (VBBC) setup (VBBST.VBBSTUP = 1) before transitioning.

This bit has no effect when transitioning from low-leakage current mode (VBB) to software standby mode.

SSBYACC bit (Fast transition / return select in software standby mode)

The SSBYACC bit is used to select the transition time to software standby mode under specific conditions and the reduction of the return time from software standby mode under specific conditions. See ^{*2} for the conditions to shorten the transition time and return time to software standby mode.

When this bit is set to 1, the transition time to software standby mode under specific conditions and the return time from software standby mode under specific conditions are reduced.

When this bit is set to 1 and a transition is made to software standby mode, the current consumption in software standby mode will increase.

13.2.7 PWSTF : Power Supply State Flag Register

Base address: SYSC = 0x4001_E000

Offset address: 0x421

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	VBBM	BOOSTM	MINPWONM	EXFPWONM	ALLPWONM
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	ALLPWONM	All Power Supply Mode Flag This flag indicates that the power supply state is all power supply mode (ALLPWON). See Table 13.7 .	R
1	EXFPWONM	Exclude-flash Power Supply mode Flag This flag indicates that the power supply state is exclude- flash power supply mode (EXFPWON). See Table 13.7 .	R
2	MINPWONM	Minimum Power Supply Mode Flag This flag indicates that the power supply state is minimum power supply mode (MINPWON). See Table 13.7 .	R
3	BOOSTM	Boost Mode Flag This flag indicates that the power control state is boost mode (BOOST). See Table 13.7 .	R
4	VBBM	Low Leakage Current Mode Flag This flag indicates that the power control state is low leakage current mode (VBB). See Table 13.7 .	R
7:5	—	These bits are read as 0.	R

Table 13.7 List of Power Supply State and AssociatedFlags

ALLPWONM	EXFPWONM	MINPWONM	BOOSTM	VBBM	Power Supply State of the Product	
					Power Supply Mode	Power Control Mode
1	0	0	0	0	ALLPWON	NORMAL
				1		VBB
			1	0		BOOST
0	1	0	0	0	EXFPWON	NORMAL
				1		VBB
0	0	1	0	0	MINPWON	NORMAL
				1		VBB

13.2.8 OPCCR : Operating Power Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0A0

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

Bit	Symbol	Function	R/W
1:0	OPCM[1:0]	Operating power control mode selection 0 0: High-speed mode 1 1: Low-speed mode Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	OPCMTSF	Operating power control mode (High-Speed / Low-Speed mode) transition state flag 0: Transition completed 1: During transition	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: When writing to the OPCCR register, set the PRC.PRC1 bit to 1 before writing.

The OPCCR is used to reduce power consumption for normal mode and sleep mode by specifying a lower operating frequency. For the procedure for changing the normal mode setting, see [section 13.5. Functions for Reducing Power](#).

OPCM[1:0] bits (Operating power control mode selection)

The OPCM[1:0] select the operating power control mode for normal operating mode and sleep mode. The PWSTCR.PWST[2:0] bits must be set to 000b before setting this register.

Changing the setting of these bits in boost mode (PWSTF.BOOSTM = 1) or low leakage current mode (PWSTF.VBBM = 1) is prohibited.

OPCMTSF flag (Operating power control mode (High-Speed / Low-Speed mode) transition state flag)

The OPCMTSF flag is set to 1 on a write access to the OPCM[1:0] bits and to 0 when the mode transition completes. Confirm that the flag is 0 before proceeding to the next step.

This flag indicates the transition state of high-speed/low-speed modes in normal mode. Confirm the transition state of power control modes by the associated flag of the PWSTF register.

For the operating frequency range and voltage range, see [section 51, Electrical Characteristics](#). Each operating power control mode in normal mode is shown below.

High-Speed mode

After the reset is released, this MCU will operate in this mode. After the reset is released, the power supply is in the full power supply mode (ALLPWON). This mode has the following limitations:

- There is a function that is prohibited to operate in this mode. See [Table 13.4](#) and [\(2\)Valid setting for the clock-related registers](#).
- There is a mode that can not transition from this mode. See [section 13.7.14. Notes on Switching Power Supply, Power Control, and Low Power Consumption Modes](#).

Low-Speed mode

This mode reduces power consumption compared to High-Speed mode if the same conditions (operating frequency, operating voltage, etc.) are performed the same. This mode has the following limitations:

- There is a function that is prohibited to operate in this mode. See [Table 13.4](#) and [\(2\)Valid setting for the clock-related registers](#).
- There is a mode that can not transition from this mode. See [section 13.7.14. Notes on Switching Power Supply, Power Control, and Low Power Consumption Modes](#).

13.2.9 SBYCR : Standby Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x00C

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

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

Note: When writing to the SBYCR register, set the PRC.PRC1 bit to 1 before writing.

SSBY bit (Software Standby Mode Select)

Sets the transition destination after executing the WFI instruction in combination with the setting of the DPSBYCR.DPSBY bit. If the WFI instruction is executed while PWSTCR.PWST [2:0] = 000b and this bit is set to 1, the mode changes to the mode shown in Table 13.8.

Table 13.8 Transition setting of low power consumption mode and power supply mode

State before WFI instruction execution			State after WFI instruction execution	
SBYCR. SSBY	DPSBYCR. DPSBY	Power supply mode	Low power mode	Power supply mode
0	X	ALLPWON	Sleep mode	Same as before the transition
		EXFPWON		
		MINPWON		
1	0	ALLPWON	Software standby mode	PWSTCR.SSBYPWG bit: 0:EXFPWON 1:MINPWON
		EXFPWON		
		MINPWON		
	1	ALLPWON	Deep Software Standby Mode	Same as before the transition
		EXFPWON		
		MINPWON		

Note: When transitioning to Software Standby Mode/Deep Software Standby Mode, in addition to the above settings, it is necessary to set PWSTCR.PWST[2:0] = 000b

For switching of low power consumption modes, see also [section 13.7.14. Notes on Switching Power Supply, Power Control, and Low Power Consumption Modes](#).

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

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

13.2.10 RAMSDCR : RAM Cutoff Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x00E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	RAMS D3	RAMS D2	RAMS D1	RAMS D0

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
3:0	RAMSD3 to RAMSD0	SRAM Cutoff in Software Standby Mode 0: Does not cut off the power supply to SRAM during software standby mode (SRAM values: retained). 1: Cut off the power supply to SRAM during software standby mode (SRAM values: undefined).	R/W

Bit	Symbol	Function	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: When writing to the RAMSDCR register, set the PRC.PRC1 bit to 1 before writing.

The RAMSDCR register controls the cutoff of power supply to SRAM during software standby mode.

Current in software standby mode can be reduced by cutting off the power supply to unnecessary SRAM areas. When the MCU enters software standby mode with the RAMSDn bit set to 1 (cutting off the power supply to SRAM during software standby mode), the power supply to the target SRAM is cut off and SRAM values are lost. When the

SRAM values need to be retained, set the RAMSDn bit to 0 (the power supply to SRAM is not cut off during software standby mode).

Even if the RAMSDn bit is set to 1 (the power supply to SRAM is cut off during software standby mode) during normal operating mode, sleep, or snooze mode, the power supply to SRAM is not cut off. In deep software standby mode, SRAM values are undefined regardless of the setting of the RAMSDn bit.

Table 13.9 shows the correspondence between the RAMSDCR.RAMSDn bit and the SRAM area for cutting off power supply.

Table 13.9 AMSDCR.RAMSDnBit and Target SRAM Area for Cutting off Power Supply

RAMSDn bit (n = 0 to 3)	SRAM Area for Cutting off Power Supply
RAMSD0	0x2000_0000 to 0x2000_7FFF
RAMSD1	0x2000_8000 to 0x2000_FFFF
RAMSD2	0x2001_0000 to 0x2001_7FFF
RAMSD3	0x2001_8000 to 0x2001_FFFF

When cutting off multiple areas, it is not necessary to specify consecutive areas. For example, a setting such as RAMSDCR = 0000 0101b is also possible.

When transferring SRAM values after transition to snooze mode or when transferring values to SRAM in snooze mode and then returning to software standby mode, note that values are undefined in the SRAM area for which the corresponding RAMSDn bit is set to 1.

13.2.11 SNZCR : Snooze Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x092

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SNZE	—	—	—	—	—	SNZD TCEN	RXDREQEN

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	RXDREQEN	RXD0 Snooze Request Enable 0: Ignore RXD0 falling edge in Software Standby mode 1: Detect RXD0 falling edge in Software Standby mode	R/W
1	SNZDTCEN	DTC Enable in Snooze mode 0: Disable DTC operation 1: Enable DTC operation	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	SNZE	Snooze mode Enable 0: Disable Snooze mode 1: Enable Snooze mode	R/W

Note: When writing to the SNZCR register, set the PRC.PRC1 bit to 1 before writing.

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

RXDREQEN bit (RXD0 Snooze Request Enable)

The RXDREQEN bit specifies whether to detect a falling edge of the RXD0 pin in Software Standby mode. This bit can be used only when SCIO is operating in asynchronous mode. To detect a falling edge of the RXD0 pin, set this bit before entering Software Standby mode. When this bit is set to 1, a falling edge of the RXD0 pin in Software Standby mode causes the MCU to enter Snooze mode.

SNZDTCEN bit (DTC Enable in Snooze mode)

The SNZDTCEN bit specifies whether to use the DTC and SRAM in Snooze mode. To use the DTC and SRAM in Snooze mode, set this bit to 1 before entering Software Standby mode. When this bit is set to 1, the DTC can be activated by setting IELSRn register.

SNZE bit (Snooze mode Enable)

The SNZE bit specifies whether to enable a transition from Software Standby mode to Snooze mode. To use Snooze mode, set this bit to 1 before entering Software Standby mode. When this bit is set to 1, a trigger as shown in [Table 13.15](#) in Software Standby mode causes the MCU to enter Snooze mode. After the MCU transitions from Software Standby mode or Snooze mode to Normal operating mode, set 0 to the SNZE bit once then set it before re-entering Software Standby mode. For details, see [section 13.6.3. Snooze Mode](#).

13.2.12 SNZEDCR0 : Snooze End Control Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x094

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SCIOUMTED	—	—	AD0UMTED	AD0MATED	DTCNZRED	DTCZRED	AGTUNFED
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AGTUNFED	AGT1 Underflow Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
1	DTCZRED	Last DTC Transmission Completion Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
2	DTCNZRED	Not Last DTC Transmission Completion Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
3	AD0MATED	AD Compare Match Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
4	AD0UMTED	AD Compare Mismatch Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	SCIOUMTED	SCIO Address Mismatch Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W

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

The SNZEDCR0 register controls the condition of switching from Snooze mode to Software Standby mode. In order to use a trigger shown in [Table 13.16](#) as a condition to switch from Snooze mode to Software Standby mode, the corresponding bit in the SNZEDCR0 register must be set to 1.

For the event to return from snooze mode to normal operating mode as shown in [Table 13.6](#), set the SNZEDCR0 register so that there is only one transition destination.

Because the ISO2 domain is powered off in minimum power supply mode (MINPWON), the snooze end requests must not be enabled for the functions in the ISO2 domain. For the power domain of each function, see [Table 13.2](#).

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 on the trigger conditions, see [section 25, Low Power Asynchronous General Purpose Timer \(AGT, AGTW\)](#).

DTCZRED bit (Last DTC Transmission Completion Snooze End Enable)

The DTCZRED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on completion of the last DTC transmission, that is, when CRA or CRB registers in the DTC is 0. For details on the trigger conditions, see [section 20, Data Transfer Controller \(DTC\)](#).

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

The DTCNZRED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on completion of each DTC transmission, that is, when CRA or CRB registers in the DTC is not 0. For details on the trigger conditions, see [section 20, Data Transfer Controller \(DTC\)](#).

AD0MATED bit (AD Compare Match Snooze End Enable)

The AD0MATED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on an AD0 event when a conversion result matches the expected data. For details on the trigger conditions, see [section 44, 14-Bit A/D Converter \(ADC14\)](#).

AD0UMTED bit (AD Compare Mismatch Snooze End Enable)

The AD0UMTED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on an AD0 event when the conversion result does not match the expected data. For details on the trigger conditions, see [section 44, 14-Bit A/D Converter \(ADC14\)](#).

SCI0UMTED bit (SCI0 Address Mismatch Snooze End Enable)

The SCI0UMTED bit specifies whether to enable a transition from Snooze mode to Software Standby mode 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 32, Serial Communications Interface \(SCI\)](#). Only set this bit to 1 when SCI0 operates in asynchronous mode.

13.2.13 SNZREQCR0 : Snooze Request Control Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x098

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	SNZR EQEN 30	SNZR EQEN 29	SNZR EQEN 28	—	—	SNZR EQEN 25	SNZR EQEN 24	SNZR EQEN 23	—	—	—	—	—	SNZR EQEN 17	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SNZR EQEN 7	SNZR EQEN 6	SNZR EQEN 5	SNZR EQEN 4	SNZR EQEN 3	SNZR EQEN 2	SNZR EQEN 1	SNZR EQEN 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SNZREQEN0	Enable IRQ0 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W

Bit	Symbol	Function	R/W
1	SNZREQEN1	Enable IRQ1 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
2	SNZREQEN2	Enable IRQ2 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
3	SNZREQEN3	Enable IRQ3 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
4	SNZREQEN4	Enable IRQ4 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
5	SNZREQEN5	Enable IRQ5 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
6	SNZREQEN6	Enable IRQ6 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
7	SNZREQEN7	Enable IRQ7 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
16:8	—	These bits are read as 0. The write value should be 0.	R/W
17	SNZREQEN17	Enable KINT snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
22:18	—	These bits are read as 0. The write value should be 0.	R/W
23	SNZREQEN23	AGTW0 compare match A snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
24	SNZREQEN24	Enable RTC alarm snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
25	SNZREQEN25	Enable RTC period snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
28	SNZREQEN28	Enable AGT1 underflow snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
29	SNZREQEN29	Enable AGT1 compare match A snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
30	SNZREQEN30	Enable AGT0 compare match A snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: When writing to the SNZREQCR0 register, set the PRC.PRC1 bit to 1 before writing.

The SNZREQCR0 register controls which triggers cause the MCU to switch from software standby to snooze mode. If a trigger is selected as a request to cancel software standby mode in the WUPEN register (see [section 16, Interrupt Controller Unit \(ICU\)](#)), the MCU enters normal operating mode when the trigger is generated even when the associated bit of the SNZREQCR register is 1. The WUPEN register setting always has higher priority than the SNZREQCR0 register setting. For details, see [section 13.6.3. Snooze Mode](#), and [section 16, Interrupt Controller Unit \(ICU\)](#).

Since the ISO2 domain is powered off in software standby mode in minimum power supply mode (MINPWON), the snooze requests must not be enabled for the functions under the ISO2 domain. For the power domain for each function, see [Table 13.2](#).

13.2.14 DPSBYCR : Deep Software Standby Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x400

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

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	IOKEEP	I/O Port Rentention 0: When the Deep Software Standby mode is canceled, clear the I/O port pins to the reset state. 1: When the Deep Software Standby mode is canceled, keep the I/O port pins in the same state as in Deep Software Standby mode.	R/W
7	DPSBY	Deep Software Standby Mode Selection 0: Sleep mode (SBYCR.SSBY=0) / Software Standby mode (SBYCR.SSBY=1) 1: Sleep mode (SBYCR.SSBY=0) / Deep Software Standby mode (SBYCR.SSBY=1)	R/W

Note: When writing to the DPSBYCR register, set the PRC.PRC1 bit to 1 before writing.

The DPSBYCR register controls the Deep Software Standby mode.

DPSBYCR is not initialized by the internal reset signal that is the source to cancel the Deep Software Standby mode. For details, see [section 6, Resets](#).

IOKEEP bit (I/O Port Rentention)

In Deep Software Standby mode, I/O ports keep the same states as in the Software Standby mode. The IOKEEP bit specifies whether to reset the state of the I/O ports or not when the Deep Software Standby mode is canceled.

DPSBY bit (Deep Software Standby Mode Selection)

The DPSBY bit controls transitions to Deep Software Standby mode.

When the WFI instruction is executed while SBYCR.SSBY bit and DPSBYCR.DPSBY bit are both 1, the MCU enters Deep Software Standby mode.

The DPSBY bit remains 1 when Deep Software Standby mode is canceled by certain pins which are sources of external pin interrupts (NMI, IRQ0_DS to IRQ3_DS) or a peripheral interrupt (voltage monitoring 1, or voltage monitoring BAT). Write 0 to this bit to clear it.

When the OFS0.IWDTSTRT bit is 0 (auto-start mode) and the OFS0.IWDTSTPCTL bit is 0 (counting continues), the setting in the DPSBY bit is ignored. Even when SBYCR.SSBY bit is 1 and the DPSBY bit is 1, the transition after the execution of a WFI instruction is to Software Standby mode.

The setting of the DPSBY bit is invalid when voltage monitoring 1 reset is enabled (LVD1CR0.RI = 1) or when a voltage monitoring BAT reset is enabled (LVDBATCR0.RI = 1). In this case, even when the SBYCR.SSBY bit is 1 and the DPSBY bit is 1, the transition after the execution of a WFI instruction is to Software Standby mode.

13.2.15 DPSIER0 : Deep Software Standby Interrupt Enable Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x402

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	DIRQ3 E	DIRQ2 E	DIRQ1 E	DIRQ0 E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ0E	IRQ0_DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
1	DIRQ1E	IRQ1_DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
2	DIRQ2E	IRQ2_DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
3	DIRQ3E	IRQ3_DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: When writing to the DPSIER0 register, set the PRC.PRC1 bit to 1 before writing.

DPSIER0 is not initialized by the internal reset signal used as Deep Software Standby mode Cancelling source. For details, see [section 6, Resets](#).

DPSIFR0 should be cleared to 0 before entering Deep Software Standby mode.

13.2.16 DPSIER1 : Deep Software Standby Interrupt Enable Register 1

Base address: SYSC = 0x4001_E000

Offset address: 0x404

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	DCCCI E	DNMI E	DRTC AIE	DRTC IE	DLVD BATIE	DLVD1 IE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DLVD1IE	LVD1 Interrupt Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
1	DLVDBATIE	LVDBAT Interrupt Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
2	DRTCIE	RTC Cycle Interrupt Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
3	DRTCAIE	RTC Alarm Interrupt Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W

Bit	Symbol	Function	R/W
4	DNMIE	NMI Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W ¹
5	DCCCIE	CCC Periodic Interrupt and WUPT Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: When writing to the DPSIER1 register, set the PRC.PRC1 bit to 1 before writing.

Note 1. 1 can be written only once. After 1 is written to this bit, subsequent write accesses are disabled.

DPSIER1 is not initialized by the internal reset signal used as Deep Software Standby mode Cancelling source. For details, see [section 6, Resets](#).

DPSIFR1 should be cleared to 0 before entering Deep Software Standby mode.

13.2.17 DPSIFR0 : Deep Software Standby Interrupt Flag Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x406

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	DIRQ3 E	DIRQ2 E	DIRQ1 E	DIRQ0 E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ0E	IRQ0_DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W ¹
1	DIRQ1E	IRQ1_DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W ¹
2	DIRQ2E	IRQ2_DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W ¹
3	DIRQ3E	IRQ3_DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W ¹
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: When writing to the DPSIFR0 register, set the PRC.PRC1 bit to 1 before writing.

Note 1. Only 0 can be written to clear the flag. Write 1 to the bits other than bits to be cleared.

Each flag is set to 1 when a cancel request specified by DPSIEGR0 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not even Deep Software Standby mode) or when the setting of DPSIER0 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR0 is cleared to 0x00.

To clear DPSIFR0 to 0x00 after modifying DPSIER0, wait for at least six PCLKB cycles, and then write 0 to DPSIFR0.

DPSIFR0 is not initialized by the internal reset signal used as Deep Software Standby mode Cancelling source. For details, see [section 6, Resets](#).

DIRQnF flags (IRQn_DS Pin Deep Standby Cancel Flag) (n = 0 to 3)

These flags indicate that a cancel request by the IRQn_DS pin has been generated.

[Setting condition]

A cancel request by the IRQn_DS pin specified by DPSIEGR0 is generated.

[Clearing condition]

Writing 0 to the flag.

13.2.18 DPSIFR1 : Deep Software Standby Interrupt Flag Register 1

Base address: SYSC = 0x4001_E000

Offset address: 0x408

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	DCCCI F	DNMIF	DRTC AIF	DRTC IF	DLVD BATIF	DLVD1 IF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DLVD1IF	LVD1 Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W ^{*1}
1	DLVDBATIF	LVDBAT Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W ^{*1}
2	DRTCIF	RTC Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W ^{*1}
3	DRTCAIF	RTC Alarm Interrupt Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W ^{*1}
4	DNMIF	NMI Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W ^{*1}
5	DCCCIF	CCC Periodic Interrupt Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W ^{*1}
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: When writing to the DPSIFR1 register, set the PRC.PRC1 bit to 1 before writing.

Note 1. Only 0 can be written to clear the flag. Write 1 to the bits other than bits to be cleared.

The flags in the DPSIFR1 register are set to 1 when the associated cancel request specified in DPSIEGR1 is generated. Each flag can be set to 1 when a cancel request is generated in any mode, not only in deep software standby mode, or when the setting in DPSIER1 is changed. Clear DPSIFR1 to 0x00 before entering deep software standby mode.

To clear DPSIFR1 to 0x00 after modifying DPSIER1, wait for at least six PCLKB cycles, and then write 0 to DPSIFR1. The DPSIFR1 register is not initialized by the internal reset signal that cancels deep software standby mode. For details, see [section 6, Resets](#).

DLVD1IF flag (LVD1 Deep Software Standby Cancel Flag)

The DLVD1IF flag indicates that a cancel request is generated by the voltage monitor 1 interrupt.

[Setting condition]

- A cancel request generated by the voltage monitor 1 interrupt specified in DPSIEGR1.

[Clearing condition]

- Writing 0 to the flag.

DLVDBATIF flag (LVDBAT Deep Software Standby Cancel Flag)

The DLVDBATIF flag indicates that a cancel request is generated by the voltage monitor BAT interrupt.

[Setting condition]

- A cancel request generated by the voltage monitor BAT interrupt specified in DPSIEGR1.

[Clearing condition]

- Writing 0 to the flag.

DRTCIF flag (RTC Deep Software Standby Cancel Flag)

Indicates that a deep software standby mode release request is generated by an RTC periodic interrupt signal.

[Setting condition]

- When a release request by the RTC periodic interrupt signal selected by the DPSIEGR1 register is generated.

[Clearing condition]

- Writing 0 to the flag.

DRTCAIF flag (RTC Alarm Interrupt Deep Software Standby Cancel Flag)

Indicates the generation of deep software standby mode release request by the RTC alarm interrupt signal.

[Setting condition]

- When a cancel request occurs by the RTC alarm interrupt signal selected by the DPSIEGR1 register.

[Clearing condition]

- Writing 0 to the flag.

DNMIF flag (NMI Pin Deep Software Standby Cancel Flag)

The DNMIF flag indicates that a cancel request is generated by the NMI pin.

[Setting condition]

- A cancel request generated by the NMI pin specified in DPSIEGR1.

[Clearing condition]

- Writing 0 to the flag.

DCCCIF flag (CCC Periodic Interrupt Deep Software Standby Cancel Flag)

The DCCCIF flag indicates that a cancel request is generated by the CCC periodic interrupt.

[Setting condition]

- A cancel request generated by the CCC periodic interrupt.

[Clearing condition]

- Writing 0 to the flag.

13.2.19 DPSIEGR0 : Deep Software Standby Interrupt Edge Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x40A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	DIRQ3 EG	DIRQ2 EG	DIRQ1 EG	DIRQ0 EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ0EG	IRQ0_DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

Bit	Symbol	Function	R/W
1	DIRQ1EG	IRQ1_DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
2	DIRQ2EG	IRQ2_DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
3	DIRQ3EG	IRQ3_DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: When writing to the DPSIEGR0 register, set the PRC.PRC1 bit to 1 before writing.

DPSIEGR0 is not initialized by the internal reset signal that is the source to cancel the Deep Software Standby mode. For details, see [section 6, Resets](#).

13.2.20 DPSIEGR1 : Deep Software Standby Interrupt Edge Register 1

Base address: SYSC = 0x4001_E000

Offset address: 0x40C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DNMI EG	—	—	DLVD BATE G	DLVD1 EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DLVD1EG	LVD1 Edge Select 0: Generate cancel request when $VCC < V_{det1}$ 1: Generate cancel request when $VCC \geq V_{det1}$ (voltage rise) is detected.	R/W
1	DLVDBATEG	LVDBAT Edge Select 0: Generate cancel request when $VBAT\_EHC < V_{detBAT}$ (voltage fall) is detected. 1: Generate cancel request when $VBAT\_EHC \geq V_{detBAT}$ (voltage rise) is detected.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	DNMIEG	NMI Pin Edge Select 0: Generate cancel request on falling edge on the NMI pin. 1: Generate cancel request on rising edge on the NMI pin.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: When writing to the DPSIEGR1 register, set the PRC.PRC1 bit to 1 before writing.

DPSIEGR1 is not initialized by the internal reset signal that is the source to cancel the Deep Software Standby mode. For details, see [section 6, Resets](#).

13.2.21 SYOCDCCR : System Control OCD Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x040E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DBGE N	—	—	—	—	—	—	DOCD F
Value after reset:	0	0	0	0	0	0	0	x

Bit	Symbol	Function	R/W
0	DOCDF	Deep Software Standby OCD flag 0: DBIRQ is not generated 1: DBIRQ is generated	R/W ^{*1}
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	DBGEN	Debugger Enable bit Set to 1 first in on-chip debug mode. 0: On-chip debugger is disabled 1: On-chip debugger is enabled	R/W

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

Note 1. Writing 0 clears the flag. Writing 1 is ignored

SYOCDCCR can be written when DBGSTR.CDBGPWRUPREQ = 1 (the debugger is connected).

SYOCDCCR is not initialized by the internal reset signal that is the source to cancel the Deep Software Standby mode.

DOCDF flag (Deep Software Standby OCD flag)

DOCDF flag indicates that a cancel request of Deep Software Standby mode by the MCUCTRL.DBIRQ bit has been generated. DOCDF flag is set to 1 when a cancel request is generated. This flag may be set to 1 when a cancel request is generated in any mode (not only Deep Software Standby mode). Therefore, a transition to Deep Software Standby mode must be made after DOCDF flag is cleared to 0.

[Setting condition]

- A cancel request by the MCUCTRL.DBIRQ is generated

[Clearing condition]

- Writing 0 to the flag
- When DBGEN bit is 0

DBGEN bit (Debugger Enable bit)

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

[Setting condition]

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

[Clearing condition]

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

Note: Certain restrictions apply in terms of the MCU states in which the DBGEN bit can be set to 1. For details, see [section 2.7.2. Restrictions on Connecting an OCD emulator](#).

13.2.22 VOCCR : Power Supply Open Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x440

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	IV1CT L	IV0CT L	—	AV0CT L
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	AV0CTL	AVCC0 Supply Control 0: The undefined value is propagated from the AVCC0 power domain to other power domains. 1: The undefined value is not propagated from the AVCC0 power domain to other power domains	R/W
1	—	This bit is read as 1. The write value should be 1.	R/W
2	IV0CTL	IOVCC0 Supply Control 0: The undefined value is propagated from the IOVCC0 power domain to other power domains. 1: The undefined value is not propagated from the IOVCC0 power domain to other power domains.	R/W
3	IV1CTL	IOVCC1 Supply Control 0: The undefined value is propagated from the IOVCC1 power domain to other power domains. 1: The undefined value is not propagated from the IOVCC1 power domain to other power domains.	R/W
7:4	—	This bit is read as 1. The write value should be 1.	R/W

Note: When writing to the VOCCR register, set the PRC.PRC1 bit to 1 before writing.

The VOCCR register is used for control such that, when power is not being supplied to the AVCC0, IOVCC0, or IOVCC1 power supply, the operation of circuits that operate with the pins to which power is not being supplied does not affect the circuits that operate with the pins to which power is being supplied.

(a) and (b) in [Figure 13.3](#) show the cases where power is not being supplied to the IOVCC1 pin, and (c) and (d) show the cases where power is being supplied to the IOVCC1 pin. [Table 13.10](#) shows the settings of the IV1CTL bit and circuit operation when power is and is not being supplied to the IOVCC1 pin. When power is not being supplied to the IOVCC1 pin, undefined values are propagated from circuit 1 operating with the power supplied through the IOVCC1 pin to other circuits. In the case of (a) in [Figure 13.3](#), the propagation of undefined values from circuit 1 to circuit 2 affects circuit 2. As shown in (b) in [Figure 13.3](#), setting the IV1CTL bit of the VOCCR register to 1 can suppress the propagation of undefined values from circuit 1 to circuit 2. As shown in (d) in [Figure 13.3](#), when power is being supplied to the IOVCC1 pin and the setting of the IV1CTL bit is 1, the propagation of input values from circuit 1 to circuit 2 is suppressed. As shown in (c) in [Figure 13.3](#), we recommend setting the IV1CTL bit to 0. However, output from circuit 1 is possible regardless of the setting of the IV1CTL bit.

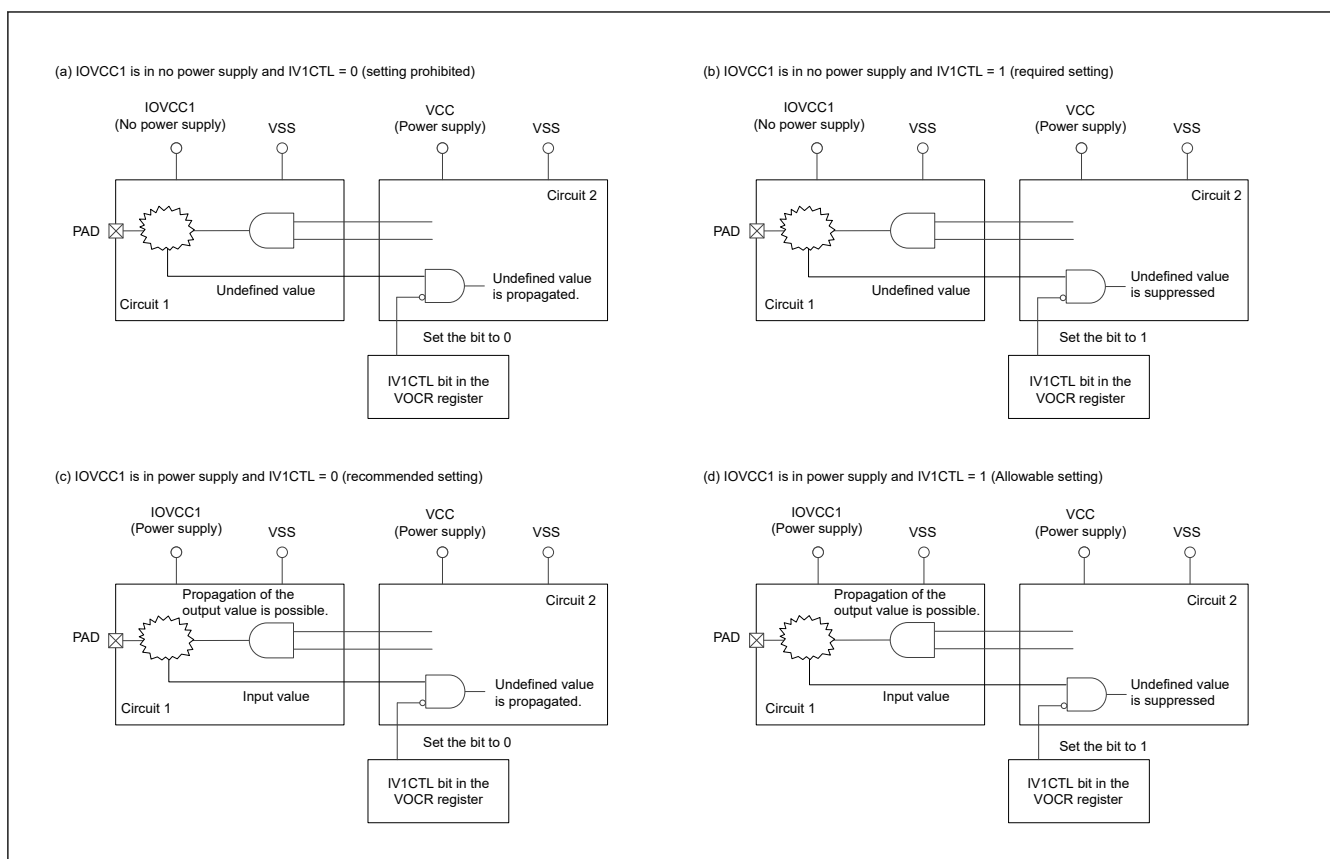


Figure 13.3 Usage Example of the VOCR Register (in Terms of the Setting of the IOVCC1 Pin)

Table 13.10 Examples of VOCR Register Settings and Circuit Operation (in case of the IOVCC1)

State of IOVCC1	Setting of the IV1CTL Bit	Operation	Case ^{*1}
No power supplied	0 (prohibited setting)	The propagation of undefined values from circuit 1 affects circuit 2.	(a)
	1 (required setting)	The propagation of undefined values from circuit 1 to circuit 2 is suppressed.	(b)
Power supplied	0 (recommended setting)	The propagation of input and output values from circuit 1 is possible. When the IOVCC1 pin changes from the state where power is not being supplied to the state where power is being supplied, change the setting of the IV1CTL bit from 1 to 0.	(c)
	1 (allowable setting)	Input values are not propagated from circuit 1 to circuit 2. The propagation of output values from circuit 1 is possible.	(d)

Note 1. See in [Figure 13.3](#)

The following shows the modules assigned to individual power supplies.

Modules assigned to AVCC0 power supply: I/O functions assigned to P000 to P007

Modules assigned to IOVCC0 power supply: I/O functions assigned to Port 8 , P010 to P015

Modules assigned to IOVCC1 power supply: I/O functions assigned to Port 1, Port 3, Port 5, port 6, port 7, and P202 to P205

AV0CTL bit (AVCC0 Supply Control)

The AV0CTL bit are used for control such that, when power is not being supplied to the corresponding AVCC0 power supply pin, the operation of the circuit that operates with the pin to which power is not being supplied does not affect the circuits that operate with the pins to which power is being supplied.

IVmCTL bit (IOVCCm Supply Control (m = 0 to 1))

The IVmCTL bits are used for control such that, when power is not being supplied to the corresponding IOVCCm (m = 0 to 1) power supply pin, the operation of the circuit that operates with the pin to which power is not being supplied does not affect the circuits that operate with the pins to which power is being supplied

The procedure for setting each bit is as follows.

When power is not to be supplied to a power supply pin:

1. Place the corresponding circuit or circuits which operate with the power supply pins to which the supply of power is to be stopped in the module stop state.
2. Set the bits corresponding to the power supply pins to which the supply of power is to be stopped to 1.
3. Read the bit set in step 2 to confirm that the value is 1, and then turn off the external regulator (power supply) by the CPU or external device.

When power is to be supplied to a power supply pin:

1. Start the external regulator (power supply) that has been turned off by the CPU or external device.
2. Confirm that the power supply started in step 1 has reached the values specified in the electrical characteristics, and then clear the bits corresponding to the pins to which power is to be supplied.
3. Cancel the module stop of each function and allow the functions to operate.

For how to handle pins when setting the AV0CTL/IVmCTL bit to 1 and turning off the power supply during deep software standby mode, see [Table 13.11](#).

Table 13.11 Handling of Pins when the Supply of Power is to be Stopped in Deep Software Standby Mode

Pins to which Power is to be Stopped	Pin Handling	
AVCC0, IOVCC0 to IOVCC1	Pxxx	Open, or if the power supply is fixed to GND, the pin can be fixed to GND.
	Dedicated pin	Open

When the supply of power to the power supply pin is to be stopped in deep software standby mode, the I/O pin (Pxxx) cannot retain the output state. Additionally, pins must be in the following state for input.

- Open when power is not supplied
- Open or fixed to GND when the power supply is fixed to GND

13.2.23 LDOCR : Regulator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x428

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

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	LDOCUT	Regulator (LDO) stop 0: LDO works Supply LDO output voltage to internal circuit 1: LDO stops Supply externally applied voltage to internal circuit	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: When writing to the LDOCR register, set the PRC.PRC1 bit to 1 before writing.

The LDOCR register controls the operation of the regulator (LDO) built into this MCU and the voltage supply source to the internal circuits except for the I/O ports, analog function.

LDOCUT bit (Regulator (LDO) stop)

Select the operation of the internal regulator (LDO) and the voltage supply source to the internal circuit. When this bit is set to 0, the LDO operates and the LDO output voltage is supplied to the internal circuit. When this bit is set to 1, the LDO stops and the externally applied voltage to the VCL and VCLH pins is supplied to the internal circuit.

When this bit is set to 1, the externally applied voltage to the VCL and VCLH pins must be within the specified range. When this bit is changed from 1 to 0, a voltage within the specified range must be applied to the VCL and VCLH pins until the LDO stabilization time (t_{LDO}) has elapsed. For details, see [section 51, Electrical Characteristics](#).

Transition to deep software standby mode with this bit set to 1 is prohibited.

13.2.24 VBCCR : Back Bias Voltage Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x4E0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VBBEN	IVDIS	—	—	—	—	—	CLKSEL
Value after reset:	*1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLKSEL ^{*2}	Back Bias Voltage Control (VBBC) Clock Select 0: LOCO clock 1: SOSC clock	R/W
5:1	—	These bits are read as 0. The write value should be 0.	R/W
6	IVDIS	Internal Voltage Discharge 0: The internal voltage is not discharged 1: The internal voltage is discharged	R/W
7	VBBEN	Back Bias Voltage Control (VBBC) Enable 0: Disable the VBBC circuit operation 1: Enable the VBBC circuit operation	R/W

Note: When writing to the VBCCR register, set the PRC.PRC1 bit to 1 before writing.

Note 1. The initial value of the VBBEN bit depends on the startup mode as follows:

- Normal startup: 0 (VBBC stops)
- Energy harvesting startup: 1 (VBBC operates)

Note 2. The same clock source as ICLK in low leakage current mode must be selected for the CLKSEL bit.

The VBCCR register enables or disables the back bias voltage control (VBBC), controls discharging of the internal voltage, and selects the operating clock.

When entering low leakage current mode (VBB), the back bias voltage control (VBBC) must be enabled by this register.

CLKSEL bit (Back Bias Voltage Control (VBBC) Clock Select)

The CLKSEL bit selects the LOCO or SOSC clock as the clock for the VBBC circuit.

Operation of LOCO or SOSC must be enabled to stabilize oscillation before selecting the clock for the VBBC circuit. The setting of this bit can also be changed while the VBBC circuit operation is enabled (VBBEN = 1). However, changing the ICLK and the clock for the VBBC circuit is prohibited while VBBC is set up (VBBEN = 1 and VBBST.VBBSTUP = 0) or after entry to low leakage current mode (VBB).

While the operation of the VBBC circuit is enabled (VBBEN = 1), stopping the clock selected by this bit is prohibited.

IVDIS bit (Internal Voltage Discharge)

The IVDIS bit is used to discharge the MCU's internal voltage.

Only set this bit when low leakage current mode (VBB) is to be entered via the following sequence of power control modes.

Applicable sequence of mode transitions: boost mode (BOOST) → normal mode (NORMAL) → low leakage current mode (VBB)

When the sequence of mode transitions stated above is made, be sure to discharge the internal voltage in normal mode before entry to low leakage current mode (VBB). Regarding the procedure for mode transitions when the internal voltage is discharged, see (5) [To switch from normal mode \(NORMAL\) to low leakage current mode \(VBB\)](#), [B] The mode transition is from boost mode (BOOST) → normal mode (NORMAL) → low leakage current mode (VBB).

The setting of this bit can be changed only when the system clock source is changed to the sub clock oscillator or LOCO in normal mode (NORMAL), and this bit can be changed only when all the oscillators except the sub clock oscillator and LOCO are stopped. Do not change in other mode.

The setting of this bit must be 0 in case of the clock setting is other than the above with boost mode (BOOST), low leakage current mode (VBB), and normal mode (NORMAL).

VBEN bit (Back Bias Voltage Control (VBBC) Enable)

The VBEN bit enables or disables operation of the VBBC circuit.

Setting this bit to 1 on normal startup allows operation of the VBBC circuit to start. Note that the VBBC circuit requires the initial setup time, so when entering low leakage current mode (VBB), we recommend setting the VBEN bit to 1 in advance after a reset release. For the setting procedure, see [section 13.5.2. Setting Power Control Mode \(BOOST/NORMAL/VBB\)](#), For the initial setup time for the VBBC circuit, see [section 51, Electrical Characteristics](#).

Since the VBBC circuit starts in an operating state on energy harvesting startup mode, this bit does not need to be set to 1. Clearing this bit to 0 allows operation of the VBBC circuit to stop.

Setting this bit to 1 is only allowed when operation of the clock selected by the CLKSEL bit is enabled (LOCOCR.LCSTP = 0 when CLKSEL = 0, or SOSCCR.SOSTP = 0 when CLKSEL = 1).

Clearing this bit to 0 is allowed except in low leakage current mode (VBB) (PWSTF.VBBM = 0). Additionally, this bit is cleared (0) automatically after transition to deep software standby mode.

13.2.25 VBBST : Back Bias Voltage Status Register

Base address: SYSC = 0x4001_E000

Offset address: 0x4E1

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	VBBSTUP
Value after reset:	0	0	0	0	0	0	0	0 ^{*1}

Bit	Symbol	Function	R/W
0	VBBSTUP	Back Bias Voltage Control (VBBC) Initial Setup Complete Flag 0: During the VBBC circuit stop or VBBC setup 1: VBBC initial setup completed	R
7:1	—	These bits are read as 0.	R

Note 1. The initial value of the VBBSTUP flag is 0. The state of the back bias voltage control (VBBC) depends on the startup mode as follows:

Normal startup: 0 (The VBBC circuit stops)

Energy harvesting startup mode: 0 during VBBC setup; 1 on completion of VBBC setup
In startup for energy harvesting mode, the MCU is started while VBEN = 1. The setting of the VBBST bit is changed from 0 to 1 on completion of setup. The value read may not be 0, which is the initial value, but 1 instead.

The VBBST register is used to confirm the completion of initial setup for the back bias voltage control (VBBC).

VBBSTUP flag (Back Bias Voltage Control (VBBC) Initial Setup Complete Flag)

The VBBSTUP bit is a status flag that indicates the completion of the VBBC initial setup.

The VBBSTUP bit is set to 0 during the VBBC circuit stop (VBBCR.VBBEN = 0) or VBBC setup, and is set to 1 on completion of the VBBC setup.

When this bit is 1, transition to Software standby in low leakage current mode (VBB) is possible.

13.3 Reducing Power Consumption by Switching Clock Signals

When the SCKDIVCR register is set, the clock frequency changes.

About the module and clock associations, see [section 9.2.1. SCKDIVCR : System Clock Division Control Register](#).

13.4 Module-Stop Function

The module stop function can stop the clock supply set for each peripheral module.

When the MSTPmi bit (m = A to D, i = 31 to 0) in MSTPCRn (n = A to D) is set to 1, the specified module stops operating and enters the module-stop state, but the CPU continues to operate independently. Setting the MSTPmi bit to 0 cancels the module-stop state, allowing the module to resume operation at the end of the bus cycle.

After a reset is canceled, all modules other than the DMAC and DTC is placed in the module-stop state. Do not access the module while the corresponding MSTPmi bit is 1.

Additionally, do not set 1 to the MSTPmi bit while the corresponding module is accessed.

13.5 Functions for Reducing Power

Power consumption can be reduced in normal operating, sleep, snooze, and software standby modes by selecting an appropriate power supply mode and power control mode for the given operating frequency and operating voltage.

13.5.1 Setting Power Supply Mode (ALLPWON/EXFPWON/MINPWON)

In normal mode (NORMAL)/low leakage current mode (VBB), the power supply mode can be changed. Boost mode (BOOST) can only be entered in all power supply mode (ALLPWON), so transition to exclude-flash power supply mode (EXFPWON) or minimum power supply mode (MINPWON) is not possible in boost mode.

In normal mode (NORMAL)/low leakage current mode (VBB), transition to the specified power supply mode is achieved by setting the given power supply mode in the PWSTCR.PWST[2:0] bits and executing a WFE instruction. Because switching the power supply mode must be performed by transition mode equivalent to Software Standby mode, the followings must be set first, and then, a WFE instruction must be executed.

- SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0 (Software Standby mode set)
- SNZCR.SNZE = 0 (Snooze mode disabled)

If an interrupt occurs during mode transition, the interrupt processing is performed according to the setting of the interrupt-related control register after completion of transition to the destination power supply mode. When there is no interrupt, the processing before the mode transition continues.

Note: When changing the power supply mode, be sure to use the following procedure.

(1) To switch from all power supply mode (ALLPWON) to exclude-flash power supply mode (EXFPWON)

1. Move programs to be executed in exclude-flash power supply mode (EXFPWON) from the code flash memory to SRAM.
2. If there is access to the flash memory by the DMAC or DTC, change the transfer settings so that the flash memory is not accessed.
3. Set SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0 (enabling Software Standby mode).^{*1}
4. Set SNZCR.SNZE = 0 (disabling Snooze mode).^{*1}
5. Set PWSTCR.PWST[2:0] = 010b (exclude flash power supply mode (EXFPWON)).
6. Confirm that the PWSTCR.PWST[2:0] bits are set to 010b.
7. Execute a WFE instruction.
8. Check the PWSTF.EXFPWONM flag.
If PWSTF.EXFPWONM = 1, transition to exclude-flash power supply mode (EXFPWON) is complete. If PWSTF.EXFPWONM = 0, return to step 7.

Note: When the MCU operates in boost mode (BOOST) in all power supply mode (ALLPWON), transition to exclude-flash power supply mode (EXFPWON) is not possible.

Note 1. If the settings in steps 3 and 4 are confirmed, there is no need to set them again.

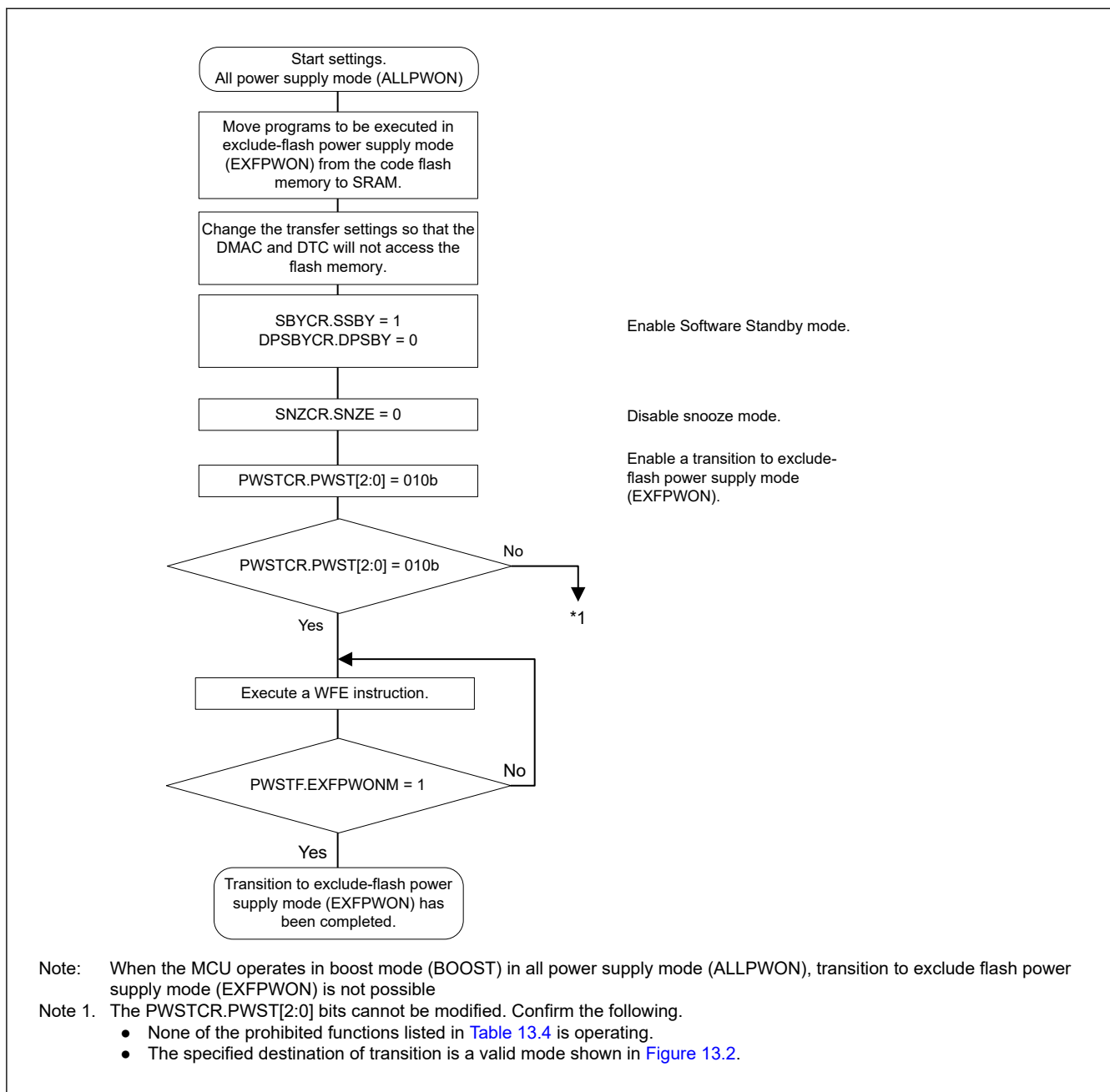


Figure 13.4 Procedure for Switching from All Power Supply Mode (ALLPWON) to Exclude-flash Power Supply Mode (EXFPWON)

(2) To switch from exclude-flash power supply mode (EXFPWON) to all power supply mode (ALLPWON)

1. Set SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0 (enabling software standby mode).^{*1}
2. Set SNZCR.SNZE = 0 (disabling snooze mode).^{*1}
3. Set PWSTCR.PWST[2:0] = 001b (all power supply mode (ALLPWON)).
4. Confirm that the PWSTCR.PWST[2:0] bits are set to 001b.
5. Set ICLK ≤ 4MHz.

6. Execute a WFE instruction.
7. Check the PWSTF.ALLPWONMflag.
If PWSTF.ALLPWONM = 1, the transition to all power supply mode (ALLPWON) has been completed. If PWSTF.ALLPWONM = 0, return to step 6.

Note 1. If the settings in steps 1 and 2 are confirmed, there is no need to set them again.

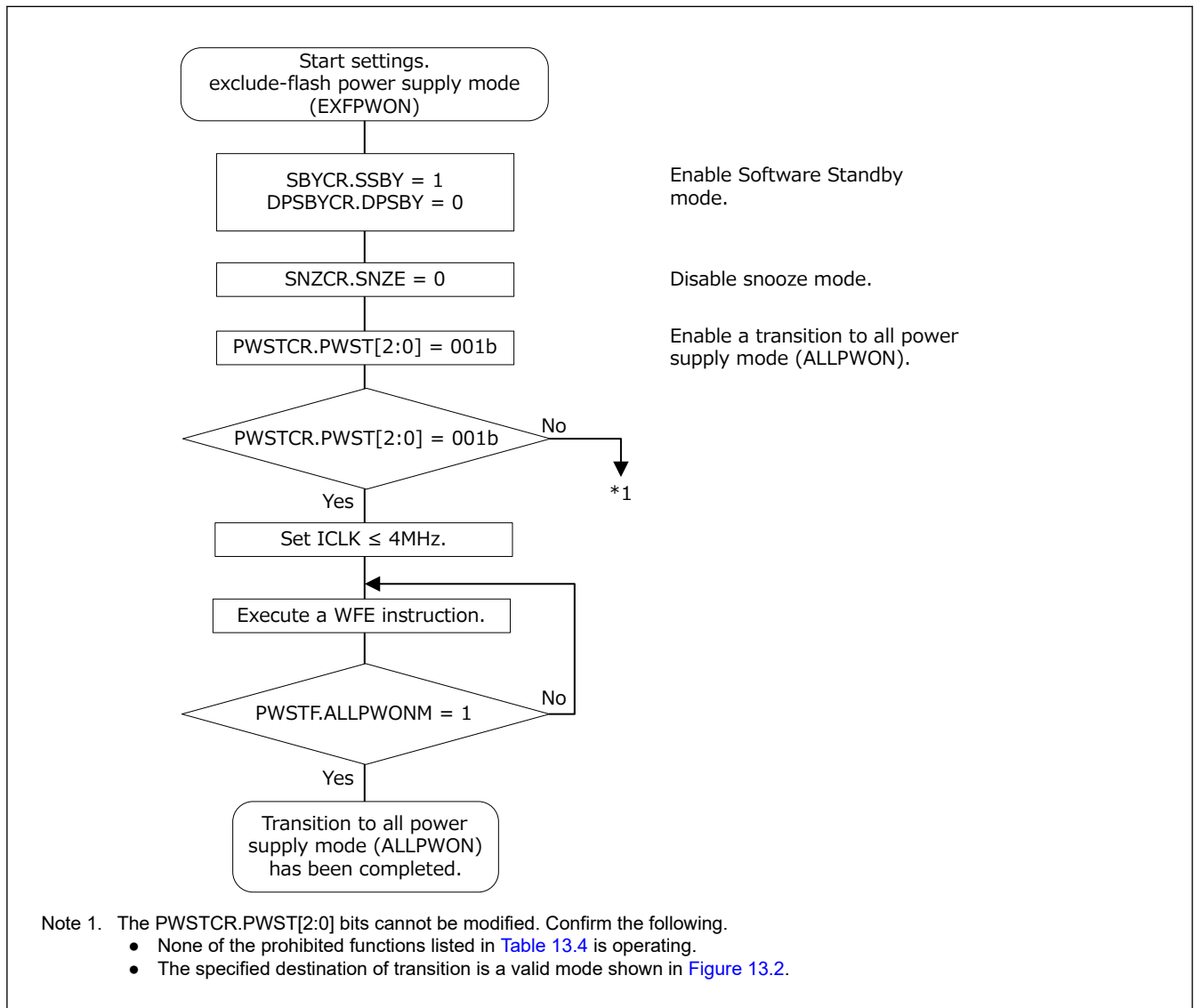


Figure 13.5 Procedure for Switching from Exclude-flash Power Supply Mode (EXFPWON) to All Power Supply Mode (ALLPWON)

(3) To switch from exclude-flash power supply mode (EXFPWON) to minimum power supply mode (MINPWON)

1. If there is DMAC or DTC access to the functions to which power supply is to be cut off in minimum power supply mode (MINPWON), change the transfer settings so that they are not accessed.
2. If there is event link access to the functions to which power supply is to be cut off in minimum power supply mode (MINPWON), change the event link settings.
3. Set SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0 (enabling Software Standby mode).^{*1}
4. Set SNZCR.SNZE = 0 (disabling Snooze mode).^{*1}
5. Set PWSTCR.PWST[2:0] = 011b (minimum power supply mode (MINPWON)).
6. Confirm that the PWSTCR.PWST[2:0] bits are set to 011b.

7. Execute a WFE instruction.
8. Check the PWSTF.MINPWONM flag.
If PWSTF.MINPWONM = 1, the transition from power supply mode to the minimum power supply mode (MINPWON) is complete. If PWSTF.MINPWONM = 0, return to step 7.

Note 1. If the settings in steps 3 and 4 are confirmed, there is no need to set them again.

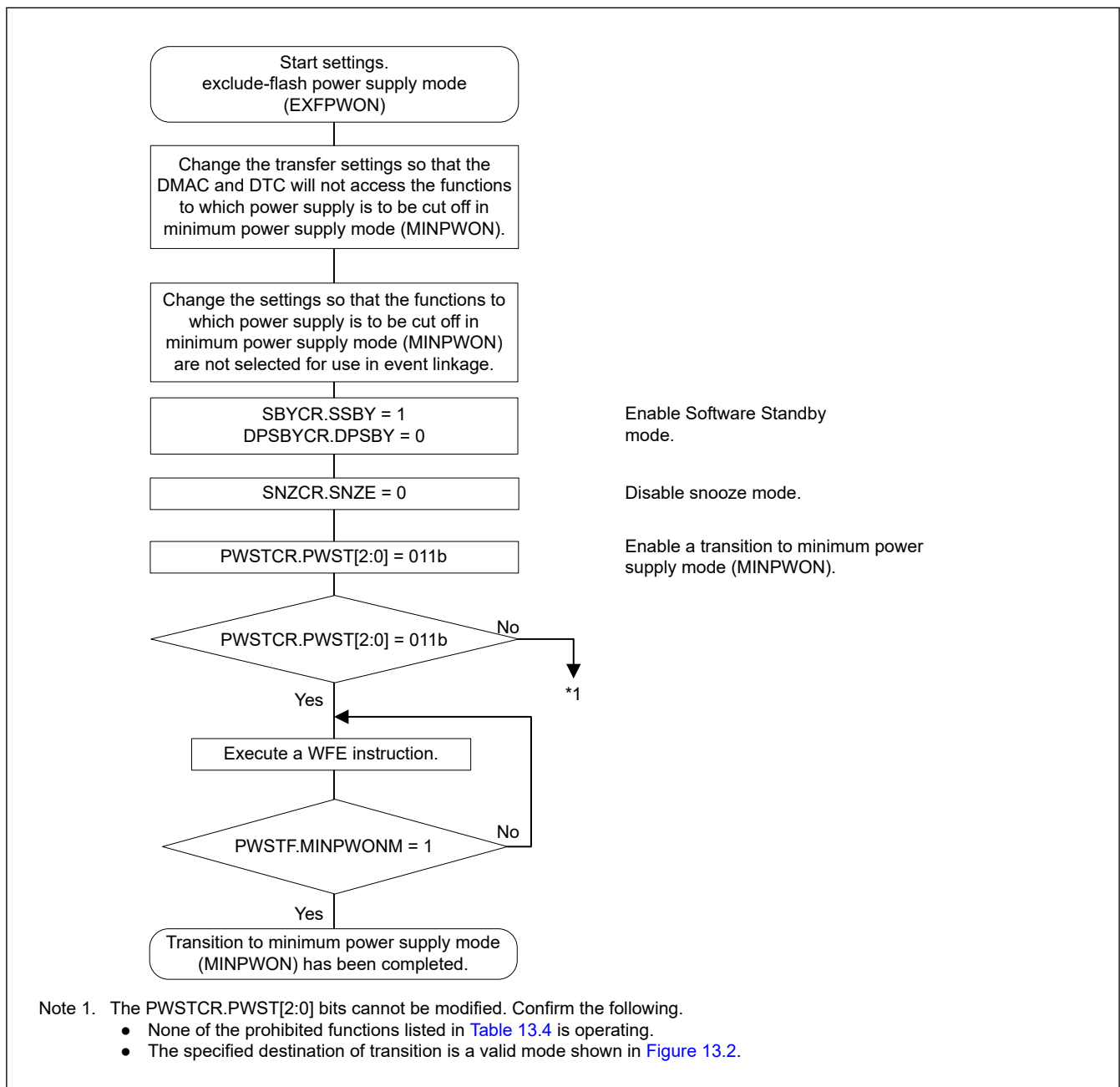


Figure 13.6 Procedure for Switching from Exclude-flash Power Supply Mode (EXFPWON) to Minimum Power Supply Mode (MINPWON)

(4) To switch from minimum power supply mode (MINPWON) to exclude-flash power supply mode (EXFPWON)

1. Set SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0 (enabling software standby mode).^{*2}
2. Set SNZCR.SNZE = 0 (disabling snooze mode).^{*2}
3. Set PWSTCR.PWST[2:0] = 010b (exclude-flash power supply mode (EXFPWON)).

4. Confirm that the PWSTCR.PWST[2:0] bits are set to 010b.
5. Execute a WFE instruction.
6. Check the PWSTF.EXFPWONM flag.
 If PWSTF.EXFPWONM = 1, the transition to exclude-flash power supply mode (EXFPWON) has been completed.^{*1*3}
 If PWSTF.EXFPWONM = 0, return to step 5.

Note: The timing of release from a reset differs between the CPU and the functions under the ISO2 domain. After mode transitions completed, repeat writing to and reading from a register in a function under the ISO2 domain until the written value is correctly read from the register if the frequency division ratio for PCLKB is set to a value from 4 to 64. A/D Emulator Debug Function Dummy Data Setting Register 0 (ADEDDMY0) is recommended as a safe confirmation register. Read/Write is possible even when the 14-bit A/D converter is in the module stop state

Note 1. After the transition to exclude-flash power supply mode (EXFPWON), the function under the ISO2 domain to which power supply was cut off in minimum power supply mode (MINPWON) needs to be set again because it was initialized.

Note 2. If the settings in steps 1 and 2 are confirmed, there is no need to set them again.

Note 3. If the frequency division ratio of ICLK and PCLKB is 1:4 or more, the peripheral functions in the ISO2 domain may be continuously reset by MINPWON even after the transition to EXFPWON mode is completed and the CPU starts operating. When resetting peripheral functions under the ISO2 domain, repeat the register write until the value written in the first register write cycle can be read, and confirm that the MINPWON reset has been released.

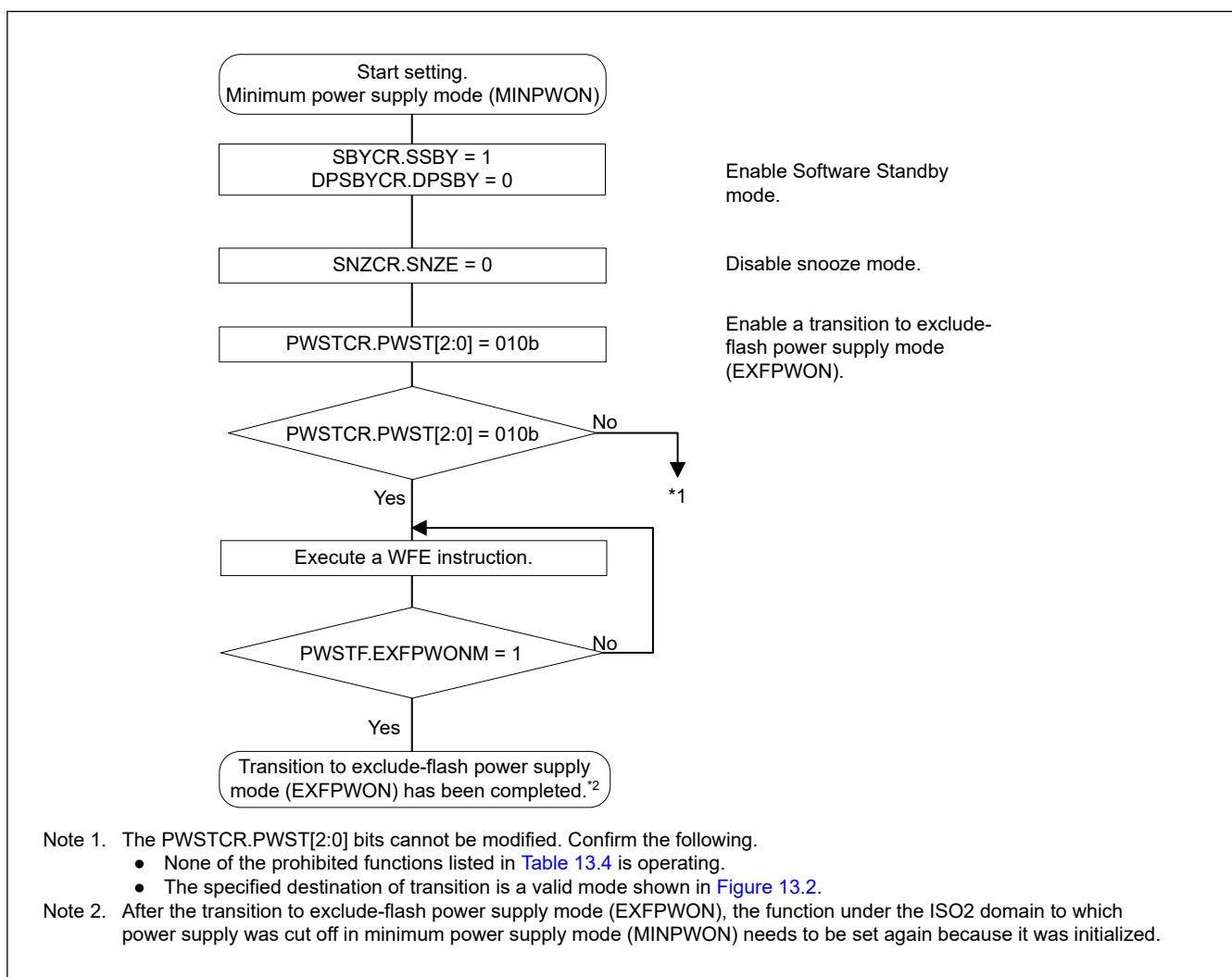


Figure 13.7 Procedure for Switching from Minimum Power Supply Mode (MINPWON) to Exclude-flash Power Supply Mode (EXFPWON)

(5) To Switch from all power supply mode (ALLPWON) to minimum power supply mode (MINPWON)

1. Store the program to be executed in the minimum power supply mode (MINPWON) from flash memory to SRAM.
2. Function to cut off power supply in the minimum power supply mode (MINPWON) is being accessed by DMAC / DTC, change the transfer settings to prevent access.
3. Change the event link setting when using the function to cut off power supply in the minimum power supply mode (MINPWON) with the event link.
4. Set SBYCR.SSBY = 1, DPSBYCR.DPSBY = 0 (Software standby mode setting).^{*1}
5. Set SNZCR.SNZE = 0 (Prohibit snooze mode).^{*1}
6. Set PWSTCR.PWST[2:0] = 011b (Minimum power supply mode (MINPWON)).
7. Confirm the PWSTCR.PWST[2:0] bits is set to 011b.
8. Execute WFE instruction.
9. Confirm the PWSTF.MINPWONM flag. The transition from the power supply mode to the minimum power supply mode (MINPWON) is complete. When PWSTF.MINPWONM = 0, return to step 8.

Note 1. If the settings in steps 4 and 5 are confirmed, there is no need to set them again.

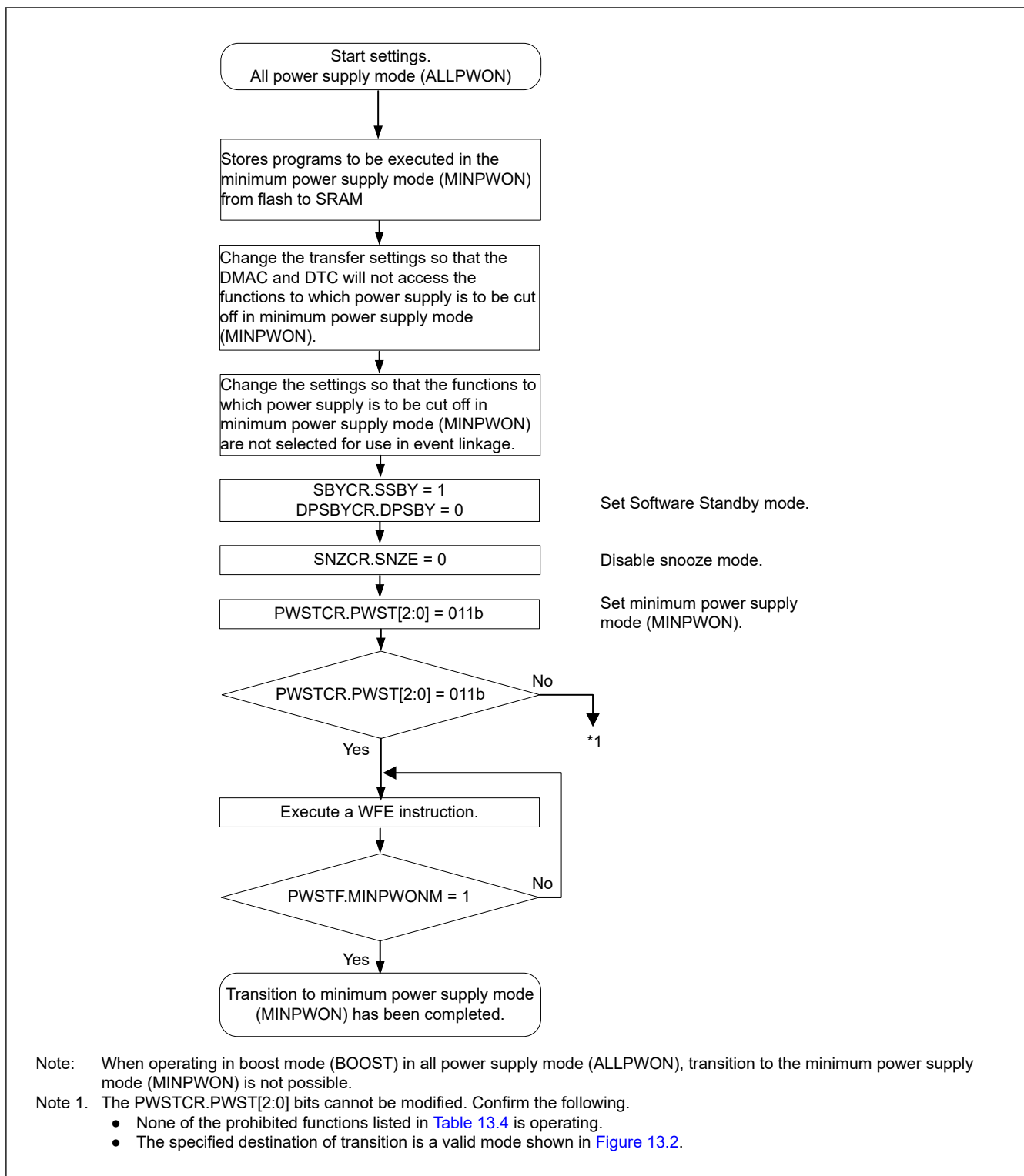


Figure 13.8 Switching from all power supply mode (ALLPWON) to minimum power supply mode (MINPWON)

(6) To Switch from minimum power supply mode (MINPWON) to all power supply mode (ALLPWON)

1. Set SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0 (Software standby mode setting).^{*1}
2. Set SNZCR.SNZE = 0 (Prohibit snooze mode).^{*1}
3. Set PWSTCR.PWST[2:0] = 001b (all power supply mode (ALLPWON)).
4. Confirm that PWSTCR.PWST[2:0] bits are set to 001b.

5. Execute WFE instruction.
6. Confirm the PWSTF.ALLPWONM flag. When PWSTF.ALLPWONM = 1, the power supply mode transitions to all power supply mode (ALLPWON). When PWSTF.ALLPWONM = 0, return to step 5.

Note: The reset release timing differs between the CPU and the function under the ISO2 domain. If the PCLKB frequency division setting is 4 to 64, after the mode transition is completed, continue writing and reading to the register until the first register write of the function under the ISO2 domain can be read correctly. A/D Emulator Debug Function Dummy Data Setting Register 0 (ADEDDMY0) is recommended as a safe confirmation register. Reading and writing are possible even when the 14-bit A/D converter is in the module stop state.

Note 1. If the settings in steps 1 and 2 are confirmed, there is no need to set them again.

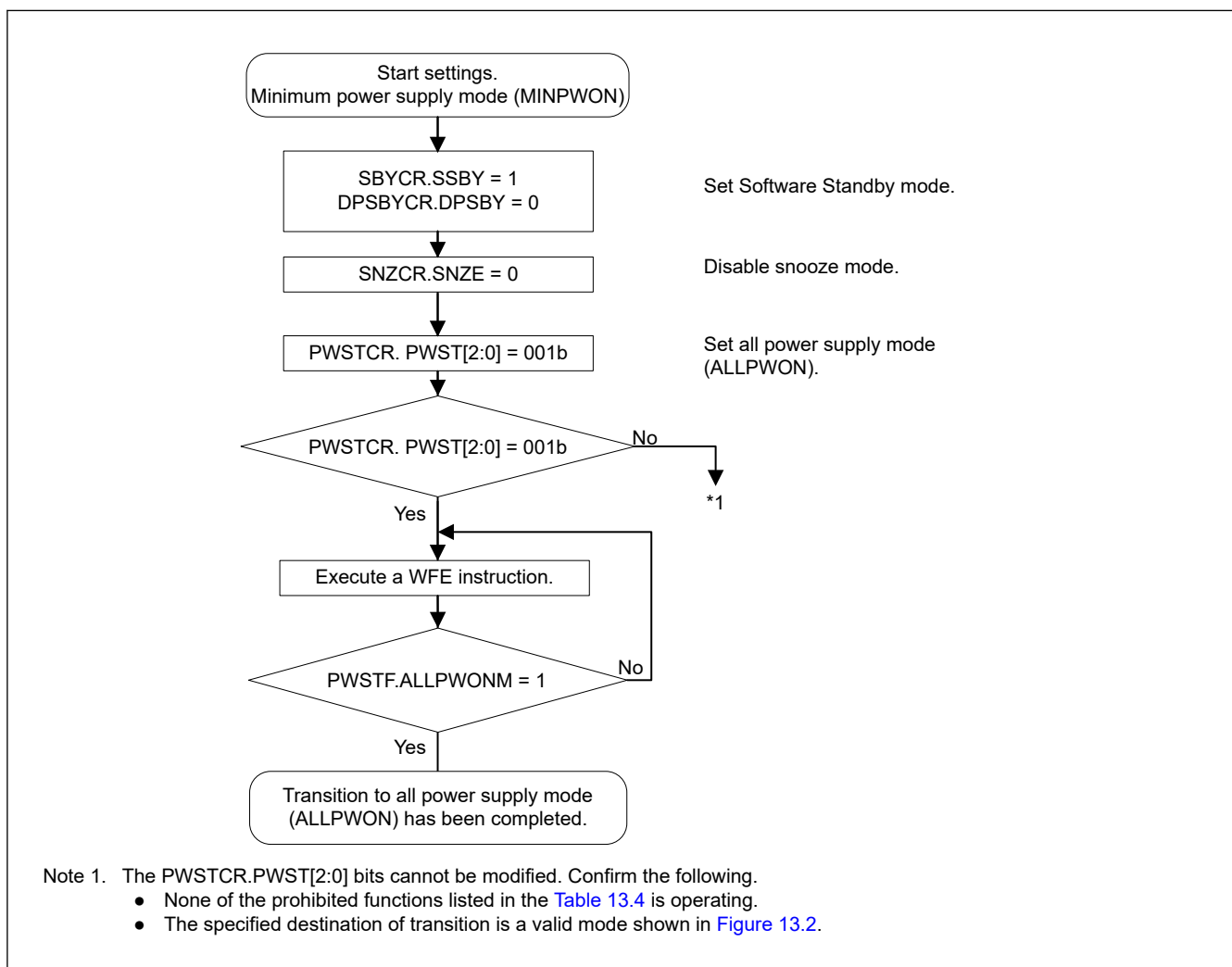


Figure 13.9 To switch from minimum power supply mode (MINPWON) to full power supply mode (ALLPWON)

13.5.2 Setting Power Control Mode (BOOST/NORMAL/VBB)

Make sure that the operating condition such as the frequency is always within the specified range before and after switching boost (BOOST)/normal (NORMAL)/low leakage current (VBB) modes, and high-speed/low-speed modes in normal (NORMAL) mode.

To switch between boost (BOOST), normal (NORMAL), and low leakage current (VBB) modes, specify a desired power control mode in the PWSTCR.PWST[2:0] bits and execute a WFE instruction.

Because switching the power control mode must be performed via transition mode equivalent to software standby mode, the followings must be set first, and then, a WFE instruction must be executed.

- SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0 (software standby mode set)

- SNZCR.SNZE = 0 (snooze mode disabled)

If an interrupt occurs during mode transition, the interrupt processing is performed according to the setting of the interrupt-related control register after completion of transition to the destination power control mode. When there is no interrupt, the processing before the mode transition continues.

To switch between high-speed, low-speed modes in normal (NORMAL) mode, specify a desired mode in the OPCCR.OPCM[1:0] bits. Execution of a WFE instruction or setting via transition mode is not necessary.

Note: When changing the power control mode, be sure to follow the following procedure.

(1) To switch from normal mode (NORMAL) to boost mode(BOOST)

1. Enter all power supply mode (ALLPWON).
2. Set SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0 (enabling SSTBY).^{*1}
3. Set SNZCR.SNZE = 0 (disabling snooze mode).^{*1}
4. Set PWSTCR.PWST[2:0] = 101b (boost mode (BOOST)).
5. Confirm that the PWSTCR.PWST[2:0] bits are set to 101b.
6. Execute a WFE instruction.
7. Check the PWSTF.BOOSTM flag.
If PWSTF.BOOSTM = 1, the transition to boost mode (BOOST) has been completed.
If PWSTF.BOOSTM = 0, return to step 6.
8. Selecting 48 MHz/64 MHz for the HOCO are enabled after transition to boost mode (BOOST).

Note 1. If the settings in steps 2 and 3 are confirmed, there is no need to set them again.

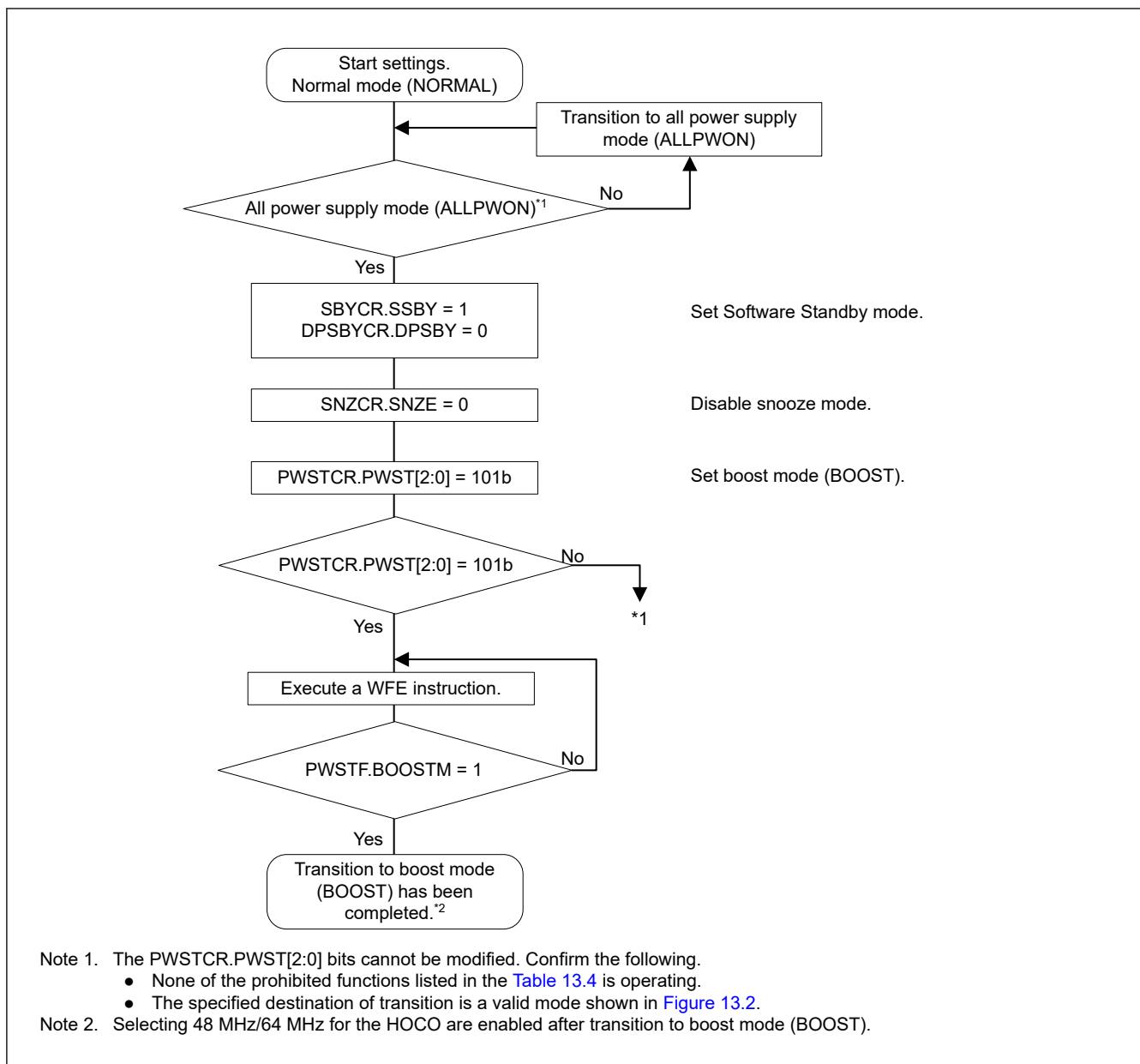


Figure 13.10 Procedure for Switching from Normal Mode (NORMAL) to Boost Mode (BOOST)

(2) To switch from boost mode (BOOST) to normal mode (NORMAL)

1. Set the HOCO frequency to 24 MHz or 32 MHz (HOCOMCR.HCFRQ[1:0] = 00b or 01b).
Set the system clock to less than the maximum operating frequency in High-Speed mode.
2. Set OPCCR.OPCM[1:0] = 00b (High-Speed mode).^{*1}
3. Set SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0 (enabling SSTBY).^{*2}
4. Set SNZCR.SNZE = 0 (disabling snooze mode).^{*2}
5. Set PWSTCR.PWST[2:0] = 100b (normal mode (NORMAL)).
6. Confirm that the PWSTCR.PWST[2:0] bits are set to 100b.
7. Execute a WFE instruction.
8. Check the PWSTF.BOOSTM and PWSTF.VBBM flags.
If PWSTF.BOOSTM and PWSTF.VBBM are both 0, the transition to normal mode (NORMAL) has been completed.
If PWSTF.BOOSTM and PWSTF.VBBM are not both 0, return to step 7.

Note 1. No transition to normal mode except for High-Speed mode.

Note 2. If the settings in steps 3 and 4 are confirmed, there is no need to set them again.

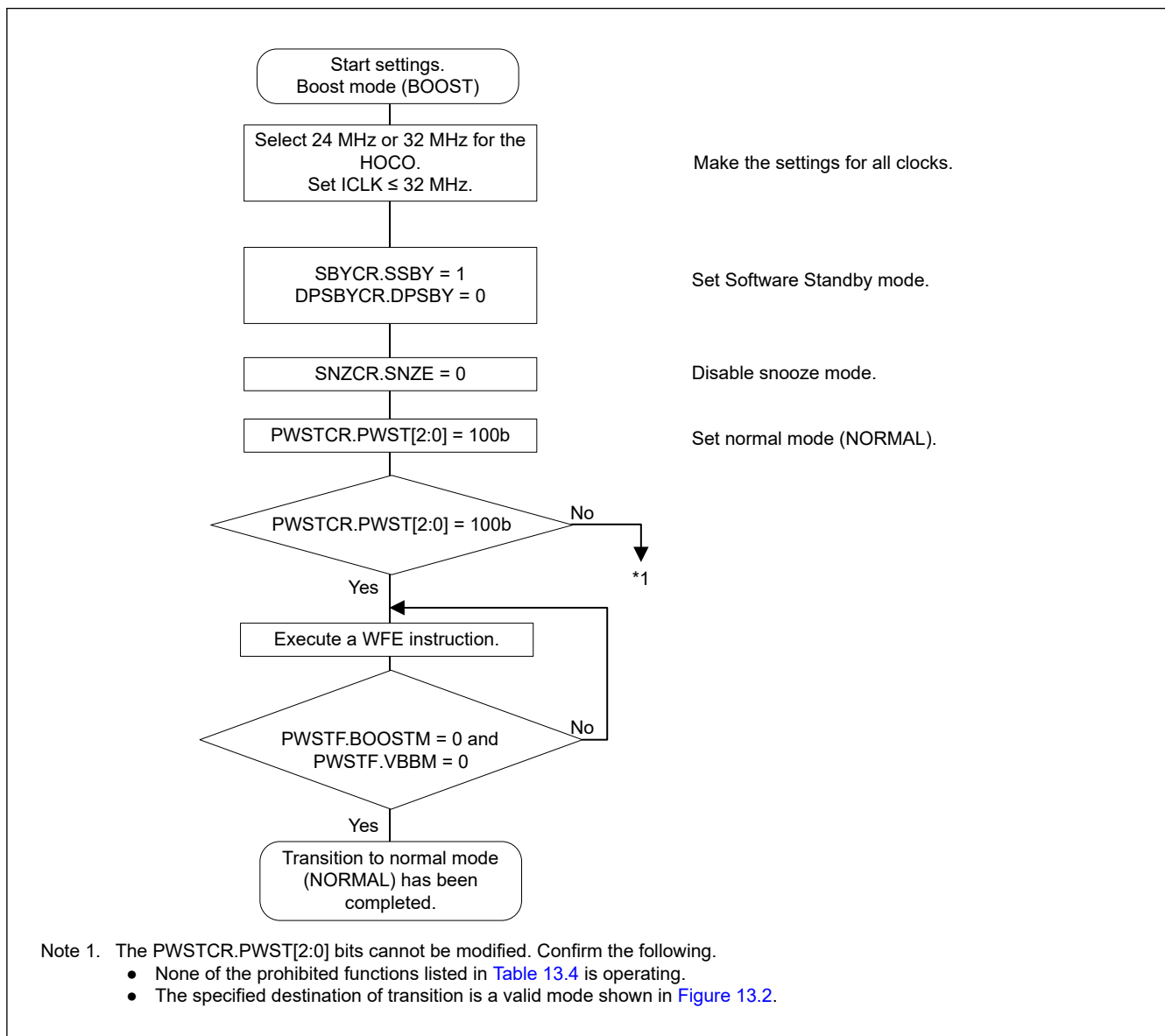


Figure 13.11 Procedure for Switching from Boost Mode (BOOST) to Normal Mode (NORMAL)

(3) To Switch from low leakage current mode (VBB) to boost mode (BOOST)

1. Transition to all power supply mode (VBB ALLPWON).^{*1}
2. Set SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0 (SSTBY setting).^{*2}
3. Set SNZCR.SNZE = 0 (prohibit snooze mode).^{*2}
4. Set PWSTCR.PWST[2:0] = 101b (Boost mode (BOOST)).
5. Confirm that 101b is set in the PWSTCR.PWST[2:0] bits.
6. Execute WFE instruction.
7. Check the PWSTF.BOOSTM flag.
When PWSTF.BOOSTM = 1, transition to boost mode (BOOST) is complete.
If PWSTF.BOOSTM = 0, return to step 6.
8. HOCO (48MHz/64MHz) can be selected after switching to boost mode (BOOST).

Note 1. The transition from power supply mode other than Flash (VBB EXFPWON) or minimum power supply mode (VBB MINPWON) to boost mode (BOOST) is not possible.

Note 2. If the settings in steps 2 and 3 are confirmed, there is no need to set them again.

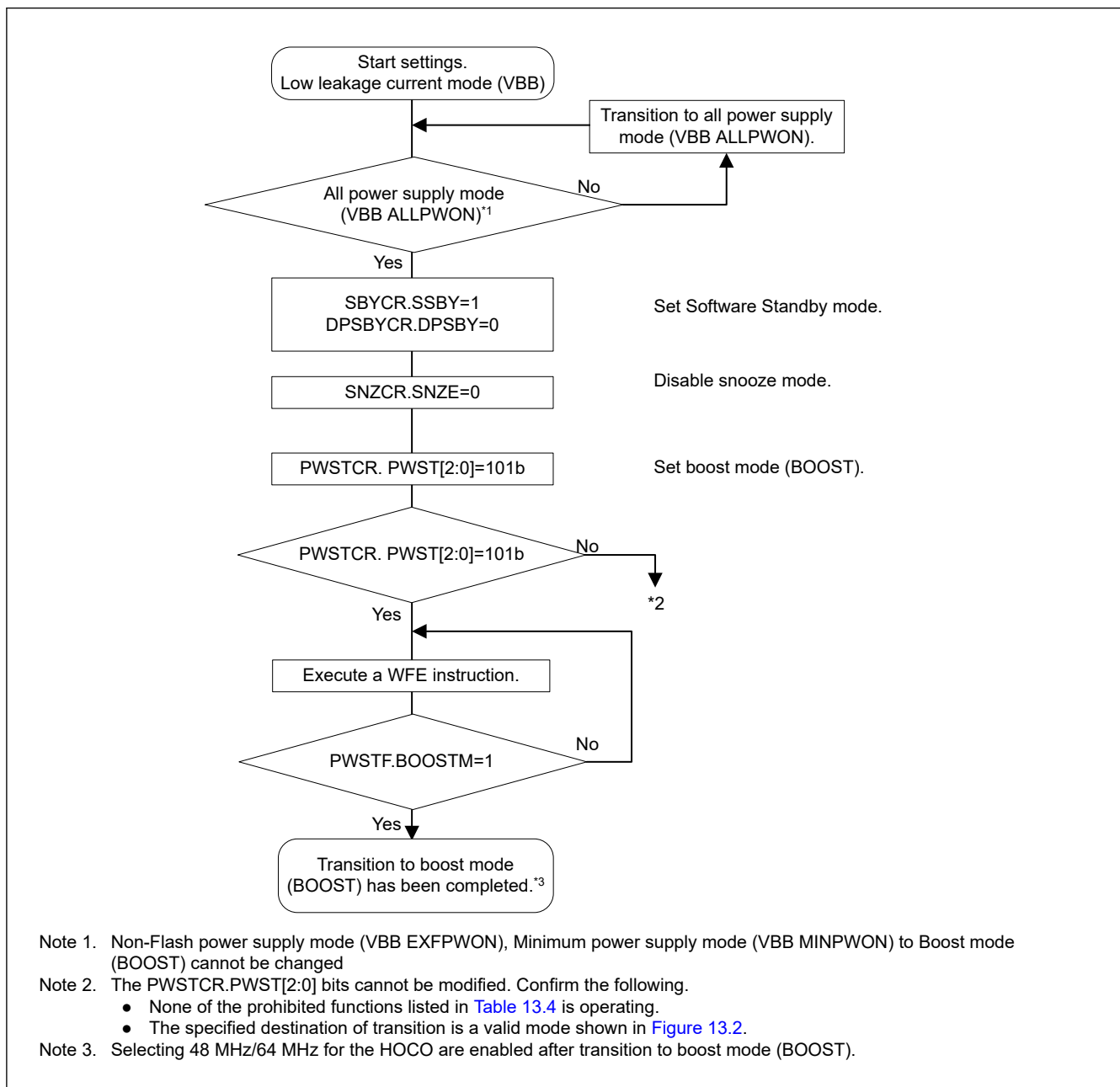


Figure 13.12 The switching procedure from low leakage current mode (VBB) to boost mode (BOOST)

(4) When switching from boost mode (BOOST) to low leakage current mode (VBB)

1. VBBCR.VBBEN = 1 set (VBBC circuit operation enabled).^{*1}
2. Confirm that VBBST.VBBSTUP = 1 (VBBC setup complete).^{*1}
3. Set SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0 (SSTBY setting).^{*2}
4. Set SNZCR.SNZE = 0 (prohibit snooze mode).^{*2}
5. Change system clock to SOSC clock or LOCO clock.
6. Stop oscillator other than system clock source.
7. Set PWSTCR. PWST[2:0] = 110b (low leakage current mode (VBB)).
8. Confirm that 110b is set in the PWSTCR. PWST[2:0] bits.

9. Execute WFE instruction.

10. Confirm PWSTF.BOOSTM flag and PWSTF.VBBM flag.

When PWSTF.BOOSTM = 0 and PWSTF.VBBM = 1, transition to low leakage current mode (VBB) is completed.

When PWSTF.BOOSTM = 0 and PWSTF.VBBM = 1, return to step 9.

Note 1. When the VBBC circuit has started operation, it is not necessary to set VBBCR.VBBEN = 1.

Note 2. If the settings in steps 3 and 4 are confirmed, there is no need to set them again.

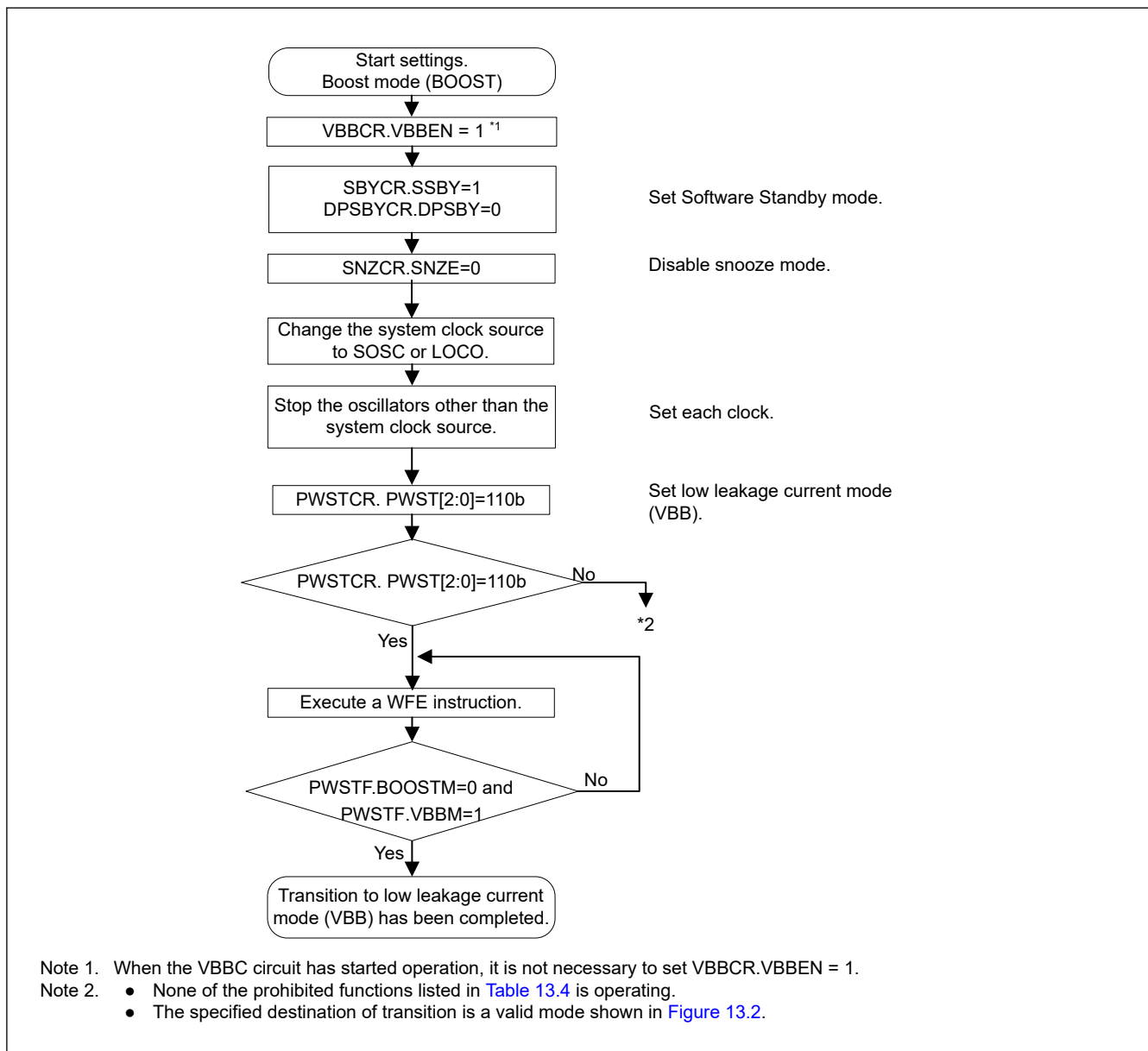


Figure 13.13 Switching procedure from boost mode (BOOST) to low leakage current mode (VBB)

(5) To switch from normal mode (NORMAL) to low leakage current mode (VBB)

Whether procedure [A] or [B] below and on the following pages is applicable depends on whether the MCU is in boost mode (BOOST) before entry to normal mode (NORMAL).

[A] The mode transition is not from boost mode (BOOST) → normal mode (NORMAL) → low leakage current mode (VBB).

1. Set VBBCR.VBBEN = 1 (enabling the VBBC circuit operation).*1
2. Confirm that VBBST.VBBSTUP is set to 1 (VBBC initial setup completed).*1

3. Set SBYCR.SSBY = 1, DPSBYCR.DPSBY = 0 (enabling SSTBY).^{*2}
4. Set SNZCR.SNZE = 0 (disabling snooze mode).^{*2}
5. Change system clock to SOSC clock or LOCO clock.
6. Stop oscillator other than system clock source.
7. Set PWSTCR.PWST[2:0] = 110b (low leakage current mode(VBB)).
8. Confirm that the PWSTCR.PWST[2:0] bits are set to 110b.
9. Execute a WFE instruction.
10. Check the PWSTF.VBBM flag.
 - If PWSTF.VBBM = 1, the transition to low leakage current mode (VBB) has been completed.
 - If PWSTF.VBBM = 0, return to step 9.

Note 1. Setting VBBCR.VBBEN = 1 is not necessary at energy harvesting startup mode because the VBBC circuit starts operation during the reset period.

Note 2. If the settings in steps 3 and 4 are confirmed, there is no need to set them again

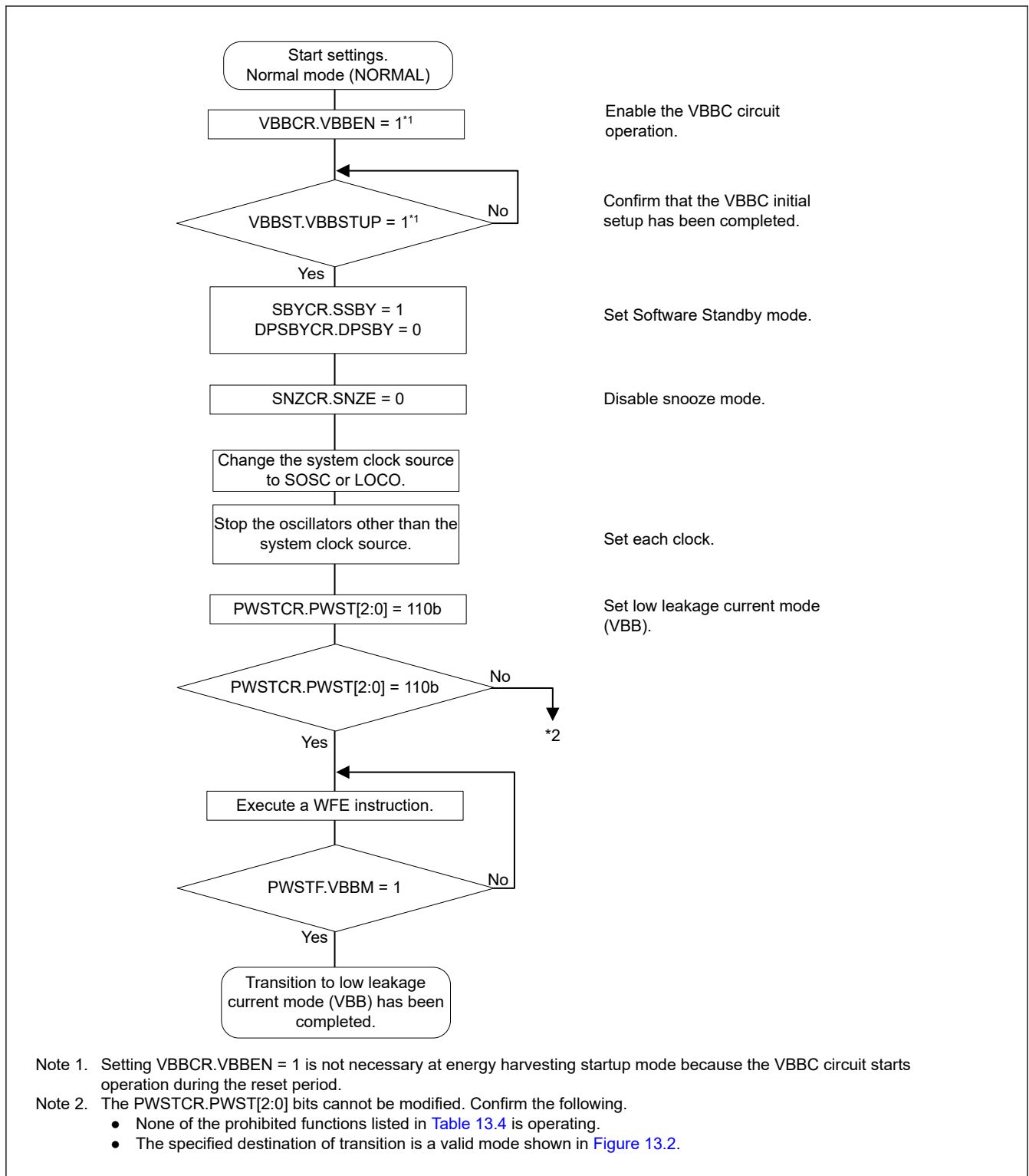


Figure 13.14 Procedure[A] for Switching from Normal Mode (NORMAL) to Low Leakage Current Mode (VBB)

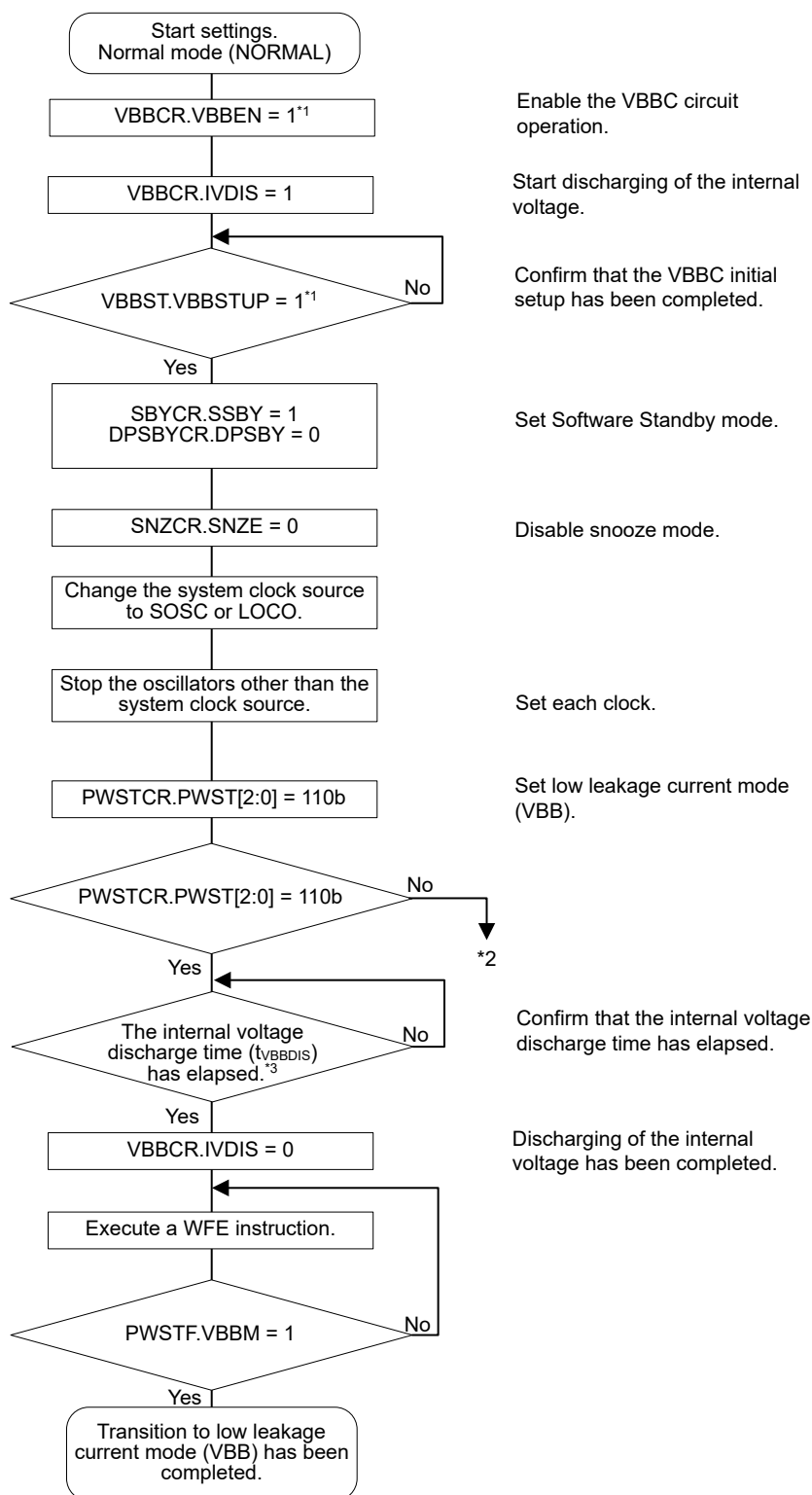
[B] The mode transition is from boost mode (BOOST) → normal mode (NORMAL) → low leakage current mode (VBB)

1. Set VBBCR.VBBEN = 1 (enabling VBBC operation)*1.
2. Set VBBCR.IVDIS = 1 (to start discharging of the internal voltage)*2.
3. Confirm that VBBST.VBBSTUP is set to 1 (VBBC initial setup completed)*1.
4. Set SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0 (enabling SSTBY).
5. Set SNZCR.SNZE = 0 (disabling snooze mode).

6. Change system clock to SOSC clock or LOCO clock.
7. Stop oscillator other than system clock source.
8. Set PWSTCR.PWST[2:0] = 110b (low leakage current mode(VBB)).
9. Confirm that the PWSTCR.PWST[2:0] bits are set to 110b.
10. Confirm that the internal voltage discharge time (t_{VBBDIS}) has elapsed^{*2}.
11. Set VBBCR.IVDIS = 0 (discharging of the internal voltage has been completed)^{*2}.
12. Execute a WFE instruction.
13. Check the PWSTF.VBBM flag.
 - If PWSTF.VBBM = 1, the transition to low leakage current mode (VBB) has been completed.
 - If PWSTF.VBBM = 0, return to step 12.

Note 1. Setting VBBCR.VBBEN = 1 is not necessary at energy harvesting startup mode because the VBBC circuit starts operation during the reset period.

Note 2. For the internal voltage discharge time (t_{VBBDIS}), see [section 51, Electrical Characteristics](#).



Note 1. Setting VBBCR.VBBEN = 1 is not necessary at energy harvesting startup because the VBBC circuit starts operation during the reset period.

Note 2. The PWSTCR.PWST[2:0] bits cannot be modified. Confirm the following.

- None of the prohibited functions listed in [Table 13.4](#) is operating.
- The specified destination of transition is a valid mode shown in [Figure 13.2](#).

Note 3. For the internal voltage discharge time (t_{VBBDIS}), see [section 51, Electrical Characteristics](#).

Figure 13.15 Procedure[B] for Switching from Normal Mode (NORMAL) to Low Leakage Current Mode (VBB)

(6) To switch from low leakage current mode (VBB) to normal mode (NORMAL)

1. Set SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0 (enabling SSTBY).^{*1}
2. Set SNZCR.SNZE = 0 (disabling snooze mode).^{*1}
3. Set PWSTCR.PWST[2:0] = 100b (normal mode (NORMAL)).
4. Confirm that the PWSTCR.PWST[2:0] bits are set to 100b.
5. Execute a WFE instruction.
6. Check the PWSTF.BOOSTM and PWSTF.VBBM flags.
If PWSTF.BOOSTM and PWSTF.VBBM are both 0, the transition to normal mode (NORMAL) has been completed.
If PWSTF.BOOSTM and PWSTF.VBBM are not both 0, return to step 5.

Note 1. If the settings in steps 1 and 2 are confirmed, there is no need to set them again.

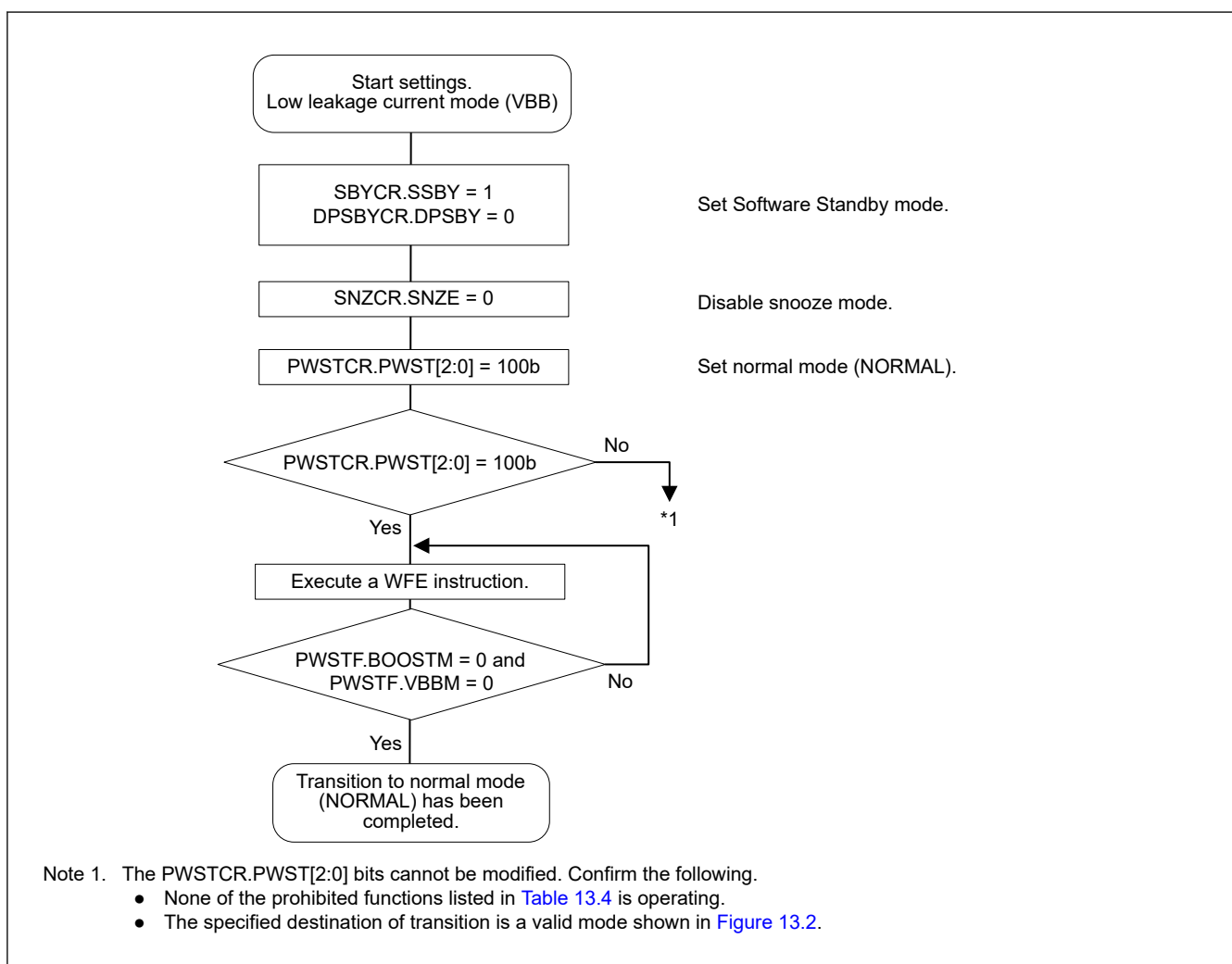


Figure 13.16 Procedure for Switching from Low Leakage Current Mode (VBB) to Normal Mode (NORMAL)

13.5.2.1 Setting High-Speed/Low-Speed Mode in Normal Mode (NORMAL)

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

To switch from high-speed mode to low-speed mode:

1. Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed).
2. Change the oscillator to use in Low-Speed mode. Set the frequency of each clock to be lower than the maximum operating frequency in Low-Speed mode.

3. Stop unnecessary oscillators in Low-Speed mode.
4. Set PWSTCR.PWST[2:0] = 000b.
5. Confirm that the PWSTCR.PWST[2:0] bits are set to 000b.
6. Set the OPCCR.OPCM[1:0] bits to 11b (low-speed mode).
7. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed)

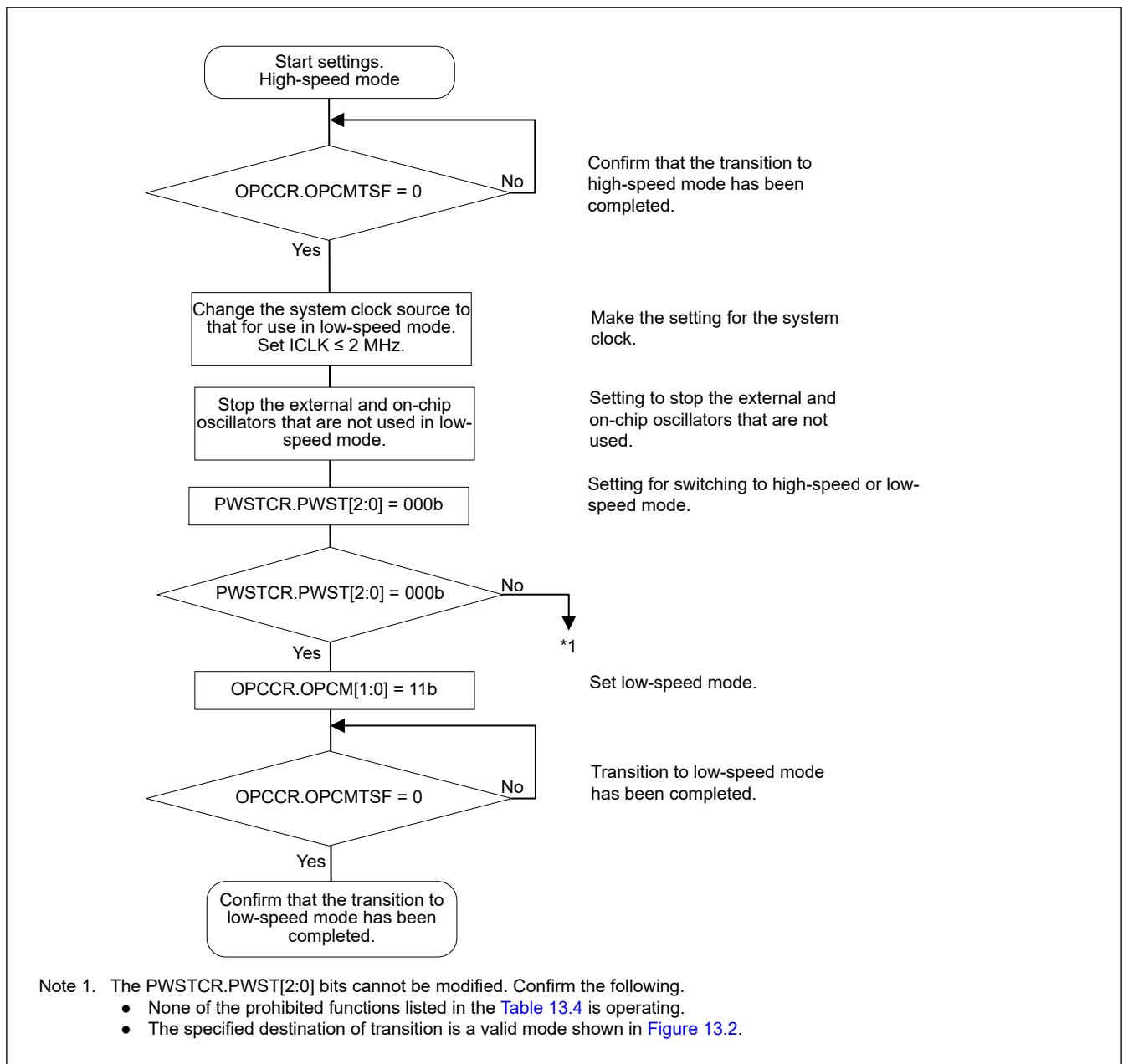


Figure 13.17 Procedure for Switching from High-speed Mode to Low-speed Mode

(2) In case of switching from low power mode to high power mode

To Switch from Low-Speed mode to High-Speed mode:

1. Confirm that the OPCCR.OPCMTSF flag is 0 (transition complete).
2. Set PWSTCR.PWST[2:0] = 000b.
3. Confirm that the PWSTCR.PWST[2:0] bits are set to 000b.
4. Set OPCCR.OPCM[1: 0] bits to 00b (High-Speed mode).

5. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
6. Run all required oscillators in High-Speed mode
7. Set the frequency of each clock equal to or less than the maximum operating frequency in High-Speed mode.

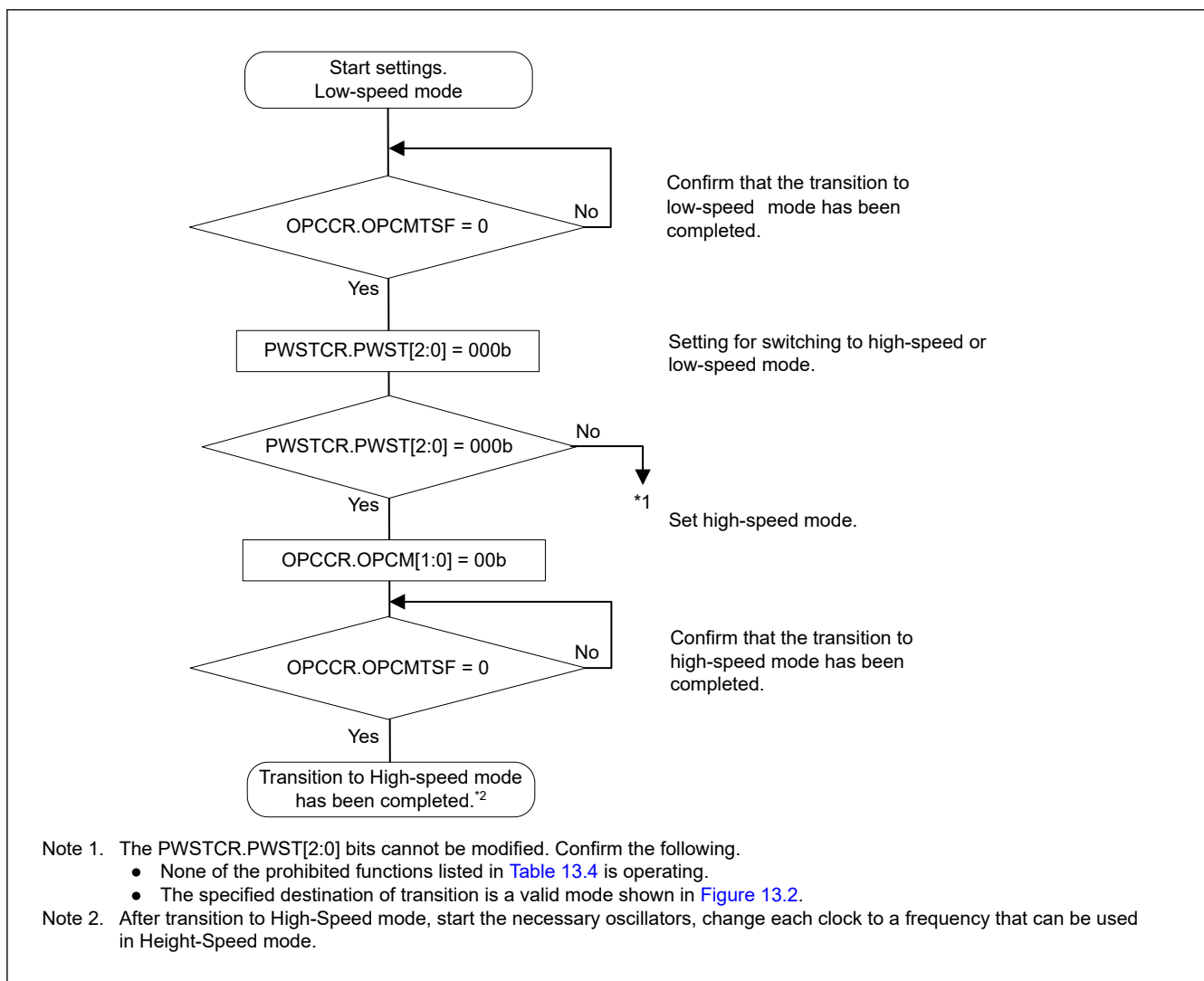


Figure 13.18 Procedure for Switching from Low-speed Mode to High-speed Mode

13.5.2.2 Low Leakage Current Mode (VBB)

Low leakage current mode (VBB) is for controlling back bias voltage to reduce leakage current.

When using low leakage current mode, the back bias voltage control (VBBC) circuit must be activated. The VBBC circuit is stopped on normal startup, so initial setup is required. Setting the VBBCR.VBBEN bit to 1 allows the VBBC circuit to start the initial setup operation. When the initial setup has completed, the VBBST.VBBSTUP flag is set to 1 and the VBBC circuit is ready to transition to low leakage current mode. Note that the VBBC circuit requires the initial setup time, so when using low leakage current mode, we recommend activating and setting up the VBBC circuit in advance after a reset release. For the initial setup time for the VBBC circuit, see [section 51, Electrical Characteristics](#).

The VBBCR.VBBEN bit does not need to be set to 1 on energy harvesting startup mode because the VBBC circuit starts operation during the internal reset period. When making a transition to low leakage current mode immediately after a reset release, we recommend using energy harvesting startup mode.

When switching from normal mode (NORMAL) to low leakage current mode (VBB), the initial setup of the VBBC circuit must be completed, and normal mode must be used for the power control mode. In this state, setting the PWSTCR.PWST[2:0] bits to 110b (VBB) and executing a WFE instruction allows transition to low leakage current mode. For more details, see [\(5\)To switch from normal mode \(NORMAL\) to low leakage current mode \(VBB\)](#).

When switching from low leakage current mode (VBB) to normal mode (NORMAL), low leakage current mode must be used for the power control mode. In this state, setting the PWSTCR.PWST[2:0] bits to 100b (normal mode) and executing a WFE instruction allows transition to normal mode. For more details, see [\(6\) To switch from low leakage current mode \(VBB\) to normal mode \(NORMAL\)](#).

When a transition to deep software standby mode occurs, the VBBCR.VBBEN bit is set to 0 and the VBBC circuit stops operating.

13.6 Low Power Consumption Mode

13.6.1 Sleep Mode

13.6.1.1 Transition to Sleep Mode

When a WFI instruction is executed while SBYCR.SSBY bit is 0, the MCU enters Sleep mode. In this mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop. Available resets or interrupts in Sleep mode cause the MCU to cancel Sleep mode. All interrupt sources are available. If using an interrupt to cancel Sleep mode, you must set the associated IELSRn register before executing a WFI instruction. For details, see [section 16, Interrupt Controller Unit \(ICU\)](#).

[Table 13.12](#) shows the IWDTC count operation when entering Sleep mode.

[Table 13.13](#) shows the WDT count operation when entering Sleep mode.

Table 13.12 IWDTC Count during Low Power Consumption Modes

Register Value	Low Power Consumption Mode	IWDTC Count
OFS0.IWDTCSTPCTL = 0	Sleep/snooze/software standby mode	Counting continues
OFS0.IWDTCSTPCTL = 1	Sleep/snooze/software standby mode	Counting stops

Table 13.13 WDT Count during Low Power Consumption Modes

Register Value	Low Power Consumption Mode	WDT Count		
			When WDTCLK = PCLKB (OFS0.WDTCLKSEL = 1)	When WDTCLK = CCC_2K (OFS0.WDTCLKSEL = 0)
In register start mode (OFS0.WDTSTRT = 1)	WDTCSSTPR. SLCSTP = 0	Sleep mode	Counting continues	Counting continues
		Snooze mode	Counting stops	
		Software standby mode		
	WDTCSSTPR. SLCSTP = 1	Sleep mode	Counting stops	Counting stops
		Snooze mode		
		Software standby mode		
In auto start mode (OFS0.WDTSTRT = 0)	OFS0.WDTSTPCTL = 0	Sleep mode	Counting continues	Counting continues
		Snooze mode	Counting stops	
		Software standby mode		
	OFS0.WDTSTPCTL = 1	Sleep mode	Counting stops	Counting stops
		Snooze mode		
		Software standby mode		

13.6.1.2 Canceling Sleep Mode

Sleep mode is canceled by any interrupt, a RES# pin reset, a power-on reset, a voltage monitor 0 reset, a voltage monitor 1 reset, a voltage monitor BAT reset, a bus master MPU error reset, a bus slave MPU error reset, an independent watchdog timer reset, or a watchdog timer reset. The operation is as follows:

- Canceling by an interrupt
When an interrupt request is generated, Sleep mode is canceled and the MCU starts the interrupt handling.
- Canceling by RES# pin reset
When RES# pin is driven low, the MCU enters the reset state. You must keep the RES# pin low for the time period specified in [section 51, Electrical Characteristics](#). When the RES# pin is driven high after the specified time period, the CPU starts the reset exception handling.
- Canceling by IWDTC reset
Sleep mode is canceled by an internal reset generated by an IWDTC underflow and the MCU starts the reset exception handling. Under the conditions in [Table 13.12](#) in which IWDTC counting stops in Sleep mode, an internal reset for canceling Sleep mode is not generated.
- 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. Under the conditions in [Table 13.13](#) in which WDT counting stops in Sleep mode, an internal reset for canceling Sleep mode is not generated.
- Canceling by other resets available in Sleep mode
Sleep mode is canceled by the associated resets, and the MCU starts reset exception handling

Note: For details on interrupt settings, see [section 16, Interrupt Controller Unit \(ICU\)](#)

13.6.2 Software Standby Mode

13.6.2.1 Transition to Software Standby Mode

When a WFI instruction is executed while the PWSTCR.PWST[2:0] bits are 000b, the SBYCR.SSBY bit is 1, and the DPSBYCR.DPSBY bit is 0, the MCU enters software standby mode. In this mode, the CPU, most of the on-chip peripheral functions, and oscillators stop. However, the values of the CPU internal registers, and the states of on-chip peripheral functions and the I/O port pins are retained. When the RAMSDn bit in the RAMSDCR register is 0, the values of the corresponding SRAM area are retained. When the RAMSDn bit is 1, the power supply to the SRAM area corresponding to the bit is shut off, so values in the area are not retained. However, current can be reduced. Software standby mode allows a significant reduction in power consumption because most of the oscillators stop in this mode.

[Table 13.5](#) shows the status of the on-chip peripheral functions and oscillators. Available resets or interrupts in software standby mode cause the MCU to cancel software standby mode. See [Table 13.6](#) for available interrupt sources and [section 16.2.9. WUPEN : Wake Up Interrupt Enable Register](#) for information on waking 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 16, Interrupt Controller Unit \(ICU\)](#).

Clear the DMAST.DMST and DTCST.DTCST bits to 0 before executing a WFI instruction, except when using the DTC in snooze mode. When using the DTC in snooze mode, set the DTCST.DTCST bit to 1 before executing a WFI instruction.

[Table 13.12](#) shows the IWDTC count operation when entering software standby mode.

[Table 13.13](#) shows the WDT count operation when entering software standby mode.

Do not enter software standby mode while the OSTDCR.OSTDE bit is 1 (oscillation stop detection interrupt function enabled). When transitioning to software standby mode, after setting the OSTDCR.OSTDE bit to 0, execute the WFI instruction. If a WFI instruction is executed while the OSTDCR.OSTDE bit is 1, the MCU enters sleep mode even when the SBYCR.SSBY bit is 1.

Do not enter software standby mode while the flash memory is undergoing a programming or erasing procedure. To enter software standby mode, execute a WFI instruction after the programming or erasure procedure completes.

Transition to software standby mode with the PWSTCR.SSBYVBB bit set to 1 (soft standby mode selection bit for low-leakage current mode) can reduce current consumption during software standby mode. Set PWSTCR.SSBYVBB to 1 while the back bias voltage control (VBBC) setup is completed (VBBST.VBBSTUP = 1), and transition to software standby.

Transition to software standby mode with the PWSTCR.SSBYPWG bit set to 1 (the software standby selection bit in the minimum power supply mode) can reduce current consumption during software standby mode. When transition from all power supply mode (ALLPWON), power supply mode other than Flash (EXFPWON), or boost mode (BOOST) to software standby mode with PWSTCR.SSBYPWG set to 1, all power supply modes (ALLPWON) are generated by interrupts It

returns to the power supply mode (EXFPWON) and boost mode (BOOST) except for Flash, but if you use the functions in the Iso2 and Iso3 areas, you need to reset them. When transitioning from the minimum power supply mode (MINPWON) to the software standby mode, the setting of PWSTCR.SSBYPWG becomes invalid.

If the PWSTCR.SSBYACC bit is set to 1 (software standby mode transition / return time reduction selection bit), the transition time to software standby mode and the return time from software standby mode are reduced. When PWSTCR.SSBYACC is set to 1, the current consumption during software standby mode is larger than when PWSTCR.SSBYACC is set to 0.

Table 13.14 shows the setting of the related control bits and the modes entered on execution of a WFI instruction.

Table 13.14 Bit Settings and Modes Entered on WFI Instruction Execution

Bit Settings		SBYCR.SSBY and DPSBYCR.DPSBY Bit Settings			
		SSBY = 0, DPSBY = 0	SSBY = 0, DPSBY = 1	SSBY = 1, DPSBY = 0	SSBY = 1, DPSBY = 1
OSTDCR.OSTDE	0	Sleep mode	Sleep mode	Software standby mode	Deep software standby mode
	1			Sleep mode	Sleep mode
FENTRYR.FENTRYC	0	Sleep mode	Sleep mode	Software standby mode	Deep software standby mode
	1			Sleep mode	Sleep mode
OFS0.IWDTSTRT, OFS0.IWDTSTPCTL	0, 0	Sleep mode	Sleep mode	Software standby mode	Software standby mode
	Other settings				Deep software standby mode
LVD1CR0.RI	0	Sleep mode	Sleep mode	Software standby mode	Deep software standby mode
	1				Software standby mode
LVDBATCR0.RI	0	Sleep mode	Sleep mode	Software standby mode	Deep software standby mode
	1				Software standby mode

13.6.2.2 Canceling Software Standby Mode

Software standby mode is canceled by the interrupts shown in Table 13.6, a RES# pin reset, a power-on reset, a voltage monitor reset, or a reset caused by an IWDG underflow. When software standby mode is canceled, Return to normal operating mode before transition to software standby mode. The oscillator that was operating before the transition to software standby mode resumes operation. After all of these oscillators are stabilized, the MCU cancels software standby mode.

When PWSTCR.SSBYPWG is 1, the transition from all power supply mode (ALLPWON), non-Flash power supply mode (EXFPWON), boost mode (BOOST) to software standby mode. When the software standby mode is released by an interrupt, it returns to the all power supply mode (ALLPWON), non-Flash power supply mode (EXFPWON), and boost mode (BOOST), respectively. When using the functions in the Iso2 and Iso3 areas, reconfiguration is required.

See section 16.2.9. WUPEN : Wake Up Interrupt Enable Register, for information on waking up the MCU from software standby mode.

You can cancel software standby mode from any of the following ways:

- Canceling by an interrupt
When an available interrupt request (see Table 13.6) is generated, all oscillators that were operating before the transition to software standby mode restart. After all of these oscillators are stabilized, the MCU cancels software standby mode and starts the interrupt handling.
- Canceling by RES# pin reset
When the RES# pin is driven low, the MCU enters the reset state and the oscillators start operating in their initial status. Be sure to keep the RES# pin low for the time period specified in section 51, Electrical Characteristics. When the RES# pin is driven high after the specified time period, the CPU starts the reset exception handling.
- Canceling by IWDG reset

Software standby mode is canceled by an internal reset generated by an IWDT underflow, and the MCU starts reset exception handling. Under the conditions in [Table 13.12](#) in which IWDT counting stops in software standby mode, an internal reset for canceling software standby mode is not generated.

- **Canceling by WDT reset**
Software standby mode is canceled by an internal reset generated by a WDT underflow, and the MCU starts reset exception handling. Under the conditions in [Table 13.13](#) in which WDT counting stops in software standby mode, an internal reset for canceling software standby mode is not generated.
- **Canceling by other reset available in software standby mode**
Software standby mode is canceled by the associated resets, and the MCU starts reset exception handling.

13.6.2.3 Example of Software Standby Mode Application

[Figure 13.19](#) shows an example of entry to Software Standby mode on detection of a falling edge of the IRQn pin, and exit from Software Standby mode by a rising edge of the IRQn pin.

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

Setting the ICU is also required to exit Software Standby mode. For details, see [section 16, Interrupt Controller Unit \(ICU\)](#). The oscillation stabilization time in [Figure 13.19](#) is specified in [section 51, Electrical Characteristics](#).

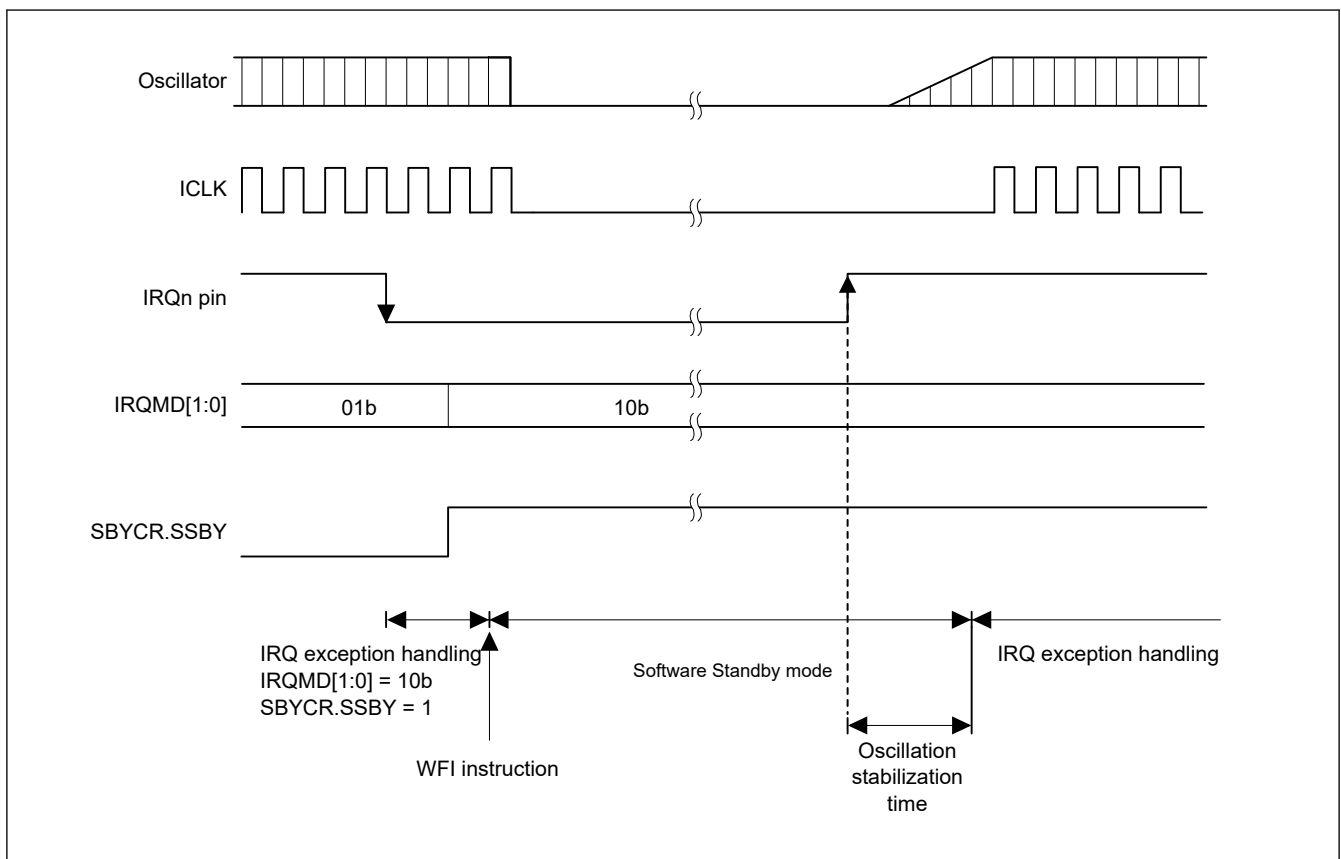


Figure 13.19 Example of Software Standby mode application

13.6.3 Snooze Mode

13.6.3.1 Transition to Snooze Mode

When the snooze control circuit receives a snooze request in software standby mode, the MCU transitions to snooze mode (snooze mode entry). In this mode, some peripheral modules operate without waking up the CPU. The peripheral modules

that can operate in snooze mode are shown in [Table 13.5](#). Also, the DTC operation can be selected in snooze mode by setting the SNZCR.SNZDTCEN bit.

[Table 13.15](#) shows the snooze requests to switch the MCU from software standby mode to snooze mode. To use a 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.

Table 13.15 Available Events for Invoking Snooze Mode

Snooze request	Control Register	
	Register	Bit*1
PORT_IRQn (n = 0 to 7)	SNZREQCR	SNZREQENn (n = 0 to 7)
KEY_INTKR	SNZREQCR	SNZREQEN17
AGTW0_AGTCMAI	SNZREQCR	SNZREQEN23
RTC_ALM	SNZREQCR	SNZREQEN24
RTC_PRD	SNZREQCR	SNZREQEN25
AGT1_AGTI	SNZREQCR	SNZREQEN28
AGT1_AGTCMAI	SNZREQCR	SNZREQEN29
AGT0_AGTCMAI	SNZREQCR	SNZREQEN30
RXD0 falling edge	SNZCR	RXDREQEN*2

Note 1. Do not enable multiple snooze requests at the same time

Note 2. This request can be used with the RXDREQEN bit set to 1 only when SCI0 operates in asynchronous mode

13.6.3.2 Canceling Snooze Mode

Snooze mode is canceled by an interrupt request that is available in Software Standby mode or a reset. [Table 13.6](#) shows the requests that can be used to exit each mode. After canceling the Snooze mode, the MCU enters Normal operating mode and proceeds with exception processing for the given interrupt or reset. The action triggered by the interrupt requests, selected in SELSR0, cancels Snooze mode. Interrupt canceling Snooze mode must be selected in IELSRn to link to the NVIC for the corresponding interrupt handling. See [section 16, Interrupt Controller Unit \(ICU\)](#) for information on SELSR0 and IELSRn registers.

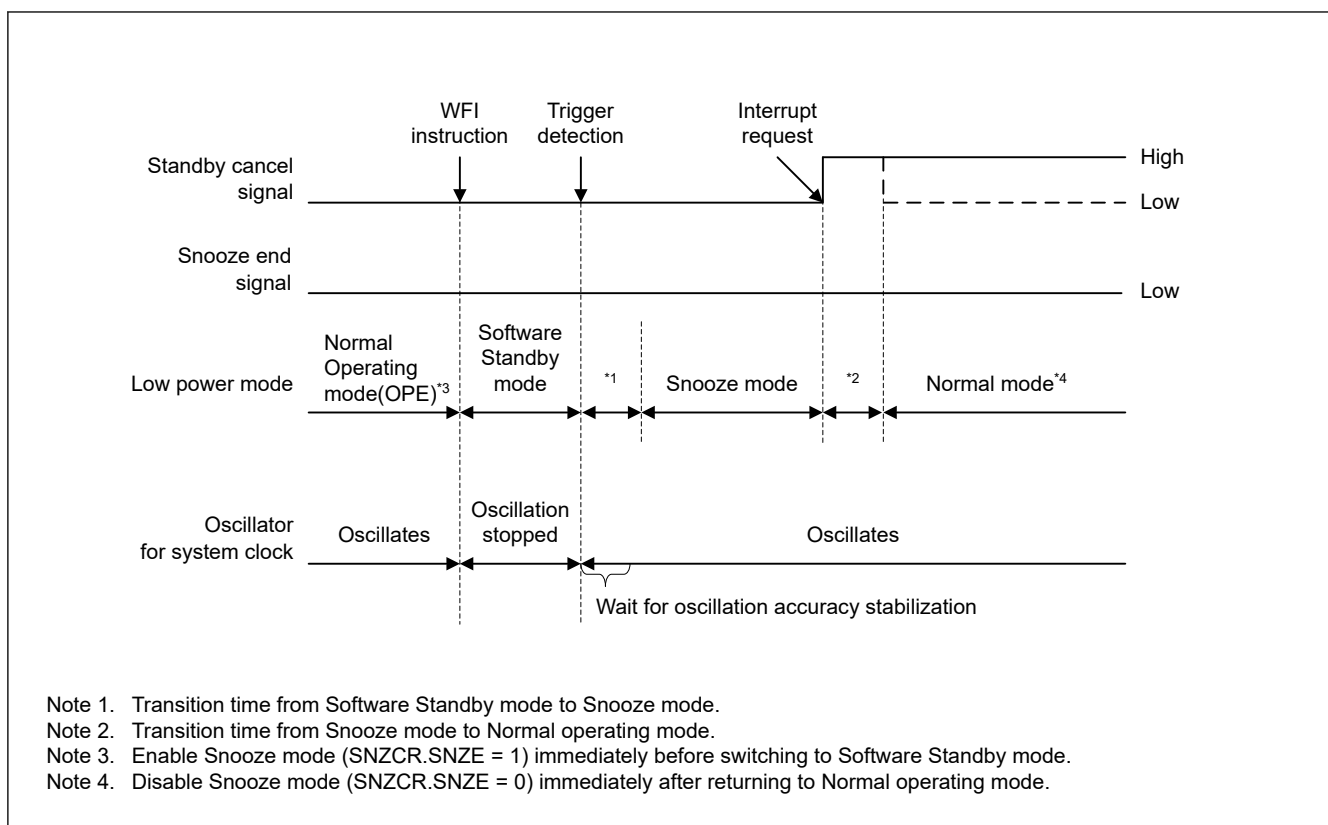


Figure 13.20 When interrupt request signal is generated in Snooze mode

13.6.3.3 Return from Snooze Mode to Software Standby Mode

Table 13.16 shows the snooze end request that can be used as triggers to return to Software Standby mode. The snooze end requests are available only in Snooze mode. If the requests are generated when the MCU is not in Snooze mode, they are ignored. When multiple requests are selected, each of the requests invokes transition to Software Standby mode from Snooze mode.

Table 13.17 shows the snooze end conditions that consist of the snooze end requests and the conditions of the peripheral modules. The SCI0, ADC14, and DTC modules can keep the MCU in Snooze mode until they complete the operation. However, an AGTn (n = 1) underflow as a trigger to return to Software Standby mode cancels Snooze mode without waiting for the completion of SCI0 operation.

Figure 13.21 shows the timing diagram for the transition from Snooze mode to Software Standby mode. This mode transition occurs according to which snooze end requests are set in the SNZEDCR register. A snooze request is cleared automatically after returning to Software Standby mode.

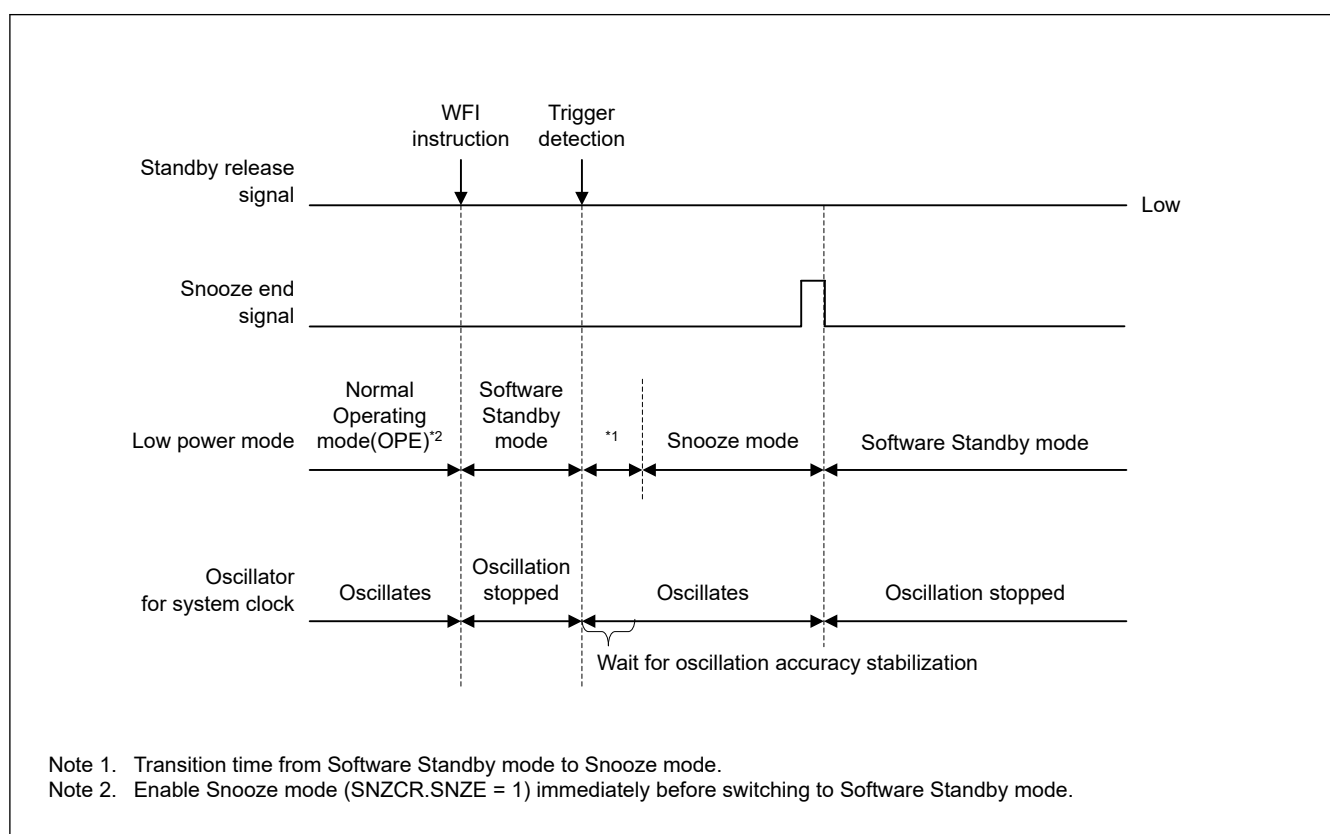
Table 13.16 Available snooze end requests (triggers to return to Software Standby mode)

Peripheral Module	Snooze end request	Enable/Disable Control	
		Register	Symbol
AGT1	AGT1 underflow (AGT1_AGTI)	SNZEDCR0	AGTUNFED
DTC	Last DTC transmission completion (DTC_COMPLETE)	SNZEDCR0	DTCZRED
DTC	Not Last DTC Transmission Completion (DTC_TRANSFER)	SNZEDCR0	DTCNZRED
ADC140	Window A/B compare match (ADC140_WCMPPM)	SNZEDCR0	AD0MATED
ADC140	Window A/B compare mismatch (ADC140_WCMPUM)	SNZEDCR0	AD0UMTED
SCI0	SCI0 address mismatch (SCI0_DCUF)	SNZEDCR0	SCI0UMTED

Table 13.17 Snooze end conditions

Operating module when a snooze end request occurs	Snooze end request	
	AGT1 underflow	All except AGT1 underflow
DTC	After the module completes operation.	After the module completes operation
ADC14		
SCI0	Immediately after a snooze end request is generated	
Other than specified	Immediately after a snooze end request is generated.	

Note: If the DTC is used to activate the ADC14 or SCI0, the MCU transitions to Software Standby mode immediately after a Snooze end request is generated.

**Figure 13.21 When interrupt request signal is not generated in Snooze mode**

13.6.3.4 Snooze Operation Example

Figure 13.22 shows an example setting for using ELC in Snooze mode.

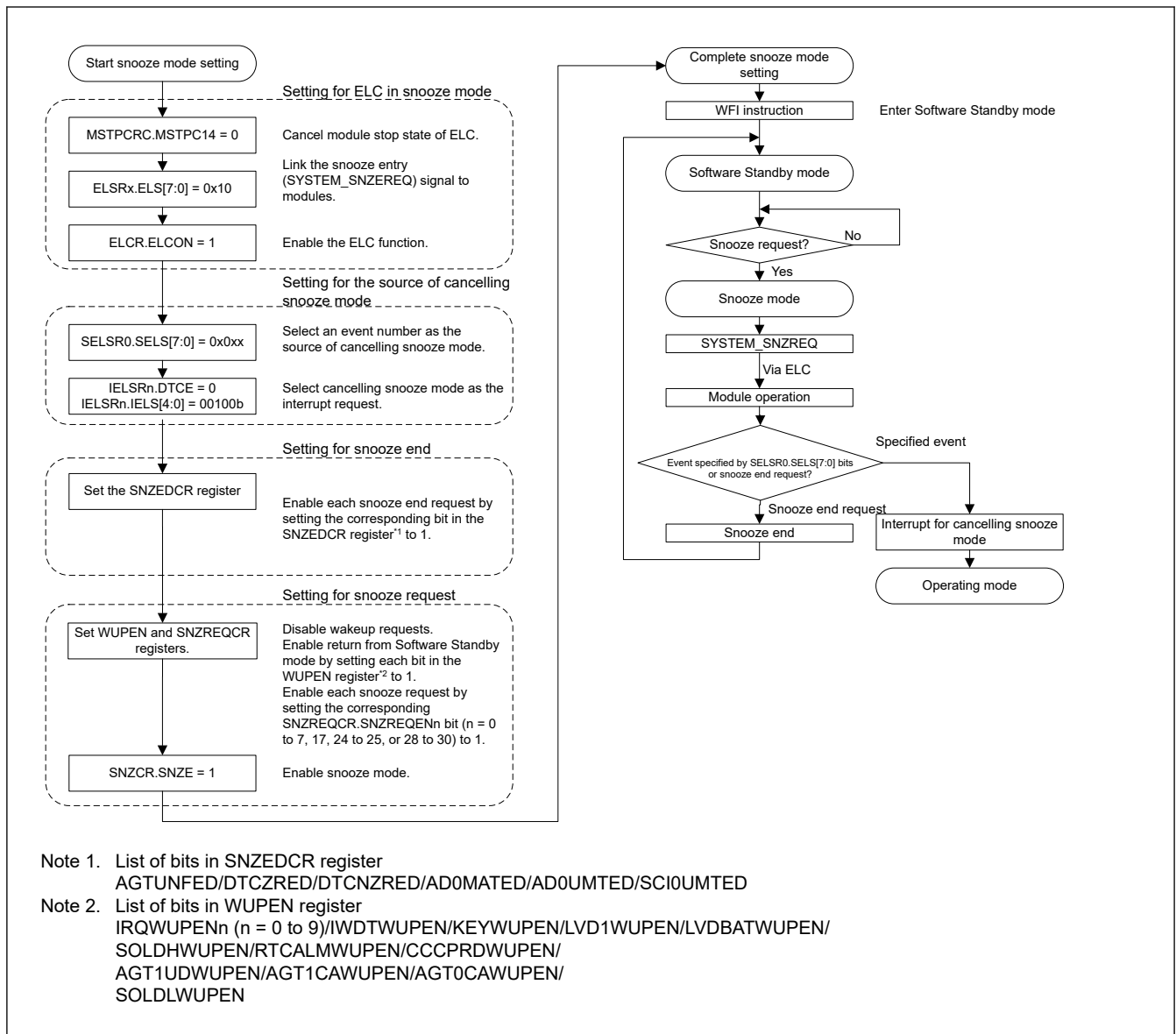


Figure 13.22 Setting example of using ELC in Snooze mode

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
- Low-speed mode

The SCI0.SEMR register must be set up as follows:

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

See [section 32, Serial Communications Interface \(SCI\)](#) for information on these bits.

[Figure 13.23](#) shows a setting example for using SCI0 in Snooze mode entry.

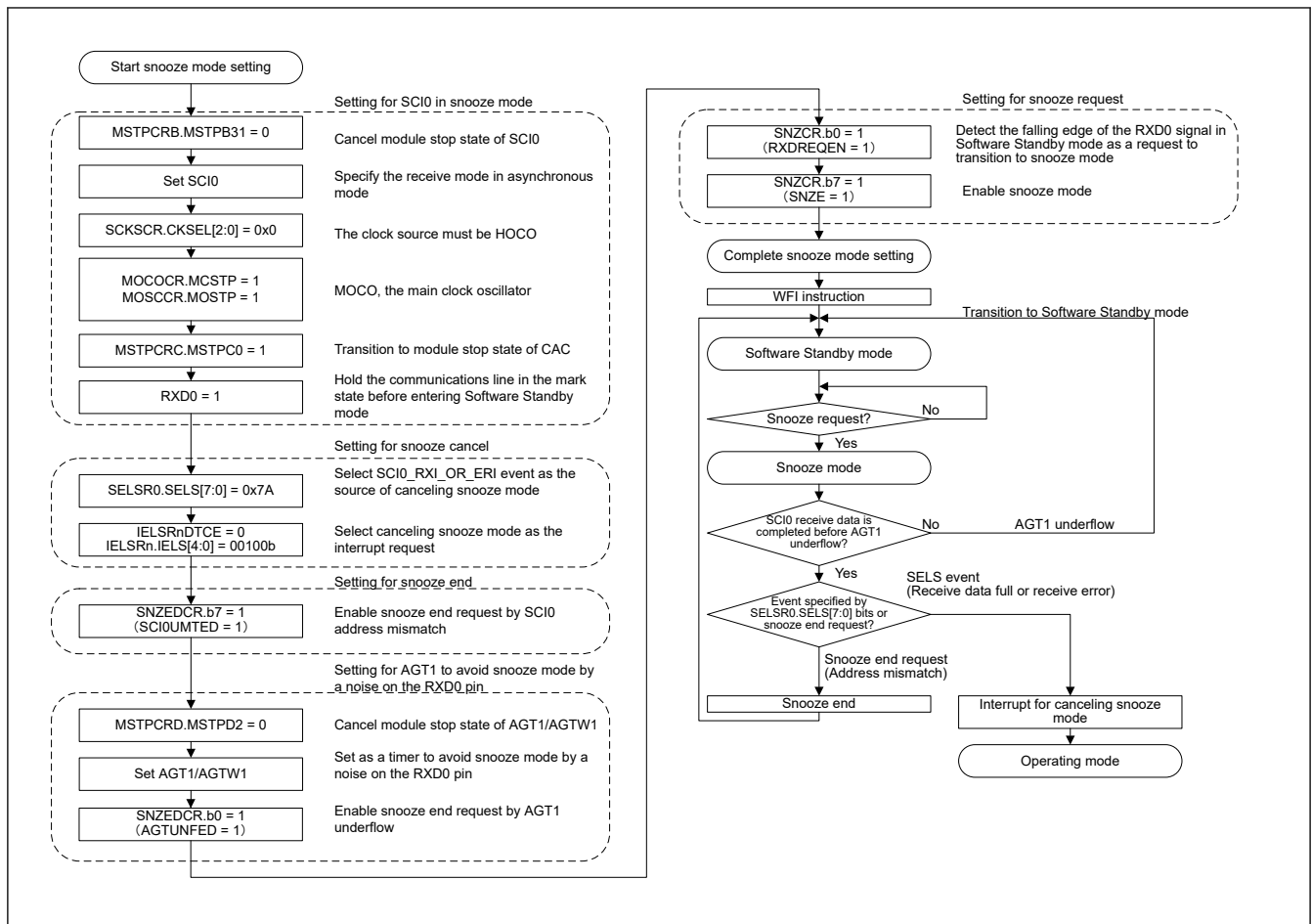


Figure 13.23 Setting example of using SCI0 in Snooze mode entry

13.6.4 Deep Software Standby Mode

13.6.4.1 Transition to Deep Software Standby Mode

When a WFI instruction is executed with the SBYCR.SSBY bit set to 1 and DPSBYCR.DPSBY bit set to 1, the MCU enters Deep Software Standby mode. See [section 13.6.2.1. Transition to Software Standby Mode](#) for the setting of the related control bits. In this mode, the CPU, on-chip peripheral functions (except for CCC, RTC, and WUPT), and all oscillators (except for Sub-clock oscillator) are stopped; furthermore, since the internal power supply to these modules is stopped, power consumption is reduced. The contents of all the registers of the CPU and internal peripheral modules (except for CCC, RTC, and WUPT) become undefined.

When the MCU enters Deep Software Standby mode while the IWDT is in auto-start mode and the OFS0.IWDTSTPCTL bit is 1, power supply to the IWDT-dedicated clock and the IWDT is cut off, and counting by the IWDT stops.

In case OFS0.IWDTSTPCTL bit is 0, the MCU enters Software Standby mode instead of Deep Software Standby mode, regardless of the setting of DPSBYCR.DPSBY bit. If OFS0.IWDTSTPCTL bit is 0 while OFS0.IWDTSTRT bit is 0 (auto-start mode), IWDT-dedicated clock and IWDT continues the operation.

When LVD1CR0.RI = 1 (selecting the voltage monitoring 1 reset) or LVD2CR0.RI = 1 (selecting the voltage monitoring 2 reset), the MCU enters Software Standby mode instead of Deep Software Standby mode. The I/O port states are same as Software Standby mode.

Note: Conditions on the DTC, DMAC, and IWDT for transition to Software Standby mode should be met before the WFI instruction is executed. For details, see [section 13.6.2. Software Standby Mode](#).

13.6.4.2 Cancelling Deep Software Standby Mode

Deep Software Standby mode is canceled by the interrupt shown in [Table 13.6](#), RES# pin reset, a power-on reset, or a voltage monitor 0 reset.

(1) Cancelling by an interrupt

Cancelling by interrupts is controlled by DPSIERn (n = 0 to 1) and DPSIFRn (n = 0 to 1). When a Deep Software Standby Cancelling interrupt is generated, the corresponding flag in DPSIFRn is set to 1. If the interrupt is enabled in DPSIERn, Deep Software Standby mode is canceled. Rising edge or falling edge can be selected by DPSIEGRn (n = 0 to 1). The interrupts for which an edge can be selected are the NMI, IRQn_DS (n = 0 to 3), voltage monitor 1, and voltage monitor BAT interrupts. When a Deep Software Standby mode canceling request occurs, the internal power is supplied and MOCO starts oscillating, and then the internal reset (Deep Software Standby reset) is generated for the entire MCU.

The stable MOCO clock is then supplied to the entire MCU and Deep Software Standby reset is canceled, and the MCU starts reset exception handling.

When Deep Software Standby mode is canceled by an external interrupt pin or internal interrupt signal, the RSTSR0.DPSRSTF flag is set to 1.

(2) Cancelling by RES# pin reset

When the RES# pin is driven low, the MCU cancels Deep Software Standby mode and enters the reset state. Keep the RES# pin low for the time specified in [section 51, Electrical Characteristics](#). When RES# pin is driven high after the specified time period, the CPU starts the reset exception handling.

(3) Cancelling by a power-on reset

Deep Software Standby mode is canceled by a power-on reset and the MCU starts the reset exception handling.

(4) Cancelling by a voltage monitor 0 reset

Deep Software Standby mode is canceled by a voltage monitor 0 reset from the voltage detection circuit and the MCU starts the reset exception handling.

13.6.4.3 Pin States when Deep Software Standby mode is Canceled

In Deep Software Standby mode, the I/O ports retain the same states from Software Standby mode. The MCU is initialized by an internal reset generated when Deep Software Standby mode is canceled. Upon cancellation of Deep Software Standby mode, the reset exception handling starts immediately. The following shows the states of I/O ports at this time.

Whether to initialize the I/O ports or to retain the I/O port states at the time of Software Standby mode can be selected by the DPSBYCR.IOKEEP bit.

- When the DPSBYCR.IOKEEP bit = 0
I/O ports are initialized by an internal reset generated when Deep Software Standby mode is canceled.
- When the DPSBYCR.IOKEEP bit = 1
This device is initialized by an internal reset generated when Deep Software Standby mode is canceled. The I/O ports maintains the state in software standby mode regardless of the internal state of this MCU. Even if the I/O ports or peripheral module settings are made, the pin state of the I/O ports remains unchanged from software standby mode. Setting the DPSBYCR.IOKEEP bit to 0 releases the held I/O port pin states. The DPSBYCR.IOKEEP bit is not initialized by an internal reset when exiting deep software standby mode.

13.6.4.4 Example of Deep Software Standby Mode Application

(1) Entering and exiting Deep Software Standby mode

[Figure 13.24](#) shows an example where a transition to Deep Software Standby mode is made at the falling edge of the IRQn_DS pin, and exiting Deep Software Standby mode is made at the rising edge of the IRQn_DS pin. In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 01b (falling edge). Then, after the DPSIEGR0.DIRQnEG (n = 0 to 3) bit is set to 1 (rising edge) and the SBYCR.SSBY bit and DPSBYCR.DPSBY bit are both set to 1, the WFI instruction is executed. Thus a transition to Deep Software Standby mode is made. After that, exiting Deep Software Standby mode is made at the rising edge of the IRQn_DS pin.

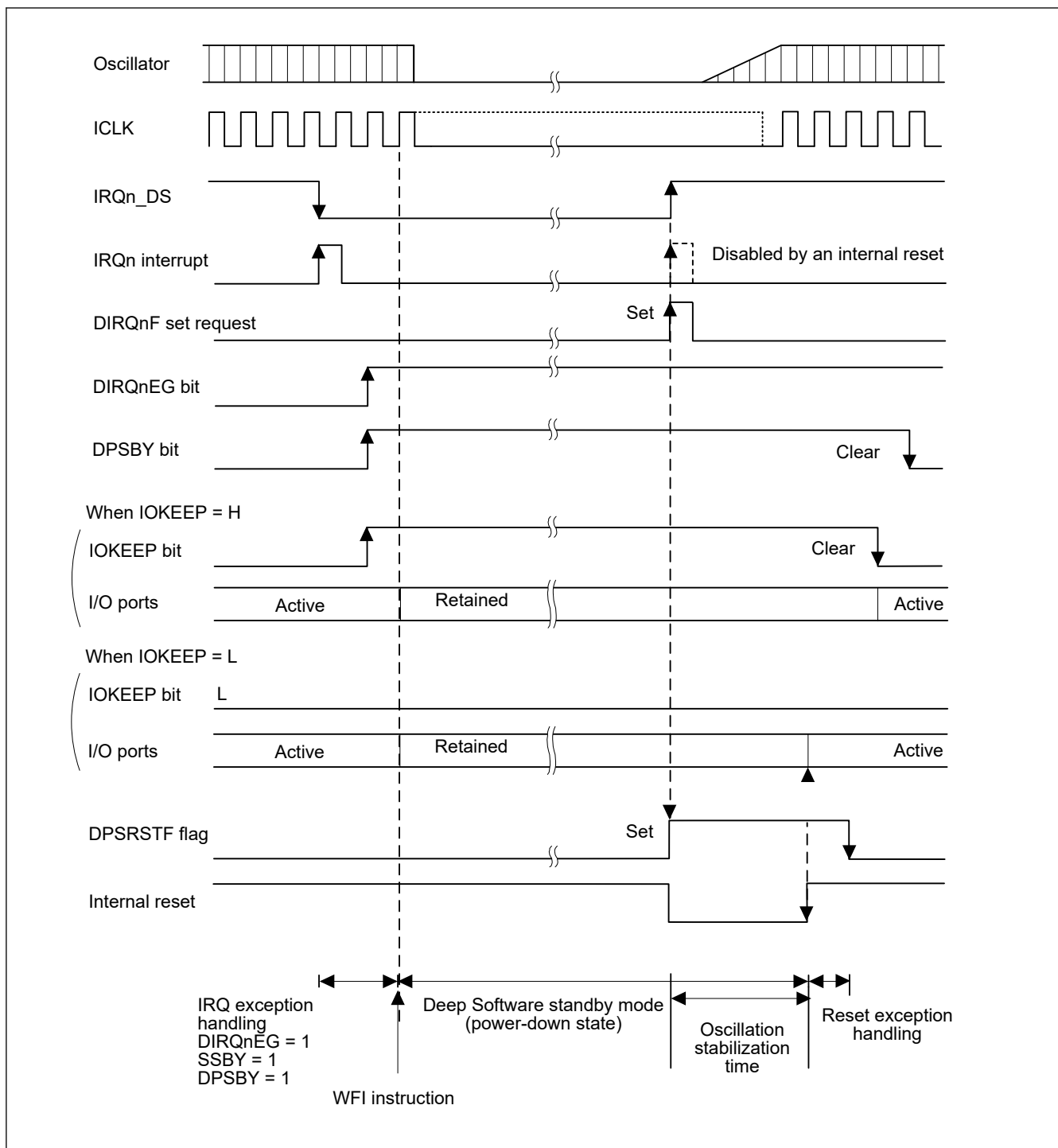


Figure 13.24 Example of Deep Software Standby Mode Application

13.6.4.5 Usage Flow for Deep Software Standby Mode

Figure 13.25 shows an example of a flowchart to use Deep Software Standby mode.

In this example, the $RSTSR0.DPSRSTF$ flag of the reset function is read after the reset exception handling to determine whether a reset was generated by the $RES\#$ pin or by the cancellation of Deep Software Standby mode.

In the case of a reset by the $RES\#$ pin, a transition to Deep Software Standby mode is made after the required register settings have been made.

In the case of a reset by the cancellation of Deep Software Standby mode, the $DPSBYCR.IOKEEP$ bit is cleared to 0 after the I/O port settings have been made.

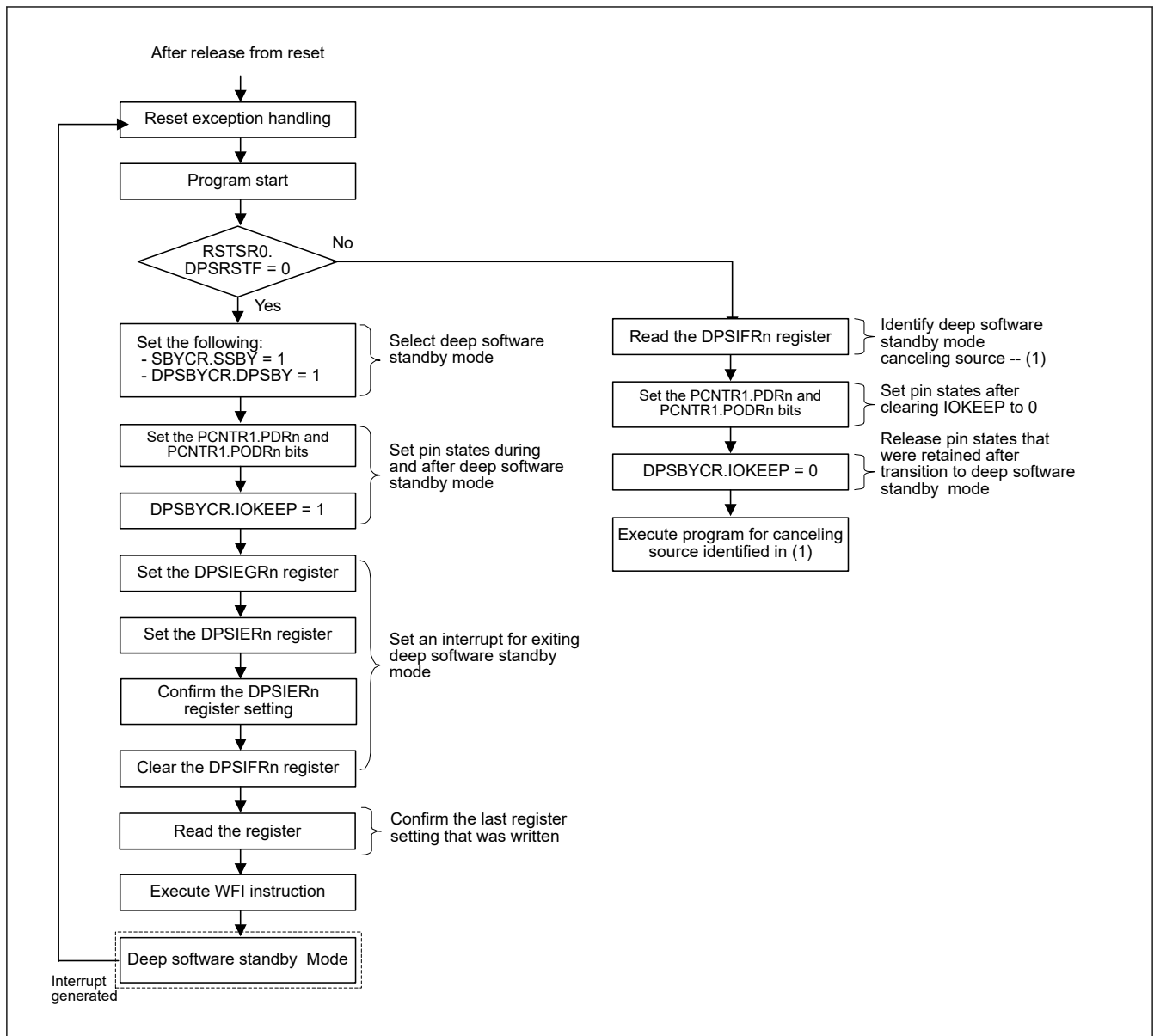


Figure 13.25 Example of Flowchart to Use Deep Software Standby Mode

13.7 Usage Notes

13.7.1 Register Access

(1) Write access to registers during specific modes or transitions

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

[Registers]

- All registers with a peripheral name of SYSC.

[Conditions]

- OPCCR.OPCMTSF = 1 or SOPCCR.SOPCMTSF = 1 (during transition of the operating power control mode)
- During the time period from executing a WFI instruction to returning to Normal mode
- FENTRYR.FENTRYC = 1 (In flash P/E mode)

(2) Valid setting for the clock-related registers

Table 13.18 and Table 13.19 show the valid settings for the clock-related registers in each operating power control mode. Do not write any value other than the valid setting. Additionally, each register has some prohibited settings under certain 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 13.18 Valid settings for the clock-related registers (1)

Mode		Valid settings						
		SCKSCR. CKSEL[2:0]	SCKDIVCR. ICK[2:0]	MOSCCR. MOSTP	HOCOCR. HCSTP	HOCOMCR. HCFRQ[1:0]	MOCOCR. MCSTP	LOCOCR. LCSTP
Boost mode (BOOST)		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 (stopped)	0 (operating) 1 (stopped) *1	00b: 24 MHz 01b: 32 MHz 10b: 48 MHz 11b: 64 MHz	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)
Normal mode (NORMAL)	High-speed/ low-speed	000b (HOCO) 001b (MOCO) 010b (LOCO) 011b (MOSC) 100b (SOSC)				00b: 24 MHz 01b: 32 MHz		
Low leakage current mode (VBB)		010b (LOCO) 100b (SOSC)	—	1 (stopped)	1 (stopped)	—	1 (stopped)	

Note 1. For transition to minimum power supply mode (MINPWON), HOCOCR.HCSTP must be set to 1 (HOCO stop).

Table 13.19 Valid settings for the clock-related registers (2)

Clock Source	Valid Settings	
	PWSTCR. PWST[2:0]	OPCCR. OPCM[1:0]
HOCO (48-MHz/64-MHz oscillation)	— *1	00b
HOCO (24-MHz/32-MHz oscillation)	000b 001b 010b 100b 101b	00b, 11b
MOSC MOCO	000b 001b 010b 011b 100b 101b	00b, 11b
SOSC. LOCO	000b 001b 010b 011b 100b 101b 110b	00b, 11b

Note 1. Writing to the PWSTCR register is prohibited while the HOCO is operating at 48 MHz/64 MHz

(3) Write access to registers by the DMAC or DTC

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

[Registers]

- MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD.

(4) Write access to registers 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, SNZEDCR0, SNZREQCR0.

(5) Write access to registers when the PRCR.PRC1 bit is 0

Do not write to the following registers when the PRCR.PRC1 bit is 0.

[Registers]

- Registers protected by the PRCR.PRC1 bit (see [section 15, Register Write Protection](#)).

13.7.2 I/O Port pin states

The I/O port pin states in Software Standby mode, Deep Software Standby and Snooze mode, unless modifying in Snooze mode, are the same before entering the modes. Therefore, power consumption is not reduced while the output signals are held high.

13.7.3 Module-Stop State of DMAC, DTC

Before writing 1 to MSTPCRA.MSTPA22, clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0. For details, see [section 19, DMA Controller \(DMAC\)](#) and [section 20, Data Transfer Controller \(DTC\)](#).

13.7.4 Internal Interrupt Sources

Interrupts do not operate in the module-stop state. If setting the module-stop bit while an interrupt request is generated, a CPU interrupt source or a DMAC or DTC startup source cannot be cleared. For this reason, make sure you disable the corresponding interrupts before setting the module-stop bits.

13.7.5 Transition to Low Power Modes

To transition to low power consumption modes (sleep mode, software standby mode, or deep software standby mode), be sure to execute the WFI instruction. To transition to the power state set in the PWSTCR register, execute the WFE instruction. Do not set the SLEEPDEEP bit of the system control register in the Cortex®-M0+ core because the MCU does not support the low power consumption modes specified by the SLEEPDEEP bit. For details, see the Cortex™-M0 Devices Generic User Guide (ARM DUI 0662B).

13.7.6 Input Buffer Control by DIRQnE Bit (n = 0 to 3)

Setting the DPSIER0.DIRQnE (n = 0 to 3) bit to 1 enables the input buffer of the IRQ0_DS to IRQ3_DS pins. Note that, although inputs to these pins are sent to the DPSIFR0.DIRQnF (n = 0 to 3) flags, they are not sent to the interrupt controller (ICU), peripheral modules, and I/O ports.

13.7.7 Timing of WFI Instruction

It is possible for the WFI instruction to be executed before I/O register write is completed, in which case operation might not be as intended. This can happen if the WFI is placed immediately after a write to an I/O register. To avoid this problem, read back the register that was written to confirm that the write completed.

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

Do not write registers in WDT or IWDT by DMAC or DTC while WDT or IWDT is stopped by entering Sleep mode or Snooze mode.

13.7.9 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, you must disable oscillators that are not required in Snooze mode before entering Software Standby mode. Otherwise, the transition from Software Standby mode to Snooze mode takes longer.

13.7.10 Snooze Mode Entry by RXD0 Falling Edge

When the SNZCR.RXDREQEN bit is 1, noise on the RXD0 pin might cause the MCU to transfer from Software Standby mode to Snooze mode. Any subsequent RXD0 data can be received in Snooze mode by a noise on the RXD0 pin. If the MCU does not receive RXD0 data after the noise, an interrupt such as SCI0_ERI or SCI0_RXI, or an address mismatch event is not generated and the MCU stays in Snooze mode. To avoid this, use an AGTn (n = 1) underflow interrupt to return to Software Standby mode or Normal operating mode when using SCI0 in Snooze mode. However, do not use the AGTn (n = 1) underflow as a source to return to Software Standby mode during an SCI communication.

13.7.11 Using SCI0 in Snooze Mode

When using SCI0 in snooze mode, do not use a request of snooze cancel other than AGTn (n = 1) underflow.

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 be stopped before entering Software Standby mode
- The RXD0 pin must be kept high before entering Software Standby mode
- A transition to Software Standby mode must not occur during an SCI0 communication
- The MSTPCRC.MSTPC0 bit must be 1 before entering Software Standby mode.

13.7.12 Conditions of A/D Conversion Start in Snooze Mode

ADC14 can only be triggered by the ELC in Snooze mode. Do not use software trigger or ADTRGn (n = 0) pin.

13.7.13 ELC Event in Snooze Mode

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

- Snooze mode entry (SYSTEM_SNZREQ)
- DTC transfer end (DTC_DTCEND)
- ADC14 window A/B compare match (ADC140_WCMPPM)
- ADC14 window A/B compare mismatch (ADC140_WCMPUM)
- Data operation circuit interrupt (DOC_DOPCI).

13.7.14 Notes on Switching Power Supply, Power Control, and Low Power Consumption Modes

When switching the power supply modes (ALLPWON/EXFPWON/MINPWON), the power control modes (BOOST, NORMAL (high-speed/low-speed), VBB), and the low power consumption modes (SLEEP, SSTBY, DSTBY), settings must be made as specified. [Table 13.20](#) shows the switching of power supply, power control, and low power consumption modes and setting numbers, and [Table 13.21](#) shows the details of setting for each number. Switching other than those shown in [Table 13.20](#) cannot be performed. In addition, settings other than those shown in [Table 13.21](#) are prohibited.

13.7.15 Notes on mode transition

When changing from all power supply mode (ALLPWON) to all power supply mode (ALLPWON), it is necessary to set $ICLK \leq 4$ MHz before executing the WFE instruction. When transitioning from all power supply mode (ALLPWON) to software standby mode (SSTBY), it is necessary to set $ICLK \leq 4$ MHz before executing the WFI instruction.

Table 13.20 Switching of Power Supply, Power Control, and Low Power Consumption Modes and Setting Numbers (1 of 5)

Before Transition			After Transition			
Power Supply Mode	Operating/Low Power Consumption Mode		Power Supply Mode	Operating/Low Power Consumption Mode		Setting Number
ALLPWON	OPE	NORMAL	ALLPWON	OPE	VBB	4
					BOOST	3
				SLEEP	NORMAL	12
			EXFPWON	OPE	NORMAL	8
				SSTBY	NORMAL	13
					VBB	13-2
			MINPWON	OPE	NORMAL	10
				SSTBY	NORMAL	15
					VBB	15-2
			DSTBY			16
		BOOST	ALLPWON	OPE	NORMAL	5
					VBB	6
				SLEEP	BOOST	12
			EXFPWON	SSTBY	NORMAL	13
					VBB	13-2
				SSTBY	NORMAL	15
			MINPWON	SSTBY	NORMAL	15
					VBB	15-2
				DSTBY		
			VBB	ALLPWON	OPE	NORMAL
		BOOST				3
		SLEEP			VBB	12
		EXFPWON		OPE	VBB	8
				SSTBY	VBB	13
					VBB	13
		MINPWON		OPE	VBB	10
				SSTBY	VBB	15
					VBB	15
		DSTBY			16	
		SLEEP	NORMAL	ALLPWON	OPE	NORMAL
	BOOST		BOOST			17
	VBB		VBB			17
	SNOOZE	NORMAL	ALLPWON	OPE	NORMAL	18
			EXFPWON	SSTBY	NORMAL	19
			MINPWON	SSTBY	NORMAL	19
		BOOST	ALLPWON	OPE	BOOST	18
			EXFPWON	SSTBY	NORMAL	19
			MINPWON	SSTBY	NORMAL	19
		VBB	ALLPWON	OPE	VBB	18
			EXFPWON	SSTBY	VBB	19
			MINPWON	SSTBY	VBB	19

Table 13.20 Switching of Power Supply, Power Control, and Low Power Consumption Modes and Setting Numbers (2 of 5)

Before Transition				After Transition				
Power Supply Mode	Operating/Low Power Consumption Mode			Power Supply Mode	Operating/Low Power Consumption Mode			Setting Number
EXFPWON	OPE	NORMAL	High-speed	ALLPWON	OPE	NORMAL		9
				EXFPWON	OPE	NORMAL	Low-speed	1
						VBB		4
					SLEEP	NORMAL		12
					SSTBY	NORMAL		13
						VBB		13-2
				MINPWON	OPE	NORMAL	High-speed	10
					SSTBY	NORMAL		15
						VBB		15-2
				DSTBY			16	
			Low-speed	ALLPWON	OPE	NORMAL		9
				EXFPWON	OPE	NORMAL	High-speed	2
						VBB		4
					SLEEP	NORMAL		12
					SSTBY	NORMAL		13
						VBB		13-2
				MINPWON	OPE	NORMAL	Low-speed	10
					SSTBY	NORMAL		15
						VBB		15-2
				DSTBY			16	

Table 13.20 Switching of Power Supply, Power Control, and Low Power Consumption Modes and Setting Numbers (3 of 5)

Before Transition				After Transition					
Power Supply Mode	Operating/Low Power Consumption Mode			Power Supply Mode	Operating/Low Power Consumption Mode		Setting Number		
EXFPWON	OPE	VBB		ALLPWON	OPE	VBB		9	
				EXFPWON	OPE	NORMAL	High-speed	7	
							Low-speed	7	
					SLEEP	VBB		12	
					SSTBY	VBB		13	
				MINPWON	OPE	VBB		10	
					SSTBY	VBB		15	
				DSTBY					
	SLEEP	NORMAL	EXFPWON	OPE	NORMAL	High-speed	17		
						Low-speed	17		
		VBB	EXFPWON	OPE	VBB		17		
	SSTBY	NORMAL	ALLPWON	OPE	NORMAL		20		
					BOOST		20		
					SNOOZE	NORMAL		23	
						BOOST		23	
				EXFPWON	OPE	NORMAL	High-speed	21	
							Low-speed	21	
					SNOOZE	NORMAL		24	
					VBB	ALLPWON	OPE	NORMAL	
			VBB					20	
			SNOOZE	NORMAL			23		
				VBB			23		
			EXFPWON	OPE		NORMAL		21	
						VBB		21	
				SNOOZE		NORMAL		21	
						VBB		24	
			SNOOZE	NORMAL	EXFPWON	OPE	NORMAL	High-speed	18
	Low-speed	18							
	SSTBY	NORMAL				19			
	VBB	EXFPWON				OPE	VBB		18
						SSTBY	VBB		19
	MINPWON	OPE	NORMAL	High-speed	ALLPWON	OPE	NORMAL		9
EXFPWON					OPE	NORMAL	High-speed	11	
MINPWON					OPE	NORMAL	Low-speed	1	
							VBB		4
					SLEEP	NORMAL		12	
					SSTBY	NORMAL		14	
VBB						14-2			
DSTBY							16		

Table 13.20 Switching of Power Supply, Power Control, and Low Power Consumption Modes and Setting Numbers (4 of 5)

Before Transition				After Transition					
Power Supply Mode	Operating/Low Power Consumption Mode			Power Supply Mode	Operating/Low Power Consumption Mode			Setting Number	
MINPWON	OPE	NORMAL	Low-speed	ALLPWON	OPE	NORMAL		9	
				EXFPWON	OPE	NORMAL	Low-speed	11	
				MINPWON	OPE	NORMAL	High-speed	2	
						VBB		4	
					SLEEP	NORMAL		12	
					SSTBY	NORMAL		14	
						VBB		14-2	
				DSTBY			16		
		VBB	ALLPWON	OPE	VBB		9		
			EXFPWON	OPE	VBB		11		
			MINPWON	OPE	NORMAL	High-speed	7		
					Low-speed		7		
				SLEEP	VBB		12		
				SSTBY	VBB		14		
			DSTBY			16			
	SLEEP	NORMAL		MINPWON	OPE	NORMAL	High-speed	17	
		Low-speed				17			
		VBB				VBB		17	
	SSTBY	NORMAL		ALLPWON	OPE	NORMAL		20	
						BOOST		20	
						SNOOZE	NORMAL		23
							BOOST		23
					EXFPWON	OPE	NORMAL	High-Speed	21
							Low-Speed		21
						SNOOZE	NORMAL		24
							MINPWON	OPE	NORMAL
				Low-speed		22			
				SNOOZE	NORMAL			25	
					VBB	ALLPWON		OPE	NORMAL
				VBB			20		
		SNOOZE	NORMAL				23		
			VBB				23		
EXFPWON		OPE	NORMAL			21			
			VBB			21			
		SNOOZE	NORMAL			24			
			VBB			24			
MINPWON		OPE	NORMAL		22				
			VBB		22				
	SNOOZE	NORMAL		25					
		VBB		25					

Table 13.20 Switching of Power Supply, Power Control, and Low Power Consumption Modes and Setting Numbers (5 of 5)

Before Transition			After Transition				
Power Supply Mode	Operating/Low Power Consumption Mode		Power Supply Mode	Operating/Low Power Consumption Mode			Setting Number
MINPWON	SNOOZE	NORMAL	MINPWON	OPE	NORMAL	High-speed	18
						Low-speed	18
				SSTBY	NORMAL		19
		VBB		MINPWON	OPE	VBB	
	SSTBY		VBB		19		

Table 13.21 Numbers and Details of Settings for Switching of Power Supply, Power Control, and Low Power Consumption Modes (1 of 2)

Setting Number	Register that Needs to be Set	Flag that Needs to be Checked	WFI/WFE Instruction Execution	Required Interrupt
1	PWSTCR.PWST[2:0] = 000b OPCCR.OPCM[1:0] = 11b	—	—	—
2	PWSTCR.PWST[2:0] = 000b OPCCR.OPCM[1:0] = 00b	—	—	—
3	SBYCR.SSBY = 1 DPSBYCR.DPSBY = 0 SNZCR.SNZE = 0 PWSTCR.PWST[2:0] = 101b	—	WFE instruction	—
4	VBBCR.VBBEN = 1 SBYCR.SSBY = 1 DPSBYCR.DPSBY = 0 SNZCR.SNZE = 0 PWSTCR.PWST[2:0] = 110b	VBBST.VBBSTUP = 1	WFE instruction	—
5	SBYCR.SSBY = 1 DPSBYCR.DPSBY = 0 SNZCR.SNZE = 0 PWSTCR.PWST[2:0] = 100b	—	WFE instruction	—
6	SBYCR.SSBY = 1 DPSBYCR.DPSBY = 0 SNZCR.SNZE = 0 PWSTCR.PWST[2:0] = 110b	—	WFE instruction	—
7	SBYCR.SSBY = 1 DPSBYCR.DPSBY = 0 SNZCR.SNZE = 0 PWSTCR.PWST[2:0] = 100b	—	WFE instruction	—
8	SBYCR.SSBY = 1 DPSBYCR.DPSBY = 0 SNZCR.SNZE = 0 PWSTCR.PWST[2:0] = 010b	—	WFE instruction	—
9	SBYCR.SSBY = 1 DPSBYCR.DPSBY = 0 SNZCR.SNZE = 0 PWSTCR.PWST[2:0] = 001b	—	WFE instruction	—
10	SBYCR.SSBY = 1 DPSBYCR.DPSBY = 0 SNZCR.SNZE = 0 PWSTCR.PWST[2:0] = 011b	—	WFE instruction	—
11	SBYCR.SSBY = 1 DPSBYCR.DPSBY = 0 SNZCR.SNZE = 0 PWSTCR.PWST[2:0] = 010b	—	WFE instruction	—
12	SBYCR.SSBY = 0	—	WFI instruction	—

Table 13.21 Numbers and Details of Settings for Switching of Power Supply, Power Control, and Low Power Consumption Modes (2 of 2)

Setting Number	Register that Needs to be Set	Flag that Needs to be Checked	WFI/WFE Instruction Execution	Required Interrupt
13	SBYCR.SSBY = 1 DPSBYCR.DPSBY = 0 SNZCR.SNZE = 0 or 1 ^{*1} PWSTCR.PWST[2:0] = 000b PWSTCR.SSSBYPWG = 0 PWSTCR.SSBYVBB = 0	—	WFI instruction	—
13-2	SBYCR.SSBY = 1 DPSBYCR.DPSBY = 0 SNZCR.SNZE = 0 or 1 ^{*1} PWSTCR.PWST[2:0] = 000b PWSTCR.SSSBYPWG = 0 PWSTCR.SSBYVBB = 1	—	WFI instruction	—
14	SBYCR.SSBY = 1 DPSBYCR.DPSBY = 0 SNZCR.SNZE = 0 or 1 ^{*1} PWSTCR.PWST[2:0] = 000b PWSTCR.SSBYVBB = 0	—	WFI instruction	—
14-2	SBYCR.SSBY = 1 DPSBYCR.DPSBY = 0 SNZCR.SNZE = 0 or 1 ^{*1} PWSTCR.PWST[2:0] = 000b PWSTCR.SSBYVBB = 1	—	WFI instruction	—
15	SBYCR.SSBY = 1 DPSBYCR.DPSBY = 0 SNZCR.SNZE = 0 or 1 ^{*1} PWSTCR.PWST[2:0] = 000b PWSTCR.SSSBYPWG = 1 PWSTCR.SSBYVBB = 0	—	WFI instruction	—
15-2	SBYCR.SSBY = 1 DPSBYCR.DPSBY = 0 SNZCR.SNZE = 0 or 1 ^{*1} PWSTCR.PWST[2:0] = 000b PWSTCR.SSSBYPWG = 1 PWSTCR.SSBYVBB = 1	—	WFI instruction	—
16	PWSTCR.PWST[2:0] = 000b SBYCR.SSBY = 1 DPSBYCR.DPSBY = 1	—	WFI instruction	—
17	—	—	—	Interrupt by one of all sources
18	—	—	—	Interrupt as listed in Table 13.6
19	—	—	—	Interrupt as listed in Table 13.6
20	—	—	—	Interrupt as listed in Table 13.6
21	—	—	—	Interrupt as listed in Table 13.6
22	—	—	—	Interrupt as listed in Table 13.6
23	—	—	—	Interrupt as listed in Table 13.6
24	—	—	—	Interrupt as listed in Table 13.6
25	—	—	—	Interrupt as listed in Table 13.6

Note 1. To disable transition from SSTBY to SNOOZE, SNZCR.SNZE = 0
To enable transition from SSTBY to SNOOZE, SNZCR.SNZE = 1

14. Energy Harvesting Controller (EHC)

14.1 Overview

When a power generating element is connected to the VSC_VCC pin, the Energy Harvesting Controller (EHC) starts operation before release from the reset state, and can supply power to a storage capacitor connected to the VCC_SU pin, a secondary battery connected to the VBAT_EHC pin, and the VCC domain within the MCU. Table 14.1 lists the functions of EHC and Figure 14.1 shows a block diagram.

Table 14.1 EHC functions

Function	Description
Automatic charge and discharge function	The power generating element connected to the VSC_VCC pin automatically supplies power to the MCU and charges the secondary battery including the storage capacitor according to the current drawn in the MCU.
Secondary battery overcharging protection	If the voltage of the secondary battery connected to the VBAT_EHC pin or the storage capacitor connected to the VCC_SU pin exceeds a predetermined value, this function automatically prevents overcharging by diverting the charging current from the power generating element connected to the VSC_VCC pin to VSS.
Reverse current flow prevention	This function automatically prevents reverse current flow from the secondary battery connected to the VBAT_EHC pin or the storage capacitor connected to the VCC_SU pin to the power generating element connected to the VSC_VCC pin.
Level of power generation detection	This function detects the power level that is generated by the power generating element connected to the VSC_VCC pin.

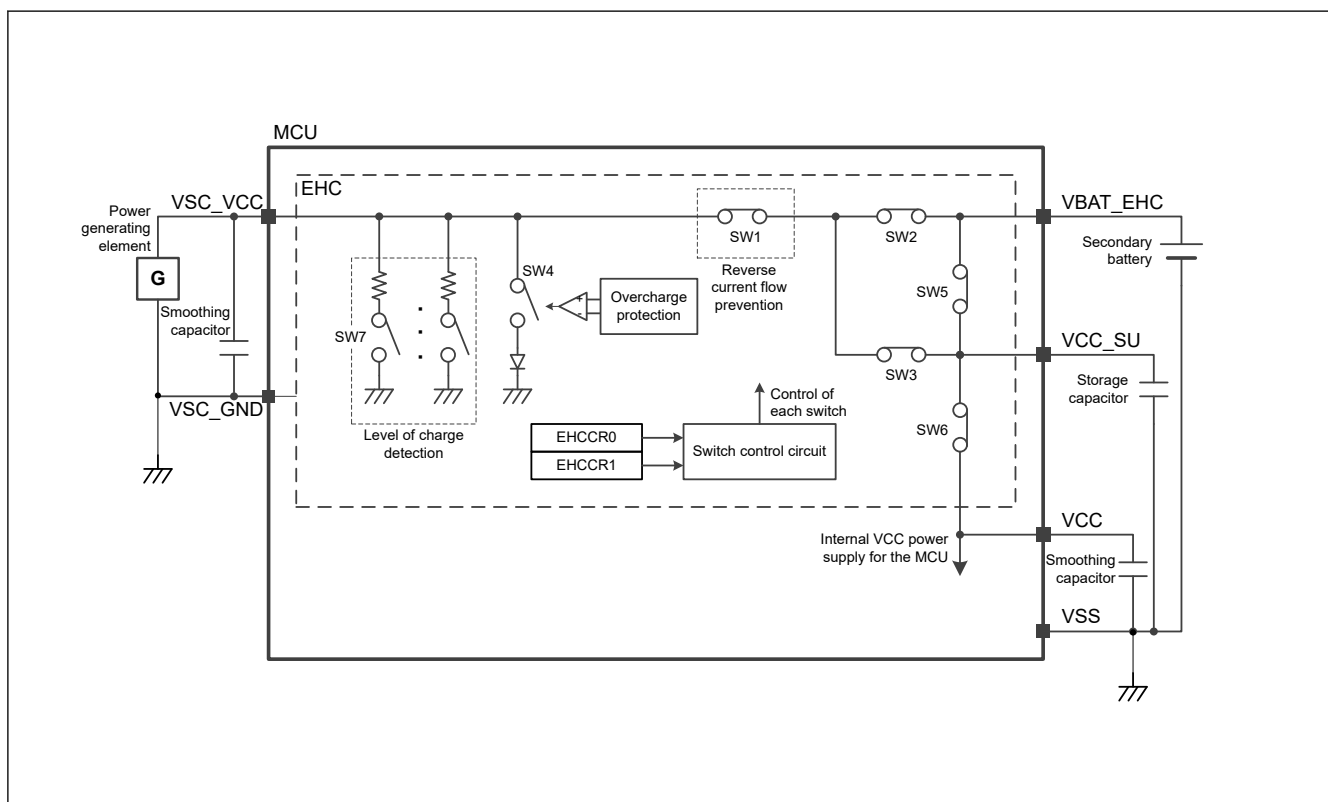


Figure 14.1 EHC block diagram

Table 14.2 lists the input and output pins of the EHC.

Table 14.2 Input and Output Pins of the EHC (1 of 2)

Pin name	I/O	Function
VSC_VCC	Input	Power supply pin for the voltage supplied by the power generating element. Connect a 4.7- to 47-nF smoothing capacitor in parallel with the power generating element. Connect this pin to the VSC_GND pin if the EHC is not used.

Table 14.2 Input and Output Pins of the EHC (2 of 2)

Pin name	I/O	Function
VSC_GND	Input	Ground pin. Connect this to the equivalent pin of the system power supply (0 V).
VCC_SU	I/O	Pin for the supply of power from a storage capacitor. When using a photovoltaic cell as a power generating element, connect a storage capacitor with a capacitance value in accordance with an operating temperature, and with a value of at least 10 times VCC. A capacitance value of 47 μ F is required at 25°C. As a temperature becomes higher, a larger capacitance value is required. See the EHC characteristics in section 51.9. EHC Characteristics . Connect this pin to a 100- μ F storage capacitor in the case where other power generating elements are used.
VBAT_EHC	Input	Pin for the supply of power from a secondary battery. In Energy Harvest start mode, connect 2.4V, 2.5V, 2.6V, 2.7V, 2.8V, 2.9V, 3.0V, 3.1V secondary battery or super capacitor.
VCC, IOVCC0/1	I/O	Power supply pin. Connect this to VSS through a 0.1- μ F smoothing capacitor (1). Place the smoothing capacitor close to this pin. In addition, connect this to VSS through a smoothing capacitor (2) having capacity of at least 1/10 of capacity of a storage capacitor connected to the VCC_SU pin to improve robustness against external noise and obtain stable operation of the circuit. For instance, connect a 4.7- μ F smoothing capacitor in the case where a 47- μ F storage capacitor is connected to the VCC_SU pin. If placing the smoothing capacitor (2) close to this pin is possible, the smoothing capacitor (1) is not required. For more details, see Appendix B. Connecting the Capacitors to the Power Supply Pins.
VSS	Input	Ground pin. Connect this to the equivalent pin of the system power supply (0 V).

14.2 Register Descriptions

14.2.1 EHCCR0 : EHC Control Register 0

Base address: EHC = 0x4001_E180

Offset address: 0x00

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ENOUT	CMPOUT	—	FORCEFIX	SWSELECT	VBATCTL	—	CHGDETEN	—	RSEL[5:0]					
Value after reset:	0	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	RSEL[5:0]	Level of Power Generation Detecting Resistance Select 0x00: Not in use 0x01: 100 Ω 0x02: 1 k Ω 0x04: 10 k Ω 0x08: 100 k Ω 0x10: 1 M Ω 0x20: 2 M Ω Others: Settings prohibited	W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	CHGDETEN	Power Generating Element Level of Power Generation Detection Enable 0: Disable detection of the level of power generation from the power generating element connected to the VSC_VCC pin 1: Enable detection of the level of power generation from the power generating element connected to the VSC_VCC pin	W
8	—	This bit is read as 0. The write value should be 0.	R/W
9	VBATCTL	VBAT_EHC Charge Control 0: Do not charge the secondary battery connected to the VBAT_EHC pin 1: Charge the secondary battery connected to the VBAT_EHC pin	W
10	SWSELECT	SW Select 0: Connect the VSC_VCC pin to the VCC_SU pin 1: Connect the VSC_VCC pin to the VBAT_EHC pin.	W

Bit	Symbol	Function	R/W
11	FORCEFIX	Forced Fix SW 0: Disable control by the SWSELECT bit 1: Enable control by the SWSELECT bit.	W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	CMPOUT	Charging Target Monitoring Flag 0: The storage capacitor connected to the VCC_SU pin from the power generating element connected to the VSC_VCC pin is being charged. 1: The secondary battery connected to the VBAT_EHC pin from the power generating element connected to the VSC_VCC pin is being charged.	R
14	ENOUT	Power Generating Element Status Flag 0: The power generating element connected to the VSC_VCC pin is not generating power. 1: The power generating element connected to the VSC_VCC pin is generating power.	R
15	—	This bit is read as 0. The write value should be 0.	R/W

Before writing to the EHCCR0 register, release the registers from protection by writing 1 to the PRCR.PRC1 bit.

The EHCCR0 register contains bits that are write-only. The read values of these write-only bits are reset values regardless of the previous settings. Therefore, the result of the logical operation on the read value should not be directly written to this register.

Setting example: EHC → EHCCR0 = 'setting value';

RSEL[5:0] bits (Level of Power Generation Detecting Resistance Select)

RSEL[5:0] bits select the resistance for use in detecting the level of power generation. Figure 14.2 shows how the resistance is selected when the RSEL[5:0] setting is 0x01. For how to use level of power generation detection, see [section 14.3.7. Level of Power Generation Detection](#).

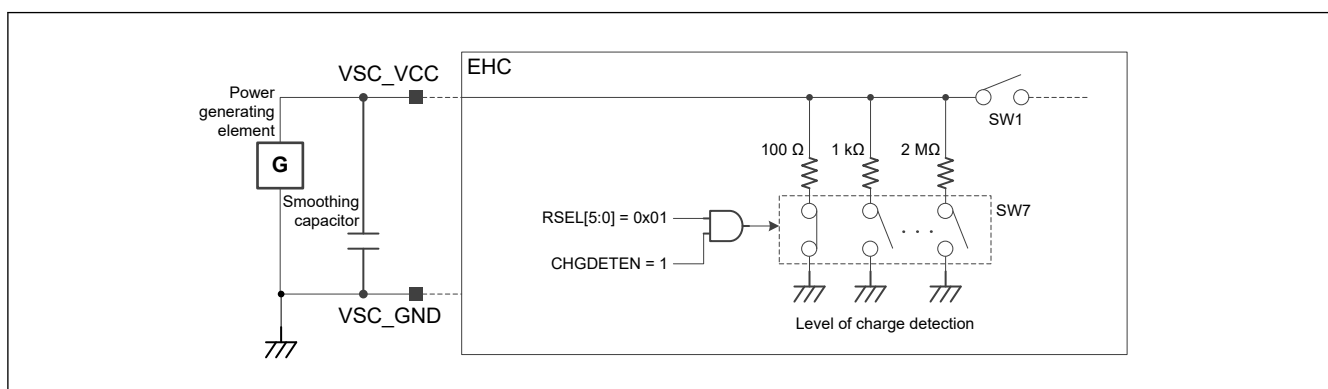


Figure 14.2 Example of selecting the resistance for use in detecting the level of power generation (when detection of the level of power generation is enabled and 100 Ω is selected)

CHGDETEN bit (Power Generating Element Level of Power Generation Detection Enable)

The CHGDETEN bit enables or disables detection of the level of power generation from the power generating element connected to the VSC_VCC pin. Writing 1 to the CHGDETEN bit enables detection of the level of power generation from the power generating element connected to the VSC_VCC pin. When detection of the level of power generation from the power generating element is enabled, the VBAT_EHC and VCC_SU pins are separated from the VSC_VCC pin and the power generation element is connected to the resistance selected by the RSEL[5:0] bits for detecting the level of power generation.

VBATCTL bit (VBAT_EHC Charge Control)

The VBATCTL bit enables or disables charging of the secondary battery connected to the VBAT_EHC pin from the power generating element connected to the VSC_VCC pin. The value set in the VBATCTL bit is only valid when the FORCEFIX

and the EHCCR1.QUICKMODE bits are set to 0. When the VBATCTL and the EHCCR0.CMPIOUT bits are set to 1, the secondary battery is charged from the VSC_VCC pin.

SWSELECT bit (SW Select)

The SWSELECT bit compulsorily assigns the connection destination of the VSC_VCC pin during the secondary battery charging period (EHCCR1.QUICKMODE = 0). This bit can be set only when FORCEFIX = 1 and EHCCR1.QUICKMODE = 0 to provide the estimation function.

FORCEFIX bit (Forced Fix SW)

The FORCEFIX bit enables or disables control by the SWSELECT bit. The value set by the FORCEFIX bit is only valid when the EHCCR1.QUICKMODE bit is set to 0.

CMPOUT flag (Charging Target Monitoring Flag)

The CMPOUT flag indicates the target for charging by the power generating element connected to the VSC_VCC pin. This flag is only valid when the EHCCR1.QUICKMODE bit is set to 0.

[Setting condition]

- The EHCCR1.QUICKMODE bit is set to 0 and the power generating element connected to the VSC_VCC pin is charging the secondary battery connected to the VBAT_EHC pin.

[Clearing conditions]

- The power generating element connected to the VSC_VCC pin is charging the storage capacitor connected to the VCC_SU pin
- The EHCCR1.QUICKMODE bit is set to 1.

ENOUT flag (Power Generating Element Status Flag)

The ENOUT flag indicates the state of power generation by the power generating element connected to the VSC_VCC pin. For the threshold voltage to determine whether power is being generated, see [section 51, Electrical Characteristics](#).

[Setting condition]

- The power generating element connected to the VSC_VCC pin is generating power.

[Clearing condition]

- The power generating element connected to the VSC_VCC pin is not generating power.

14.2.2 EHCCR1 : EHC Control Register 1

Base address: EHC = 0x4001_E180

Offset address: 0x02

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

Bit	Symbol	Function	R/W
0	QUICKMODE	Steady Operation Setting 0: Initial charging or EHC initialization setting 1: Steady operation or EHC unused setting	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: This register is initialized by a power-on reset.

Before writing to the EHCCR1 register, release the registers from protection by writing 1 to the PRCR.PRC1 bit.

QUICKMODE bit (Steady Operation Setting)

During the initial charging period, set the QUICKMODE bit to 0 and the EHCCR0.VBATCTL bit to 1, and charge the secondary battery.

By setting this bit to 1 in steady state, the power generating element connected to the VSC_VCC pin automatically supplies power directly to the MCU and charges the secondary battery and the storage capacitor according to the current drawn in the MCU.

When the VBAT_EHC voltage decreases and the operation of the MCU stops, set this bit from 1 to 0 and initialize the EHC circuit so that the initial charging can be restarted.

When the EHC is not used, set QUICKMODE = 1 for external noise tolerance and current reduction.

14.2.3 EHCRMR : EHC resistance Monitoring Register

Address: 0x0100_8142

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:

RCFMF1[7:0]	RCFMF0[7:0]

Value after reset:

Chip-specific value	Chip-specific value

Bit	Symbol	Function	R/W
7:0	RCFMF0[7:0]	Resistance Correction Coefficient Monitoring Flags 0 0x00: 0/512 0x01: +1/512 0x02: +2/512 ⋮ 0x7F: +127/512 0x80: -128/512 0x81: -127/512 ⋮ 0xFF: -1/512	R
15:8	RCFMF1[7:0]	Resistance Correction Coefficient Monitoring Flags 1 0x00: 0/512 0x01: +1/512 0x20: +2/512 ⋮ 0x7F: +127/512 0x80: -128/512 0x81: -127/512 ⋮ 0xFF: -1/512	R

The EHCRMR register stores the correction coefficient used for correcting variations in the resistance for detecting the level of power generation. When 100 Ω, 1 kΩ, or 10 kΩ is selected by the level of power generation detecting resistance selection bits of EHC control register 0 (EHCCR0.RSEL[5:0]), use the value indicated by the RCFMF0[7:0] flags as the correction coefficient. When 100 kΩ, 1 MΩ, or 2 MΩ is selected, use the value indicated by the RCFMF1[7:0] flags as the correction coefficient.

RCFMF0[7:0] flags (Resistance Correction Coefficient Monitoring Flags 0)

RCFMF0[7:0] flags indicate the correction coefficient for the resistance used in detecting the level of power generation. A correction coefficient measured for each chip before shipment is stored in the bits. The accuracy of detection by the level of power generation detection can be improved by using this correction coefficient. Use the following equation to obtain the resistance after correction.

Resistance after correction = resistance set by EHCCR0.RSEL[5:0] bits × (1 + correction coefficient set by RCFMF0[7:0] flags)

The following shows a usage example.

When EHCCR0.RSEL[5:0] = 0x01 (100 Ω) and RCFMF0[7:0] = 0x28 (+40/512),

$$100\ \Omega \times (1 + 40/512) \approx 107.8\ \Omega$$

Therefore, the resistance for detecting the level of power generation is 107.8 Ω .

RCFMF1[7:0] flags (Resistance Correction Coefficient Monitoring Flags 1)

RCFMF1[7:0] flags indicate the correction coefficient for the resistance used in detecting the level of power generation. A correction coefficient measured for each chip before shipment is stored in the bits. The accuracy of detection by the level of power generation detection can be improved by using this correction coefficient.

Use the following equation to obtain the resistance after correction.

Resistance after correction = resistance set by EHCCR0.RSEL[5:0] bits $\times$ (1 + correction coefficient set by RCFMF1[7:0] flags)

The following shows a usage example.

When EHCCR0.RSEL[5:0] = 0x08 (100 k Ω) and RCFMF1[7:0] = 0x28 (+40/512),

$$100\ \text{k}\Omega \times (1 + 40/512) \approx 107.8\ \text{k}\Omega$$

Therefore, the resistance for detecting the level of power generation is 107.8 k Ω .

14.3 Operation

14.3.1 Basic Operation

The EHC charges a secondary battery connected to the VBAT_EHC pin and a storage capacitor connected to the VCC_SU pin with the current generated by the power generating element connected to the VSC_VCC pin. The EHC supplies power to the VCC domain in the MCU. Even when the power generating element connected to the VSC_VCC pin is generating less power than is required for the current being drawn by the MCU, the MCU can still continue to operate with the current from the secondary battery connected to the VBAT_EHC pin or the current from the storage capacitor connected to the VCC_SU pin. The EHC has several switches. Turning the switches on and off changes the state of the EHC.

Figure 14.3 shows an example of basic EHC operation.

Initial charging period

When the power generating element connected to the VSC_VCC pin starts to generate power (initial charging period), the EHC starts to charge the storage capacitor connected to the VCC_SU pin. When the voltage of VCC_SU exceeds that of VCC_SU_H, the initial charging is complete. This operation proceeds before release from the power-on reset state.

Secondary battery charging period

When the initial charging is finished, the EHC starts to supply power to the VCC domain in the MCU from the storage capacitor connected to the VCC_SU pin. A power-on reset is generated when the voltage in the VCC domain exceeds V_{POR}^{*1} . After a power-on reset and the MCU is released from the internal reset state, start the charging of the secondary battery connected to the VBAT_EHC pin from the power generating element connected to the VSC_VCC pin by setting the EHCCR0.VBATCTL bit to 1 (charging of the secondary battery connected to the VBAT_EHC pin).

If the voltage of the storage capacitor connected to the VCC_SU pin drops to the lower internal threshold voltage ($V_{CC\_SU\_L}^{*1}$) due to current drawn by the MCU while the secondary battery connected to the VBAT_EHC pin is being charged, the EHC automatically switches the target for charging from the secondary battery to the storage capacitor. When charging of the storage capacitor connected to the VCC_SU pin causes its voltage to reach the upper internal threshold voltage ($V_{CC\_SU\_H}^{*1}$), the EHC determines that charging is complete and automatically switches the target for charging to the secondary battery connected to the VBAT_EHC pin.

During the secondary battery charging period, the secondary battery connected to the VBAT_EHC pin and the storage capacitor connected to the VCC_SU pin are alternately charged. The present target for charging can be checked by reading the EHCCR0.CMPOUT flag. Use the voltage monitoring BAT circuit (LVDBAT) to detect the charging state of the secondary battery.

Figure 14.4 shows an example of the sequence of operations when charging the secondary battery connected to the VBAT_EHC pin during the secondary battery charging period.

Note 1. For the voltage detection level V_{POR} , the thresholds of the internal threshold voltages ($V_{CC\_SU\_L}$ and $V_{CC\_SU\_H}$), and the values of overcharging detection voltages (V_{CC_CHG} and V_{BAT_CHG}), see [section 51, Electrical Characteristics](#).

Steady operation period

The EHC enters the steady operation state when the EHCCR1.QUICKMODE bit is set to 1 (steady operation setting) after the secondary battery connected to the VBAT_EHC pin is charged. In the steady operation state, the EHC charges the secondary battery connected to the VBAT_EHC pin from the power generating element connected to the VSC_VCC pin and, at the same time, supplies power to the VCC area in the MCU from the secondary battery connected to the VBAT_EHC pin.

The state of charging of the secondary battery can be checked by using the 14-bit A/D converter (ADC14) and the reference voltage generation circuit (VREF) and by directly measuring the VCC pin voltage. For details, see [section 14.3.3. Checking the Voltage on the VBAT_EHC Pin](#).

The VCC pin voltage can be indirectly measured by using the voltage monitoring BAT circuit (LVDBAT) or the voltage monitoring 1 circuit instead of the 14-bit A/D converter.

Period of maintaining operation driven by the VBAT_EHC voltage

If the power generating element connected to the VSC_VCC pin stops generating power during the steady operation period, the MCU is supplied with power from the secondary battery connected to the VBAT_EHC pin for continued operation. To prevent a reverse current flow to the power generating element at this time, the EHC automatically cuts off the connections between the VSC_VCC pin and VBAT_EHC pin and between the VSC_VCC pin and VCC_SU pin. When the power generating element restarts power generation, the EHC returns to the steady operation state.

Operation stopping due to dropping of the VBAT_EHC voltage

While operation of the MCU is maintained by the secondary battery connected to the VBAT_EHC pin, the voltage of the secondary battery drops little by little as the current is drawn for this purpose. Dropping of the voltage from the secondary battery connected to the VBAT_EHC pin can be checked by the level of the LVD1 signal. Because the MCU can no longer continue to operate once the active level of the LVD1 signal is detected, stop the MCU by setting the EHCCR1.QUICKMODE bit to 0.

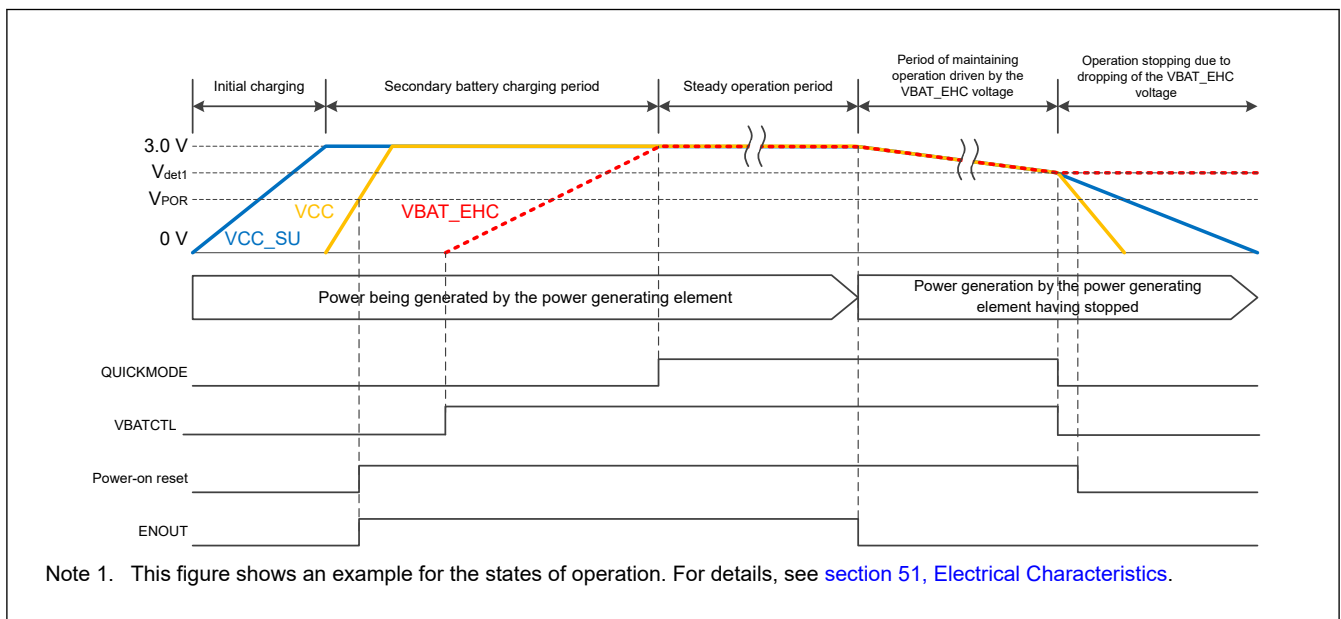


Figure 14.3 Example of EHC operations (VBAT_EHC charging voltage = 3.0 V)

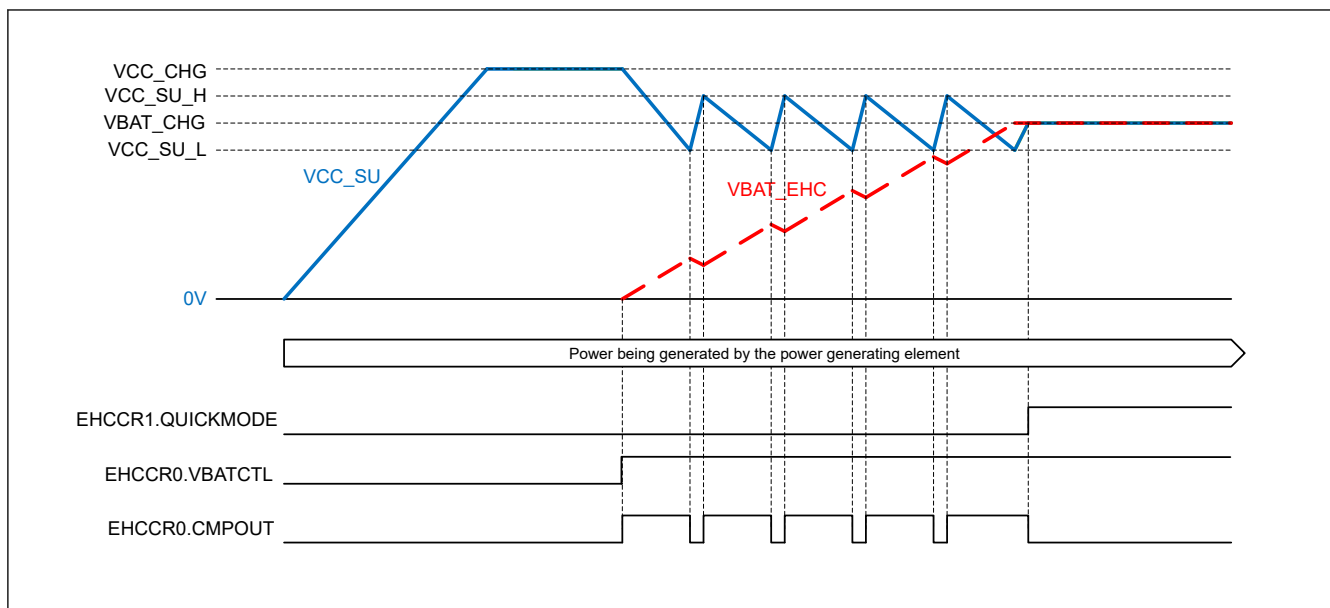


Figure 14.4 Charging operations for the VBAT_EHC pins during the secondary battery charging period

14.3.2 Initial Settings

Figure 14.5 shows the flow of initial settings when the EHC is used.

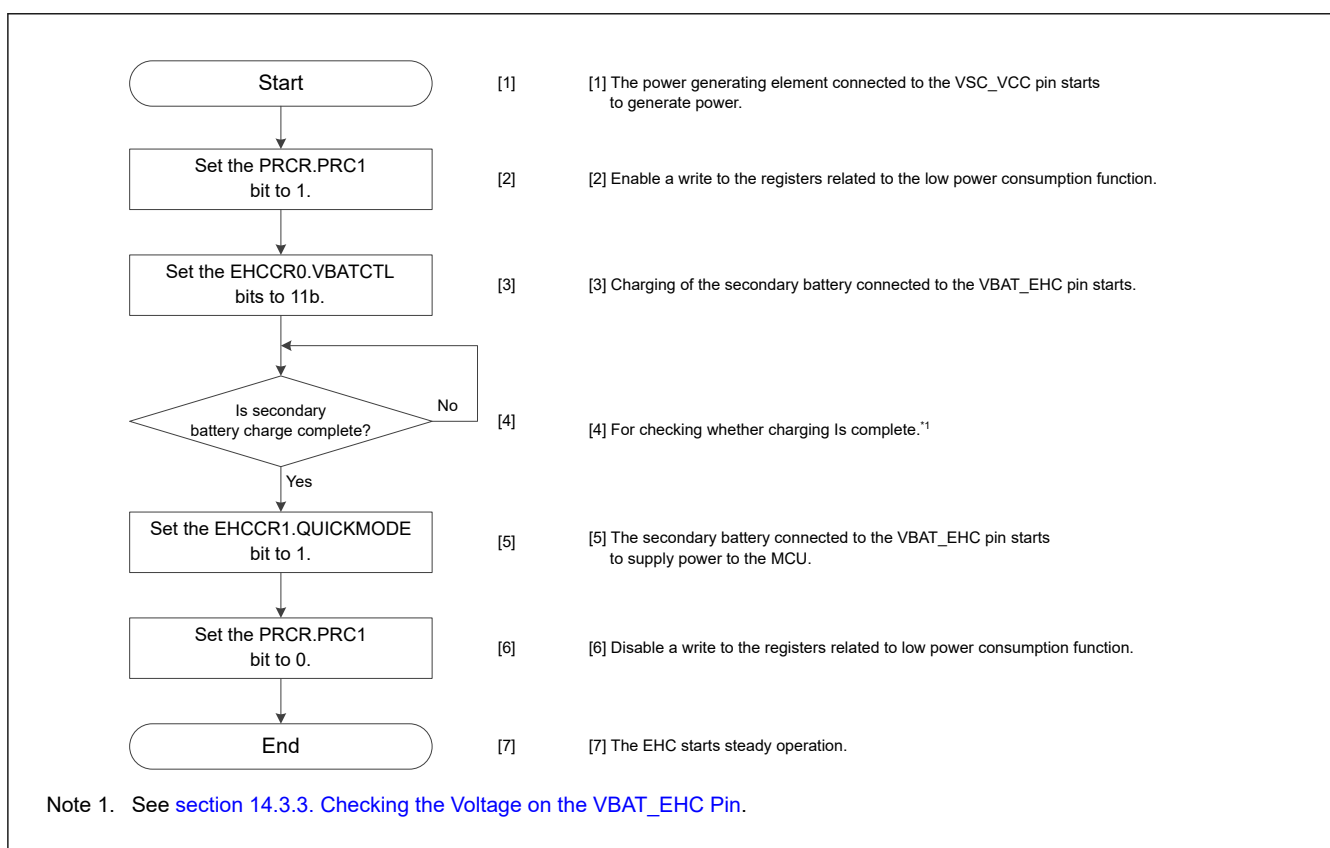


Figure 14.5 Flow of EHC initial settings

14.3.3 Checking the Voltage on the VBAT_EHC Pin

The voltage of the secondary battery connected to the VBAT_EHC pin can be checked by using the voltage monitor circuit (LVD1 or LVDBAT), 14-bit A/D converter (ADC14), and reference voltage generation circuit (VREF). Measurement by LVDBAT is effective in the system that requires a low-power operation. If the measurement with higher accuracy is

required, use the 14-bit A/D converter. Be sure to use the measurement by the voltage monitor BAT circuit while the secondary battery is being charged (EHCCR1.QUICKMODE = 0). Although the discharge cutoff voltage of the secondary battery is lower than its charging voltage (2.6 V or 3.0 V), the voltage on the VBAT_EHC pin must be divided and input to the analog input pin of the ADC14 because the VREF output voltage is 1.25 V or 2.5 V. We also recommend connecting a MOS switch to suppress the drawing of current if this check is unnecessary. Use a MOS switch with low leakage current.

Figure 14.6 and Figure 14.7 show examples of connections for checking the voltage of the secondary battery. The former is a case where the VCC (= AVCC) voltage is to be measured by resistive division and the latter is a case where the internal VREF voltage is to be measured with reference to the VCC (= AVCC) voltage. In the figure, the VCC pin is used for measurement instead of the VBAT_EHC pin because directly measuring the VBAT_EHC pin may cause some leakage current to flow from the secondary battery (in the steady operation (EHCCR1.QUICKMODE = 1), the levels of the VBAT_EHC and VCC voltage are the same). Note the following points regarding measurement of the voltage of the secondary battery:

- Wait for about 0.001 seconds till the voltage of AN0xx becomes stable after turning the MOSFET on.
- Measurement of VCC must proceed while the secondary battery is not being charged. Take the following two steps to stop the charging.
 1. Set the EHCCR0.RSEL[5:0] bits to 0x01 (selecting 100 Ω).
 2. Set the EHCCR0.CHGDETEN to 1 (enabling detection of the current being driven by the power generating element).
- Use an ADC14 input port AN0xx at high precision.

For details on the ADC14 and VREF circuits, see [section 44, 14-Bit A/D Converter \(ADC14\)](#) and [section 45, Reference Voltage Generation Circuit \(VREF\)](#).

This check is not required if sufficient charging time can be secured by estimating the charging time based on the current being driven by the power generating element connected to the VSC_VCC pin and the capacity of the secondary battery.

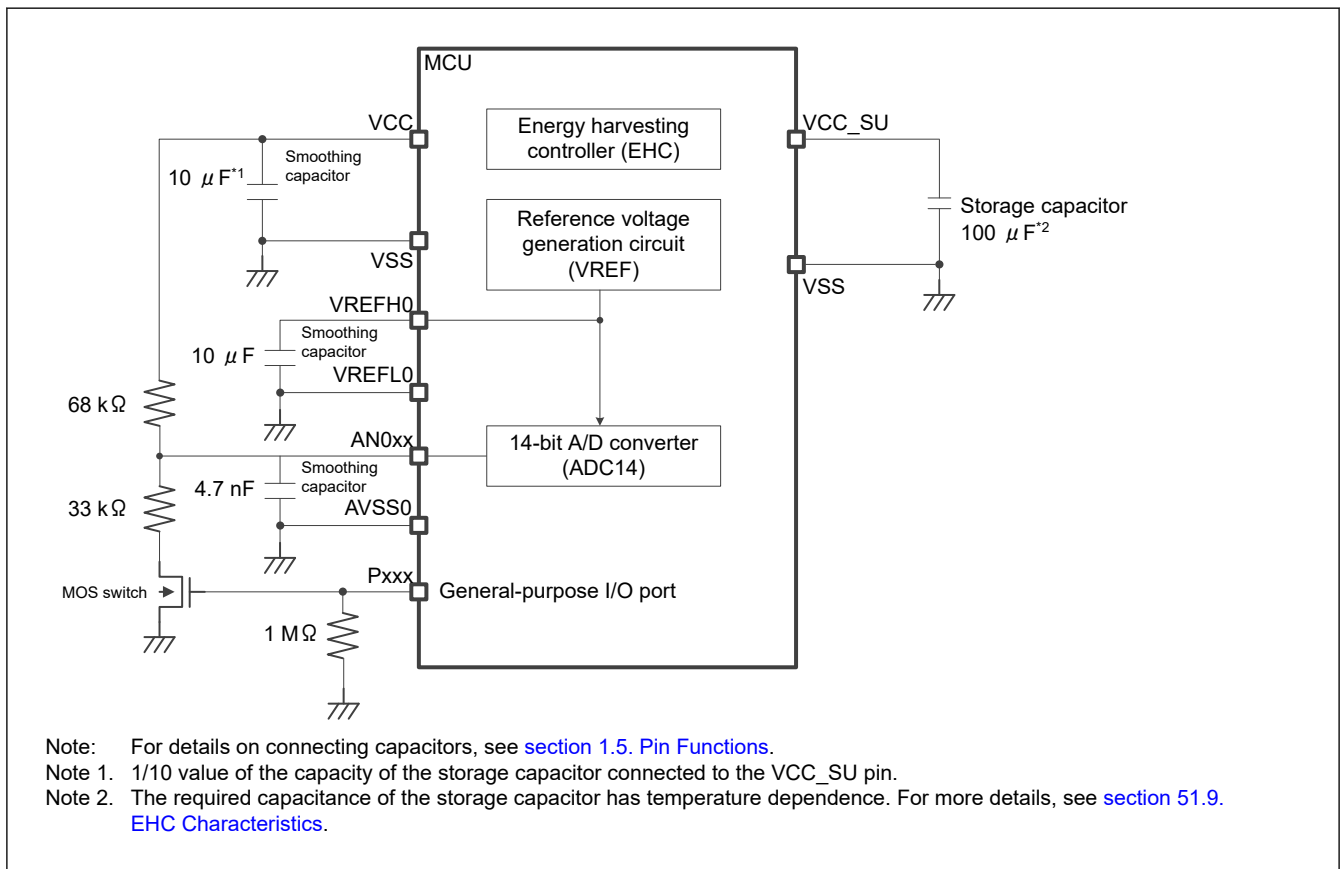


Figure 14.6 Example 1 of connections for checking the voltage on the VBAT_EHC pin

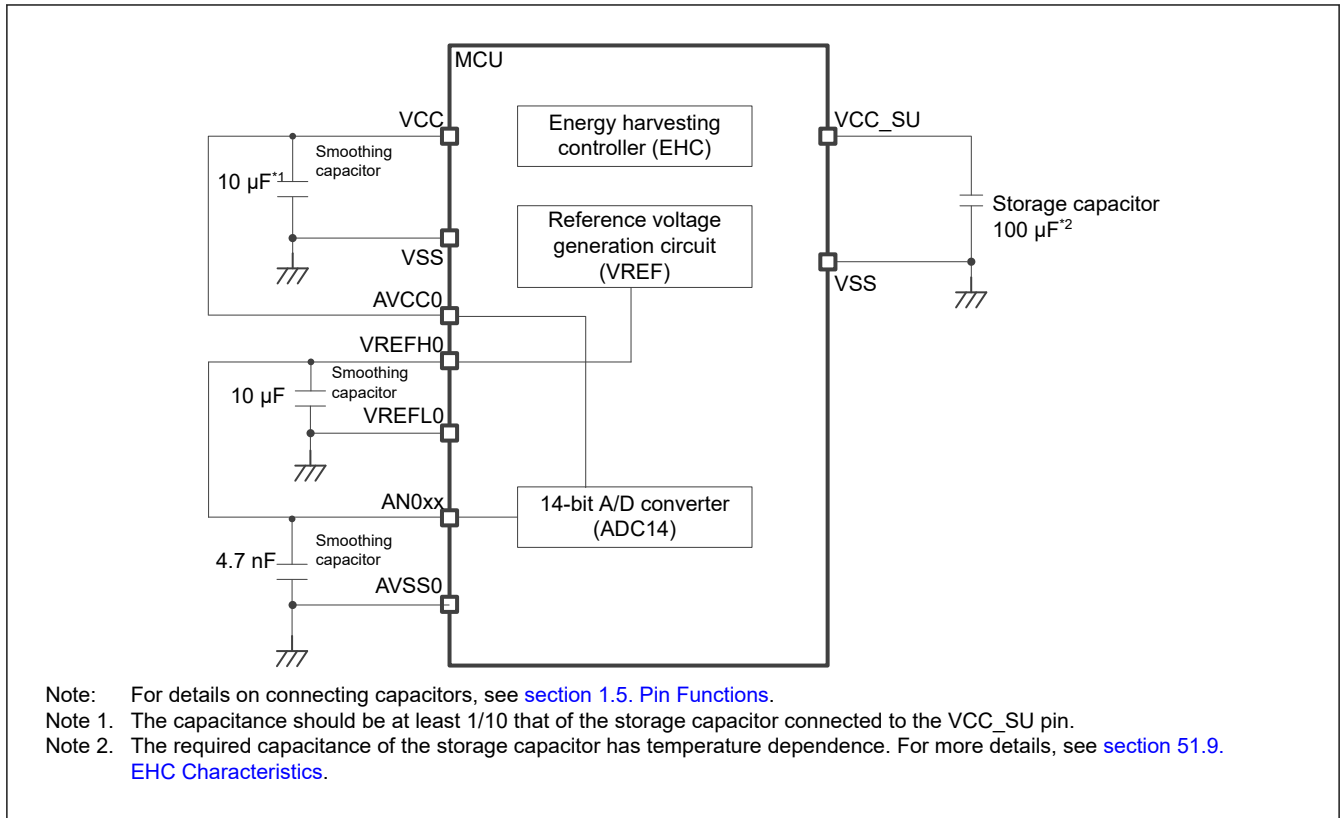


Figure 14.7 Example 2 of Connections for Checking the Voltage on the VBAT_EHC Pin

14.3.4 Processing in Response to Excessive Dropping of the Secondary Battery Voltage

If the power generating element connected to the VSC_VCC pin stops generating power during the steady operation period, the MCU continues to operate with power supplied from the secondary battery connected to the VBAT_EHC pin. If the power generating element stops generating power for a long enough time, the voltage of the secondary battery drops little by little as current is drawn by the MCU operation. Typically, the secondary battery becomes excessively discharged if it discharges more than a predetermined amount of current, which might affect the battery life. For this reason, if the voltage of the secondary battery drops below a predetermined value, stop the MCU in such cases. Since the voltage of the secondary battery connected to the VBAT_EHC pin is equivalent to the VCC voltage, a voltage drop can be detected by the output signal of the voltage monitor 1 circuit.

If the LVD1SR.MON flag indicates that the VBAT_EHC pin voltage has dropped, set the EHCCR1.QUICKMODE bit to 0 to stop the MCU. When the QUICKMODE bit is set to 0, all EHC registers are initialized. This cuts off the supply of power to the VCC domain in the MCU. To prevent too abrupt of a drop in the voltage to the VCC domain in the MCU, place the MCU in a low power state by changing the settings as follows before setting the QUICKMODE bit to 0:

- Lower the clock frequency to 2 MHz or less (it is recommended that the clock frequency is 32.768 kHz.).
- Set the Module Stop Control Registers n (MSTPCRn, n = A to D) to halt the peripheral modules.

For details of the settings, see [section 9, Clock Generation Circuit](#) and [section 13, Power-Saving Functions](#).

If the QUICKMODE bit is set to 0, the MCU does not resume operation even when the bit is set to 1 again, regardless of the level of the output signal of the voltage monitoring 1 circuit. Restart it from the initial charging state.

14.3.5 Secondary Battery Overcharging Protection

The EHC has a secondary battery overcharge protection circuit to prevent overcharging of the secondary battery connected to the VBAT_EHC pin and the storage capacitor connected to the VCC_SU pin by the power generating element connected to the VSC_VCC pin. The secondary battery overcharge protection circuit has an internal comparator that automatically detects the voltage of the secondary battery or the storage capacitor exceeding a predetermined value and, if it does, passes the charging current supplied from the power generating element to VSS.

14.3.6 Reverse Current Flow Prevention

While the power generating element connected to the VSC_VCC pin is not generating power, the voltage on the VSC_VCC pin becomes lower than the voltage of the secondary battery connected to the VBAT_EHC pin or the storage capacitor connected to the VCC_SU pin. Accordingly, current may flow in the opposite direction such as from the secondary battery or the storage capacitor to the power generating element. The EHC therefore has an internal comparator and switches to automatically prevent such reverse current flows.

14.3.7 Level of Power Generation Detection

When the power generating element is a constant current source such as a solar cell, the power generation can be measured by changing the resistance of the VSC_VCC pin.

For power generation detection of the power generating element, two methods are used:

- Directly measure the VSC_VCC pin voltage by the 14-bit A/D converter
- Confirm the Power Generating Element Status flag (EHCCR0.ENOUT).

14.3.7.1 Measurement with High Accuracy by 14-bit A/D Converter

Set the EHCCR0.CHGDETEN bit to 1 (enabling power generation detection of the power generation amount element connected to the VSC_VCC pin), and set the EHCCR0.RSEL[5:0] bits (Level of Power Generation Detecting Resistance Select bits) as the VSC_VCC pin voltage $0.1\text{ V} < \text{VSC\_VCC} < 1.0\text{ V}$. The accuracy of detecting resistance for power generation detection can be improved by using the correction coefficient indicated by the EHC Resistance Monitoring Register (EHCRMR). The built-in 14-bit A/D converter has an analog input path that can directly measure the VSC_VCC pin voltage. When the output voltage (AVTRO) of the reference voltage generation circuit is set to 1.25 V and the AVTRO is used as the reference voltage, the power generation current can be detected.

The current generated by the power generating element can be calculated by the following equation when using the resistance value correction coefficient and executing 14-bit A/D conversion.

Generated current =

$$\frac{\text{14-bit A/D conversion result}}{16384} \times \frac{\text{Output voltage of reference voltage generation circuit}}{\text{Resistance for the power generation amount detection} \times (1 + \text{Resistance correction coefficient})}$$

14-bit A/D conversion result: the ADVSCR.AD[13:0] value

Voltage output from the reference voltage generating circuit: 1.25 V

Resistance for the power generation amount detection: Resistance set by EHCCR0.RSEL[5:0] bits

Resistance correction coefficient: Correction coefficient indicated by EHCRMR.RCFMF[n][7:0] flag (n = 0, 1)

For details on the 14-bit A/D converter and the reference voltage generation circuit, see [section 44, 14-Bit A/D Converter \(ADC14\)](#) and [section 45, Reference Voltage Generation Circuit \(VREF\)](#).

When the power generation amount detection function is enabled (EHCCR0.CHGDETEN = 1), the secondary battery connected to the VBAT_EHC pin and the storage capacitor connected to the VCC_SU pin cannot be charged. Use this function in the state where the secondary battery and the storage capacitor are fully charged.

14.3.7.2 Simple Measurement by ENOUT Flag

Inside the EHC circuit, the comparator is built to compare the VSC_VCC pin voltage and the voltage that lowers the VCC_SU pin voltage to 0.146 times. The comparison result of the comparator is output to the Power Generating Element Status flag (EHCCR0.ENOUT). The value of the EHCCR0.ENOUT flag is based on the calculation result of the following I_{VSC} relationship and outputs the following comparison determination result of the power generation amount.

- ENOUT = 0: power generation is less than I_{VSC}
- ENOUT = 1: power generation is more than or equal to I_{VSC}

$$I_{VSC} = \frac{0.146 \times \text{VCC\_SU pin voltage}}{\text{Resistance for power generation amount detection}^{*1} \times (1 + \text{Resistance correction coefficient}^{*2})}$$

Note 1. Resistance value selected by the EHCCR0.RSEL[5:0] bits.

Note 2. EHCRMR value (fixed value per chip)

Table 14.3 Power generation using ENOUT pin

Value of RSEL[5:0] bits		Power generation
0x01	100 Ω	5 mA or more
0x02	1 k Ω	500 μ A to 5 mA
0x04	10 k Ω	50 μ A to 500 μ A
0x08	100 k Ω	5 μ A to 50 μ A
0x10	1 M Ω	500 nA to 5 μ A
0x20	2 M Ω	250 nA to 500 nA

14.4 Interrupt Sources

The EHC has two interrupt sources: secondary battery charging detection and storage capacitor charging detection. [Table 14.4](#) lists details of these interrupts.

Table 14.4 Interrupt sources

Interrupt name	Interrupt source	Interrupt for CPU	Activation of DMAC	Activation of DTC
SOL_DH	Secondary battery charging detection	Possible	Not Possible	Not possible
SOL_DL	Storage capacitor charging detection	Possible	Not Possible	Not possible

14.4.1 Secondary Battery Charging Detection Interrupt

This interrupt is triggered on detection of connection to the secondary battery of the power generating element connected to the VSC_VCC pin. The EHCCR0.CMPOUT bit is the interrupt flag. So the EHCCR0.CMPOUT bit being 1 triggers the interrupt.

This interrupt only operates when the EHCCR1.QUICKMODE bit is set to 0.

14.4.2 Storage Capacitor Charging Detection

This interrupt is triggered on detection of connection to the storage capacitor of the power generating element connected to the VSC_VCC pin. The EHCCR0.CMPOUT bit is the interrupt flag. So the EHCCR0.CMPOUT bit being 1 triggers the interrupt.

This interrupt only operates when the EHCCR1.QUICKMODE bit is set to 0.

14.5 Usage Notes

14.5.1 Register Write Protection

The registers described in this chapter are protected by the register write protection function. To access the registers, cancel the protection. For details, see [section 15, Register Write Protection](#).

14.5.2 Handling of the VCC, IOVCC0/1, and AVCC0 Pins when using the EHC

Do not apply power to the VCC pin when using the EHC. A smoothing capacitor with a value from 1 to 10 μ F should be connected between the VCC and VSS pins.

The EHC does not supply power to the IOVCC0/1, and AVCC0 pins. Supply power to these pins from an external power supply or from the VCC pin. [Figure 14.8](#) shows an example of power connections when the EHC is in use.

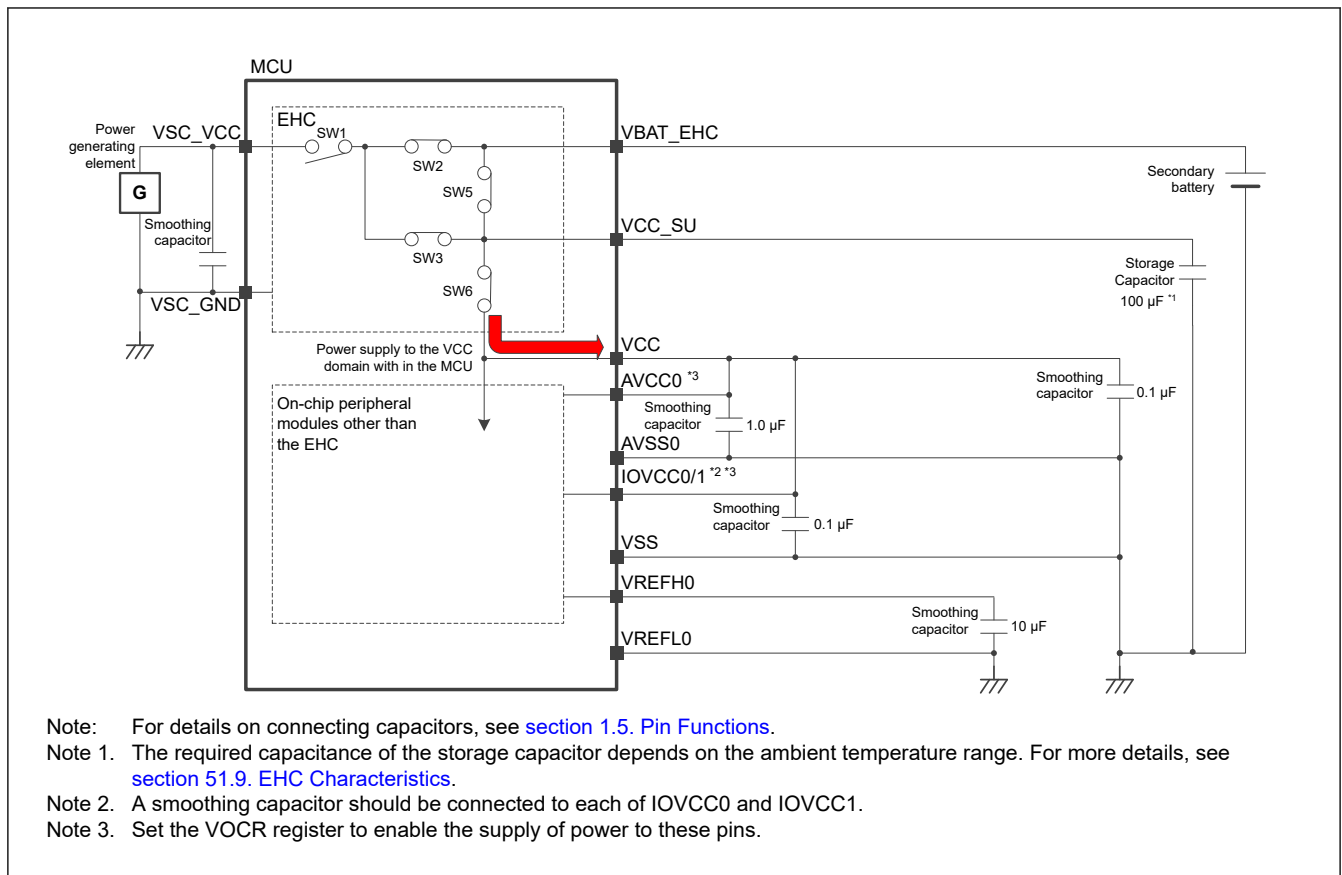


Figure 14.8 Example of power connections when the EHC is in use

14.5.3 Handling of Pins when the EHC is not Used

When the EHC is not used, connect the VSC_VCC pin to the VSC_GND pin. When the VSC_VCC pin is connected to the VSC_GND pin, the EHC stops operating. Connect the VBAT_EHC and VCC_SU pins to the VCC pin.

[Figure 14.9](#) shows an example of power connections when the EHC is not used.

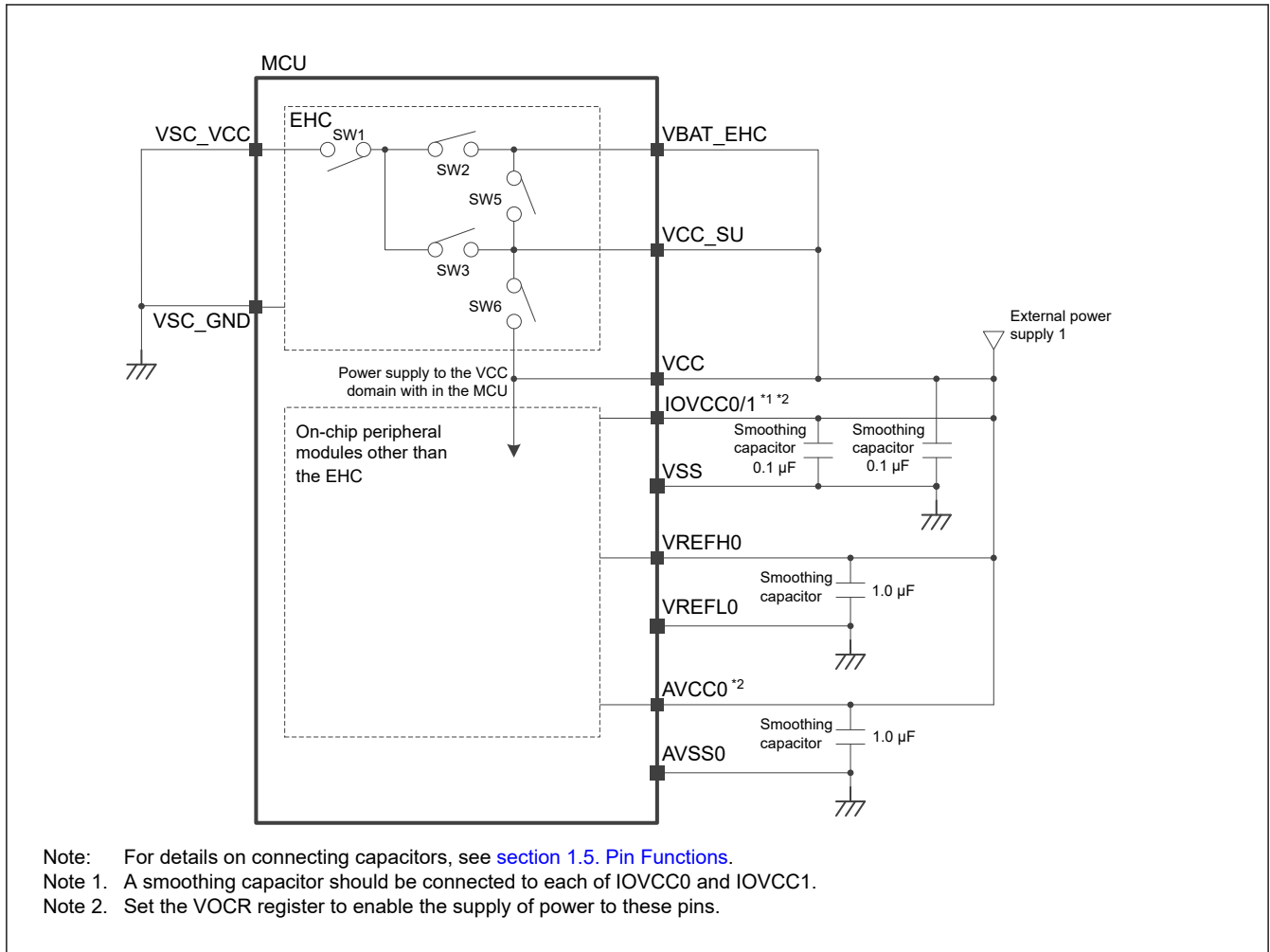


Figure 14.9 Example of power connections when the EHC is not to be used

14.5.4 Selecting a Power Generating Element

As the power generating element connected to the VSC_VCC pin, use a power generating element with an open-circuit voltage of up to 5.4 V and a short-circuit current of up to 10 mA, that satisfies the following condition:

- When the voltage of the power generating element is equal to the charging voltage of the secondary battery, the current driven by the power generating element is at least 3 μ A.

Although the upper limit on the open-circuit voltage of the power generating element is 5.4 V, use a power generating element that produces a voltage on the VSC_VCC pin which does not exceed the absolute maximum rating.

Note: For the absolute maximum rating, see [section 51, Electrical Characteristics](#).

14.5.5 Selecting a Secondary Battery

Connect a secondary battery with a charging voltage of 2.4 V or 3.1 V to the VBAT_EHC pin. Set the OFS1.VATSEL[2:0] bits of the option-setting memory to correspond to the given charging voltage. For details, see [section 7.2.2. OFS1 : Option Function Select Register 1](#).

14.5.6 Selecting a Secondary Battery when the MLCD is Used

The MIP liquid crystal display controller (MLCD) requires an MLCD pin output voltage of at least 2.7 V. Consequently, when using the EHC to supply power to the VCC domain in the MCU from the secondary battery connected to the VBAT_EHC pin and the MLCD is used, connect a secondary battery with a charging voltage of at least 2.7 V to the VBAT_EHC pin.

14.5.7 Setting the Frequency for the Secondary Battery Charging Period

For the secondary battery charging period, set a clock frequency no greater than 2 MHz to reduce the current drawn by the storage capacitor connected to the VCC_SU pin.

There is no limitation on the clock frequency after the EHC enters the steady operation period. The time during which the MCU can continue to operate depends on:

- The states of the MCU operation
- The amount and state of the power generated by the power generating element connected to the VSC_VCC pin
- The discharge capacity of the secondary battery connected to the VBAT_EHC pin.

Use the EHC after carefully evaluating these conditions.

14.5.8 Reset range

EHC has a different reset range depending on a reset factor. See to [Table 14.5](#).

Table 14.5 Reset factor and initialization range

Reset factor	SW1	SW2	SW3	SW4	SW5	SW6	SW7	Initialized register
Reset by RES# pin (When QIOCKMIDE = 0)	On	Off	On	Off	Same as before	Same as before	Off	EHCCR0
Reset by RES# pin (When QIOCKMIDE = 1)	On	On	On	Off	Same as before	Same as before	Off	EHCCR0
Power on reset	On	Off	On	Off	Off	Same as before	Off	EHCCR0/1
EHCCR1.QUICKMODE change from 1 to 0	On	Off	On	Off	Off	Off	Off	EHCCR0/1

15. Register Write Protection

15.1 Overview

The register write protection function protects important registers from being overwritten due to software errors. The registers to be protected are set with the Protect Register (PRCR).

Table 15.1 lists the association between the bits in the PRCR register and the registers to be protected.

Table 15.1 Association between the bits in the PRCR register and registers to be protected

PRCR bit	Register to be protected
PRC0	Registers related to the clock generation circuit: SCKDIVCR, SCKSCR, MOSCCR, HOCOCR, HOCOMCR, MOCOCR, FLLCR, FLLCR2, CKOCR, CKO32CR, OSTDCR, OSTDSR, MOSCWTCR, MOMCR, SOSCCR, SOMCR, LOCOCR
PRC1	Registers related to the functions for reducing power consumption: SBYCR, RAMSDCR, SNZCR, SNZEDCR, SNZREQCR, OPCCR, DPSBYCR, DPSIER0/1, DPSIFR0/1, DPSIEGR0/1, SYOCDCR, FSTPCR, EHCCR0/1, VOCCR, PWSTCR, VBBCR
PRC3	Registers related to the LVD: LVD1CR1, LVD1SR, LVDBATCR1, LVDBATSR, LVCMPCR, LVDLVL, LVD1CR0, LVDBATCR0
PRC4	Registers related to the functions for reducing power consumption: LDOCR

15.2 Register Descriptions

15.2.1 PRCR : Protect Register

Base address: SYSC = 0x4001_E000

Offset address: 0x3FE

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								—	—	—	PRC4	PRC3	—	PRC1	PRC0

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	PRC0	Enable writing to the registers related to the clock generation circuit 0: Disable writes 1: Enable writes	R/W
1	PRC1	Enable writing to the registers related to the functions for reducing power consumption 0: Disable writes 1: Enable writes	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	PRC3	Enable writing to the registers related to the LVD 0: Disable writes 1: Enable writes	R/W
4	PRC4	Enables writing to the LDOCR register 0: Disable writes 1: Enable writes	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	PRC Key Code These bits control the write access to the PRCR register. To modify the PRCR register, write 0xA5 to the upper 8 bits and the target value to the lower 8 bits as a 16-bit unit.	W

PRCn bits (Protect bit n) (n = 0, 1, 3, 4)

The PRCn bits enable or disable writing to the protected registers listed in [Table 15.1](#). Setting the PRCn bits to 1 enables writing, and 0 disables writing.

The register controlled by PRC4 may not reflect the PRC4 change when PRCR and its controlled registers are continuously written access. Avoid continuous write access or read the PRCR after PRC4 change, and then write access the PRC4-controlled register.

16. Interrupt Controller Unit (ICU)

16.1 Overview

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

[Table 16.1](#) lists the ICU specifications.

Table 16.1 ICU specifications

Parameter		Description
Maskable interrupts	Peripheral function interrupts	- Interrupts from peripheral modules - Number of sources: 168
	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 10 sources, with interrupts from IRQi (i = 0 to 9) pins.
	Interrupt requests to CPU (NVIC)	32 interrupt requests are output to NVIC. - Maskable interrupt sources are classified into 8 groups, and one source can be selected individually from 31 sources that are classified into groups.
	DMAC control	- The DMAC can be activated using interrupt sources^{*1} - The target interrupt source can be selected individually for every DMAC channels.
	DTC control	- The DTC can be activated using interrupt sources^{*1} - The method for selecting an interrupt source is the same as that of the interrupt request to NVIC.
Non-maskable interrupts ^{*2}	NMI pin interrupt	- Interrupt from the NMI pin - Interrupt detection on falling edge or rising edge
	WDT underflow/refresh error ^{*3}	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	IWDT underflow/refresh error ^{*3}	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	Low voltage detection 1 ^{*3}	Voltage monitor 1 interrupt of the voltage monitor 1 circuit (LVD_LVD1)
	Low voltage detection BAT ^{*3}	Voltage monitor BAT interrupt of the voltage monitor BAT circuit (LVD_LVDBAT)
	Bus slave MPU error	Interrupt on MPU bus slave error
	Bus master MPU error	Interrupt on MPU bus master error
	CPU stack pointer monitor error	Interrupt on CPU stack pointer monitor error
	Oscillation stop detection interrupt ^{*3}	Interrupt on detecting that the main oscillation has stopped
Low power modes		- Sleep mode: return is initiated by non-maskable interrupts or any other interrupt source - Software Standby mode: return is initiated by non-maskable interrupts. Interrupt can be selected in the WUPEN register. - Snooze mode: return is initiated by non-maskable interrupts. Interrupt can be selected in the SELSR0 and WUPEN registers. See section 16.2.8. SELSR0 : SYS Event Link Setting Register and section 16.2.9. WUPEN : Wake Up Interrupt Enable Register.

Note 1. For the DMAC and DTC activation sources, see [Table 16.4](#).

Note 2. Non-maskable interrupts can be enabled only once after a reset release. For details, see [section 16.2.3. NMIER : Non-Maskable Interrupt Enable Register](#)

Note 3. These non-maskable interrupts can also be used as maskable interrupts. When used as maskable interrupts, do not change the value of the NMIER register from the reset state. To enable voltage monitor 1 and voltage monitor BAT interrupts, set the LVD1CR1.IRQSEL and LVDBATCR1.IRQSEL bits to 1.

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

[Figure 16.1](#) shows the ICU block diagram.

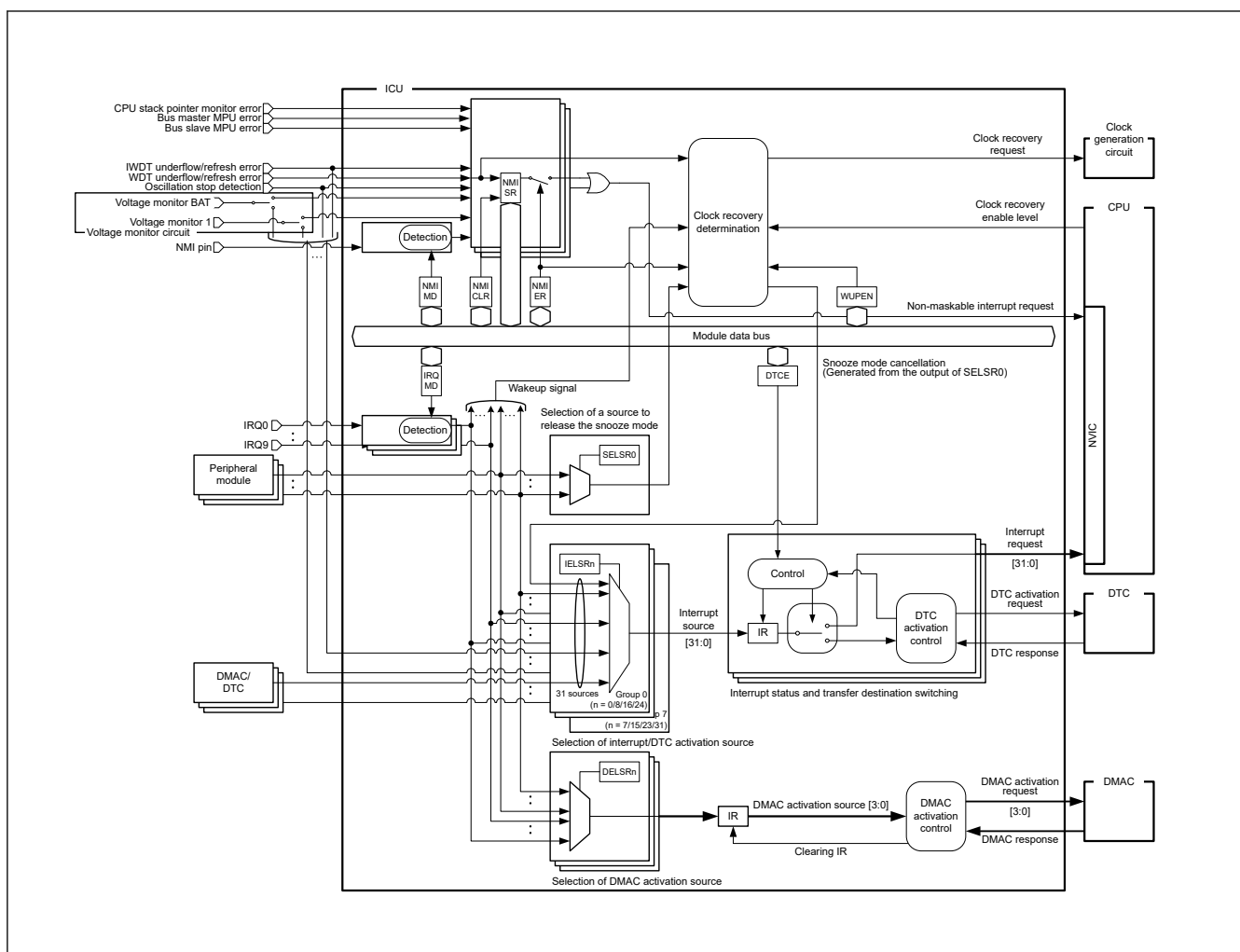


Figure 16.1 ICU block diagram

Table 16.2 lists the ICU input/output pins.

Table 16.2 ICU I/O pins

Pin name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQi (i = 0 to 9)	Input	External interrupt request pins

16.2 Register Descriptions

This chapter does not describe the Arm[®] NVIC internal registers. For information about these registers, see Cortex[™]-M0+ Technical Reference Manual (ARM DDI 0484C).

16.2.1 IRQCRi : IRQ Control Register (i = 0 to 9)

Base address: ICU = 0x4000_6000

Offset address: 0x000 + 0x1 × i

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	IRQMD[1:0]

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
1:0	IRQMD[1:0]	IRQi Detection Sense Select 0 0: Falling edge 0 1: Rising edge 1 0: Rising and falling edges 1 1: Low level	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

IRQCRi register changes must satisfy the following conditions:

- For a CPU interrupt or DTC trigger:
Change the IRQCRi register value and then set the target IELSRn register (n = 0 to 31).
You can change the register values only when the value of the target IELSRn register is 0x0000.
- For a DMAC trigger:
Change the IRQCRi register value and then set the target DELSRn register (n = 0 to 3).
You can change the register values only when the value of the target DELSRn register is 0x0000.
- For a wakeup enable signal:
Change the IRQCRi register setting before setting the target WUPEN.IRQWUPEN[n] (n = 0 to 9). 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 more information about the settings, see [section 16.5.5. External Pin Interrupts](#).

16.2.2 NMISR : Non-Maskable Interrupt Status Register

Base address: ICU = 0x4000_6000

Offset address: 0x140

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SPES T	BUSM ST	BUSS ST	—	—	NMIST	OSTS T	—	—	LVDB ATST	LVD1S T	WDTS T	IWDT ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTST	IWDT Underflow/Refresh Error Status Flag 0: Interrupt not requested 1: Interrupt requested	R
1	WDTST	WDT Underflow/Refresh Error Status Flag 0: Interrupt not requested 1: Interrupt requested	R
2	LVD1ST	Voltage Monitor 1 Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
3	LVDBATST	Voltage Monitor BAT Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
5:4	—	These bits are read as 0.	R
6	OSTST	Oscillation Stop Detection Interrupt Status Flag 0: Interrupt not requested for main oscillation stop 1: Interrupt requested for main oscillation stop	R
7	NMIST	NMI Status Flag 0: Interrupt not requested 1: Interrupt requested	R
8	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
9	—	This bit is read as 0.	R
10	BUSST	MPU Bus Slave Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested.	R
11	BUSMST	MPU Bus Master Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
12	SPEST	CPU Stack Pointer Monitor Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
13	—	This bit is read as 0.	R
14	—	This bit is read as 0.	R
15	—	This bit is 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.

LVDBATST flag (Voltage Monitor BAT Interrupt Status Flag)

The LVDBATST flag indicates a request for voltage monitor BAT interrupt. It is read-only and cleared by the NMICLR.LVDBATCLR bit.

[Setting condition]

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

[Clearing condition]

When 1 is written to the NMICLR.LVDBATCLR bit.

OSTST flag (Oscillation Stop Detection Interrupt Status Flag)

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

[Setting condition]

When the main oscillation stop detection interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.OSTCLR bit.

NMIST flag (NMI Status Flag)

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

[Setting condition]

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

[Clearing condition]

When 1 is written to the NMICLR.NMICLR bit.

BUSST flag (MPU Bus Slave Error Interrupt Status Flag)

The BUSST 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.

16.2.3 NMIER : Non-Maskable Interrupt Enable Register

Base address: ICU = 0x4000_6000

Offset address: 0x120

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SPEE N	BUSM EN	BUSS EN	—	—	NMIE N	OSTE N	—	—	LVDB ATEN	LVD1E N	WDTE N	IWDT EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTEN	IWDT Underflow/Refresh Error Interrupt Enable 0: Disabled 1: Enabled.	R/W ^{*1} *2
1	WDTEN	WDT Underflow/Refresh Error Interrupt Enable 0: Disabled 1: Enabled	R/W ^{*1} *2
2	LVD1EN	Voltage monitor 1 Interrupt Enable 0: Disabled 1: Enabled	R/W ^{*1} *2
3	LVDBATEN	Voltage monitor BAT Interrupt Enable 0: Disabled 1: Enabled	R/W ^{*1} *2
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	OSTEN	Oscillation Stop Detection Interrupt Enable 0: Disabled 1: Enabled	R/W ^{*1} *2
7	NMIEN	NMI Pin Interrupt Enable 0: Disabled 1: Enabled	R/W ^{*1}
8	—	This bit is read as 0. The write value should be 0.	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	BUSSEN	MPU Bus Slave Error Interrupt Enable 0: Disabled 1: Enabled	R/W ^{*1}
11	BUSMEN	MPU Bus Master Error Interrupt Enable 0: Disabled 1: Enabled	R/W ^{*1}
12	SPEEN	CPU Stack Pointer Monitor Interrupt Enable 0: Disabled 1: Enabled	R/W ^{*1}
13	—	This bit is read as 0. The write value should be 0.	R/W
15:14	—	This bit is 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.

LVDBATEN bit (Voltage monitor BAT Interrupt Enable)

The LVDBATEN bit enables voltage monitor BAT interrupt as an NMI trigger.

OSTEN bit (Oscillation Stop Detection Interrupt Enable)

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

NMIEN bit (NMI Pin Interrupt Enable)

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

BUSSEN bit (MPU Bus Slave Error Interrupt Enable)

The BUSSEN bit enables bus slave MPU error interrupt as an NMI trigger.

BUSMEN bit (MPU Bus Master Error Interrupt Enable)

The BUSMEN bit enables bus master MPU 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.

16.2.4 NMICLR : Non-Maskable Interrupt Status Clear Register

Base address: ICU = 0x4000_6000

Offset address: 0x130

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SPECLR	BUSMCLR	BUSSCLR	—	—	NMICLR	OSTCLR	—	—	LVDBATCLR	LVD1CLR	WDTCLR	IWDTCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTCLR	IWDT Clear 0: No effect 1: Clear the NMISR.IWDTST flag	R/W ¹
1	WDTCLR	WDT Clear 0: No effect 1: Clear the NMISR.WDTST flag	R/W ¹
2	LVD1CLR	LVD1 Clear 0: No effect 1: Clear the NMISR.LVD1ST flag	R/W ¹
3	LVDATCLR	LVDAT Clear 0: No effect 1: Clear the NMISR.LVDATST flag.	R/W ¹
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	OSTCLR	OST Clear 0: No effect 1: Clear the NMISR.OSTST flag	R/W ¹
7	NMICLR	NMI Clear 0: No effect 1: Clear the NMISR.NMIST flag	R/W ¹
8	—	This bit is read as 0. The write value should be 0.	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	BUSSCLR	Bus Slave Error Clear 0: No effect 1: Clear the NMISR.BUSSST flag	R/W ¹
11	BUSMCLR	Bus Master Error Clear 0: No effect 1: Clear the NMISR.BUSMST flag	R/W ¹
12	SPECLR	CPU Stack Pointer Monitor Interrupt Clear 0: No effect 1: Clear the NMISR.SPEST flag	R/W ¹
13	—	This bit is read as 0. The write value should be 0.	R/W
15:14	—	This bit is 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.

LVDBATCLR bit (LVDBAT Clear)

Writing 1 to the LVDBATCLR bit clears the NMISR.LVDBATST 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.

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.

16.2.5 NMICR : NMI Pin Interrupt Control Register

Base address: ICU = 0x4000_6000

Offset address: 0x100

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

Bit	Symbol	Function	R/W
0	NMIMD	NMI Detection Set 0: Falling edge 1: Rising edge	R/W
7:1	—	These bits are read as 0. The write value should be 0.	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.

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

Base address: ICU = 0x4000_6000

Offset address: 0x300 + 0x4 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

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

Bit	Symbol	Function	R/W
4:0	IELS[4:0]	ICU Event Link Select 0x00: Disable interrupts to the associated NVIC or DTC module Others: Event signal number to be linked. For details, see section 16.3.3. ICU and DTC Event Number .	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W
16	IR	Interrupt Status Flag 0: No interrupt request generated. 1: An interrupt request is generated.	R/W ¹
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	DTCE	DTC Activation Enable 0: DTC activation is disabled. 1: DTC activation is enabled.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: This register requires halfword or word access.

Note 1. For edge detection, only 0 can be written. Writing 1 to the IR flag is prohibited. For level detection, writing is prohibited.

The IELSRn register selects the IRQ source used by the NVIC. IELSRn, where n = 0 to 31, corresponds to the NVIC-IRQ input source numbers 0 to 31.

IELS[4:0] bits (ICU Event Link Select)

The IELS[4:0] bits link an event signal to the associated NVIC or DTC module. Event options are classified into 8 groups (groups 0 to 7). For details, see [Table 16.3](#) and [Table 16.4](#).

IR flag (Interrupt Status Flag)

The IR status flag indicates an individual interrupt request from the event specified in IELS[4:0].

[Setting condition]

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

[Clearing condition]

(1) For edge detection

When 0 is written to the IR flag. DTCE must be set to 0 before writing 0 to the IR flag.

(2) For level detection

The flag becomes 0 by clearing the interrupt request output of the peripheral module. (even if the interrupt request destination receives an interrupt request, the flag does not become 0.)

To clear an interrupt request for every peripheral module, see sections of each peripheral modules.

Note: See [Table 16.7](#) for edge detection or level detection.

DTCE bit (DTC Activation Enable)

When the DTCE bit is set to 1, the associated event is selected as the source for DTC activation.

[Setting condition]

- When 1 is written to the DTCE bit.

[Clearing condition]

- When the specified number of transfers is complete. For chain transfers, when the specified number of transfers for the last chain transfer is complete.
- When 0 is written to the DTCE bit.

16.2.7 DELSRn : DMAC Event Link Setting Register n (n = 0 to 3)

Base address: ICU = 0x4000_6000

Offset address: 0x280 + 0x4 × n

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

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

Bit	Symbol	Function	R/W
7:0	DELS[7:0]	DMAC Event Link Select 0x00: Disable interrupts to the associated DMAC module Others: Event signal number to be linked. For details, see Table 16.4 .	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	IR	DMAC Activation Request Status flag 0: No DMAC activation request occurred 1: DMAC activation request occurred.	R/W ^{*1}
31:17	—	These bits are read as 0. The write value should be 0.	R/W

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

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

The DELS[7:0] bits specify an event signal to link to the associated DMAC module. Do not set the same event number in multiple DELSRn registers.

IR flag (DMAC Activation Request Status flag)

This flag is a status flag for a DMAC activation request.

This flag is associated with the DELS[7:0] bits of this register.

[Setting condition]

When a DMAC activation request occurs from the associated peripheral module or IRQi pin.

[Clearing conditions]

- When 0 is written to the flag
- When the DMA transfer is started after a DMAC activation request occurs.

16.2.8 SELSR0 : SYS Event Link Setting Register

Base address: ICU = 0x4000_6000

Offset address: 0x200

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

Bit	Symbol	Function	R/W
7:0	SELS[7:0]	SYS Event Link Select 0x00: Disable event output to the associated low-power mode module Others: Event signal number to be linked. For details, see Table 16.4 .	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

The SELSR0 register selects the events that wake up the CPU from Snooze mode. You can use only the events listed in [Table 16.4](#) checked as “Canceling Snooze mode using SELSR0”. Events specified in this register are defined as ICU_SNZCANCEL in [Table 16.4](#). When ICU_SNZCANCEL is selected in the IELSRn.IELS[4:0] bits, the SELSR0 event interrupt occurs.

16.2.9 WUPEN : Wake Up Interrupt Enable Register

Base address: ICU = 0x4000_6000

Offset address: 0x1A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SOLD LWUP EN	AGT0 CAWU PEN	AGT1 CAWU PEN	AGT1 UDWU PEN	AGTW 1CAW UPEN	CCCP RDWU PEN	RTCP RDWU PEN	RTCA LMWU PEN	AGTW 0CAW UPEN	SOLD HWUP EN	—	—	LVDB ATWU PEN	LVD1 WUPE N	KEYW UPEN	IWDT WUPE N
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	IRQW UPEN 9	IRQW UPEN 8	IRQW UPEN 7	IRQW UPEN 6	IRQW UPEN 5	IRQW UPEN 4	IRQW UPEN 3	IRQW UPEN 2	IRQW UPEN 1	IRQW UPEN 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IRQWUPEN0	PORT_IRQ0 Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by IRQ0 interrupts 1: Enable Software Standby returns by IRQ0 interrupts	R/W
1	IRQWUPEN1	PORT_IRQ1 Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by IRQ1 interrupts 1: Enable Software Standby returns by IRQ1 interrupts.	R/W
2	IRQWUPEN2	PORT_IRQ2 Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by IRQ2 interrupts 1: Enable Software Standby returns by IRQ2 interrupts.	R/W
3	IRQWUPEN3	PORT_IRQ3 Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by IRQ3 interrupts 1: Enable Software Standby returns by IRQ3 interrupts	R/W
4	IRQWUPEN4	PORT_IRQ4 Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by IRQ4 interrupts 1: Enable Software Standby returns by IRQ4 interrupts.	R/W
5	IRQWUPEN5	PORT_IRQ5 Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by IRQ5 interrupts 1: Enable Software Standby returns by IRQ5 interrupts.	R/W
6	IRQWUPEN6	PORT_IRQ6 Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by IRQ6 interrupts 1: Enable Software Standby returns by IRQ6 interrupts.	R/W
7	IRQWUPEN7	PORT_IRQ7 Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by IRQ7 interrupts 1: Enable Software Standby returns by IRQ7 interrupts.	R/W
8	IRQWUPEN8	PORT_IRQ8 Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by IRQ8 interrupts 1: Enable Software Standby returns by IRQ8 interrupts.	R/W

Bit	Symbol	Function	R/W
9	IRQWUPEN9	PORT_IRQ9 Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by IRQ9 interrupts 1: Enable Software Standby returns by IRQ9 interrupts.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	IWDTWUPEN	IWDT_NMIUNDF Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by IWDT_NMIUNDF interrupts 1: Enable Software Standby returns by IWDT_NMIUNDF interrupts.	R/W
17	KEYWUPEN	KEY_INTKR Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by KEY_INTKR interrupts 1: Enable Software Standby returns by KEY_INTKR interrupts.	R/W
18	LVD1WUPEN	LVD_LVD1 Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by LVD_LVD1 interrupts 1: Enable Software Standby returns by LVD_LVD1 interrupts.	R/W
19	LVDBATWUPEN	LVD_LVDBAT Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by LVD_LVDBAT interrupts 1: Enable Software Standby returns by LVD_LVDBAT interrupts.	R/W
21:20	—	These bits are read as 0. The write value should be 0.	R/W
22	SOLDHWUPEN	SOL_DH Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by SOL_DH (rising of CMPOUT) interrupts 1: Enable Software Standby returns by SOL_DH (rising of CMPOUT) interrupts.	R/W
23	AGTW0CAWUPEN	AGTW0_AGTCAI Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by AGTW0_AGTCAI interrupts 1: Enable Software Standby returns by AGTW0_AGTCAI interrupts.	R/W
24	RTCALMWUPEN	RTC_ALM Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by RTC_ALM alarm interrupts 1: Enable Software Standby returns by RTC_ALM alarm interrupts.	R/W
25	RTCPRDWUPEN	RTC_PRD Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by RTC_PRD period interrupts 1: Enable Software Standby returns by RTC_PRD period interrupts.	R/W
26	CCCPRDWUPEN	CCC_PRD Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by CCC_PRD period interrupts 1: Enable Software Standby returns by CCC_PRD period interrupts.	R/W
27	AGTW1CAWUPEN	AGTW1_AGTCAI Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by AGTW1_AGTCAI interrupts 1: Enable Software Standby returns by AGTW1_AGTCAI interrupts.	R/W
28	AGT1UDWUPEN	AGT1_AGTI Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by AGT1_AGTI interrupts 1: Enable Software Standby returns by AGT1_AGTI interrupts.	R/W
29	AGT1CAWUPEN	AGT1_AGTCAI Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by AGT1_AGTCAI interrupts 1: Enable Software Standby returns by AGT1_AGTCAI interrupts.	R/W
30	AGT0CAWUPEN	AGT0_AGTCAI Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by AGT0_AGTCAI interrupts 1: Enable Software Standby returns by AGT0_AGTCAI interrupts.	R/W
31	SOLDLWUPEN	SOL_DL Interrupt Software Standby Returns Enable 0: Disable Software Standby returns by SOL_DL (falling of CMPOUT) interrupts 1: Enable Software Standby returns by SOL_DL (falling of CMPOUT) interrupts.	R/W

The bits in this register control whether the associated interrupt can wake up the CPU from Software Standby mode or not.

IRQWUPEN_i bit (PORT_IRQ_i Interrupt Software Standby Returns Enable) (i = 0 to 9)

The IRQWUPEN_i bit enables the use of PORT_IRQ_i interrupts to cancel Software Standby mode.

IWDTWUPEN bit (IWDT_NMIUNDF Interrupt Software Standby Returns Enable)

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

KEYWUPEN bit (KEY_INTKR Interrupt Software Standby Returns Enable)

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

LVD1WUPEN bit (LVD_LVD1 Interrupt Software Standby Returns Enable)

The LVD1WUPEN bit enables the use of LVD_LVD1 interrupts to cancel Software Standby mode.

LVDBATWUPEN bit (LVD_LVDBAT Interrupt Software Standby Returns Enable)

The LVDBATWUPEN bit enables the use of LVD_LVDBAT interrupts to cancel Software Standby mode.

SOLDHWUPEN bit (SOL_DH Interrupt Software Standby Returns Enable)

The SOLDHWUPEN bit enables the use of SOL_DH (rising of CMPOUT) interrupts to cancel Software Standby mode.

AGTW0CAWUPEN bit (AGTW0_AGTCMAI Interrupt Software Standby Returns Enable)

The AGTW0CAWUPEN bit enables the use of AGTW0_AGTCMAI interrupts to cancel Software Standby mode.

RTCALMWUPEN bit (RTC_ALM Interrupt Software Standby Returns Enable)

The RTCALMWUPEN bit enables the use of RTC_ALM interrupts to cancel Software Standby mode.

RTCPRDWUPEN bit (RTC_PRD Interrupt Software Standby Returns Enable)

The RTCPRDWUPEN bit enables the use of RTC_PRD interrupts to cancel Software Standby mode.

CCCPRDWUPEN bit (CCC_PRD Interrupt Software Standby Returns Enable)

The CCCPRDWUPEN bit enables the use of CCC_PRD interrupts to cancel Software Standby mode.

AGTW1CAWUPEN bit (AGTW1_AGTCMAI Interrupt Software Standby Returns Enable)

The AGTW1CAWUPEN bit enables the use of AGTW1_AGTCMAI interrupts to cancel Software Standby mode.

AGT1UDWUPEN bit (AGT1_AGTI Interrupt Software Standby Returns Enable)

The AGT1UDWUPEN bit enables the use of AGT1_AGTI interrupts to cancel Software Standby mode.

AGT1CAWUPEN bit (AGT1_AGTCMAI Interrupt Software Standby Returns Enable)

The AGT1CAWUPEN bit enables the use of AGT1_AGTCMAI interrupts to cancel Software Standby mode.

AGT0CAWUPEN bit (AGT0_AGTCMAI Interrupt Software Standby Returns Enable)

The AGT0CAWUPEN bit enables the use of AGT0_AGTCMAI interrupts to cancel Software Standby mode.

SOLDLWUPEN bit (SOL_DL Interrupt Software Standby Returns Enable)

The SOLDLWUPEN bit enables the use of SOL_DL (falling of CMPOUT) interrupts to cancel Software Standby mode.

16.3 Vector Table

The ICU detects maskable and non-maskable interrupts. Interrupt priorities are set up in the Arm NVIC. For information about these registers, see [section 16.8. Reference](#).

16.3.1 Interrupt Vector Table

[Table 16.3](#) describes the interrupt vector table. The interrupt vector addresses conform to the NVIC specifications.

Table 16.3 Interrupt vector table (1 of 3)

Exception number	IRQ number	Type	Vector offset	Source	Description
0	—	Initial SP	0x00	Arm	Initial value of the stack pointer (SP)
1	—	Reset	0x04	Arm	Reset vector
2	—	NMI	0x08	Arm	Non-maskable interrupt
3	—	HardFault	0x0C	Arm	Hard fault (exception caused by an error)

Table 16.3 Interrupt vector table (2 of 3)

Exception number	IRQ number	Type	Vector offset	Source	Description
4	—	Reserved	0x10	Arm	Reserved
5	—	Reserved	0x14	Arm	Reserved
6	—	Reserved	0x18	Arm	Reserved
7	—	Reserved	0x1C	Arm	Reserved
8	—	Reserved	0x20	Arm	Reserved
9	—	Reserved	0x24	Arm	Reserved
10	—	Reserved	0x28	Arm	Reserved
11	—	SVCall	0x2C	Arm	Supervisor call (an SVC command that calls system service)
12	—	Reserved	0x30	Arm	Reserved
13	—	Reserved	0x34	Arm	Reserved
14	—	PendSV	0x38	Arm	Pendable request for system service
15	—	SysTick	0x3C	Arm	SysTick timer alarm that informs the counter reaching 0
16	0	Interrupt 0	0x40	ICU.IELSR0	Event selected in the ICU.IELSR0 register
17	1	Interrupt 1	0x44	ICU.IELSR1	Event selected in the ICU.IELSR1 register
18	2	Interrupt 2	0x48	ICU.IELSR2	Event selected in the ICU.IELSR2 register
19	3	Interrupt 3	0x4C	ICU.IELSR3	Event selected in the ICU.IELSR3 register
20	4	Interrupt 4	0x50	ICU.IELSR4	Event selected in the ICU.IELSR4 register
21	5	Interrupt 5	0x54	ICU.IELSR5	Event selected in the ICU.IELSR5 register
22	6	Interrupt 6	0x58	ICU.IELSR6	Event selected in the ICU.IELSR6 register
23	7	Interrupt 7	0x5C	ICU.IELSR7	Event selected in the ICU.IELSR7 register
24	8	Interrupt 8	0x60	ICU.IELSR8	Event selected in the ICU.IELSR8 register
25	9	Interrupt 9	0x64	ICU.IELSR9	Event selected in the ICU.IELSR9 register
26	10	Interrupt 10	0x68	ICU.IELSR10	Event selected in the ICU.IELSR10 register
27	11	Interrupt 11	0x6C	ICU.IELSR11	Event selected in the ICU.IELSR11 register
28	12	Interrupt 12	0x70	ICU.IELSR12	Event selected in the ICU.IELSR12 register
29	13	Interrupt 13	0x74	ICU.IELSR13	Event selected in the ICU.IELSR13 register
30	14	Interrupt 14	0x78	ICU.IELSR14	Event selected in the ICU.IELSR14 register
31	15	Interrupt 15	0x7C	ICU.IELSR15	Event selected in the ICU.IELSR15 register
32	16	Interrupt 16	0x80	ICU.IELSR16	Event selected in the ICU.IELSR16 register
33	17	Interrupt 17	0x84	ICU.IELSR17	Event selected in the ICU.IELSR17 register
34	18	Interrupt 18	0x88	ICU.IELSR18	Event selected in the ICU.IELSR18 register
35	19	Interrupt 19	0x8C	ICU.IELSR19	Event selected in the ICU.IELSR19 register
36	20	Interrupt 20	0x90	ICU.IELSR20	Event selected in the ICU.IELSR20 register
37	21	Interrupt 21	0x94	ICU.IELSR21	Event selected in the ICU.IELSR21 register
38	22	Interrupt 22	0x98	ICU.IELSR22	Event selected in the ICU.IELSR22 register
39	23	Interrupt 23	0x9C	ICU.IELSR23	Event selected in the ICU.IELSR23 register
40	24	Interrupt 24	0xA0	ICU.IELSR24	Event selected in the ICU.IELSR24 register
41	25	Interrupt 25	0xA4	ICU.IELSR25	Event selected in the ICU.IELSR25 register
42	26	Interrupt 26	0xA8	ICU.IELSR26	Event selected in the ICU.IELSR26 register
43	27	Interrupt 27	0xAC	ICU.IELSR27	Event selected in the ICU.IELSR27 register

Table 16.3 Interrupt vector table (3 of 3)

Exception number	IRQ number	Type	Vector offset	Source	Description
44	28	Interrupt 28	0xB0	ICU.IELSR28	Event selected in the ICU.IELSR28 register
45	29	Interrupt 29	0xB4	ICU.IELSR29	Event selected in the ICU.IELSR29 register
46	30	Interrupt 30	0xB8	ICU.IELSR30	Event selected in the ICU.IELSR30 register
47	31	Interrupt 31	0xBC	ICU.IELSR31	Event selected in the ICU.IELSR31 register

16.3.2 Event Table

The following table lists heading details for [Table 16.4](#), which describes each event number.

Heading	Description
Interrupt request source	Name of the interrupt request source
Name	Name of the interrupt
CPU interrupts	"✓" indicates that the event can be connected to the NVIC (IELSRn setting)
Invoke DTC	"✓" indicates that the event can be used to request DTC activation (IELSRn setting)
Invoke DMAC	"✓" indicates that the event can be used to request DMAC activation (DELSRn setting)
Canceling Snooze mode	"✓" indicates that the event can be used to cancel from Snooze mode
Canceling Software Standby mode	"✓" indicates that the event can be used to request a return from Software Standby mode
Method for detecting interrupts	"Edge" and "level" indicate a method for detecting interrupts

Table 16.4 Event table (1 of 6)

Event number*4	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze mode	Canceling Software Standby mode	Method for detecting interrupts
			CPU interrupts	Invoke DTC	Invoke DMAC			
0x01	Port	PORT_IRQ0	✓	✓	✓	✓	✓	Edge/level
0x02		PORT_IRQ1	✓	✓	✓	✓	✓	Edge/level
0x03		PORT_IRQ2	✓	✓	✓	✓	✓	Edge/level
0x04		PORT_IRQ3	✓	✓	✓	✓	✓	Edge/level
0x05		PORT_IRQ4	✓	✓	✓	✓	✓	Edge/level
0x06		PORT_IRQ5	✓	✓	✓	✓	✓	Edge/level
0x07		PORT_IRQ6	✓	✓	✓	✓	✓	Edge/level
0x08		PORT_IRQ7	✓	✓	✓	✓	✓	Edge/level
0x09	DMAC0	DMAC0_INT	✓	✓				Edge
00xA	DMAC1	DMAC1_INT	✓	✓				Edge
0x0B	DMAC2	DMAC2_INT	✓	✓				Edge
0x0C	DMAC3	DMAC3_INT	✓	✓				Edge
0x0D	DTC	DTC_COMPLETE	✓			✓*3		Edge
0x0F	ICU	ICU_SNZCANCEL	✓			✓		Edge
0x10	FCU	FCU_FIFERR	✓					Level*6
0x11		FCU_FRDYI	✓					Edge
0x12	LVD	LVD_LVD1	✓			✓	✓	Edge
0x13		LVD_LVDBAT	✓			✓	✓	Edge
0x14	MOSC	MOSC_STOP	✓					Level

Table 16.4 Event table (2 of 6)

Event number*4	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze mode	Canceling Software Standby mode	Method for detecting interrupts
			CPU interrupts	Invoke DTC	Invoke DMAC			
0x15	Snooze mode control circuit	SYSTEM_SNZREQ		✓				Edge
0x16	EHC	SOL_DH	✓			✓	✓	Edge
0x17		SOL_DL	✓			✓	✓	Edge
0x18	AGT0	AGT0_AGTI	✓	✓	✓			Edge
0x1A		AGT0_AGTCMBI	✓	✓	✓			Edge
0x1B	AGT1	AGT1_AGTI	✓	✓	✓	✓	✓	Edge
0x1C		AGT1_AGTCMAI	✓	✓	✓	✓	✓	Edge
0x1D	AGT0	AGT0_AGTCMAI	✓	✓	✓	✓	✓	Edge
0x1E	IWDT	IWDT_NMIUNDF	✓			✓	✓	Edge
0x1F	WDT	WDT_NMIUNDF	✓					Edge
0x20	RTC	RTC_ALM	✓			✓	✓	Edge
0x21		RTC_PRD	✓			✓	✓	Edge
0x22		RTC_CUP	✓					Edge
0x23	ADC14	ADC140_ADI	✓	✓	✓			Edge
0x24		ADC140_GBADI	✓	✓	✓			Edge
0x25		ADC140_CMPAI	✓					Level
0x26		ADC140_CMPBI	✓					Level
0x27		ADC140_WCMPM	✓	✓	✓	✓*3		Edge
0x28		ADC140_WCMPUM	✓	✓	✓	✓*3		Edge
0x29		ADC140_GCADI	✓	✓	✓			Edge
0x2A	AGTW0	AGTW0_AGTCMAI	✓	✓	✓	✓	✓	Edge
0x2B		AGTW0_AGTI	✓	✓	✓			Edge
0x2C		AGTW0_AGTCMBI	✓	✓	✓			Edge
0x2D	AGTW1	AGTW1_AGTI	✓	✓	✓			Edge
0x2E		AGTW1_AGTCMAI	✓	✓	✓	✓	✓	Edge
0x2F	IIC0	IIC0_RXI	✓	✓	✓			Edge
0x30		IIC0_TXI	✓	✓	✓			Edge
0x31		IIC0_TEI	✓					Level
0x32		IIC0_EEI	✓					Level
0x33	IIC1	IIC1_RXI	✓	✓	✓			Edge
0x34		IIC1_TXI	✓	✓	✓			Edge
0x35		IIC1_TEI	✓					Level
0x36		IIC1_EEI	✓					Level
0x37	KINT	KEY_INTKR	✓			✓*5	✓*5	Edge/level
0x38	DOC	DOC_DOPCI	✓			✓		Edge
0x39	CAC	CAC_FEERI	✓					Level
0x3A		CAC_MENDI	✓					Level
0x3B		CAC_OVFI	✓					Level

Table 16.4 Event table (3 of 6)

Event number*4	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze mode	Canceling Software Standby mode	Method for detecting interrupts
			CPU interrupts	Invoke DTC	Invoke DMAC			
0x3C	I/O port	IOPORT_GROUP3	✓	✓*1	✓*1			Edge
0x3D		IOPORT_GROUP2	✓	✓*1	✓*1			Edge
0x3E	ELC	ELC_SWEVT0	✓*2	✓				Edge
0x3F		ELC_SWEVT1	✓*2	✓				Edge
0x40	POE	POEG_GROUPA	✓					Level
0x41		POEG_GROUPB	✓					Level
0x42	TMR	TMR_CMIA0	✓	✓	✓			Edge
0x43		TMR_CMIB0	✓	✓	✓			Edge
0x44		TMR_OVF0	✓					Edge
0x45		TMR_CMIA1	✓	✓	✓			Edge
0x46		TMR_CMIB1	✓	✓	✓			Edge
0x47		TMR_OVF1	✓					Edge
0x48	CCC	CCC_PRD	✓	✓	✓	✓	✓	Edge
		WUPT_OVI*7	✓	✓	✓	✓	✓	Edge
0x49		CCC_CUP	✓	✓	✓			Edge
0x4A		CCC_ERR	✓	✓	✓			Edge
0x4E	ELC	ELC_INT0	✓	✓	✓			Edge
0x4F		ELC_INT1	✓	✓	✓			Edge
0x50	GPT320	GPT0_CCMPA	✓	✓	✓			Edge
0x51		GPT0_CCMPB	✓	✓	✓			Edge
0x52		GPT0_CMPC	✓	✓	✓			Edge
0x53		GPT0_CMPD	✓	✓	✓			Edge
0x54		GPT0_OVF	✓	✓	✓			Edge
0x55		GPT0_UDF	✓	✓	✓			Edge
0x56	GPT321	GPT1_CCMPA	✓	✓	✓			Edge
0x57		GPT1_CCMPB	✓	✓	✓			Edge
0x58		GPT1_CMPC	✓	✓	✓			Edge
0x59		GPT1_CMPD	✓	✓	✓			Edge
0x5A		GPT1_OVF	✓	✓	✓			Edge
0x5B		GPT1_UDF	✓	✓	✓			Edge
0x5C	GPT162	GPT2_CCMPA	✓	✓	✓			Edge
0x5D		GPT2_CCMPB	✓	✓	✓			Edge
0x5E		GPT2_CMPC	✓	✓	✓			Edge
0x5F		GPT2_CMPD	✓	✓	✓			Edge
0x60		GPT2_OVF	✓	✓	✓			Edge
0x61		GPT2_UDF	✓	✓	✓			Edge

Table 16.4 Event table (4 of 6)

Event number*4	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze mode	Canceling Software Standby mode	Method for detecting interrupts
			CPU interrupts	Invoke DTC	Invoke DMAC			
0x62	GPT163	GPT3_CCMPA	✓	✓	✓			Edge
0x63		GPT3_CCMPB	✓	✓	✓			Edge
0x64		GPT3_CMPC	✓	✓	✓			Edge
0x65		GPT3_CMPD	✓	✓	✓			Edge
0x66		GPT3_OVF	✓	✓	✓			Edge
0x67		GPT3_UDF	✓	✓	✓			Edge
0x68	GPT164	GPT4_CCMPA	✓	✓	✓			Edge
0x69		GPT4_CCMPB	✓	✓	✓			Edge
0x6A		GPT4_CMPC	✓	✓	✓			Edge
0x6B		GPT4_CMPD	✓	✓	✓			Edge
0x6C		GPT4_OVF	✓	✓	✓			Edge
0x6D		GPT4_UDF	✓	✓	✓			Edge
0x6E	GPT165	GPT5_CCMPA	✓	✓	✓			Edge
0x6F		GPT5_CCMPB	✓	✓	✓			Edge
0x70		GPT5_CMPC	✓	✓	✓			Edge
0x71		GPT5_CMPD	✓	✓	✓			Edge
0x72		GPT5_OVF	✓	✓	✓			Edge
0x73		GPT5_UDF	✓	✓	✓			Edge
0x74	GPT	GPT_UVWEDGE	✓					Edge
0x75	SCI0	SCI0_RXI	✓	✓	✓			Edge
0x76		SCI0_TXI	✓	✓	✓			Edge
0x77		SCI0_TEI	✓					Level
0x78		SCI0_ERI	✓					Level
0x79		SCI0_AM	✓			✓*3		Edge
0x7A		SCI0_RXI_OR_ERI				✓*3		Edge
0x7B	SCI1	SCI1_RXI	✓	✓	✓			Edge
0x7C		SCI1_TXI	✓	✓	✓			Edge
0x7D		SCI1_TEI	✓					Level
0x7E		SCI1_ERI	✓					Level
0x7F		SCI1_AM	✓					Edge
0x80	SCI2	SCI2_RXI	✓	✓	✓			Edge
0x81		SCI2_TXI	✓	✓	✓			Edge
0x82		SCI2_TEI	✓					Level
0x83		SCI2_ERI	✓					Level
0x84		SCI2_AM	✓					Edge

Table 16.4 Event table (5 of 6)

Event number*4	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze mode	Canceling Software Standby mode	Method for detecting interrupts
			CPU interrupts	Invoke DTC	Invoke DMAC			
0x85	SCI3	SCI3_RXI	✓	✓	✓			Edge
0x86		SCI3_TXI	✓	✓	✓			Edge
0x87		SCI3_TEI	✓					Level
0x88		SCI3_ERI	✓					Level
0x89		SCI3_AM	✓					Edge
0x8A	SCI4	SCI4_RXI	✓	✓	✓			Edge
0x8B		SCI4_TXI	✓	✓	✓			Edge
0x8C		SCI4_TEI	✓					Level
0x8D		SCI4_ERI	✓					Level
0x8E		SCI4_AM	✓					Edge
0x8F	SCI5	SCI5_RXI	✓	✓	✓			Edge
0x90		SCI5_TXI	✓	✓	✓			Edge
0x91		SCI5_TEI	✓					Level
0x92		SCI5_ERI	✓					Level
0x93		SCI5_AM	✓					Edge
0x94	SCI9	SCI9_RXI	✓	✓	✓			Edge
0x95		SCI9_TXI	✓	✓	✓			Edge
0x96		SCI9_TEI	✓					Level
0x97		SCI9_ERI	✓					Level
0x98		SCI9_AM	✓					Edge
0x99	SPI0	SPI0_SPRI	✓	✓	✓			Edge
0x9A		SPI0_SPTI	✓	✓	✓			Edge
0x9B		SPI0_SPII	✓					Level
0x9C		SPI0_SPEI	✓					Level
0x9D		SPI0_SPTEND	✓					Edge
0x9E	SPI1	SPI1_SPRI	✓	✓	✓			Edge
0x9F		SPI1_SPTI	✓	✓	✓			Edge
0xA0		SPI1_SPII	✓					Level
0xA1		SPI1_SPEI	✓					Level
0xA2		SPI1_SPTEND	✓					Edge
0xA3	QSPI	QSPI_INTR	✓					Level
0xA4	DIV	DIV_CALCCOMP	✓	✓	✓			Edge
0xA6	MLCD	MLCD_TEI	✓	✓	✓			Edge
0xA7		MLCD_TEMI	✓	✓	✓			Edge
0xA8	GDT	GDT_DATII	✓	✓	✓			Edge
0xA9		GDT_DATOI	✓	✓	✓			Edge
0xAA		GDT_FDCENDI	✓	✓	✓			Edge

Table 16.4 Event table (6 of 6)

Event number*4	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze mode	Canceling Software Standby mode	Method for detecting interrupts
			CPU interrupts	Invoke DTC	Invoke DMAC			
0xAB	TSIP-Lite	PROC_BUSY	✓					Edge
0xAC		ROMOK	✓					Edge
0xAD		LONG_PLG	✓					Edge
0xAE		TEST_BUSY	✓					Edge
0xAF		WRRDY0	✓					Edge
0xB0		WRRDY4	✓					Edge
0xB1		RDRDY0	✓					Edge
0xB2		INTEGRATE_WRRDY	✓	✓	✓			Edge
0xB3		INTEGRATE_RDRDY	✓	✓	✓			Edge
0xB4	Input on the pins	PORT_IRQ8	✓	✓	✓	✓	✓	Edge/level
0xB5		PORT_IRQ9	✓	✓	✓	✓	✓	Edge/level

Note 1. When interrupts are input continuously, only the first edge detection is valid.

Note 2. Only interrupts after DTC transfer are supported.

Note 3. By setting the SNZEDCR register, the snooze mode is canceled, followed by transition to the software standby mode.

Note 4. The event number is used only for the selection of a DMAC activation source with DELSRn and the selection of a return source from the snooze mode with SELSR0. The event number is not used to set a CPU interrupt with the IELSRn register and the activation of the DTC. For details, see [section 16.2.6. IELSRn : ICU Event Link Setting Register n \(n = 0 to 31\)](#) and [Table 16.7](#).

Note 5. Supported only when KRCTL.KRMD = 1.

Note 6. Issue the FACL command of status clear by the flash sequencer as follows:

- Set the FACL command processing head address with FSADDR.
- Set the code flash P/E mode with FENTRYR.

FENTRYR.KEY = 0xAA, FENTRYR.FENTRYC = 1

- Write 0x50 to the FACL command issue area.

For details, see the section of Flash Memory.

Note 7. Use CCC_PRD and WUPT_OVI exclusively.

16.3.3 ICU and DTC Event Number

[Table 16.5](#), [Table 16.6](#) indicates the IELSRn.IELS[4:0] set values on the CPU interrupt or on DTC activation request. [Table 16.7](#) indicates register set values of each event selection.

Table 16.5 ICU Event Link Select (1/2) (1 of 2)

IELS[4:0]	Group 0 (IELSR0/8/16/24)	Group 1 (IELSR1/9/17/25)	Group 2 (IELSR2/10/18/26)	Group 3 (IELSR3/11/19/27)
0x00	Interrupt disabled	Interrupt disabled	Interrupt disabled	Interrupt disabled
0x01	PORT_IRQ0	PORT_IRQ1	PORT_IRQ2	PORT_IRQ3
0x02	DMAC0_INT	DMAC1_INT	DMAC2_INT	DMAC3_INT
0x03	DTC_COMPLETE*1	FCU_FIFERR*1	FCU_FRDYI*1	SYSTEM_SNZREQ*2
0x04	ICU_SNZCANCEL*1	LVD_LVDBAT*1	AGT0_AGTCAI	IWDT_NMIUNDF*1
0x05	LVD_LVD1*1	AGT1_AGTCAI	RTC_PRD*1	RTC_CUP*1
0x06	AGT1_AGTI	RTC_ALM*1	ADC140_CMPAI*1	ADC140_CMPBI*1
0x07	WDT_NMIUNDF*1	ADC140_GBADI	ADC140_GCADI	AGTW0_AGTCAI*1
0x08	ADC140_ADI	ADC140_WCMPUM*2	IIC0_TEI*1	IIC0_EEI*1
0x09	ADC140_WCMPM*2	IIC0_TXI	CAC_MENDI*1	CAC_OVFI*1

Table 16.5 ICU Event Link Select (1/2) (2 of 2)

IELS[4:0]	Group 0 (IELSR0/8/16/24)	Group 1 (IELSR1/9/17/25)	Group 2 (IELSR2/10/18/26)	Group 3 (IELSR3/11/19/27)
0x0A	IIC0_RXI	DOC_DOPCI* ¹	ELC_SWEVT0	ELC_SWEVT1
0x0B	CCC_PRD WUPT_OVI* ³	CAC_FEERI* ¹	POEG_GROUPA* ¹	POEG_GROUPB* ¹
0x0C	GPT0_CCMPA	CCC_CUP	GPT0_CMPC	Setting prohibited
0x0D	GPT0_OVF	GPT0_CCMPB	GPT2_CMPC	GPT0_CMPD
0x0E	GPT2_CCMPA	GPT0_UDF	GPT2_OVF	GPT2_CMPD
0x0F	GPT_UVWEDGE* ¹	GPT2_CCMPB	SCI0_TEI* ¹	GPT2_UDF
0x10	SCI0_RXI	SCI0_TX1	SPI0_SPII* ¹	SCI0_ER1* ¹
0x11	SCI0_AM* ¹	SPI0_SPTI	SPI0_SPTEND* ¹	SPI0_SPEI* ¹
0x12	SPI0_SPRI	SOL_DL* ¹	AGTW1_AGTI* ¹	QSPI_INTR* ¹
0x13	SOL_DH* ¹	AGT0_AGTICMBI	IOPORT_GROUP2	AGT0_AGTI
0x14	Setting prohibited	AGTW0_AGTICMBI	TMR_CMIA0	AGTW1_AGTICMAI* ¹
0x15	AGTW0_AGTI	IIC1_TXI	GPT1_CMPC	TMR_CMIB0
0x16	IIC1_RXI	IOPORT_GROUP3	GPT4_CMPC	ELC_INT0
0x17	KEY_INTKR* ¹	GPT1_CCMPB	GPT5_OVF	GPT1_CMPD
0x18	GPT1_CCMPA	GPT3_UDF	SCI1_TEI* ¹	GPT4_CMPD
0x19	GPT3_OVF	GPT4_CCMPB	SCI3_TEI* ¹	GPT5_UDF
0x1A	GPT4_CCMPA	SCI1_TXI	SCI4_AM* ¹	SCI1_ERI* ¹
0x1B	SCI1_RXI	SCI2_AM* ¹	SCI5_TEI* ¹	SCI3_ERI* ¹
0x1C	SCI3_RXI	SCI3_TXI	SPI1_SPII* ¹	SCI5_ERI* ¹
0x1D	SCI5_RXI	SCI5_TXI	MLCD_TEMI	SCI9_AM* ¹
0x1E	PORT_IRQ8	MLCD_TEI	LONG_PLG* ¹	GDT_DATII
0x1F	PROC_BUSY* ¹	ROMOK* ¹	PORT_IRQ9	TEST_BUSY* ¹

Note 1. Available only for linking to the NVIC (not available for activating the DTC)

Note 2. Available only for activating the DTC (not available for linking to the NVIC)

Note 3. Use CCC_PRD and WUPT_OVI exclusively.

Table 16.6 ICU Event Link Select (2/2) (1 of 2)

IELS[4:0]	Group 4 (IELSR4/12/20/28)	Group 5 (IELSR5/13/21/29)	Group 6 (IELSR6/14/22/30)	Group 7 (IELSR7/15/23/31)
0x00	Interrupt disabled	Interrupt disabled	Interrupt disabled	Interrupt disabled
0x01	PORT_IRQ0	PORT_IRQ1	PORT_IRQ2	PORT_IRQ3
0x02	DMAC0_INT	DMAC1_INT	DMAC2_INT	DMAC3_INT
0x03	DTC_COMPLETE* ¹	FCU_FIFERR* ¹	FCU_FRDYI* ¹	SYSTEM_SNZREQ* ²
0x04	ICU_SNZCANCEL* ¹	LVD_LVDBAT* ¹	AGT0_AGTICMAI	IWDT_NMIUNDF* ¹
0x05	LVD_LVD1* ¹	AGT1_AGTICMAI	RTC_PRD* ¹	RTC_CUP* ¹
0x06	AGT1_AGTI	RTC_ALM* ¹	ADC140_CMPAI* ¹	ADC140_CMPBI* ¹
0x07	WDT_NMIUNDF* ¹	ADC140_GBADI	ADC140_GCADI	Setting prohibited
0x08	ADC140_ADI	ADC140_WCMPUM* ²	IIC0_TEI* ¹	IIC0_EEI* ¹
0x09	ADC140_WCMPM* ²	IIC0_TXI	CAC_MENDI* ¹	CAC_OVFI* ¹
0x0A	IIC0_RXI	DOC_DOPCI* ¹	ELC_SWEVT0	ELC_SWEVT1
0x0B	CCC_PRD	CAC_FEERI* ¹	POEG_GROUPA* ¹	POEG_GROUPB* ¹

Table 16.6 ICU Event Link Select (2/2) (2 of 2)

IELS[4:0]	Group 4 (IELSR4/12/20/28)	Group 5 (IELSR5/13/21/29)	Group 6 (IELSR6/14/22/30)	Group 7 (IELSR7/15/23/31)
0x0C	GPT0_CCMPA	CCC_CUP	GPT0_CMPC	Setting prohibited
0x0D	GPT0_OVF	GPT0_CCMPB	GPT2_CMPC	GPT0_CMPD
0x0E	GPT2_CCMPA	GPT0_UDF	GPT2_OVF	GPT2_CMPD
0x0F	GPT_UVWEDGE* ¹	GPT2_CCMPB	SCI0_TEI* ¹	GPT2_UDF
0x10	SCI0_RXI	SCI0_TX1	SPI0_SPII* ¹	SCI0_ER1* ¹
0x11	SCI0_AM* ¹	SPI0_SPTI	SPI0_SPTEND* ¹	SPI0_SPEI* ¹
0x12	SPI0_SPRI	PORT_IRQ5	PORT_IRQ6	QSPI_INTR* ¹
0x13	PORT_IRQ4	IIC1_EEI* ¹	MOSC_STOP	PORT_IRQ7
0x14	IIC1_TEI* ¹	TMR_CMIA1	TMR_CMIB1	TMR_OVF1* ¹
0x15	TMR_OVF0* ¹	Setting prohibited	CCC_ERR	ELC_INT1
0x16	Setting prohibited	GPT1_UDF	GPT3_CMPC	GPT3_CMPD
0x17	GPT1_OVF	GPT3_CCMPB	GPT4_OVF	GPT4_UDF
0x18	GPT3_CCMPA	GPT5_CCMPB	GPT5_CMPC	GPT5_CMPD
0x19	GPT5_CCMPA	SCI1_AM* ¹	SCI2_TEI* ¹	SCI2_ERI* ¹
0x1A	SCI2_RXI	SCI2_TXI	SCI3_AM* ¹	SCI4_ERI* ¹
0x1B	SCI4_RXI	SCI4_TXI	SCI4_TEI* ¹	SCI5_AM* ¹
0x1C	SCI9_RXI	SCI9_TXI	SCI9_TEI* ¹	SCI9_ERI* ¹
0x1D	SPI1_SPRI	SPI1_SPTI	SPI1_SPTEND* ¹	SPI1_SPEI* ¹
0x1E	GDT_DATOI	GDT_FDCENDI	RDRDY0* ¹	DIV_CALCCOMP
0x1F	WRRDY0* ¹	WRRDY4* ¹	INTEGRATE_RDRDY	INTEGRATE_WRRDY

Note 1. Available only for linking to the NVIC (not available for activating the DTC)

Note 2. Available only for activating the DTC (not available for linking to the NVIC)

The following list describes heading details for [Table 16.7](#).

- IELSRn.IELS[4:0]
 - Indicates the IELS[4:0] bit set values on the CPU interrupt of groups 0 to 7 or on DTC activation request selection.
 (*1): indicates CPU interrupt enabled (DTC activation disabled)
 (*2): DTC activation enabled (CPU interrupt disabled).
- DELSRn.DELS[7:0]
 - Indicates the DELS[7:0] bit set values on DMAC activation request selection. The set values indicate the event numbers listed in [Table 16.4](#).
- SELSR0.SELS[7:0]
 - Indicates the SELS[7:0] bit set values on the selection of snooze mode return request. The set values indicate the event numbers listed in [Table 16.4](#).

Table 16.7 Register set values of each event selection (1 of 6)

Name	Support products IELSRn.IELS[4:0]								DELSRn. DELS[7:0] (n = 0 to 3)	SELSR0. SELS[7:0]
	Group 0 (n = 0/8/16/24)	Group 1 (n = 1/9/17/25)	Group 2 (n = 2/10/18/26)	Group 3 (n = 3/11/19/27)	Group 4 (n = 4/12/20/28)	Group 5 (n = 5/13/21/29)	Group 6 (n = 6/14/22/30)	Group 7 (n = 7/15/23/31)		
PORT_IRQ0	0x01				0x01				0x01	
PORT_IRQ1		0x01				0x01			0x02	
PORT_IRQ2			0x01				0x01		0x03	

Table 16.7 Register set values of each event selection (2 of 6)

Name	Support products IELSRn.IELS[4:0]								DELSRn. DELS[7:0] (n = 0 to 3)	SELSR0. SELS[7:0]
	Group 0 (n = 0/8/16/24)	Group 1 (n = 1/9/17/25)	Group 2 (n = 2/10/18/26)	Group 3 (n = 3/11/19/27)	Group 4 (n = 4/12/20/28)	Group 5 (n = 5/13/21/29)	Group 6 (n = 6/14/22/30)	Group 7 (n = 7/15/23/31)		
PORT_IRQ3				0x01				0x01	0x04	
PORT_IRQ4					0x13				0x05	
PORT_IRQ5						0x12			0x06	
PORT_IRQ6							0x12		0x07	
PORT_IRQ7								0x13	0x08	
DMAC0_INT	0x02				0x02					
DMAC1_INT		0x02				0x02				
DMAC2_INT			0x02				0x02			
DMAC3_INT				0x02				0x02		
DTC_COMPLETE	0x03 (*1)				0x03 (*1)					0x0D
ICU_SNZCANCEL	0x04 (*1)				0x04 (*1)					
FCU_FIFERR		0x03 (*1)				0x03 (*1)				
FCU_FRDYI			0x03 (*1)				0x03 (*1)			
LVD_LVD1	0x05 (*1)				0x05 (*1)					
LVD_LVDBAT		0x04 (*1)				0x04 (*1)				
MOSC_STOP							0x13 (*1)			
SYSTEM_SNZREQ				0x03 (*2)				0x03 (*2)		
SOL_DH	0x13 (*1)									
SOL_DL		0x12 (*1)								
AGT0_AGTI				0x13					0x18	
AGT0_AGTCMBI		0x13							0x1A	
AGT1_AGTI	0x06				0x06				0x1B	
AGT1_AGTCMAI		0x05				0x05			0x1C	
AGT0_AGTCMAI			0x04				0x04		0x1D	
IWDT_NMIUNDF				0x04 (*1)				0x04 (*1)		
WDT_NMIUNDF	0x07 (*1)				0x07 (*1)					
RTC_ALM		0x06 (*1)				0x06 (*1)				
RTC_PRD			0x05 (*1)				0x05 (*1)			
RTC_CUP				0x05 (*1)				0x05 (*1)		
ADC140_ADI	0x08				0x08				0x23	
ADC140_GBADI		0x07				0x07			0x24	
ADC140_CMPAI			0x06 (*1)				0x06 (*1)			
ADC140_CMPBI				0x06 (*1)				0x06 (*1)		
ADC140_WCMPPM	0x09 (*2)				0x09 (*2)				0x27	0x27
ADC140_WCMPUM		0x08 (*2)				0x08 (*2)			0x28	0x28
ADC140_GCADI			0x07				0x07		0x29	
AGTW0_AGTCMAI				0x07 (*1)				0x07 (*1)	0x2A	
AGTW0_AGTI	0x15								0x2B	
AGTW0_AGTCMBI		0x14							0x2C	
AGTW1_AGTI			0x12 (*1)						0x2D	
AGTW1_AGTCMAI				0x14 (*1)					0x2E	
IIC0_RXI	0x0A				0x0A				0x2F	

Table 16.7 Register set values of each event selection (3 of 6)

Name	Support products IELSRn.IELS[4:0]								DELSRn. DELS[7:0] (n = 0 to 3)	SELSR0. SELS[7:0]
	Group 0 (n = 0/8/16/24)	Group 1 (n = 1/9/17/25)	Group 2 (n = 2/10/18/26)	Group 3 (n = 3/11/19/27)	Group 4 (n = 4/12/20/28)	Group 5 (n = 5/13/21/29)	Group 6 (n = 6/14/22/30)	Group 7 (n = 7/15/23/31)		
IIC0_TXI		0x09				0x09			0x30	
IIC0_TEI			0x08 (*1)				0x08 (*1)			
IIC0_EEI				0x08 (*1)				0x08 (*1)		
IIC1_RXI	0x16								0x33	
IIC1_TXI		0x15							0x34	
IIC1_TEI					0x14 (*1)					
IIC1_EEI						0x13 (*1)				
KEY_INTKR	0x17 (*1)									
DOC_DOPCI		0x0A (*1)				0x0A (*1)				0x38
CAC_FEERI		0x0B (*1)				0x0B (*1)				
CAC_MENDI			0x09 (*1)				0x09 (*1)			
CAC_OVFI				0x09 (*1)				0x09 (*1)		
IOPORT_GROUP3		0x16							0x3C	
IOPORT_GROUP2			0x13						0x3D	
ELC_SWEVT0			0x0A				0x0A			
ELC_SWEVT1				0x0A				0x0A		
POEG_GROUPA			0x0B (*1)				0x0B (*1)			
POEG_GROUPB				0x0B (*1)				0x0B (*1)		
TMR_CMIA0			0x14						0x42	
TMR_CMIB0				0x15					0x43	
TMR_OVF0					0x15 (*1)					
TMR_CMIA1						0x14			0x45	
TMR_CMIB1							0x14		0x46	
TMR_OVF1								0x14 (*1)		
CCC_PRD	0x0B				0x0B				0x48	
WUPT_OVI*1	0x0B				0x0B				0x48	
CCC_CUP		0x0C				0x0C			0x49	
CCC_ERR							0x15		0x4A	
ELC_INT0				0x16					0x4E	
ELC_INT1								0x15	0x4F	
GPT0_CCMPA	0x0C				0x0C				0x50	
GPT0_CCMPB		0x0D				0x0D			0x51	
GPT0_CMPC			0x0C				0x0C		0x52	
GPT0_CMPD				0x0D				0x0D	0x53	
GPT0_OVF	0x0D				0x0D				0x54	
GPT0_UDF		0x0E				0x0E			0x55	
GPT1_CCMPA	0x18								0x56	
GPT1_CCMPB		0x17							0x57	
GPT1_CMPC			0x15						0x58	
GPT1_CMPD				0x17					0x59	
GPT1_OVF					0x17				0x5A	
GPT1_UDF						0x16			0x5B	

Table 16.7 Register set values of each event selection (4 of 6)

Name	Support products IELSRn.IELS[4:0]								DELSRn. DELS[7:0] (n = 0 to 3)	SELSR0. SELS[7:0]
	Group 0 (n = 0/8/16/24)	Group 1 (n = 1/9/17/25)	Group 2 (n = 2/10/18/26)	Group 3 (n = 3/11/19/27)	Group 4 (n = 4/12/20/28)	Group 5 (n = 5/13/21/29)	Group 6 (n = 6/14/22/30)	Group 7 (n = 7/15/23/31)		
GPT2_CCMPA	0x0E				0x0E				0x5C	
GPT2_CCMPB		0x0F				0x0F			0x5D	
GPT2_CMPC			0x0D				0x0D		0x5E	
GPT2_CMPD				0x0E				0x0E	0x5F	
GPT2_OVF			0x0E				0x0E		0x60	
GPT2_UDF				0x0F				0x0F	0x61	
GPT3_CCMPA					0x18				0x62	
GPT3_CCMPB						0x17			0x63	
GPT3_CMPC							0x16		0x64	
GPT3_CMPD								0x16	0x65	
GPT3_OVF	0x19								0x66	
GPT3_UDF		0x18							0x67	
GPT4_CCMPA	0x1A								0x68	
GPT4_CCMPB		0x19							0x69	
GPT4_CMPC			0x16						0x6A	
GPT4_CMPD				0x18					0x6B	
GPT4_OVF							0x17		0x6C	
GPT4_UDF								0x17	0x6D	
GPT5_CCMPA					0x19				0x6E	
GPT5_CCMPB						0x18			0x6F	
GPT5_CMPC							0x18		0x70	
GPT5_CMPD								0x18	0x71	
GPT5_OVF			0x17						0x72	
GPT5_UDF				0x19					0x73	
GPT_UVWEDGE	0x0F (*1)				0x0F (*1)					
SCI0_RXI	0x10				0x10				0x75	
SCI0_TXI		0x10				0x10			0x76	
SCI0_TEI			0x0F (*1)				0x0F (*1)			
SCI0_ERI				0x10 (*1)				0x10 (*1)		
SCI0_AM	0x11 (*1)				0x11 (*1)					0x79
SCI0_RXI_OR_ERI										0x7A
SCI1_RXI	0x1B								0x7B	
SCI1_TXI		0x1A							0x7C	
SCI1_TEI			0x18 (*1)							
SCI1_ERI				0x1A (*1)						
SCI1_AM						0x19 (*1)				
SCI2_RXI					0x1A				0x80	
SCI2_TXI						0x1A			0x81	
SCI2_TEI							0x19			
SCI2_ERI								0x19		
SCI2_AM		0x1B								
SCI3_RXI	0x1C								0x85	

Table 16.7 Register set values of each event selection (5 of 6)

Name	Support products IELSRn.IELS[4:0]								DELSRn. DELS[7:0] (n = 0 to 3)	SELSR0. SELS[7:0]
	Group 0 (n = 0/8/16/24)	Group 1 (n = 1/9/17/25)	Group 2 (n = 2/10/18/26)	Group 3 (n = 3/11/19/27)	Group 4 (n = 4/12/20/28)	Group 5 (n = 5/13/21/29)	Group 6 (n = 6/14/22/30)	Group 7 (n = 7/15/23/31)		
SCI3_TXI		0x1C							0x86	
SCI3_TEI			0x19 (*1)							
SCI3_ERI				0x1B (*1)						
SCI3_AM							0x1A (*1)			
SCI4_RXI					0x1B				0x8A	
SCI4_TXI						0x1B			0x8B	
SCI4_TEI							0x1B (*1)			
SCI4_ERI								0x1A (*1)		
SCI4_AM			0x1A (*1)							
SCI5_RXI	0x1D								0x8F	
SCI5_TXI		0x1D							0x90	
SCI5_TEI			0x1B (*1)							
SCI5_ERI				0x1C (*1)						
SCI5_AM								0x1B (*1)		
SCI9_RXI					0x1C				0x94	
SCI9_TXI						0x1C			0x95	
SCI9_TEI							0x1C (*1)			
SCI9_ERI								0x1C (*1)		
SCI9_AM				0x1D (*1)						
SPI0_SPRI	0x12				0x12				0x99	
SPI0_SPTI		0x11				0x11			0x9A	
SPI0_SPII			0x10 (*1)				0x10 (*1)			
SPI0_SPEI				0x11 (*1)				0x11 (*1)		
SPI0_SPTEND			0x11 (*1)				0x11 (*1)			
SPI1_SPRI					0x1D				0x9E	
SPI1_SPTI						0x1D			0x9F	
SPI1_SPII			0x1C (*1)							
SPI1_SPEI								0x1D (*1)		
SPI1_SPTEND							0x1D (*1)			
QSPI_INTR				0x12 (*1)				0x12 (*1)		
DIV_CALCCOMP								0x1E	0xA4	
MLCD_TEI		0x1E							0xA6	
MLCD_TEMI			0x1D						0xA7	
GDT_INI				0x1E					0xA8	
GDT_OUTI					0x1E				0xA9	
GDT_IFI						0x1E			0xAA	
PROC_BUSY	0x1F (*1)									
ROMOK		0x1F (*1)								
LONG_PLG			0x1E (*1)							
TEST_BUSY				0x1F (*1)						
WRRDY0					0x1F (*1)					
WRRDY4						0x1F (*1)				

Table 16.7 Register set values of each event selection (6 of 6)

Name	Support products IELSRn.IELS[4:0]								DELSRn. DELS[7:0] (n = 0 to 3)	SELSR0. SELS[7:0]
	Group 0 (n = 0/8/16/24)	Group 1 (n = 1/9/17/25)	Group 2 (n = 2/10/18/26)	Group 3 (n = 3/11/19/27)	Group 4 (n = 4/12/20/28)	Group 5 (n = 5/13/21/29)	Group 6 (n = 6/14/22/30)	Group 7 (n = 7/15/23/31)		
RDRDY0							0x1E (*1)			
INTEGRATE_WRRDY								0x1F	0xB2	
INTEGRATE_RDRDY							0x1F		0xB3	
PORT_IRQ8	0x1E								0xB4	
PORT_IRQ9			0x1F						0xB5	

Note 1. Use CCC_PRD and WUPT_OVI exclusively.

16.4 Interrupt Operation

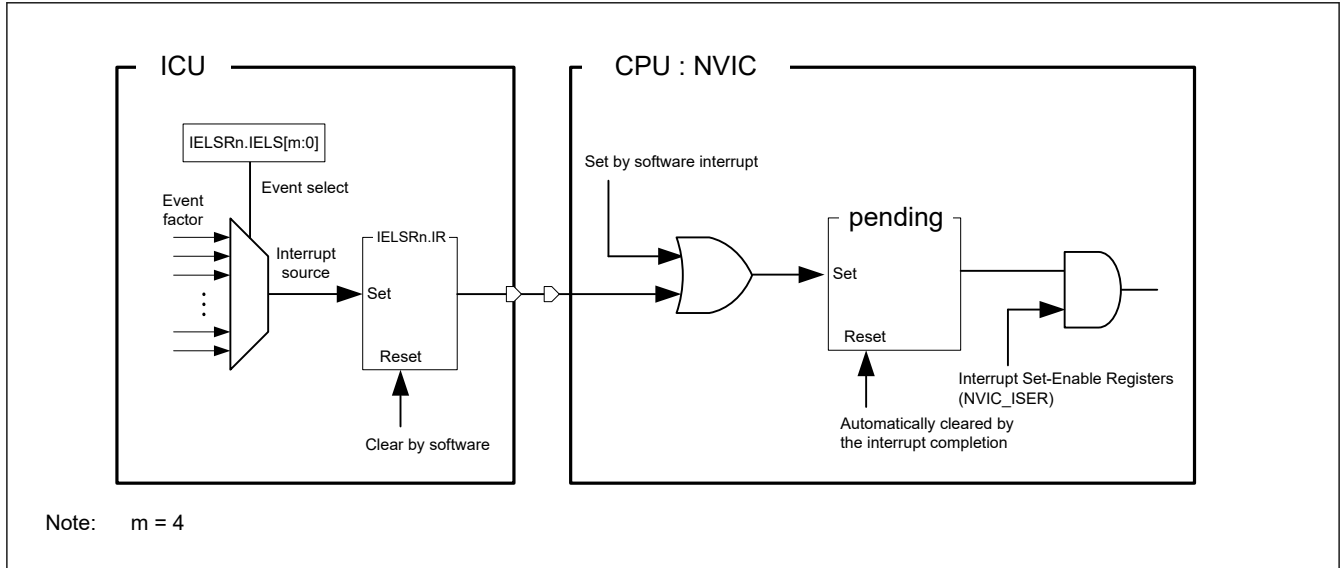
The ICU performs the following functions:

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

16.4.1 Interrupt detection selection

The ICU selects an event source input from a peripheral function interrupt or an external terminal interrupt with IELSRn.IELS [4:0].

The accepted interrupt source sets the IELSRn.IR to 1 and sends an interrupt request to the NVIC.

**Figure 16.2** Interrupt path of the ICU and CPU (NVIC)

16.4.2 Detecting Interrupts

External pin interrupt requests are detected by either the edge or level (falling edge, rising edge, rising and falling edges, or low level) of the interrupt signal. Set the IRQCIRi.IRQMD[1:0] bits to select detection mode for the IRQi pins. For interrupt sources associated with peripheral modules, see [Table 16.3](#) and [Table 16.4](#).

16.5 Interrupt setting procedure

16.5.1 Enabling Interrupt Requests

The following shows the procedure for enabling an interrupt request.

1. Set the Interrupt Set-Enable register (NVIC_ISER).
2. Set the IELSRn.IELS[4:0] bits as the interrupt source.
3. Specify the operation settings for the event source, such as DMAC activation (DELSRn.DEELS[7:0]), snooze mode cancellation (SELSR0.SELS[7:0]), software standby mode cancellation (WUPEN register setting).

16.5.2 Disabling Interrupt Requests

The procedure to disable the interrupt request is shown below.

1. Disable the operation settings for the event source, such as DMAC activation (DELSRn.DEELS[7:0]), snooze mode cancellation (SELSR0.SELS[7:0]), software standby mode cancellation (WUPEN register setting).
2. Clear the interrupt source setting (IELSRn.IELS[4:0] = 0x00).
3. Clear the interrupt status flag (IELSRn.IR = 0).
4. Clear the interrupt Clear-enable register (NVIC_ICER) and interrupt Clear-Pending register (NVIC_ICPR).

16.5.3 Polling for interrupts

The following shows the procedure for polling for interrupt requests.

1. Set the Interrupt Clear-Enable register (NVIC_ICER).
2. Set the IELSRn.IELS[4:0] bits as the interrupt source.
3. Specify the operation settings for the event source, such as DMAC activation (DELSRn.DEELS[7:0]), snooze mode cancellation (SELSR0.SELS[7:0]), software standby mode cancellation (WUPEN register setting).
4. Poll the interrupt Set-Pending register (NVIC_ISPR).

16.5.4 Selecting Interrupt Request Destinations

The interrupt output destination, CPU, DMAC, or DTC can be independently selected for each interrupt source.

The available destinations are fixed for each interrupt, as described in [Table 16.3](#), [Table 16.4](#) and [Table 16.6](#).

Use an interrupt request destination setting that is indicated by a “✓” in the event list.

Setting the same interrupt source for IELSRn and DELSRn is prohibited.

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

16.5.4.1 CPU interrupt request

When IELSRn.DTCE = 0, the event specified in the IELSRn register is output to the NVIC. Set the IELSRn.IELS[4:0] bits to the target event and set the IELSRn.DTCE bit to 0.

16.5.4.2 DTC activation

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

1. Set the IELSRn.IELS[4:0] 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 16.8](#) shows operation when the DTC is the interrupt request destination.

Table 16.8 Operation when DTC becomes interrupt request destination

Interrupt request destination	DISEL*1	Remaining transfer operations	Operation per request	IR*2	Interrupt request destination after transfer
DTC*3	1	≠ 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	CPU(IELSRn.DTCE bit is cleared)
	0	≠ 0	DTC transfer	Cleared at the start of DTC data transfer after reading DTC transfer data	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	CPU(IELSRn.DTCE bit is cleared)

Note 1. DTC.MRB.DISEL bit controls the interrupt generates timing from DTC to CPU.

Note 2. When the IELSRn.IR flag is 1, an interrupt request (DTC activation request) that occurs again is ignored.

Note 3. For chain transfers, DTC transfer continues until the last chain transfer ends. The DISEL bit state and the remaining transfer count determine whether a CPU interrupt occurs, the IELSRn.IR flag clear timing, and the interrupt request destination after transfer. See [Table 20.2](#) in [section 20, Data Transfer Controller \(DTC\)](#).

16.5.4.3 DMAC Activation

Events specified in the DELSRn registers are output to the DMAC.

To set the interrupt source for DMAC, use the following procedure:

1. Set the DELSRn.DELS[7:0] bits to the event to activate the DMAC.
2. Set the activation source for the target DMAC channel (DMACn.DMTMD.DCTG[1:0]) to 01b (interrupts from peripheral modules or external interrupt input pins).
3. Set the DMAC transfer enable bit for the target DMAC channel (DMACn.DMCNT.DTE) to 1.
4. Set the DMAC operation enable bit (DMACn.DMAST) to 1.

16.5.5 External Pin Interrupts

To use external pin interrupts:

1. Configure I/O ports settings
2. Set the IRQMD[1:0] bits of the given IRQCRi register (i = 0 to 9) to select the senses of detection.
3. Select the IRQ pin as follows:
 - If the IRQ pin is to be used for CPU interrupt requests, set the IELSRn.IELS[4:0] bits and the IELSRn.DTCE bit to 0
 - If the IRQ pin is to be used for DTC activation, set the IELSRn.IELS[4:0] bits and the IELSRn.DTCE bit to 1.
 - If the IRQ pin is to be used for DMAC activation, set the DELSRn.DELS[7:0] bits.

16.5.6 Non-Maskable Interrupt Operation

The following sources can trigger a non-maskable interrupt:

- NMI pin interrupt
- Oscillation stop detection interrupt
- WDT underflow/refresh error interrupt
- IWDT underflow/refresh error interrupt
- Voltage monitor 1 interrupt
- Voltage monitor BAT interrupt
- MPU bus master error interrupt
- MPU bus slave error interrupt

- CPU stack pointer monitor interrupt

Non-maskable interrupts can only be used with the CPU, not to activate the DTC or DMAC. Non-maskable interrupts take precedence over all other interrupts. The non-maskable interrupt states can be verified in the Non-Maskable Interrupt Status Register (NMISR). Confirm that all bits in the NMISR are 0 before returning from the NMI handler.

Non-maskable interrupts are disabled by default. To use non-maskable interrupts, use the following procedure:

To use the NMI pin, use the following steps:

1. Set the NMIMD bit of NMICR register.
2. Write 1 to the NMICLR.NMICLR bit to clear the NMISR.NMIST flag to 0.
3. 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.

16.6 Return from Low Power Modes

[Table 16.4](#) lists the interrupt sources you can use to exit Sleep, Snooze, or Software Standby mode. For more information, see [section 13, Power-Saving Functions](#).

16.6.1 Return from Sleep Mode

To return from Sleep mode in response to an interrupt:

non-maskable interrupt

- Use the NMIER register to enable the target interrupt request.

maskable interrupt

- Select the CPU as the interrupt request destination.
- Enable the interrupt in the NVIC.

16.6.2 Return from Software Standby Mode

The ICU returns from Software Standby mode using a non-maskable interrupt or a maskable interrupt. For maskable interrupt of canceling source, see [Table 16.4](#).

To return from Software Standby mode:

Non-maskable interrupt

Use the NMIER register to enable the target interrupt request.

Maskable interrupt

- Select the interrupt source that enables return from Software Standby
- Use the WUPEN register to enable the target interrupt request
- Select the CPU as the interrupt request destination
- 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.

16.6.3 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. Set the number of the required interrupt request in SELSR0.SELS[7:0]
2. Set the value 0x04 (ICU_SNZCANCEL) in IELSRn.IELS[4:0] (n = 0, 4, 8, 12, 16, 20, 24, 28).

3. Select the CPU as the interrupt request destination.
4. Enable the interrupt in the NVIC.

16.7 Usage Notes

16.7.1 When using WFI instruction with non-maskable interrupt

When executing a WFI instruction, always make sure that the status flags in the NMISR register are all 0.

16.7.2 Notes on PCLKB

When operating in a mode other than software standby mode, interrupts from the WDT / IWDG may be missed, so set the PCLKB frequency to 32 KHz or more.

16.7.3 Notes on DMAC/DTC activation source setting

Do not specify the same event number as the activation factor for DMAC and DTC.

16.7.4 Interrupt at Module-stop

For the CCC, RTC, WUPT, AGT, and AGTW with a built-in counter, even if the module is set to stop (MSTP = 1), the counter will not stop when the counter is set to operate.

However, since the interrupt circuit is stopped, do not use the interrupt function in the module stop state.

16.8 Reference

- Cortex™-M0+ Technical Reference Manual (ARM DDI 0484C)

17. Buses

17.1 Overview

Table 17.1 lists the bus specifications, Figure 17.1 shows the bus configuration, and Table 17.2 lists the addresses assigned for each bus.

Table 17.1 Bus specifications

Bus Type			Description
Bus masters	Main buses	System bus (CPU)	Connected to CPU
		DMA bus	Connected to DMAC/DTC
Bus slaves	Memory buses	Memory bus 1 (code flash memory)	Connected to code flash memory
		Memory bus 2 (SRAM0)	Connected to SRAM0
		Memory bus 3 (SRAM1)	Connected to SRAM1
	Internal peripheral buses	Internal peripheral bus 1	Connected to CPU and system control
		Internal peripheral bus 3	Connected to peripheral modules (I/O Ports, ELC, SCI2 to SCI5, SPI, POE, RTC, WDT, IWDTC, CAC, TMR, WUPT, IIC, DOC, GPT, ADC14, TSN)
		Internal peripheral bus 4	Connected to peripheral modules (SCI0, SCI1, MLCD, GDT, IrDA, SPI, and CRC) and TSIP-Lite
		Internal peripheral bus 5	Connected to peripheral modules (KINT, CCC, AGT, AGTW, LST, DIL, DIV, VREF)
		Internal peripheral bus 9	Connected to the flash memory control registers
	External bus	QSPI area	Connected to external SPI devices (QSPI area)

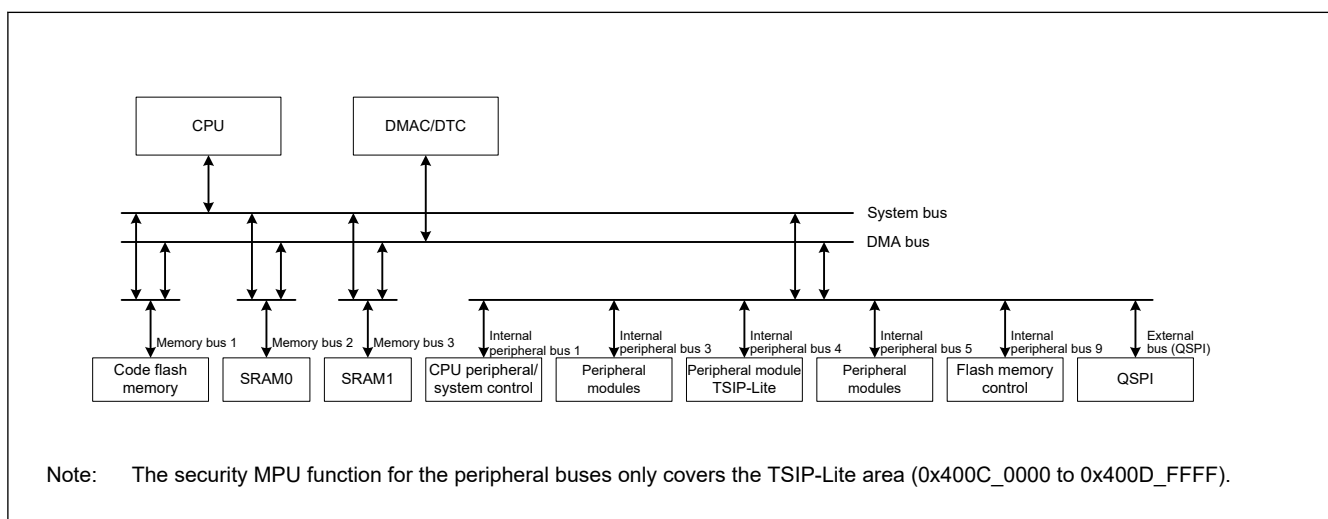


Figure 17.1 Bus configuration

Table 17.2 Addresses assigned for each bus (1 of 2)

Addresses	Slave Buses	Area
0x0000_0000 to 0x0003_FFFF	Memory bus 1	Code flash memory
0x2000_0000 to 0x2000_7FFF	Memory bus 2	SRAM0 (32 KB)
0x2000_8000 to 0x2001_FFFF	Memory bus 3	SRAM1 (96 KB)
0x4000_0000 to 0x4001_8FFF	Internal peripheral bus 1	CPU and system control registers
0x4001_9000 to 0x4001_9FFF	Memory bus 2	Peripheral I/O registers (MTB)
0x4001_A000 to 0x4001_FFFF	Internal peripheral bus 1	CPU and system control registers

Table 17.2 Addresses assigned for each bus (2 of 2)

Addresses	Slave Buses	Area
0x4004_0000 to 0x4005_FFFF	Internal peripheral bus 3	Peripheral I/O registers (peripheral modules)
0x4006_0000 to 0x4007_FFFF	Internal peripheral bus 4	
0x4008_0000 to 0x400B_FFFF	Internal peripheral bus 5	
0x400C_0000 to 0x400D_FFFF	Internal peripheral bus 4	Peripheral I/O registers (TSIP-Lite)
0x4010_0000 to 0x407F_FFFF	Internal peripheral bus 9	Peripheral I/O registers (flash memory control)
0x6000_0000 to 0x67FF_FFFF	External bus	External SPI devices (QSPI area)

17.2 Description of Buses

17.2.1 Bus Master

The bus master consist of the system bus and DMA bus. The system bus and DMA bus are connected to the following:

- Code flash memory
- SRAM0
- SRAM1
- CPU and system control
- Peripheral modules / TSIP-Lite
- Flash memory control
- QSPI

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

The DMA bus is used for DMA transfer and DTC transfer. Arbitration for the DMA bus mastership is performed between the DMAC and DTC in the fixed-priority order as follows:

DMAC0 > DMAC1 > DMAC2 > DMAC3 > DTC

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.

17.2.2 Bus Slave

For connections from the main bus to the slave interfaces, see the slave interfaces in [section 17.1. Overview](#).

Bus access from the system bus and DMA bus is arbitrated and has the following fixed priority order:

DMA bus > system bus

Different master and slave transfer combinations can proceed simultaneously.

17.2.3 Parallel Operations

Parallel operations are possible when different bus masters request access to different slave modules. [Figure 17.2](#) shows an example of parallel operations. In this example, the CPU uses the instruction and operand buses for simultaneous access to the flash and SRAM, respectively. Additionally, the DMAC or DTC simultaneously uses the DMA bus for access to a peripheral bus during access to the flash and SRAM by the CPU.

Note: The internal peripheral buses and external bus cannot be accessed from the bus masters at the same time.

Although five types of internal peripheral buses and one type of external bus are provided, the bus masters can access only either of an internal peripheral bus or external bus.

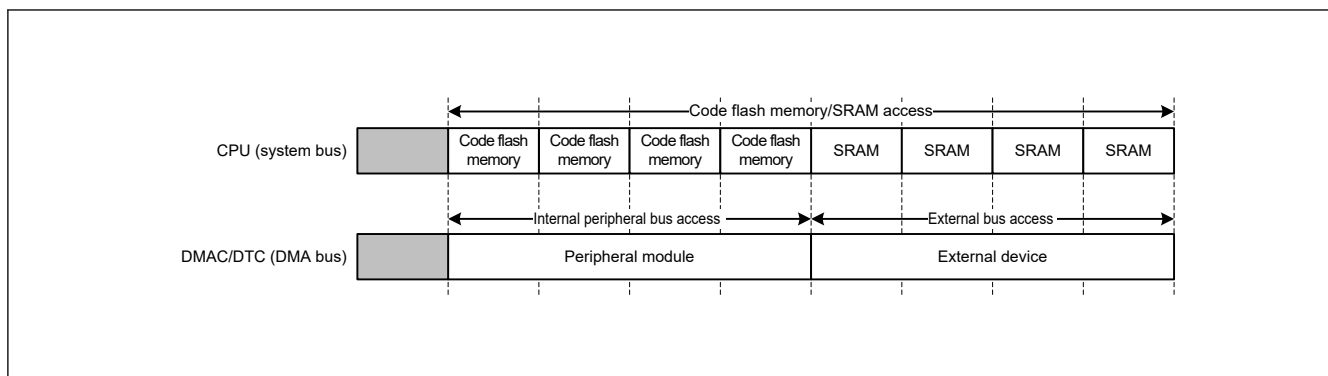


Figure 17.2 Example of parallel operations

17.3 Register Descriptions

17.3.1 BUSMCNTx : Bus Master Control Register x (x = SYS, DMA)

Base address: BUS = 0x4000_3000

Offset address: 0x1008 (BUSMCNTSYS)
0x100C (BUSMCNTDMA)

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

Bit	Symbol	Function	R/W
14:0	—	These bits are read as 0. The write value should be 0.	R/W
15	IERES	Ignore Error Responses 0: A bus error is reported. 1: A bus error is not reported.	R/W

IERES bit (Ignore Error Responses)

When IERES bit is set to 1, no hard fault occurs even if a bus error is detected.

Table 17.3 lists the registers associated with each bus type. Table 17.4 shows the bus error operation depending on the IERES bit value.

Table 17.3 Associations between bus types and registers

Bus type	Bus Master Control Register	Bus Error Address Register	Bus Error Status Register
System bus (CPU)	BUSMCNTSYS	BUS3ERRADD	BUS3ERRSTAT
DMA bus (DMAC / DTC)	BUSMCNTDMA	BUS4ERRADD	BUS4ERRSTAT

Table 17.4 Bus Error Behaviors with IERES Bit Values (1 of 2)

Bus Error	Bus Master	IERES Bit	Retention of the Address in the BUSnERRADD Register	Retention of the Error Flag in the BUSnERRSTAT Register	Hard Fault	Reset or Nonmaskable Interrupt
Illicit address access	System	0	Retained	Retained	Occurs	Does not occur
		1	Not retained	Not retained	Does not occur	Does not occur
	DMA	0	Retained	Retained	Does not occur	Does not occur
		1	Not retained	Not retained	Does not occur	Does not occur
Bus master MPU error	DMA	0	Retained	Retained	Does not occur	Occurs
		1	Retained	Retained	Does not occur	Occurs

Table 17.4 Bus Error Behaviors with IERES Bit Values (2 of 2)

Bus Error	Bus Master	IERES Bit	Retention of the Address in the BUSnERRADD Register	Retention of the Error Flag in the BUSnERRSTAT Register	Hard Fault	Reset or Nonmaskable Interrupt
Bus slave MPU error	System	0	Retained	Retained	Occurs	Occurs
		1	Retained	Retained	Does not occur	Occurs
	DMA	0	Retained	Retained	Does not occur	Occurs
		1	Retained	Retained	Does not occur	Occurs

17.3.2 BUSnERRADD : Bus Error Address Register n (n = 3, 4)

Base address: BUS = 0x4000_3000

Offset address: 0x1820 (n = 3)
0x1830 (n = 4)

Bit position: 31

0

Bit field:

BERAD[31:0]

Value after reset: x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x

Bit	Symbol	Function	R/W
31:0	BERAD[31:0]	Bus Error Address When a bus error occurs, these bits store the error address.	R

Note: BUSnERRADD is only cleared by resets other than MPU-related resets. For more information, see [section 6, Resets](#), and [section 18, Memory Protection Unit \(MPU\)](#).

Table 17.3 lists the registers associated with each bus type.

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

The BERAD[31:0] bits store the accessed address when a bus error occurred. For more information, see the description of the ERRSTAT flag in [section 17.3.3. BUSnERRSTAT : BUS Error Status Register n \(n = 3, 4\)](#) and [section 17.4. Bus Error Monitoring Section](#).

The value of the BUSnERRADD.BERAD[31:0] bits (n = 3, 4) is valid only when the BUSnERRSTAT.ERRSTAT flag (n = 3, 4) is set to 1.

17.3.3 BUSnERRSTAT : BUS Error Status Register n (n = 3, 4)

Base address: BUS = 0x4000_3000

Offset address: 0x1824 (n = 3)
0x1834 (n = 4)

Bit position: 7 6 5 4 3 2 1 0

Bit field:

ERRS TAT	—	—	—	—	—	—	ACCS TAT

Value after reset: 0 0 0 0 0 0 0 x

Bit	Symbol	Function	R/W
0	ACCSTAT	Error Access Status Access status when the error occurred: 0: Read access 1: Write access	R
6:1	—	These bits are read as 0.	R
7	ERRSTAT	Bus Error Status 0: No bus error occurred. 1: Bus error occurred.	R

Note: BUSnERRADD is only cleared by resets other than MPU-related resets. For more information, see [section 6, Resets](#), and [section 18, Memory Protection Unit \(MPU\)](#).

[Table 17.3](#) lists the registers associated with each bus type.

ACCSTAT flag (Error Access Status)

The ACCSTAT flag indicates the access status, write or read access, when a bus error occurs. For more information, see the description of the BUSnERRSTAT.ERRSTAT flag and [section 17.4. Bus Error Monitoring Section](#).

The value is valid only when the BUSnERRSTAT.ERRSTAT flag (n = 3, 4) is set to 1.

ERRSTAT flag (Bus Error Status)

The ERRSTAT flag indicates whether a bus error occurred. When a bus error occurs, the access address and status of write or read access are stored. The BUSnERRSTAT.ERRSTAT flag (n = 3, 4) is set to 1.

For more information on bus errors, see [section 17.4. Bus Error Monitoring Section](#) and [section 18, Memory Protection Unit \(MPU\)](#).

17.4 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.

17.4.1 Error Type that Occurs by Bus

Three types of errors can occur on each bus:

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

[section 17.4.3. Conditions for issuing illegal Address Access Errors](#) lists the address ranges where access leads to illegal address access errors. The reserved area in the slave does not trigger an illegal address access error.

For more information on bus master MPU and bus slave MPU, see [section 18, Memory Protection Unit \(MPU\)](#).

17.4.2 Operation when a Bus Error Occurs

When a bus error occurs, operation is not guaranteed and the error is returned to the requesting master IP. The bus error information occurred in each master is stored in the BUSnERRADD and BUSnERRSTAT registers. These registers must be cleared by reset only. For more information, see [section 17.3.2. BUSnERRADD : Bus Error Address Register n \(n = 3, 4\)](#) and [section 17.3.3. BUSnERRSTAT : BUS Error Status Register n \(n = 3, 4\)](#).

Note: DMAC / DTC does not receive bus errors. If the DMAC or DTC accesses the bus, the transfer continues.

17.4.3 Conditions for issuing illegal Address Access Errors

[Table 17.5](#) lists the address spaces for each bus that issue illegal address access errors.

Table 17.5 Conditions for issuing illegal address access errors (1 of 2)

Addresses	Area	Bus Master	
		System Bus	DMA Bus
0x0000_0000 to 0x0003_FFFF	Memory bus 1 (code flash memory)	—	—
0x0200_0000 to 0x027F_FFFF	Memory mirror area	MMF enabled: — MMF disabled: E	E
0x0280_0000 to 0x1FFF_FFFF	Reserved area	E	E
0x2000_0000 to 0x2000_7FFF	Memory bus 2 (SRAM0)	—	—
0x2000_8000 to 0x2001_FFFF	Memory bus 3 (SRAM1)	—	—

Table 17.5 Conditions for issuing illegal address access errors (2 of 2)

Addresses	Area	Bus Master	
		System Bus	DMA Bus
0x2002_0000 to 0x3FFF_FFFF	Reserved area	E	E
0x4000_0000 to 0x4001_8FFF	Internal peripheral bus 1	—	—
0x4001_9000 to 0x4001_9FFF	Memory bus 2 (MTB)	—	—
0x4001_A000 to 0x4001_FFFF	Internal peripheral bus 1	—	—
0x4002_0000 to 0x4003_FFFF	Reserved area	E	E
0x4004_0000 to 0x4005_FFFF	Internal peripheral bus 3	—	—
0x4006_0000 to 0x4007_FFFF	Internal peripheral bus 4	—	—
0x4008_0000 to 0x400B_FFFF	Internal peripheral bus 5	—	—
0x400C_0000 to 0x400D_FFFF	Internal peripheral bus 4 (TSIP-Lite)	—	—
0x400E_0000 to 0x400F_FFFF	Reserved area	E	E
0x4010_0000 to 0x407F_FFFF	Internal peripheral bus 9	—	—
0x4080_0000 to 0x5FFF_FFFF	Reserved area	E	E
0x6000_0000 to 0x67FF_FFFF	External bus (QSPI area)	—	—
0x6800_0000 to 0xDFFF_FFFF	Reserved area	E	E
0xE000_0000 to 0xFFFF_FFFF	Cortex®-M0+ system	—	E

Note: E : Indicates the bus where an illegal address access error occurs.

—: Indicates the bus where an illegal address access error does not occur or the bus where access does not occur.

Note: If MMF (Memory Mirror Function) is enabled, the access to mapped area (0x0200_0000 to 0x027F_FFFF) is switched to the user specific area (MMF output address = CPU output address + offset). The bus does not detect whether the MMF switched the address. Therefore, if the MMF is enabled and the CPU accesses 0x0200_0000 to 0x027F_FFFF, no error occurs (depends on the switched address). If the MMF is disabled and the CPU accesses 0x0200_0000 to 0x027F_FFFF, the bus detects an error.

17.5 References

1. *ARM® Cortex™-M0+ Devices Generic User Guide* (ARM DUI 0662B)
2. *ARM® AMBA® 3 AHB-Lite Protocol v1.0 Specification* (ARM IHI 0033A)

18. Memory Protection Unit (MPU)

18.1 Overview

The MCU has four Memory Protection Units (MPUs) and a CPU stack pointer monitor function are provided. The Arm[®] MPU is used for control of access by the CPU, and the bus master MPU is used for the masters other than the CPU. The bus slave MPU can be used to control access to slave functions through the main bus. [Table 18.1](#) shows the list of Master/Slave MPU. The CPU stack pointer monitors can be used to detect overflows and underflows of the stack pointers. The security MPU can be used to restrict access to the secure data so that this is only possible from a secure program.

[Table 18.2](#) lists the MPU specifications.

Table 18.1 List of Master/Slave MPU

Master/Slave	Description
Master	- Arm[®]MPU (CPU) - Bus master group A (DMA bus)
Slave	- Bus slave MPU (For details, see section 18.5. Bus Slave MPU.) - Security MPU (Code flash memory, SRAM, TSIP-Lite)

Table 18.2 MPU specifications

Classification	Error generator	Specifications
Illegal memory access	CPU	An exception interrupt occurs when an illegal access is tried to an undefined address space.
	CPU stack pointer monitor	Reset or non-maskable interrupts occurs when a stack pointer overflow or underflow is generated.
Memory protection	Arm MPU	- Protect area from CPU access (system bus). - MPU can set 8 regions. - Generates hard fault exception at illegal access.
	Bus master MPU	- Reset or non-maskable interrupts generate when access from bus master (except CPU) to protected area occurs. - Bus master MPU group A (DMA bus): MPU can set 4 regions.
	Bus slave MPU	Reset, Non-maskable interrupt or hard fault exception is generated when access from bus master to protected bus.
Security	Security MPU	Protect accesses from non-secure programs to the following protected regions: - Secure program: 2 regions - Secure data - Code flash memory: 1 region - SRAM: 1 region - TSIP-Lite: 1 region.

For information on error access for the Arm MPU, see [section 18.8. References](#). For information on error access for other MPUs, see [section 17.3. Register Descriptions](#) and [section 17.4. Bus Error Monitoring Section](#) in [section 17, Buses](#).

18.2 CPU Stack Pointer Monitor

The CPU stack pointer monitor detects underflows and overflows of the stack pointer. Because the Arm CPU has two stack pointers, a Main Stack Pointer (MSP) and a Process Stack Pointer (PSP), it supports two CPU stack pointer monitors. If a stack pointer underflow or overflow is detected, the CPU stack pointer monitor generates a reset or a non-maskable interrupt. The CPU stack pointer monitor is enabled by setting the Stack Pointer Monitor Enable bits in the Stack Pointer Monitor Access Control Register (MSPMPUCTL, PSPMPUCTL) to 1.

[Table 18.3](#) lists the specifications of the CPU stack pointer monitor. [Figure 18.1](#) shows a CPU stack pointer monitor block diagram and [Figure 18.2](#) shows the CPU stack pointer monitor register setting flow.

Table 18.3 CPU stack pointer monitor specifications (1 of 2)

Parameter	Description
Region to be monitored	SRAM region

Table 18.3 CPU stack pointer monitor specifications (2 of 2)

Parameter	Description
Number of regions	2 regions: • Main Stack Pointer (MSP) • Process Stack Pointer (PSP).
Address specification for individual regions	Specifying start and end addresses for individual regions
Stack pointer monitor enable or disable setting for individual regions	Enabling or disabling stack pointer monitor for individual regions
Operation on error detection	Reset or non-maskable interrupts
Register protection	Prevents illegal writing to the CPU stack pointer monitor register

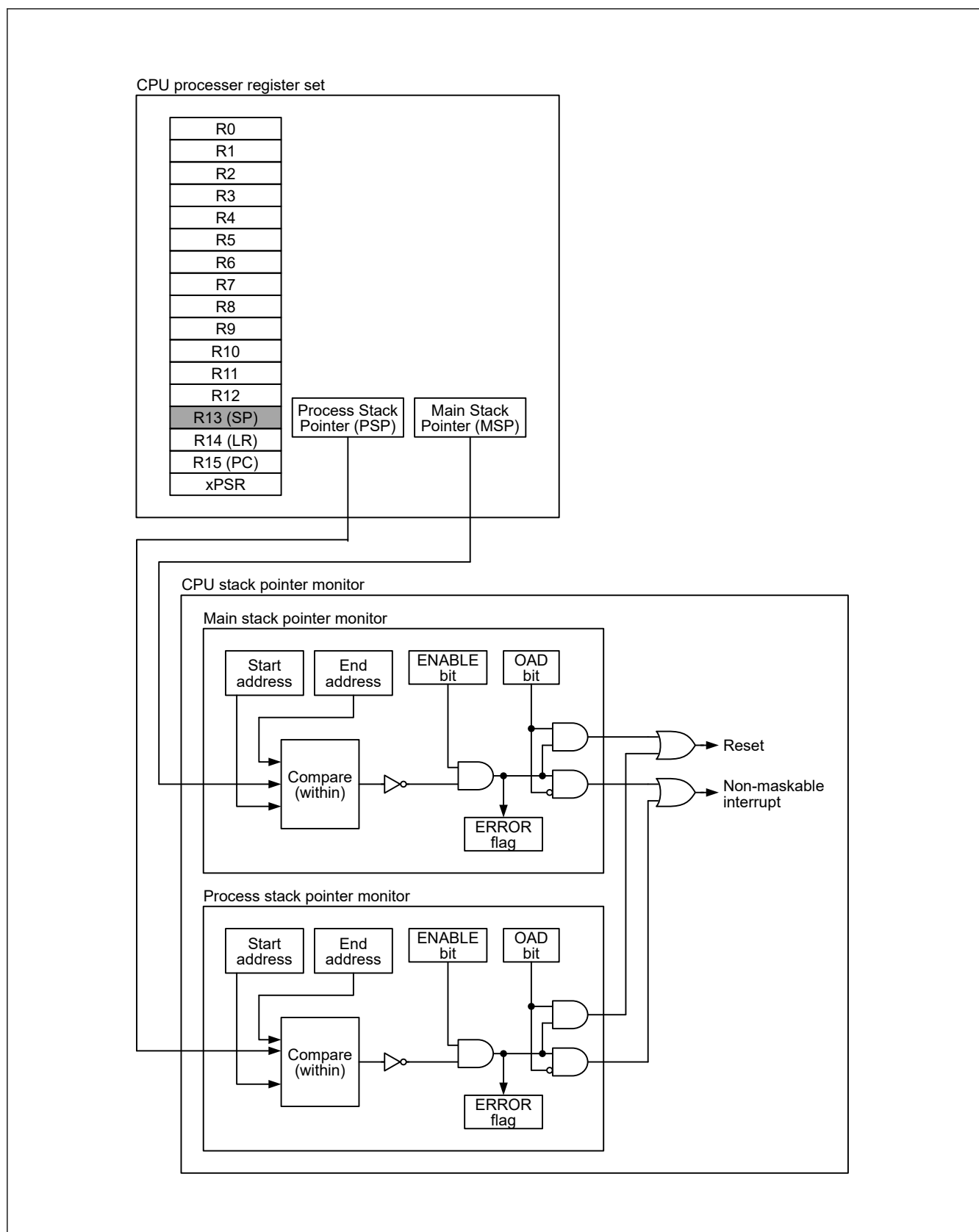


Figure 18.1 CPU stack pointer monitor block diagram

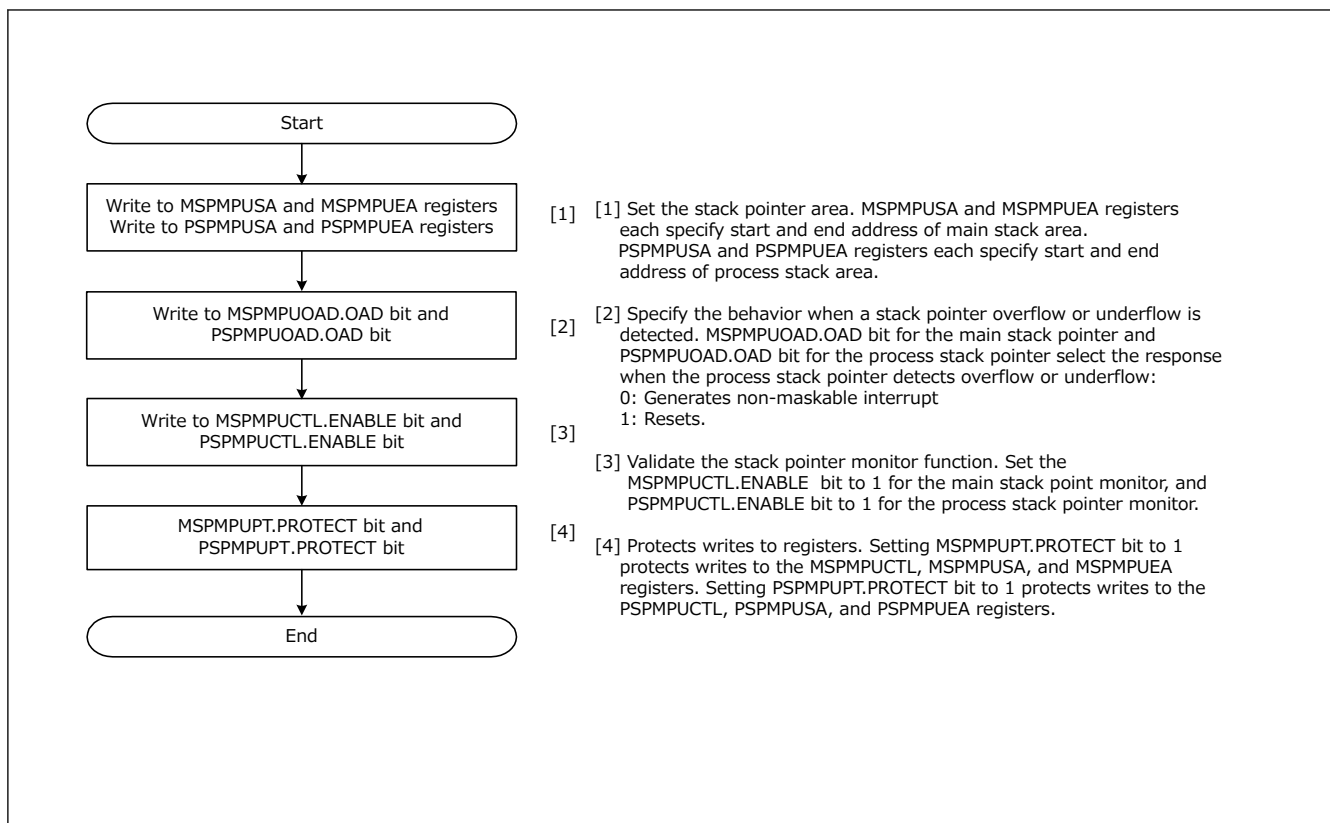


Figure 18.2 CPU stack pointer monitor register setting flow

18.2.1 Protecting the Registers

Registers related to the CPU stack pointer monitor can be protected with the associated PROTECT bit. For details, see [section 18.2.3.7. MSPMPUPT, PSPMPUPT : Stack Pointer Monitor Protection Register](#).

18.2.2 Overflow and Underflow Errors

The CPU stack pointer monitor generates an error if an overflow or underflow error is detected. [Figure 18.3](#) illustrates the concept of CPU stack pointer overflow and underflow. Set the OAD bit to select whether the error is reported as a non-maskable interrupt or reset.

The non-maskable interrupt status is indicated in ICU.NMISR.SPEST. For details, see [section 16, 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 flags in MSPMPUCTL and PSPMPUCTL registers to determine whether it is a main stack pointer monitor error or a process stack pointer monitor error.

A non-maskable interrupt is generated continuously while the stack pointer overflows or underflows. To clear the error, set the stack pointer within the specified region and then clear the non-maskable interrupt flag by setting the ICU.NMICLR.SPECLR bit to 1. Write 0 to clear the ERROR flags in the MSPMPUCTL and PSPMPUCTL registers.

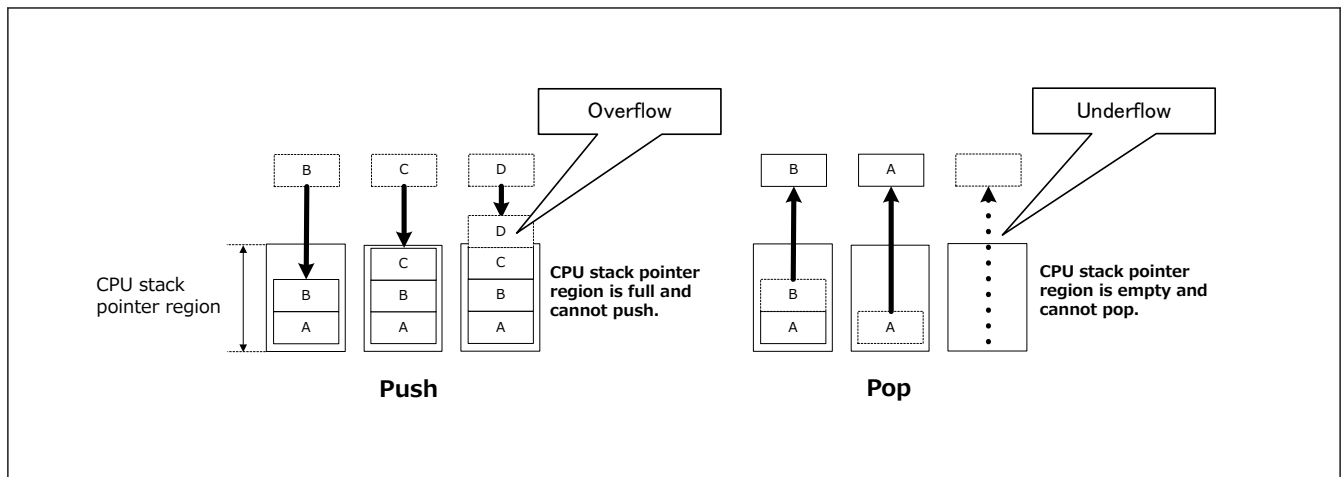


Figure 18.3 Concept of CPU stack pointer overflow and underflow

18.2.3 Register Descriptions

18.2.3.1 MSPMPUSA : Main Stack Pointer (MSP) Monitor Start Address Register

Base address: RMPU = 0x4000 0000

Offset address: 0xD08

Bit position: 31

1 0

Bit field

MSPMPUSA[31:0]

[illegible]

Bit	Symbol	Function	R/W
31:0	MSPMPUSA[31:0]	Region Start Address Address where the region starts, for use in region determination. The lower 2 bits should be 0.	R/W

Note: Set this register when the MSPMPUCTL.ENABLE = 0 (Stack pointer monitor is disabled).

The MSPMPUSA and MSPMPUEA registers specify the CPU stack region of SRAM. For SRAM area to be covered, see [section 4.1. Address Space](#).

18.2.3.2 MSPMPUEA : Main Stack Pointer (MSP) Monitor End Address Register

Base address: RMPU = 0x4000 0000

Offset address: 0xD0C

Bit position: 31

1 0

Bit field:

MSPMPUEA[31:0]

[illegible]

Bit	Symbol	Function	R/W
31:0	MSPMPUEA[31:0]	Region End Address Address where the region ends, for use in region determination. The lower 2 bits should be 1. For SRAM area to be covered, see section 4.1. Address Space.	R/W

Note: Set this register when the MSPMPUCTL.ENABLE = 0 (Stack pointer monitor is disabled).

Note: Set MSPMPUOAD register when the MSPMPUCTL.ENABLE = 0 (Stack pointer monitor is disabled). Set PSPMPUOAD register when the PSPMPUCTL.ENABLE = 0 (Stack pointer monitor is disabled).

Note 1. Write data is not retained.

OAD bit (Operation after Detection)

The OAD bit selects either a reset or a non-maskable interrupt when a stack pointer underflow or overflow is detected by the CPU stack pointer monitor.

The main and the process stack pointer monitors each uses an OAD bit to determine which signal is generated when a stack pointer underflow or overflow is detected. Write 0xA5 in KEY[7:0] bits in halfword access simultaneously when setting the OAD bit.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the OAD bit. When writing to the OAD bit, simultaneously write 0xA5 to KEY[7:0] bits. When values other than 0xA5 are written to the KEY[7:0] bits, the OAD bit is not updated. The KEY[7:0] bits are always read as 0x00.

18.2.3.6 MSPMPUCTL, PSPMPUCTL : Stack Pointer Monitor Access Control Register

Base address: RMPU = 0x4000_0000

Offset address: 0xD04 (MSPMPUCTL)
0xD14 (PSPMPUCTL)

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

Bit	Symbol	Function	R/W
0	ENABLE	Stack Pointer Monitor Enable 0: Stack pointer monitor is disabled 1: Stack pointer monitor is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	ERROR	Stack Pointer Monitor Error Flag 0: Stack pointer has not overflowed or underflowed 1: Stack pointer has overflowed or underflowed	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value depends on the reset generation sources.

ENABLE bit (Stack Pointer Monitor Enable)

The ENABLE bit enables or disables the stack pointer monitor function, independently set for the main stack pointer monitor and the process stack pointer monitor.

When the MSPMPUCTL.ENABLE bit is set to 1, the following registers are available:

- MSPMPUSA
- MSPMPUEA
- MSPMPUOAD.

When the PSPMPUCTL.ENABLE bit is set to 1, the following registers are available:

- PSPMPUSA
- PSPMPUEA
- PSPMPUOAD.

ERROR flag (Stack Pointer Monitor Error Flag)

The ERROR flag indicates the status of the stack pointer monitor. Each stack pointer monitor has an independent ERROR flag.

[Setting condition]

- Overflow or underflow of the stack pointer.

[Clearing condition]

- 0 is written to this flag
- A reset other than the stack pointer error reset. (For the details of reset factors, see [section 6, Resets.](#))

Note: Only 0 can be written to the ERROR flag.

18.2.3.7 MSPMPUPT, PSPMPUPT : Stack Pointer Monitor Protection Register

Base address: RMPU = 0x4000_0000

Offset address: 0xD06 (MSPMPUPT)
0xD16 (PSPMPUPT)

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

Bit	Symbol	Function	R/W
0	PROTECT	Protection of Register 0: Stack pointer monitor register writes are permitted. 1: Stack pointer monitor register writes are protected. Reads are permitted	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writes to the PROTECT bit	R/W ¹

Note 1. Write data is not retained.

PROTECT bit (Protection of Register)

The PROTECT bit enables or disables writes to the associated registers to be protected, independently set for the main stack pointer monitor and the process stack pointer monitor.

MSPMPUPT.PROTECT controls the following main stack pointer protection registers:

- MSPMPUCTL
- MSPMPUSA
- MSPMPUEA.

PSPMPUPT.PROTECT controls the following process stack pointer protection registers:

- PSPMPUCTL
- PSPMPUSA
- PSPMPUEA.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the PROTECT bit simultaneously. When writing to the PROTECT bit, write 0xA5 to KEY[7:0] bits. When values other than 0xA5 are written to the KEY[7:0] bits, the PROTECT bit is not updated.

The KEY[7:0] bits are always read as 0.

18.3 Arm MPU

The Arm MCU monitors the addresses accessed by the CPU across the entire address space (0x0000_0000 to 0xFFFF_FFFF) and provides support for:

- 8 protected regions
Each protected region can be divided into 8 sub-regions and can set enable or disable to each sub-region.
- Overlapping protected regions, with ascending region priority:
7 = highest priority
0 = lowest priority.
- Setting access permissions to protected region (Read, Write, Execution)

Arm MPU mismatches and permission violations invoke the programmable-priority MemManage fault (Hard Fault) handler. For details, see reference 2. in [section 18.8. References](#).

18.3.1 Priorities to the Protection Region

The Arm[®] MPU can set multipul overlapping protection regions. For overlapping protection regions, the access rights are determined according to the priority shown in [Table 18.4](#).

Table 18.4 Priorities to the Protection Region

Protection region	Priority
Region 7	High
Region 6	↓
Region 5	
Region 4	
Region 3	
Region 2	
Region 1	
Region 0	Low

18.4 Bus Master MPU

The bus master MPU monitors the addresses accessed by the bus masters (DMAC/DTC) in the entire address space (0x0000_0000 to 0xFFFF_FFFF).

The following procedure allows you to monitor access by the DMAC and DTC to the protected regions by using the bus master MPU.

1. Apply protection to the whole address range (0x0000_0000 to 0xFFFF_FFFF).
2. Specify the address ranges and access attributes (read or write) of the regions that will require access and release the protection of those regions.
3. Step 2 allows you to specify up to four regions to be released from protection. Access to the regions that are not released from protection is prohibited.

If access to a protected region is detected, the bus master MPU generates an internal reset or a non-maskable interrupt in response to the detection of access by the DMAC or DTC to a protected regions. For information on error access, see [section 17.3. Register Descriptions](#) and [section 17.4. Bus Error Monitoring Section](#) in [section 17, Buses](#).

The access control information for each area consists of protected/not-protected to read or write.

[Table 18.5](#) lists the specifications of the bus master MPU, and [Figure 18.4](#) shows a block diagram.

Table 18.5 Bus master MPU specifications (1 of 2)

Parameter	Description
Protected master groups	• Bus master MPU group A: DMA bus

Table 18.5 Bus master MPU specifications (2 of 2)

Parameter	Description
Protected regions	0x0000_0000 to 0xFFFF_FFFF
Number of regions	Bus master MPU group A: 4 regions (DMA bus)
Address specification for individual regions	Specifying start and end address for individual regions
Enable or disable setting for memory protection in individual regions	Enabling or disabling setting for the associated region
Access-control settings for individual regions	Protection or no-protection settings for read and write
Operation on error detection	Reset or non-maskable interrupts
Register protection	Protecting registers from illegal writes

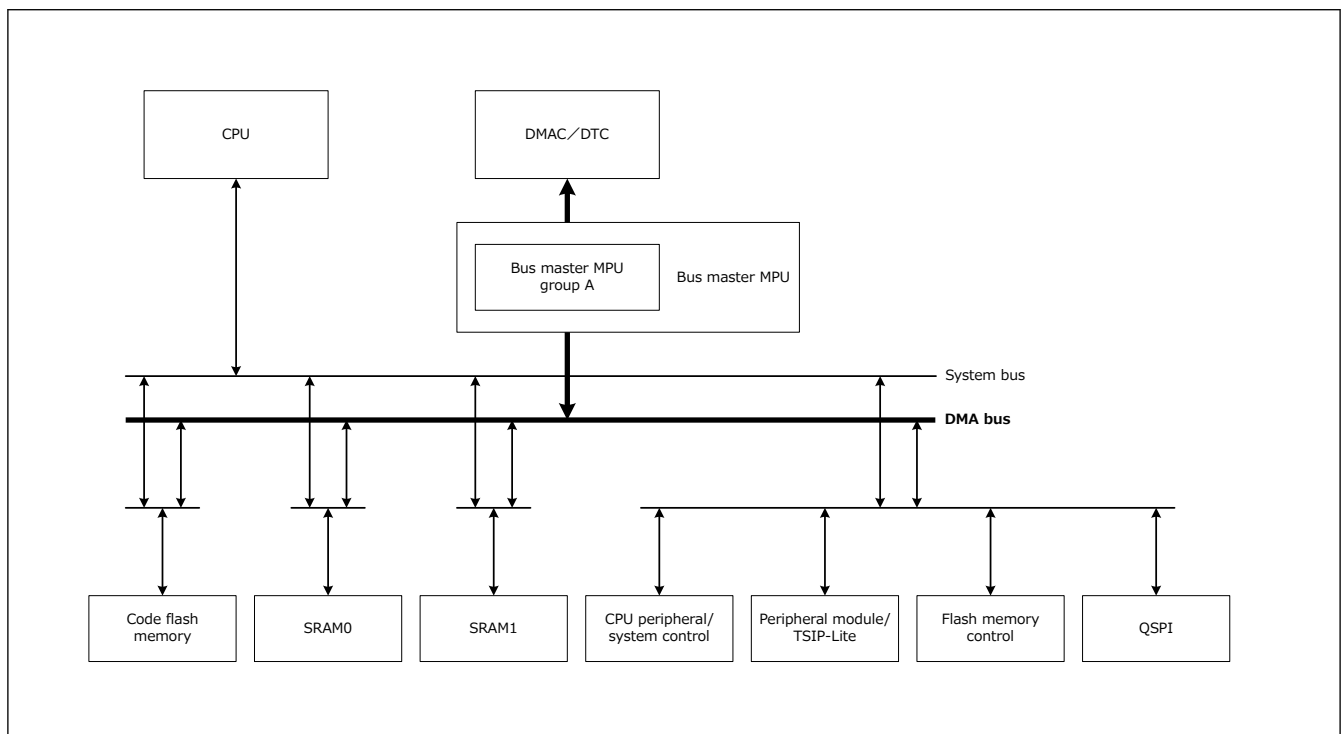


Figure 18.4 MPU bus master block diagram

18.4.1 Register Descriptions

18.4.1.1 MMPUSAn : Group A Region n Start Address Register (n = 0 to 3)

Base address: RMPU = 0x4000_0000

Offset address: $0x204 + (0x010 \times n)$

Bit position: 31

1 0

Bit field

MMPUSA[31:0]

[illegible]

Bit	Symbol	Function	R/W
31:0	MMPUSA[31:0]	Region Start Address Address where the region starts, for use in region determination. The lower 2 bits should be 0.	R/W

Base address: RMPU = 0x4000_0000

Offset address: $0x208 + 0x010 \times n$

Bit position: 31

1 0

Bit field:

MMPUEA[31:0]

Value after reset: x x x x x x x x x x x x x x x x x x x x x x x x x x x x 1 1

Bit	Symbol	Function	R/W
31:0	MMPUEA[31:0]	Region End Address Address where the region ends, for use in region determination. The lower 2 bits should be 1.	R/W

Base address: RMPU = 0x4000_0000
Offset address: $0x200 + 0x010 \times n$

Bit position:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:

—	—	—	—	—	—	—	—	—	—	—	—	—	WP	RP	ENAB LE

[illegible]

Bit	Symbol	Function	R/W
0	ENABLE	Region Enable 0: Group A region n disabled 1: Group A region n enabled	R/W
1	RP	Read Protection 0: Read permission 1: Read protection	R/W
2	WP	Write Protection 0: Write permission 1: Write protection	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

ENABLE bit (Region Enable)

When the ENABLE bit is set to 1, the RP bit and the WP bit can be set to permit or protect access to the region that is set in MMPUSAn and MMPUEAn. When the ENABLE bit is set to 0, no region is specified for group A region n access.

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.

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 18.6 Protection enable/disable settings (1 of 2)

MMPUACAn.ENABLE	MMPUACAn.RP	MMPUACAn.WP	Access	Protection enable/disable
0	—	—	—	Protection for all group A region is disabled.

Table 18.6 Protection enable/disable settings (2 of 2)

MMPUACAn.ENABLE	MMPUACAn.RP	MMPUACAn.WP	Access	Protection enable/disable
1	0	0	Read	Protection for the specified region is disabled.
			Write	Protection for the specified region is disabled.
	0	1	Read	Protection for the specified region is disabled.
			Write	Protection for the specified region is enabled.
	1	0	Read	Protection for the specified region is enabled.
			Write	Protection for the specified region is disabled.
	1	1	Read	Protection for the specified region is enabled.
			Write	Protection for the specified region is enabled.

Note: The bus master MPU generates reset or interrupts when access to protection enable region is detected. It does not cancel the access behavior.

18.4.1.4 MMPUCTLA : Bus Master MPU Control Register

Base address: RMPU = 0x4000_0000

Offset address: 0x000

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

Bit	Symbol	Function	R/W
0	ENABLE	Master Group Enable 0: Master group A disabled 1: Master group A enabled	R/W
1	OAD	Operation After Detection 0: Non-maskable interrupt 1: Reset	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writes to the OAD and ENABLE bits	R/W ^{*1}

Note 1. Write data is not retained.

ENABLE bit (Master Group Enable)

The ENABLE bit enables or disables the bus master MPU function of master group A.

When this bit is set to 1, MMPUACAn is available. When this bit is set to 0, MMPUACAn is unavailable, including permission for all regions. When writing to the ENABLE bit simultaneously, write 0xA5 to the KEY[7:0] bits using halfword access.

OAD bit (Operation After Detection)

The OAD bit generates either a reset or non-maskable interrupt when access to the protected region is detected by the bus master MPU. When writing to the OAD bit simultaneously, write 0xA5 to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the ENABLE and OAD bits. When writing to the ENABLE and OAD bits simultaneously, write 0xA5 to the KEY[7:0] bits. When other values are written to the KEY[7:0] bits, the ENABLE and the OAD bits are not updated. The KEY[7:0] bits are always read as 0x00.

18.4.1.5 MMPUPTA : Group A Protection of Register

Base address: RMPU = 0x4000_0000

Offset address: 0x102

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

Bit	Symbol	Function	R/W
0	PROTECT	Protection of Register 0: All bus master MPU group A register writes are permitted. 1: All bus master MPU group A register writes are protected. Reads are permitted.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writes to the PROTECT bit	R/W ^{*1}

Note 1. Write data is not retained.

PROTECT bit (Protection of Register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

The following registers are protected by MMPUPTA.PROTECT:

- MMPUSAn
- MMPUEAn
- MMPUACAn
- MMPUCTLA.

When the PROTECT bit is set simultaneously, write 0xA5 to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the PROTECT bit. When writing to the PROTECT bit simultaneously, write 0xA5 to the KEY[7:0] bits. When other values are written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

18.4.2 Operation

18.4.2.1 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 from the DMAC or DTC is detected, the bus master MPU generates a memory protection error.

The bus master MPU only supports the DMA bus, and the memory protection function enables protection from all access for bus master MPU group A. The protection for access to the whole memory is disabled after a reset. Setting the MMPUCTLA.ENABLE bit to 1 enables protection of the whole memory. The access protection is enabled or disabled in up to four group A regions, 0 to 3. Set the address range, enable or disable the protection for reading and writing, and enable protection by the circuits for the respective regions. In cases of overlap between an access-protected region and permitted region, or two access-protected regions, the region of overlap is access-protected, with no access allowed. When protection by the circuit for the region is disabled, access to that region is allowed.

Figure 18.5 shows the use case of a bus master MPU.

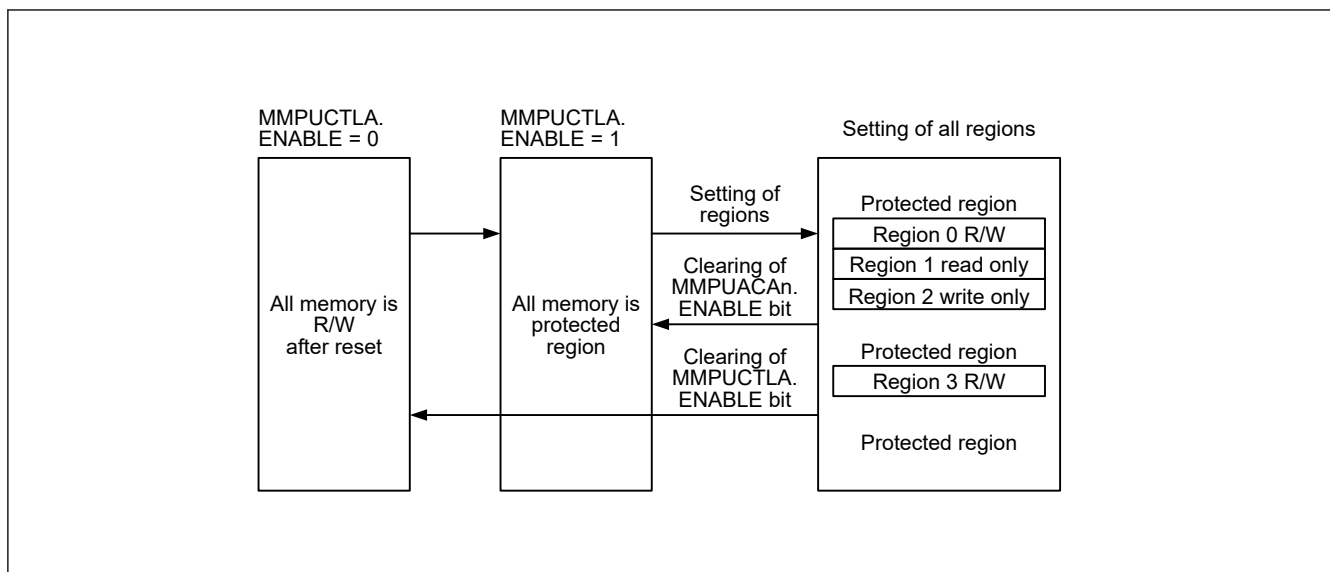


Figure 18.5 Use case of bus master MPU

Figure 18.6 shows the access permission or protection by the overlapping bus master MPU regions.

Access control for the overlapping regions is as follows:

- The overlapping region of a protected region and an unprotected region is handled as a protected region.
- The region outside of region 0 to 3 is handled as a protected region.

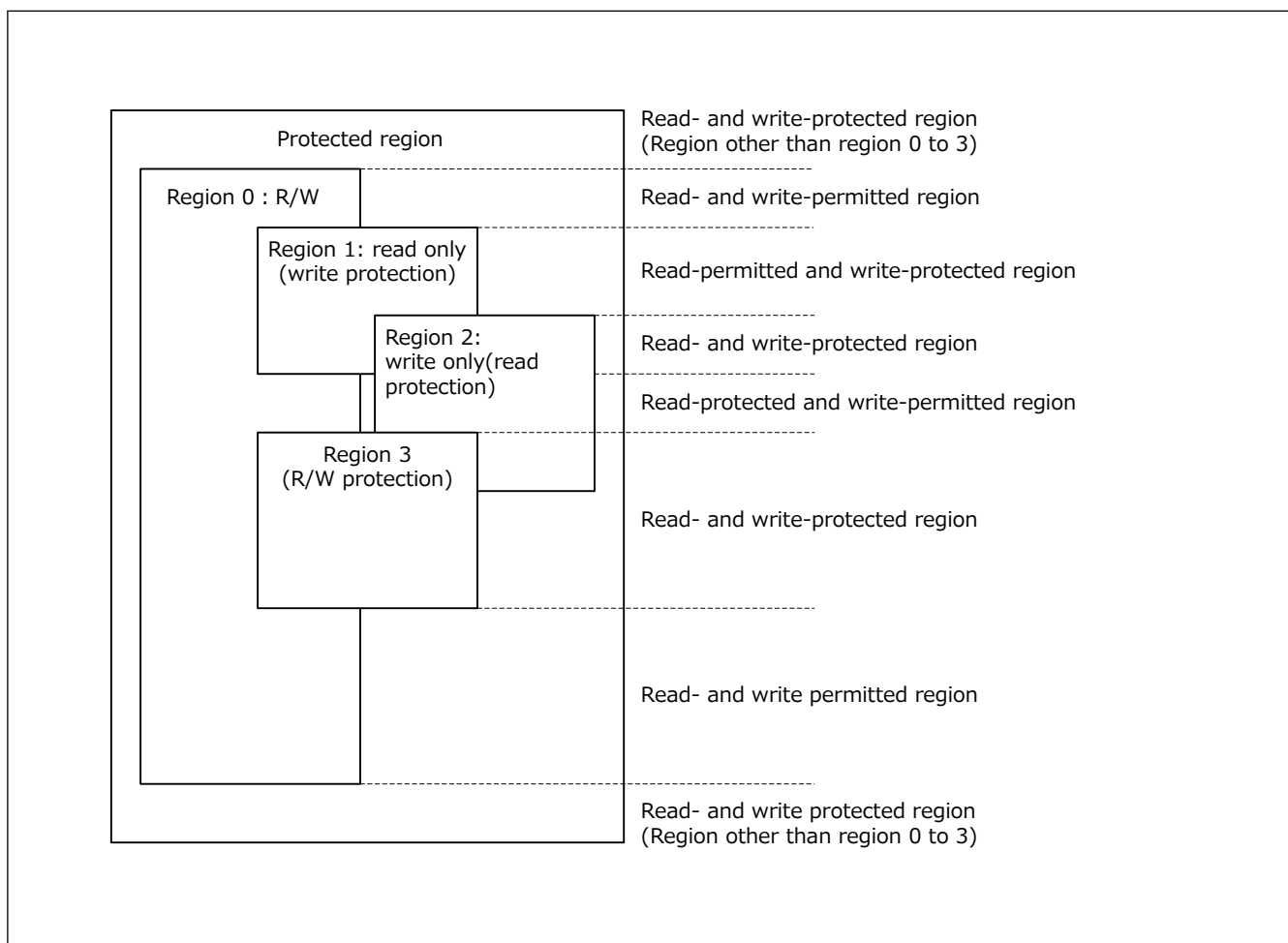


Figure 18.6 Access permission or protection by overlap of the bus master MPU regions

Figure 18.7 shows the register setting flow after reset. During this register setting, stop the bus master except the CPU.

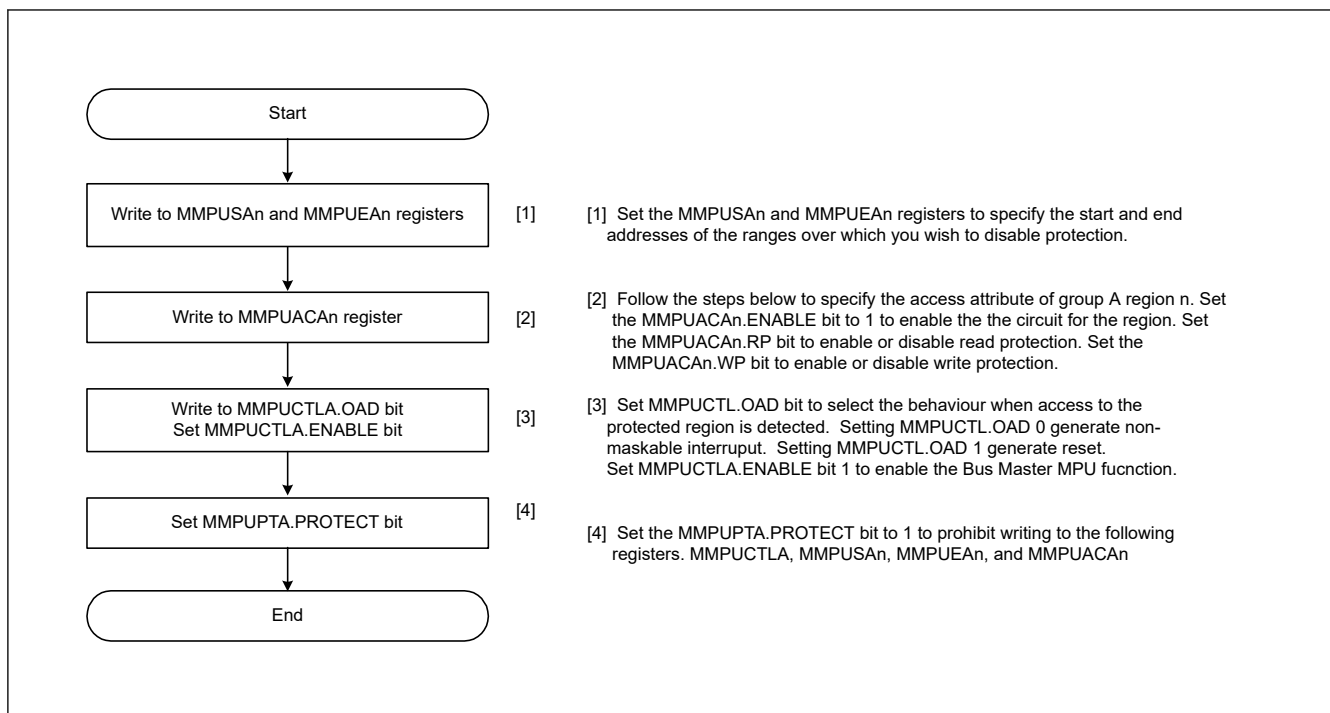


Figure 18.7 Register setting flow of bus master MPU after reset

Figure 18.8 shows the register setting flow for adding regions. During this register setting, stop the master except the CPU.

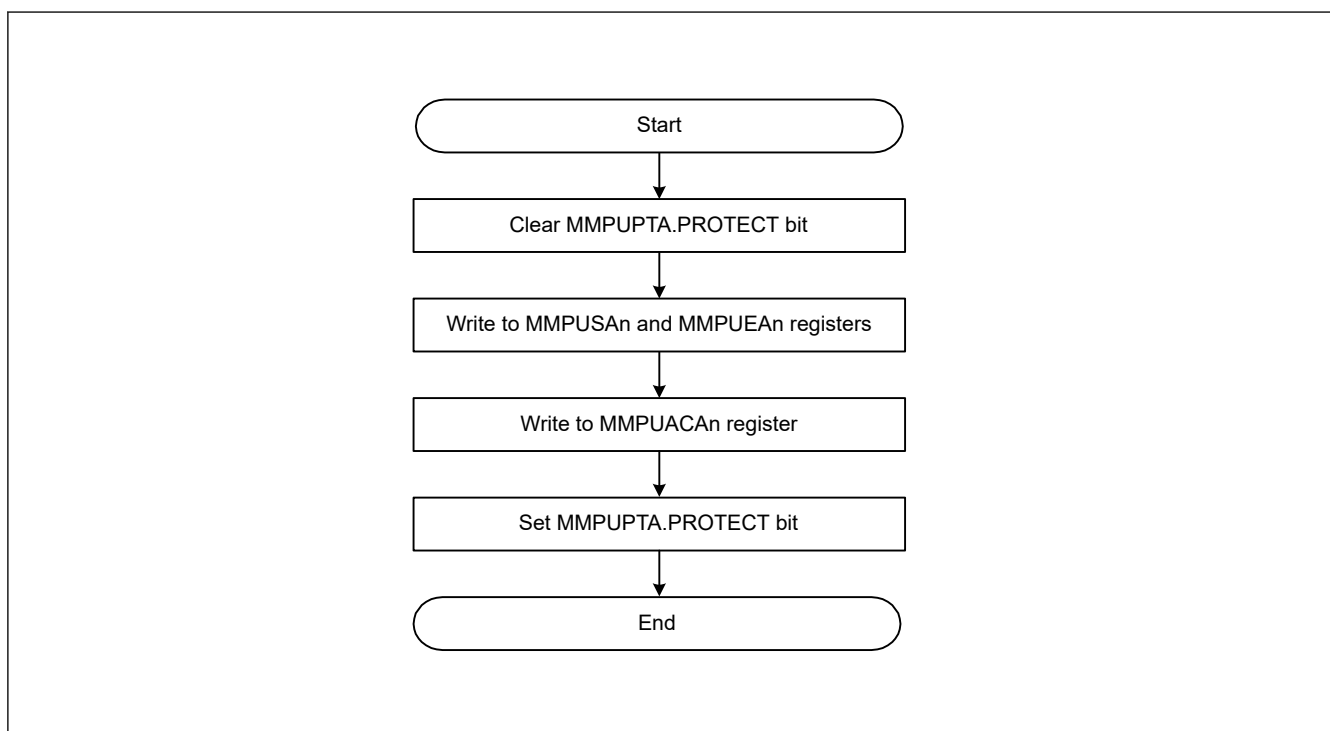


Figure 18.8 Register setting flow for region addition

18.4.2.2 Protecting the registers

To protect the registers related to the bus master MPU, set the PROTECT bit in the MMPUPTA register.

18.4.2.3 Memory protection error

If access to a protected region from the DMAC or DTC 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 16, Interrupt Controller Unit \(ICU\)](#). The reset status is indicated in SYSTEM.RSTSR1.BUSMRF. For details, see [section 6, Resets](#).

18.5 Bus Slave MPU

The bus slave MPU monitors access to the bus slave functions, such as flash memory or SRAM. The bus slave function can be accessed from two bus masters, system bus and DMA bus. The bus slave MPU has a separate protection register for each of the two bus masters, with individual access protection control. If access to a protected region is detected, the bus slave MPU generates a reset or a non-maskable interrupt, and store the bus error status, error access status, and bus error address in the I/O Registers. For details, see [section 17.3, Register Descriptions](#) and [section 17.4, Bus Error Monitoring Section](#) in [section 17, Buses](#). The supported access control information for the individual regions consists of permission to read and write.

[Table 18.7](#) lists the specifications of the bus slave MPU and [Figure 18.9](#) shows a Bus slave MPU block diagram.

Table 18.7 Specifications of bus slave MPU

Specifications	Description
Protected bus master	System bus Bus master MPU group A: DMA bus and System bus (CPU)
Protected bus slave function	Memory bus 1: Code flash memory Memory bus 2: SRAM0 Memory bus 3: SRAM1 Internal peripheral bus 1: CPU peripheral/system control Internal peripheral bus 3, 4, and 5: Peripheral module, TSIP-Lite Internal peripheral bus 9: Flash memory control External bus: QSPI
Access-control information settings for individual regions	Protection or no-protection settings for read and write
Operation on error detection	Reset or non-maskable interrupt, or exception
Protection of register	Register can be protected from illegal writes

The bus slave MPU is located on each bus slave side and controls the permission or protection of access from each bus master to each bus slave.

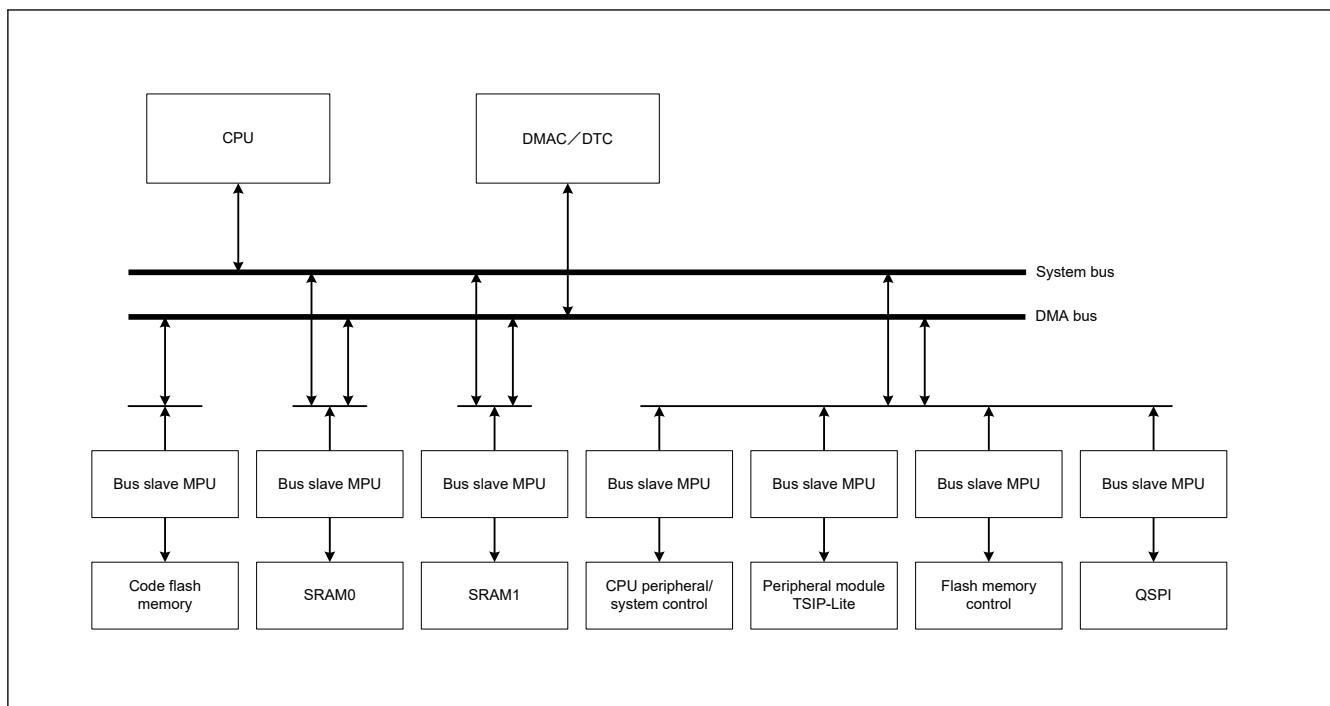


Figure 18.9 Bus slave MPU block diagram

18.5.1 Register Descriptions

18.5.1.1 SMPUMBIU : Access Control Register for Memory Bus 1

Base address: RMPU = 0x4000_0000

Offset address: 0xC10

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

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	RPGRPA	Master MPU Group A Read Protection 0: Memory protection read for master MPU group A disabled 1: Memory protection read for master MPU group A enabled	R/W
3	WPGRPA	Master MPU Group A Write Protection 0: Memory protection write for master MPU group A disabled 1: Memory protection write for master MPU group A enabled	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

RPGRPA bit (Master MPU Group A Read Protection)

The RPGRPA bit enables or disables memory protection for master MPU group A (the DMA bus) reads on memory bus 1.

WPGRPA bit (Master MPU Group A Write Protection)

The WPGRPA bit enables or disables memory protection for master MPU group A (the DMA bus) writes on memory bus 1.

18.5.1.2 SMPUSRAM0 : Access Control Register for Memory Bus 2

Base address: RMPU = 0x4000_0000

Offset address: 0xC18

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

Bit	Symbol	Function	R/W
0	RPCPU	CPU Read Protection 0: Memory protection for CPU read disabled 1: Memory protection for CPU read enabled	R/W
1	WPCPU	CPU Write Protection 0: Memory protection for CPU write disabled 1: Memory protection for CPU write enabled	R/W
2	RPGRPA	Master MPU Group A Read Protection 0: Memory protection for master MPU group A read disabled 1: Memory protection for master MPU group A read enabled	R/W
3	WPGRPA	Master MPU Group A Write Protection 0: Memory protection for master MPU group A write disabled 1: Memory protection for master MPU group A write enabled	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for CPU reads on memory bus 2.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for CPU writes on memory bus 2.

RPGRPA bit (Master MPU Group A Read Protection)

The RPGRPA bit enables or disables memory protection for master MPU group A (the DMA bus) reads on memory bus 2.

WPGRPA bit (Master MPU Group A Write Protection)

The WPGRPA bit enables or disables memory protection for master MPU group A (the DMA bus) writes on memory bus 2.

18.5.1.3 SMPUSRAM1 : Access Control Register for Memory Bus 3

Base address: RMPU = 0x4000_0000

Offset address: 0xC1C

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

Bit	Symbol	Function	R/W
0	RPCPU	CPU Read Protection 0: Memory protection for CPU read disabled 1: Memory protection for CPU read enabled	R/W
1	WPCPU	CPU Write Protection 0: Memory protection for CPU write disabled 1: Memory protection for CPU write enabled	R/W

Bit	Symbol	Function	R/W
2	RPGRPA	Master MPU Group A Read Protection 0: Memory protection for master MPU group A read disabled 1: Memory protection for master MPU group A read enabled	R/W
3	WPGRPA	Master MPU Group A Write Protection 0: Memory protection for master MPU group A write disabled 1: Memory protection for master MPU group A write enabled	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for CPU reads on memory bus 3.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for CPU writes on memory bus 3.

RPGRPA bit (Master MPU Group A Read Protection)

The RPGRPA bit enables or disables memory protection for master MPU group A (the DMA bus) reads on memory bus 3.

WPGRPA bit (Master MPU Group A Write Protection)

The WPGRPA bit enables or disables memory protection for master MPU group A (the DMA bus) writes on memory bus 3.

18.5.1.4 SMPUP0BIU : Access Control Register for Internal Peripheral Bus 1

Base address: RMPU = 0x4000_0000

Offset address: 0xC20

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

Bit	Symbol	Function	R/W
0	RPCPU	CPU Read Protection 0: Memory protection for CPU read disabled 1: Memory protection for CPU read enabled	R/W
1	WPCPU	CPU Write Protection 0: Memory protection for CPU write disabled 1: Memory protection for CPU write enabled	R/W
2	RPGRPA	Master MPU Group A Read Protection 0: Memory protection for master MPU group A read disabled 1: Memory protection for master MPU group A read enabled	R/W
3	WPGRPA	Master MPU Group A Write Protection 0: Memory protection for master MPU group A write disabled 1: Memory protection for master MPU group A write enabled	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for CPU reads on internal peripheral bus 1.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for CPU writes on internal peripheral bus 1.

RPGRPA bit (Master MPU Group A Read Protection)

The RPGRPA bit enables or disables memory protection for master MPU group A (the DMA bus) reads on internal peripheral bus 1.

WPGRPA bit (Master MPU Group A Write Protection)

The WPGRPA bit enables or disables memory protection for master MPU group A (the DMA bus) writes on internal peripheral bus 1.

Note: The read/write protection by SMPUP0BIU register is not controlled in the MTB I/O register area (0x4001_9000 to 0x4001_9FFF).

Note: When protection is set with SMPUP0BIU, MPU-related registers including this register are protected without exception. When RPCPU bit is set to 1, CPU read cannot confirm the non-maskable interrupt status (ICU.NMISR.BUSSST flag) and reset status (SYSTEM.RSTSR1.BUSSRF flag). When WPCPU bit is set to 1, CPU write cannot re-set WPCPU bit as disabled.

18.5.1.5 SMPUP2BIU : Access Control Register for Internal Peripheral Bus 3 to 5

Base address: RMPU = 0x4000_0000

Offset address: 0xC24

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

Bit	Symbol	Function	R/W
0	RPCPU	CPU Read Protection 0: Memory protection for CPU read disabled 1: Memory protection for CPU read enabled	R/W
1	WPCPU	CPU Write Protection 0: Memory protection for CPU write disabled 1: Memory protection for CPU write enabled	R/W
2	RPGRPA	Master MPU Group A Read Protection 0: Memory protection for master MPU group A read disabled 1: Memory protection for master MPU group A read enabled	R/W
3	WPGRPA	Master MPU Group A Write Protection 0: Memory protection for master MPU group A write disabled 1: Memory protection for master MPU group A write enabled	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for CPU reads on internal peripheral bus 3 to 5.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for CPU writes on internal peripheral bus 3 to 5.

RPGRPA bit (Master MPU Group A Read Protection)

The RPGRPA bit enables or disables memory protection for master MPU group A (the DMA bus) reads on internal peripheral bus 3 to 5.

WPGRPA bit (Master MPU Group A Write Protection)

The WPGRPA bit enables or disables memory protection for master MPU group A (the DMA bus) writes on internal peripheral bus 3 to 5.

18.5.1.6 SMPUFBIU : Access Control Register for Internal Peripheral Bus 9

Base address: RMPU = 0x4000_0000

Offset address: 0xC14

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

Bit	Symbol	Function	R/W
0	RPCPU	CPU Read Protection 0: Memory protection for CPU read disabled 1: Memory protection for CPU read enabled	R/W
1	WPCPU	CPU Write Protection 0: Memory protection for CPU write disabled 1: Memory protection for CPU write enabled	R/W
2	RPGRPA	Master MPU Group A Read Protection 0: Memory protection for master MPU group A read disabled 1: Memory protection for master MPU group A read enabled	R/W
3	WPGRPA	Master MPU Group A Write Protection 0: Memory protection for master MPU group A write disabled 1: Memory protection for master MPU group A write enabled	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for CPU reads on internal peripheral bus 9.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for CPU writes on internal peripheral bus 9.

RPGRPA bit (Master MPU Group A Read Protection)

The RPGRPA bit enables or disables memory protection for master MPU group A (the DMA bus) reads on internal peripheral bus 9.

WPGRPA bit (Master MPU Group A Write Protection)

The WPGRPA bit enables or disables memory protection for master MPU group A (the DMA bus) writes on internal peripheral bus 9.

18.5.1.7 SMPUEXBIU2 : Access Control Register for QSPI Area

Base address: RMPU = 0x4000_0000

Offset address: 0xC34

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

Bit	Symbol	Function	R/W
0	RPCPU	CPU Read Protection 0: Memory protection for CPU read disabled 1: Memory protection for CPU read enabled	R/W

Bit	Symbol	Function	R/W
1	WPCPU	CPU Write Protection 0: Memory protection for CPU write disabled 1: Memory protection for CPU write enabled	R/W
2	RPGRPA	Master MPU Group A Read Protection 0: Memory protection for master MPU group A read disabled 1: Memory protection for master MPU group A read enabled	R/W
3	WPGRPA	Master MPU Group A Write Protection 0: Memory protection for master MPU group A write disabled 1: Memory protection for master MPU group A write enabled	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for CPU reads on external bus.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for CPU writes on external bus.

RPGRPA bit (Master MPU Group A Read Protection)

The RPGRPA bit enables or disables memory protection for master MPU group A (the DMA bus) reads on external bus.

WPGRPA bit (Master MPU Group A Write Protection)

The WPGRPA bit enables or disables memory protection for master MPU group A (the DMA bus) writes on external bus.

18.5.1.8 SMPUCTL : Slave MPU Control Register

Base address: RMPU = 0x4000_0000

Offset address: 0xC00

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

Bit	Symbol	Function	R/W
0	OAD	Operation After Detection 0: Non-maskable interrupt 1: Reset	R/W
1	PROTECT	Protection of Register 0: All bus slave register writes are permitted 1: All bus slave register writes are protected. Reads are permitted	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writes to the OAD and PROTECT bits	R/W ^{*1}

Note 1. Write data is not retained.

OAD bit (Operation After Detection)

The OAD bit generates either a reset or non-maskable interrupt when access to the protected region is detected by the bus slave MPU. When the OAD bit is set simultaneously, write 0xA5 to the KEY[7:0] bits using halfword access.

PROTECT bit (Protection of Register)

The PROTECT bit enables or disables writes to the associated registers to be protected. SMPUCTL.PROTECT controls the following registers:

- SMPUMBIU

- SMPUFBIU
- SMPUSRAM0
- SMPUSRAM1
- SMPUP0BIU
- SMPUP2BIU
- SMPUEXBIU2

When the PROTECT bit is set, write 0xA5 to the KEY[7:0] bits simultaneously using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the OAD and PROTECT bits. When writing to the OAD and PROTECT bits simultaneously, write 0xA5 to the KEY[7:0] bits. When other values are written, the OAD and the PROTECT bits are not updated. The KEY[7:0] bits are always read as 0x00.

18.5.2 Functions

18.5.2.1 Memory protection

The bus slave MPU monitoring uses access control information that is set for the individual access control registers, whether or not access by the bus slaves violates the access control settings. If access to the protected region is detected, the bus slave MPU generates a memory protection error.

The bus slave MPU is enabled by writing 1 to the Write Protect (WPCPU or WPGRPA) bit or the Read Protect (RPCPU or RPGRPA) bit in the access control registers (SMPUMBIU, SMPUFBIU, SMPUSRAM0, SMPUSRAM1, SMPUP0BIU, SMPUP2BIU, and SMPUEXBIU2).

18.5.2.2 Protecting the registers

Registers related to the bus slave MPU can be protected with the PROTECT bit in the SMPUCTL register.

18.5.2.3 Memory protection error

If access to a protected region is detected, the bus slave MPU generates a memory protection error. Set the OAD bit to select whether the error is reported as a non-maskable interrupt or reset.

The non-maskable interrupt status is indicated in ICU.NMISR.BUSSST. For details, see [section 16, Interrupt Controller Unit \(ICU\)](#). The reset status is indicated in SYSTEM.RSTSR1.BUSSRF. For details, see [section 6, Resets](#).

18.6 Security MPU

The MCU incorporates a security MPU and has secure regions. The secure regions can be protected from non-secure program accesses. A non-secure program cannot access a protected region.

[Table 18.8](#) lists the specifications of the security MPU.

Table 18.8 Security MPU specifications

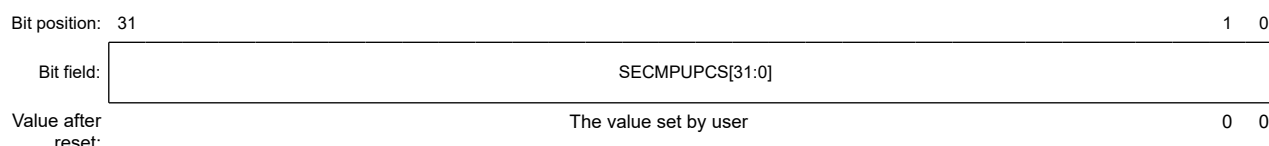
Specifications	Description
Regions to be covered by memory protection	0x0000_0000 to 0x0003_FFFF (Code flash memory) 0x2000_0000 to 0x2001_FFFF (SRAM) 0x400C_0000 to 0x400D_FFFF (Peripheral I/O register (TSIP-Lite))
Number of regions	For secure program: 2 For secure data: 3
Address specification for individual regions	Setting the address where regions start and end
Enable or disable setting for memory protection in individual regions	Settings enabled or disabled for the associated region

18.6.1 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.

18.6.1.1 SECMPUPCSn : Security MPU Program Counter Start Address Register n (n = 0, 1)

Address: 0x0000_0408 (n = 0), 0x0000_0410 (n = 1)



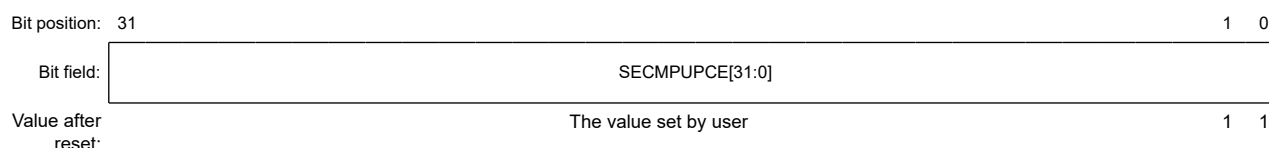
Bit	Symbol	Function	R/W
31:0	SECMPUPCS[31:0]	Region Start Address Address where the region starts, for use in region determination. The lower 2 bits are ignored (fixed to 00b by hardware).	R/W

The SECMPUPCSn and SECMPUPCEn registers specify the security fetch region of the code flash memory or SRAM. For the code flash memory and SRAM area to be covered, see [section 4.1. Address Space](#).

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 2). Place the secure program in a location at least 12 bytes away from the last instruction of a non-secure program.

18.6.1.2 SECMPUPCEn : Security MPU Program Counter End Address Register n (n = 0, 1)

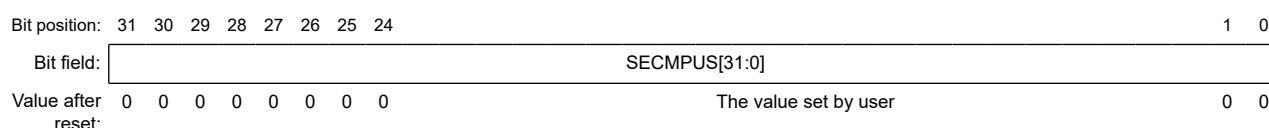
Address: 0x0000_040C (n = 0), 0x0000_0414 (n = 1)



Bit	Symbol	Function	R/W
31:0	SECMPUPCE[31:0]	Region End Address Address where the region ends, for use in region determination. The lower 2 bits are ignored (fixed to 11b by hardware).	R/W

18.6.1.3 SECMPUS0 : Security MPU Region 0 Start Address Register

Address: 0x0000_0418



Bit	Symbol	Function	R/W
31:0	SECMPIUS[31:0]	Region Start Address Address where the region starts, for use in region determination. The higher 8 bits and lower 2 bits are ignored (fixed to higher 8 bits = 0x00 and lower 2 bits = 00b by hardware).	R/W

The SECMPIUS0 and SECMPIUE0 registers specify the security program and data of the code flash memory. For the code flash memory and SRAM area to be covered, see [section 4.1. Address Space](#). The memory space defined in the SECMPIUS0 and SECMPIUE0 registers can only be accessed from the secure program set up in the SECMPIUPCSn and SECMPIUPCEn registers (n = 0, 1).

Setting of the vector table area is prohibited.

18.6.1.4 SECMPIUE0 : Security MPU Region 0 End Address Register

Address: 0x0000_041C

Bit position: 31 30 29 28 27 26 25 24

1 0

Bit field: SECMPIUE[31:0]

Value after reset: 0 0 0 0 0 0 0 0

The value set by user

1 1

Bit	Symbol	Function	R/W
31:0	SECMPIUE[31:0]	Region End Address Address where the region ends, for use in region determination. The higher 8 bits and lower 2 bits are ignored (fixed to higher 8 bits = 0x00 and lower 2 bits = 11b by hardware).	R/W

18.6.1.5 SECMPIUS1 : Security MPU Region 1 Start Address Register

Address: 0x0000_0420

Bit position: 31 30 29 28 27 26 25 24 23 22 21

1 0

Bit field: SECMPIUS[31:0]

Value after reset: 0 0 1 0 0 0 0 0 0 0 0

The value set by user

0 0

Bit	Symbol	Function	R/W
31:0	SECMPIUS[31:0]	Region Start Address Address where the region starts, for use in region determination. The higher 12 bits and lower 2 bits are ignored (fixed to higher 12 bits = 0x200 and lower 2 bits = 00b by hardware).	R/W

The SECMPIUS1 and SECMPIUE1 registers specify the security program and data of the SRAM. For the SRAM area to be covered, see [section 4.1. Address Space](#).

The memory space defined in the SECMPIUS1 and SECMPIUE1 registers can only be accessed from the secure program set up in the SECMPIUPCSn and SECMPIUPCEn registers (n = 0, 1).

Setting of the stack area and the vector table are prohibited.

18.6.1.9 SECMPUAC : Security MPU Access Control Register

Address: 0x0000_0438

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	DISPC 1	DISPC 0	—	—	—	—	—	DIS2	DIS1	DIS0
Value after reset:	1	1	1	1	1	1	The value set by user		1	1	1	1	1	The value set by user		

Bit	Symbol	Function	R/W
0	DIS0	Region 0 Disable 0: Security MPU region 0 enabled 1: Security MPU region 0 disabled	R/W
1	DIS1	Region 1 Disable 0: Security MPU region 1 enabled 1: Security MPU region 1 disabled	R/W
2	DIS2	Region 2 Disable 0: Security MPU region 2 enabled 1: Security MPU region 2 disabled	R/W
7:3	—	These bits are read as 1. When programming to the code flash, the write value should be 1.	R/W
8	DISPC0	PC Region 0 Disable 0: Security MPU PC region 0 enabled 1: Security MPU PC region 0 disabled	R/W
9	DISPC1	PC Region 1 Disable 0: Security MPU PC region 1 enabled 1: Security MPU PC region 1 disabled	R/W
15:10	—	These bits are read as 1. When programming to the code flash, the write value should be 1.	R/W

Note: When the option setting memory area is erased, the settings of the security MPU is disabled.

Note: To enable or disable the security MPU, see [section 18.6.2. Memory Protection](#).

DIS0 bit (Region 0 Disable)

The DIS0 bit enables or disables the security MPU region 0. If security MPU region 0 is enabled, the code flash memory region within the limits set up by SECMPUS0 and SECMPUE0 is secure data.

DIS1 bit (Region 1 Disable)

The DIS1 bit enables or disables the security MPU region 1. If security MPU region 1 is enabled, the SRAM region within the limits set up by SECMPUS1 and SECMPUE1 is secure data.

DIS2 bit (Region 2 Disable)

The DIS2 bit enables or disables the security MPU region 2. If security MPU region 2 is enabled, the data of the peripheral I/O registers of TSIP-Lite within the limits set up by SECMPUS2 and SECMPUE2 is secure data.

DISPC0 bit (PC Region 0 Disable)

The DISPC0 bit enables or disables the security MPU PC region 0. If security MPU PC region 0 is enabled, the code flash memory or the SRAM region within the limits set up by SECMPUPCS0 and SECMPUPCE0 contains a secure program.

DISPC1 bit (PC Region 1 Disable)

The DISPC1 bit enables or disables the security MPU PC region 1. If security MPU PC region 1 is enabled, the code flash memory or the SRAM region within the limits set up by SECMPUPCS1 and SECMPUPCE1 contains a secure program.

18.6.2 Memory Protection

The security MPU protects the secured regions (the code flash memory, the SRAM, and TSIP-Lite) from being accessed by non-secure programs. If access to a protected region is detected, the access becomes invalid.

The Figure 18.10 shows the setting flow of security MPU registers.

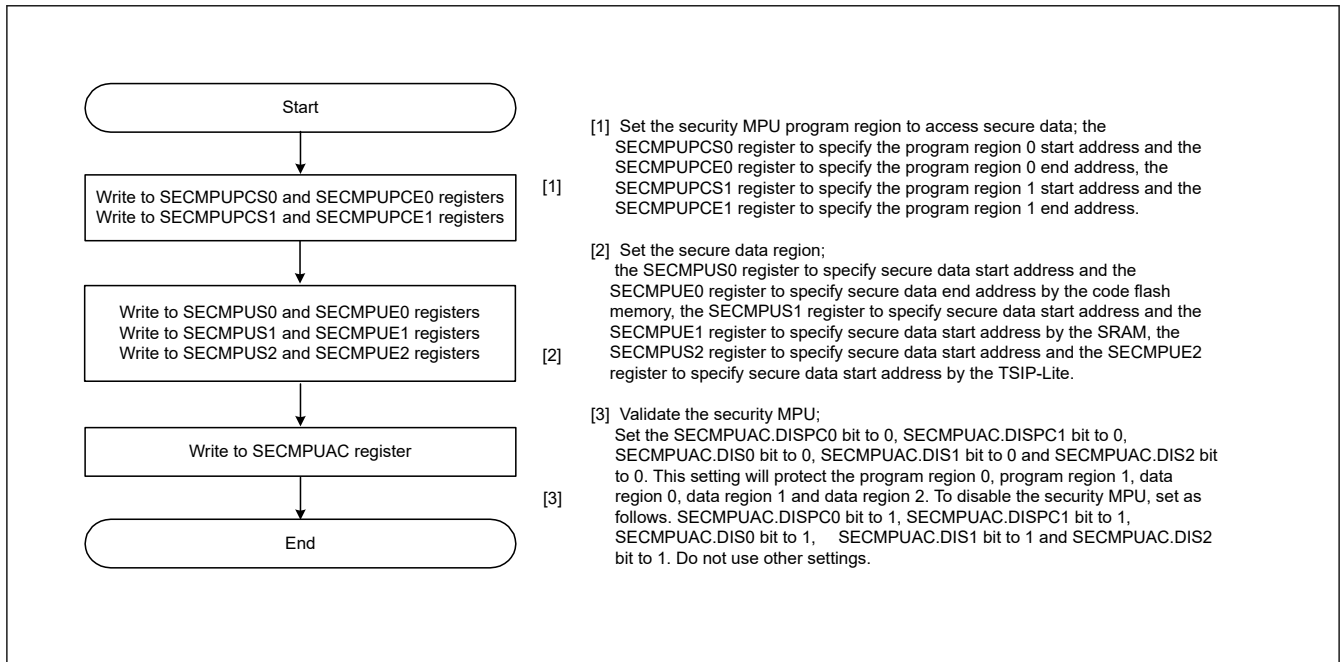


Figure 18.10 Setting flow of security MPU registers

The security MPU provides access protection in the following conditions:

- Secure data is accessed from a non-secure program
- Secure data is accessed from other than the CPU (DMAC, DTC)
- Secure data is accessed from the debugger.

Secure data is accessible only from a secure program.

The Table 18.9 shows the secure program/data settings and access protection.

Table 18.9 Secure program/data settings and access protection

Program/Data	Region name	Region setting register	Setting range	access protection description
Secure program	Program region 0	SECMPUPCS0, SECMPUPCE0	0x0000_0000 to 0x0003_FFFF (Code flash memory)	Accessible to all data (Data region 0 to 2, non-secure data)
	Program region 1	SECMPUPCS1, SECMPUPCE1	0x2000_0000 to 0x2001_FFFF (SRAM)	
Secure data	Data region 0	SECMPUS0, SECMPUE0	0x0000_0000 to 0x0003_FFFF (Code flash memory)	Accessible from secure program (Program region 0 to 1) Not accessible from non-secure program
	Data region 1	SECMPUS1, SECMPUE1	0x2000_0000 to 0x2001_FFFF (SRAM)	
	Data region 2	SECMPUS2, SECMPUE2	0x400C_0000 to 0x400D_FFFF (Peripheral I/O registers (TSIP-Lite))	
Non-secure program	—	—	Region outside of program region 0 and 1	Accessible to non-secure data Not accessible to secure data (data region 0 to 2)
Non-secure data	—	—	Region outside of data region 0 to 2	Accessible from all program (program region 0 and 1, non-secure program)

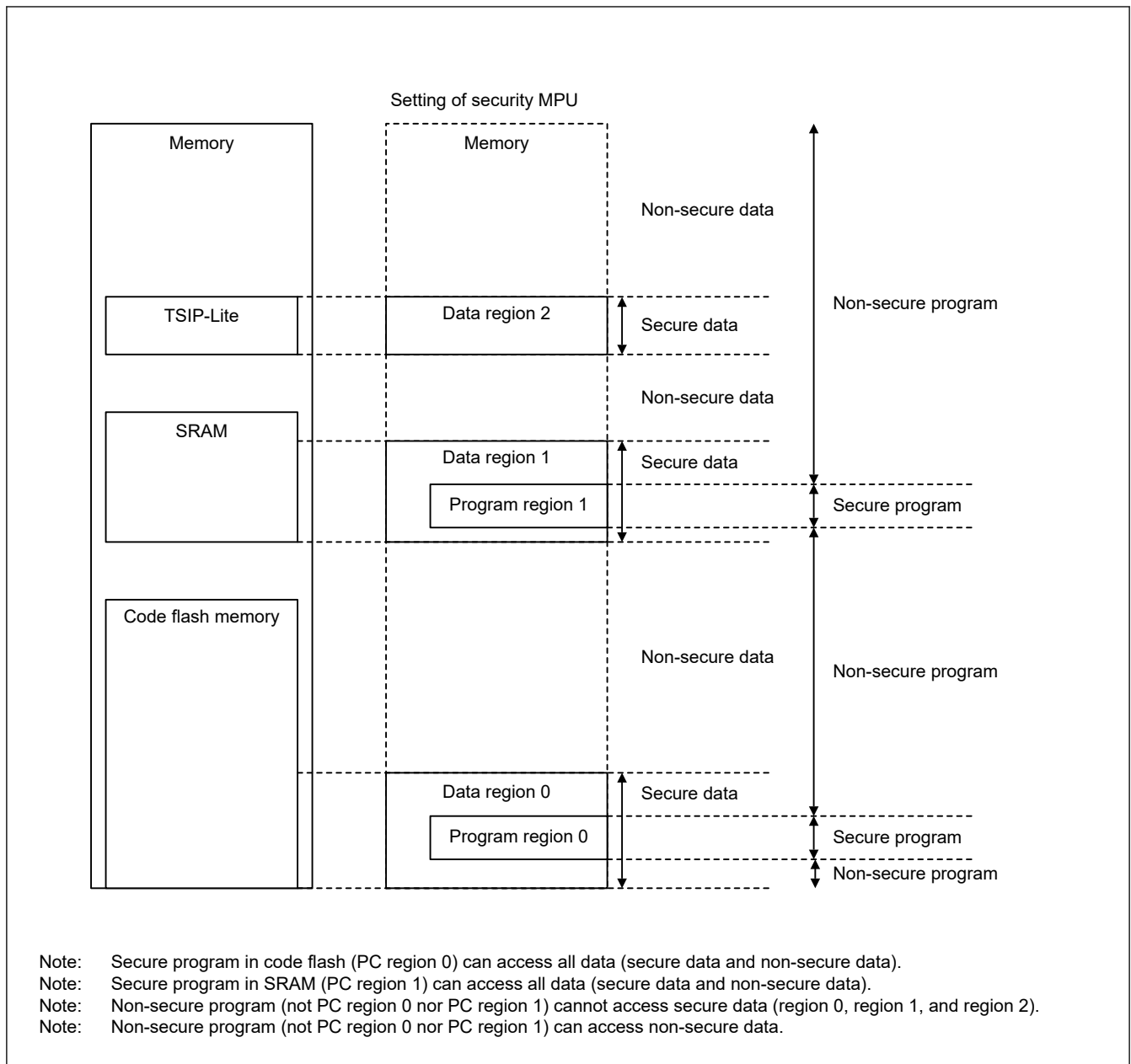


Figure 18.11 Use case of security MPU

18.7 Usage Notes

18.7.1 Notes on the Use of a Debugger

The protected memory cannot be debugged if the security MPU is enabled. Disable the security MPU when debugging a secure program.

18.7.2 Notes on Setting the Registers

Access through the bus must be suspended before writing to the following registers.

- Registers related to the CPU stack pointer monitor
- Registers related to the bus master MPU
- Registers related to the bus slave MPU

18.8 References

1. *ARM®v6-M Architecture Reference Manual* (ARM DDI 0419E)
2. *ARM® Cortex®-M0+ Technical Reference Manual* (ARM DDI 0484C)
3. *ARM® Cortex®-M0+ Devices Generic User Guide* (ARM DUI 0662B)

19. DMA Controller (DMAC)

19.1 Overview

This MCU incorporates an 4-channel direct memory access controller (DMAC). The DMAC is a module to transfer data without the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address.

[Table 19.1](#) lists the specifications of the DMAC, and [Figure 19.1](#) shows a block diagram of the DMAC.

Table 19.1 Specifications of DMAC

Item		Description
Number of channels		4 (DMACn (n = 0 to 3))
Transfer space		4 Gbytes (0x00000000 to 0xFFFFFFFF excluding reserved areas)
Maximum transfer volume		64 M data (Maximum number of transfers in block transfer mode: 1,024 data × 65,536 blocks)
DMAC activation source		Selectable for each channel: • Software trigger • Interrupt requests from peripheral modules or trigger from external interrupt input pins.^{*1}
Channel priority		Channel 0 > Channel 1 > Channel 2 > Channel 3 (Channel 0: highest)
Transfer data	Single data	Bit length: 8, 16, 32 bits
	Block size	Number of data: 1 to 1,024
Transfer mode	Normal transfer mode	• One data transfer by one DMA transfer request • Free running function (setting in which total number of data transfers is not specified) settable
	Repeat transfer mode	• One data transfer by one DMA transfer request • Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination. • Maximum settable repeat size: 1,024
	Block transfer mode	• One block data transfer by one DMA transfer request • Maximum settable block size: 1,024 data
Selective functions	Extended repeat area function	• Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed • Area of 2 bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination
Interrupt (DMACn_INT)	Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.
	Transfer escape end interrupt	• Generated when the repeat size of data transfer is completed. • Generated when the source address extended repeat area overflows. • Generated when the destination address extended repeat area overflows.
Event link activation (DMACn_INT)		An event link request is generated after each data transfer (for block transfer, after each block is transferred).
Power consumption reduction function		Module-stop state can be set.

Note 1. For details on DMAC activation sources, see [Table 16.4](#) in [section 16, Interrupt Controller Unit \(ICU\)](#).

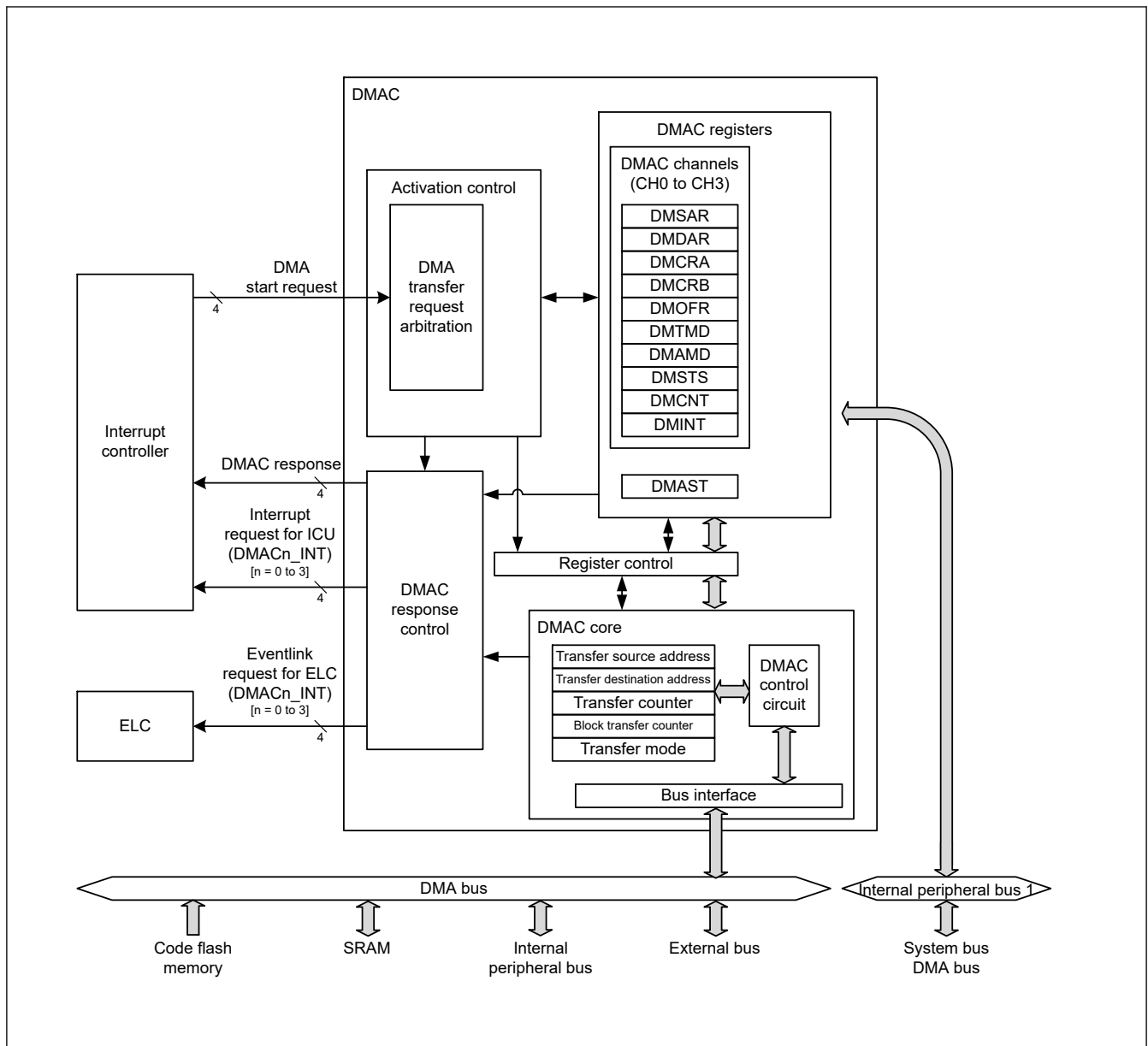


Figure 19.1 Block Diagram of DMAC

19.2 Register Descriptions

19.2.1 DMSAR : DMA Source Address Register

Base address: $\text{DMACn} = 0x4000\_5000 + 0x0040 \times n$ ($n = 0$ to 3)

Offset address: 0x00

Bit position: 31

0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer source start address Setting range is 0x00000000 to 0xFFFFFFFF (4 Gbytes).	R/W

Set DMSAR while DMAC activation is disabled (the DMAST.DMST bit = 0) or DMA transfer is disabled (the DMCNT.DTE bit = 0).

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bit.

19.2.2 DMDAR : DMA Destination Address Register

Base address: $\text{DMACn} = 0x4000\_5000 + 0x0040 \times n$ ($n = 0$ to 3)

Offset address: 0x04

Bit position: 31

0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer destination start address setting range is 0x00000000 to 0xFFFFFFFF (4 Gbytes).	R/W

Set DMDAR while DMAC activation is disabled (the DMAST.DMST bit = 0) or DMA transfer is disabled (the DMCNT.DTE bit = 0).

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bit.

19.2.3 DMCRA : DMA Transfer Count Register

Base address: $\text{DMACn} = 0x4000\_5000 + 0x0040 \times n$ ($n = 0$ to 3)

Offset address: 0x08

Bit position:

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

Bit field:

—	—	—	—	—	—	DMCRAH[9:0]									

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit position:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:

DMCRL[15:0]															

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	DMCRL[15:0]	Lower bits of transfer count Specifies the number of transfer operations.	R/W
25:16	DMCRAH[9:0]	Upper bits of transfer count Specifies the number of transfer operations.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Set the same value for DMCRAH and DMCRL in repeat transfer mode and block transfer mode. Bits 15 to 10 are fixed to 0 in repeat transfer mode and block transfer mode.

(1) Normal Transfer Mode (DMTMD.MD[1:0] = 00b)

DMCRL functions as a 16-bit transfer counter.

The number of transfer operations is one when the setting is 0x0001, and 65,535 when it is 0xFFFF. The value is decremented by one each time data is transferred.

When the setting is 0x0000, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running function).

DMCRAH is not used in normal transfer mode. Write 0x0000 to DMCRAH.

(2) Repeat Transfer Mode (DMTMD.MD[1:0] = 01b)

DMCRAH specifies the repeat size and DMCRAL functions as a 10-bit transfer counter.

The number of transfer operations is one when the setting is 0x001, 1,023 when it is 0x3FF, and 1,024 when it is 0x000. In repeat transfer mode, a value in the range of 0x000 to 0x3FF (1 to 1,024) can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 0x000, at which the value in DMCRAH is loaded into DMCRAL.

(3) Block Transfer Mode (DMTMD.MD[1:0] = 10b)

DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter.

The block size is one when the setting is 0x001, 1,023 when it is 0x3FF, and 1,024 when it is 0x000. In block transfer mode, a value in the range of 0x000 to 0x3FF can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 0x000, at which the value in DMCRAH is loaded into DMCRAL.

19.2.4 DMCRB : DMA Block Transfer Count Register

Base address: $\text{DMACn} = 0x4000\_5000 + 0x0040 \times n$ ($n = 0$ to 3)

Offset address: 0x0C

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

Bit	Symbol	Function	R/W
15:0	n/a	Specifies the number of block transfer operations or repeat transfer operations 0x0001 to 0xFFFF (1 to 65,535) 0x0000 (65,536)	R/W

DMCRB specifies the number of block transfer operations and repeat transfer operations in block and repeat transfer mode, respectively.

The number of transfer operations is one when the setting is 0x0001, 65,535 when it is 0xFFFF, and 65,536 when it is 0x0000.

In repeat transfer mode, the value is decremented by one when the final data of one repeat size is transferred.

In block transfer mode, the value is decremented by one when the final data of one block size is transferred.

In normal transfer mode, DMCRB is not used. The setting is invalid.

19.2.5 DMTMD : DMA Transfer Mode Register

Base address: $\text{DMACn} = 0x4000\_5000 + 0x0040 \times n$ ($n = 0$ to 3)

Offset address: 0x10

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

Bit	Symbol	Function	R/W
1:0	DCTG[1:0]	Transfer Request Source Select 0 0: Software request 0 1: Hardware request*1 1 0: Setting prohibited 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	SZ[1:0]	Transfer Data Size Select 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: Setting prohibited	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	DTS[1:0]	Repeat Area Select 0 0: The destination is specified as the repeat area or block area 0 1: The source is specified as the repeat area or block area 1 0: The repeat area or block area is not specified 1 1: Setting prohibited	R/W
15:14	MD[1:0]	Transfer Mode Select 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Setting prohibited	R/W

Note 1. To select the DMAC activation source, use the DELSRn registers of the ICU. For details on DMAC activation sources, see [Table 16.4](#) in [section 16, Interrupt Controller Unit \(ICU\)](#).

DCTG[1:0] bits (Transfer Request Source Select)

DCTG[1:0] bits select interrupts from software and peripheral modules or external interrupt input pins.

SZ[1:0] bits (Transfer Data Size Select)

SZ[1:0] bits select 8, 16, or 32 bit data transfer size when performing data transfer.

DTS[1:0] bits (Repeat Area Select)

DTS[1:0] select either the source or destination as the repeat area in repeat or block transfer mode. In normal transfer mode, setting these bits is invalid.

MD[1:0] bits (Transfer Mode Select)

MD[1:0] bits select normal transfer, repeat transfer and block transfer mode.

19.2.6 DMINT : DMA Interrupt Setting Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 3)

Offset address: 0x13

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable 0: Disables an interrupt request for an extended repeat area overflow on the destination address 1: Enables an interrupt request for an extended repeat area overflow on the destination address	R/W

Bit	Symbol	Function	R/W
1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable 0: Disables an interrupt request for an extended repeat area overflow on the source address 1: Enables an interrupt request for an extended repeat area overflow on the source address	R/W
2	RPTIE	Repeat Size End Interrupt Enable 0: Disables the repeat size end interrupt request 1: Enables the repeat size end interrupt request	R/W
3	ESIE	Transfer Escape End Interrupt Enable 0: Disables the transfer escape end interrupt request 1: Enables the transfer escape end interrupt request	R/W
4	DTIE	Transfer End Interrupt Enable 0: Disables the transfer end interrupt request 1: Enables the transfer end interrupt request	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

DARIE bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while DARIE bit is set to 1, the DMCNT.DTE bit is cleared to 0. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the destination address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the destination address, this bit is ignored.

SARIE bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while SARIE bit is set to 1, the DMCNT.DTE bit is cleared to 0. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the source address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the source address, this bit is ignored.

RPTIE bit (Repeat Size End Interrupt Enable)

When RPTIE bit is set to 1 in repeat transfer mode, the DMCNT.DTE bit is cleared to 0 after completion of a 1-repeat size data transfer. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DMTMD.DTS[1:0] bits are 10b (= repeat area or block area is not specified).

When this bit is set to 1 in block transfer mode, the DMCNT.DTE bit is cleared to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DMTMD.DTS[1:0] bits are 10b (= repeat area or block area is not specified).

ESIE bit (Transfer Escape End Interrupt Enable)

ESIE bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that are generated during DMA transfer.

The transfer escape end interrupt is generated when the DMSTS.ESIF flag is set to 1 with this bit set to 1. The transfer escape end interrupt is cleared by clearing this bit or the DMSTS.ESIF flag to 0.

DTIE bit (Transfer End Interrupt Enable)

DTIE bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers.

The transfer end interrupt is generated when the DMSTS.DTIF flag is set to 1 with this bit set to 1. The transfer end interrupt is cleared by clearing this bit or the DMSTS.DTIF flag to 0.

19.2.7 DMAMD : DMA Address Mode Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 3)

Offset address: 0x14

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

Bit	Symbol	Function	R/W
4:0	DARA[4:0]	Destination Address Extended Repeat Area Specifies the extended repeat area on the destination address. For details on the settings, see Table 19.2 .	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
7:6	DM[1:0]	Destination Address Update Mode 0 0: Destination address is fixed 0 1: Offset addition 1 0: Destination address is incremented 1 1: Destination address is decremented	R/W
12:8	SARA[4:0]	Source Address Extended Repeat Area Specifies the extended repeat area on the source address. For details on the settings, see Table 19.2 .	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
15:14	SM[1:0]	Source Address Update Mode 0 0: Source address is fixed 0 1: Offset addition 1 0: Source address is incremented 1 1: Source address is decremented	R/W

DARA[4:0] bits (Destination Address Extended Repeat Area)

DARA[4:0] bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer or block transfer is selected, and when DMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write 00000b in the DARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DMINT.DARIE bit set to 1. [Table 19.2](#) lists the settings and the corresponding extended repeat areas.

DM[1:0] bits (Destination Address Update Mode)

DM[1:0] bits select the mode of updating the destination address.

When increment is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the DMOFR register is added to the address.

SARA[4:0] bits (Source Address Extended Repeat Area)

SARA[4:0] bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer source, do not specify the extended repeat area on the source address. When repeat transfer or block transfer is selected, and when DMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DMINT.SARIE bit set to 1. Table 19.2 lists the settings and the corresponding extended repeat areas.

SM[1:0] bits (Source Address Update Mode)

SM[1:0] bits select the mode of updating the source address.

When increment is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, and 10b, the source address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, and 10b, the source address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the DMOFR register is added to the address.

Table 19.2 SARA[4:0] or DARA[4:0] Settings and Corresponding Repeat Areas (1 of 2)

SARA[4:0] or DARA[4:0] Settings and Corresponding Repeat Areas	Extended Repeat Area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address

Table 19.2 SARA[4:0] or DARA[4:0] Settings and Corresponding Repeat Areas (2 of 2)

SARA[4:0] or DARA[4:0] Settings and Corresponding Repeat Areas	Extended Repeat Area
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited.

19.2.8 DMOFR : DMA Offset Register

Base address: $\text{DMACn} = 0x4000\_5000 + 0x0040 \times n$ ($n = 0$ to 3)

Offset address: $0x18$

Bit position: 31 0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination 0x00000000 to 0x0FFFFFFF (0 bytes to (16 M – 1) bytes) 0xFF000000 to 0xFFFFFFFF (–16 Mbytes to –1 byte)	R/W

Write to this register while the DMAC operation is stopped or DMA transfer is disabled (not during data transfer).

Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading DMOFR returns the extended value.

19.2.9 DMCNT : DMA Transfer Enable Register

Base address: $\text{DMACn} = 0x4000\_5000 + 0x0040 \times n$ ($n = 0$ to 3)

Offset address: $0x1C$

Bit position: 7 6 5 4 3 2 1 0

Bit field:

—	—	—	—	—	—	—	DTE

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	DTE	DMA Transfer Enable 0: Disables DMA transfer 1: Enables DMA transfer	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

DTE bit (DMA Transfer Enable)

When the DMAST.DMST bit is set to 1 (DMAC activation is enabled) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.

19.2.10 DMREQ : DMA Software Start Register

Base address: $\text{DMACn} = 0x4000\_5000 + 0x0040 \times n$ ($n = 0$ to 3)

Offset address: $0x1D$

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

Bit	Symbol	Function	R/W
0	SWREQ	DMA Software Start 0: DMA transfer is not requested 1: DMA transfer is requested	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	CLRS	DMA Software Start Bit Auto Clear Select 0: SWREQ bit is cleared after DMA transfer is started by software 1: SWREQ bit is not cleared after DMA transfer is started by software	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

SWREQ bit (DMA Software Start)

When 1 is written to SWREQ bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is cleared to 0 if the CLRS bit is set to 0. This bit is not cleared to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer.

Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DMTMD.DCTG[1:0] bits are set to 00b (DMAC activation source is software).

Setting this bit is invalid when the DMTMD.DCTG[1:0] bits are set to a value other than 00b.

To start DMA transfer by software with the CLRS bit being 0, ensure that the SWREQ bit is 0, and then write 1 to the SWREQ bit.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When 0 is written to this bit.

CLRS bit (DMA Software Start Bit Auto Clear Select)

CLRS bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is cleared to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not cleared to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

19.2.11 DMSTS : DMA Status Register

Base address: $DMACn = 0x4000\_5000 + 0x0040 \times n$ ($n = 0$ to 3)

Offset address: $0x1E$

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

Bit	Symbol	Function	R/W
0	ESIF	Transfer Escape End Interrupt Flag 0: A transfer escape end interrupt has not been generated 1: A transfer escape end interrupt has been generated	R/W ¹
3:1	—	These bits are read as 0. The write value should be 0.	R
4	DTIF	Transfer End Interrupt Flag 0: A transfer end interrupt has not been generated 1: A transfer end interrupt has been generated	R/W ¹
6:5	—	These bits are read as 0. The write value should be 0.	R
7	ACT	DMAC Active Flag 0: DMAC is in the idle state 1: DMAC is operating	R

Note 1. Only 0 can be written to clear the flag.

ESIF flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- When 1-repeat size data transfer is completed in repeat transfer mode with the DMINT.RPTIE bit set to 1.
- When 1-block data transfer is completed in block transfer mode with the DMINT.RPTIE bit set to 1.
- When an extended repeat area overflow on the source address occurs while the DMINT.SARIE bit is set to 1 and the DMAMD.SARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer source address)
- When an extended repeat area overflow on the destination address occurs while the DMINT.DARIE bit is set to 1 and the DMAMD.DARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer destination address)

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DMCNT.DTE bit.

DTIF flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt has been generated.

[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of DMCRAL becoming 0 on completion of transfer)
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of DMCRB becoming 0 on completion of transfer)
- When the specified number of blocks have been transferred in block transfer mode (the value of DMCRB becoming 0 on completion of transfer)

[Clearing conditions]

- When 0 is written to this bit
- When 1 is written to the DMCNT.DTE bit

ACT flag (DMAC Active Flag)

This flag indicates whether the DMAC is in the idle or active state.

[Setting condition]

- When the DMAC starts data transfer operation

[Clearing condition]

- When data transfer in response to one transfer request is completed

19.2.12 DMAST : DMA Module Activation Register

Base address: DMA = 0x4000_5200

Offset address: 0x00

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

Bit	Symbol	Function	R/W
0	DMST	DMAC Operation Enable 0: DMAC activation is disabled 1: DMAC activation is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

DMST bit (DMAC Operation Enable)

Setting the DMAST.DMST to 1 enables DMAC activation for all channels. When the DMST bit is set to 1 (DMAC activation is enabled), and 1 is written to the DMCNT.DTE bit (DMA transfer is enabled) for multiple channels, all of the associated channels can be placed in the transfer request ready state at the same time.

When the DMST bit clears to 0 during DMA transfer, DMA transfer is suspended after the current data transfer associated with a single transfer request completes. To resume DMA transfer, set the DMST bit to 1 again.

[Setting condition]

- When 1 is written to this bit

[Clearing condition]

- When 0 is written to this bit

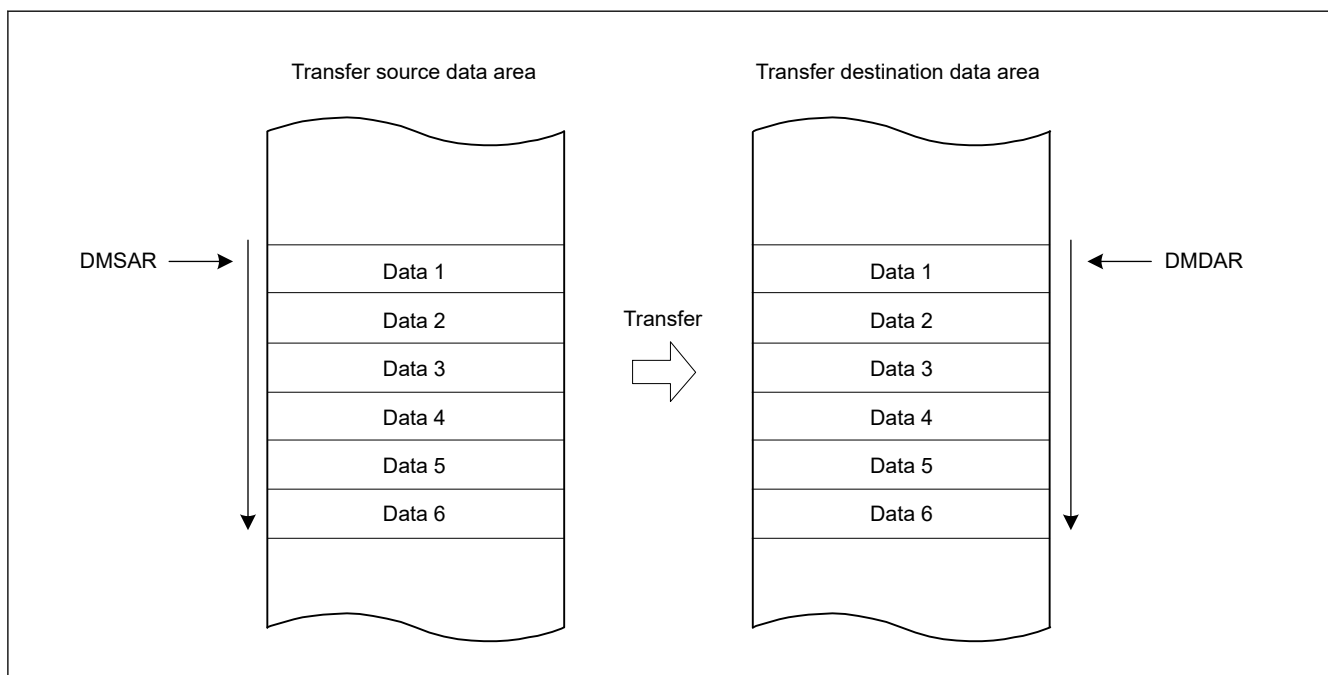
19.3 Operation**19.3.1 Transfer Mode****19.3.1.1 Normal Transfer Mode**

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using the DMCRAL register. When these bits are set to 0x0000, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running function). Setting DMCRB register is invalid in normal transfer mode. Except in free running function, a transfer end interrupt request can be generated after completion of the specified number of transfer operations.

Table 19.3 summarizes the register update operation in normal transfer mode, and Figure 19.2 shows the operation in normal transfer mode.

Table 19.3 Register Update Operation in Normal Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request
DMSAR	Transfer source address	Increment/decrement/fixed/offset addition
DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition
DMCRAL	Transfer count	Decrement by one/not updated (in free running function)
DMCRAH	—	Not updated (Not used in normal transfer mode)
DMCRB	—	Not updated (Not used in normal transfer mode)

**Figure 19.2 Operation in Normal Transfer Mode**

19.3.1.2 Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request.

A maximum of 1K data can be set as a total repeat transfer size using DMCRA register.

A maximum of 64K can be set as the number of repeat transfer operations using DMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of repeat transfer operations) can be set as a total data transfer size.

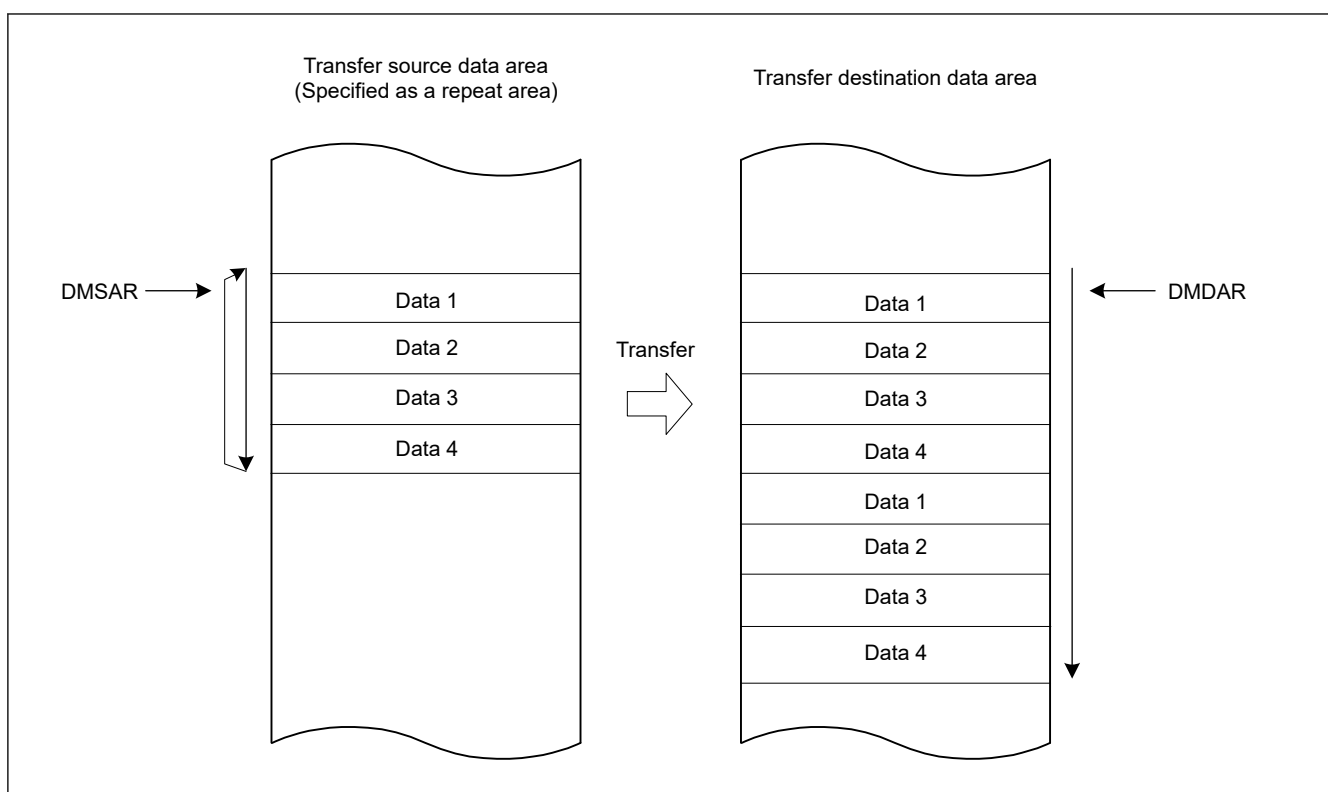
Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (DMSAR or DMDAR) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the repeat size end interrupt handling.

A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations.

Table 19.4 summarizes the register update operation in repeat transfer mode, and Figure 19.3 shows the operation in repeat transfer mode.

Table 19.4 Register Update Operation in Repeat Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request	
		When DMCRAL register is not 1	When DMCRAL register is 1 (Transfer of the Last Data in Repeat Size)
DMSAR	Transfer source address	Increment/decrement/fixed/offset addition	- DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition - DMTMD.DTS[1:0] = 01b Initial value of DMSAR - DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition	- DMTMD.DTS[1:0] = 00b Initial value of DMDAR - DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition - DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMCRAH	Repeat size	Not updated	Not updated
DMCRL	Transfer count	Decrement by one	DMCRAH
DMCRB	Count of repeat transfer operations	Not updated	Decrement by one

**Figure 19.3 Operation in Repeat Transfer Mode**

19.3.1.3 Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request.

A maximum of 1K data can be set as a total block transfer size using DMCRA register.

A maximum of 64K can be set as the number of block transfer operations using DMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (DMSAR or DMDAR) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped and the repeat size end interrupt

can be requested. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the repeat size end interrupt handling.

Transfer end interrupt request can be generated after completion of the specified number of block transfer operations.

Table 19.5 summarizes the register update operation in block transfer mode, and Figure 19.4 shows the operation in block transfer mode.

Table 19.5 Register Update Operation in Block Transfer Mode

Register	Function	Update Operation after Completion of Single-Block Transfer by One Transfer Request
DMSAR	Transfer source address	- DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition - DMTMD.DTS[1:0] = 01b Initial value of DMSAR - DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMDAR	Transfer destination address	- DMTMD.DTS[1:0] = 00b Initial value of DMDAR - DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition - DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMCRAH	Block size	Not updated
DMCRAL	Transfer count	DMCRAH
DMCRB	Count of block transfer operations	Decrement by one

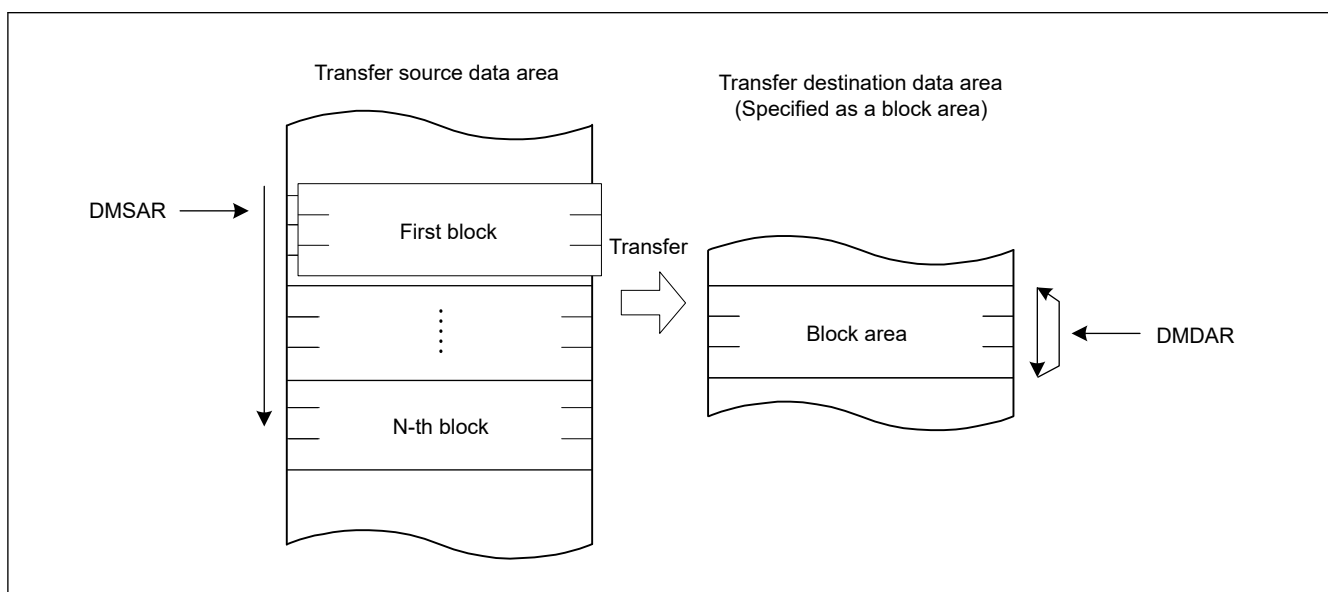


Figure 19.4 Operation in Block Transfer Mode

19.3.2 Extended Repeat Area Function

The DMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (DMSAR) and transfer destination address register (DMDAR).

The extended repeat area on the source address is specified by the DMAMD.SARA[4:0] bits. The extended repeat area on the destination address is specified by the DMAMD.DARA[4:0] bits. The size can be specified separately for the source and destination sides.

However, the area (of transfer source or transfer destination) which is specified as the repeat area or block area should not be specified as the extended repeat area.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt by an extended repeat area overflow can be requested. When an overflow occurs in the extended repeat area on the transfer source while the DMINT.SARIE bit is set to 1, the DMSTS.ESIF flag is set to 1 and the DMCNT.DTE bit is cleared to 0 to stop DMA transfer. At this time, if the DMINT.ESIE bit is set to 1, an interrupt by an extended repeat area overflow is requested. When the DMINT.DARIE bit is set to 1, the destination address register becomes a target to apply the function. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the interrupt handling.

Figure 19.5 shows an example of the extended repeat area operation.

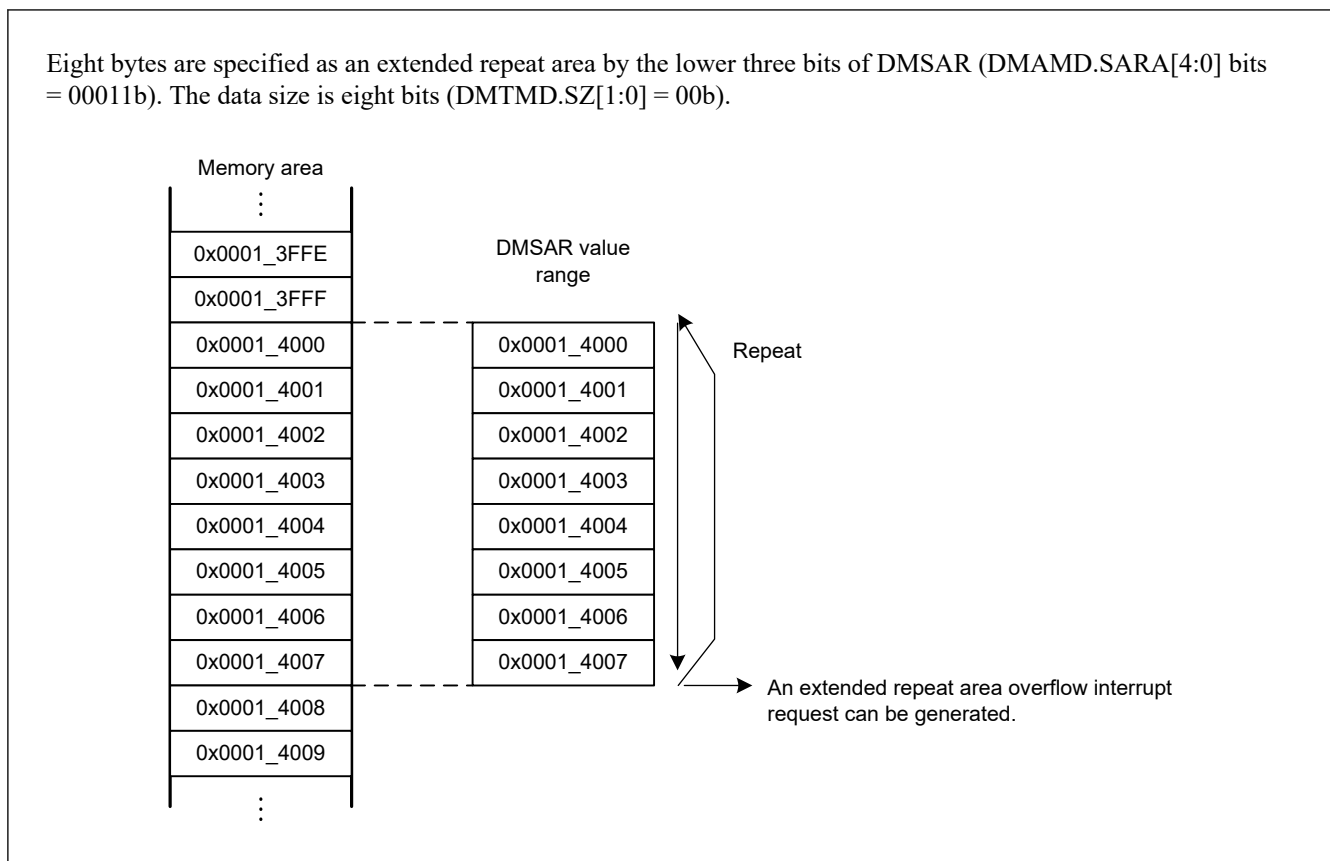


Figure 19.5 Example of Extended Repeat Area Operation

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration.

When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt by the overflow is suspended until transfer of the block is completed, and the transfer overruns.

Figure 19.6 shows an example when the extended repeat area function is used in block transfer mode.

Eight bytes are specified as an extended repeat area by the lower three bits of DMSAR (DMAMD.SARA[4:0] bits = 00011b), block transfer mode with block size 5 is set (DMCRA = 0x0005_0005), and transfer source address is not specified as a block area. Data size is eight bits (DMTMD.SZ[1:0] bits = 00b).

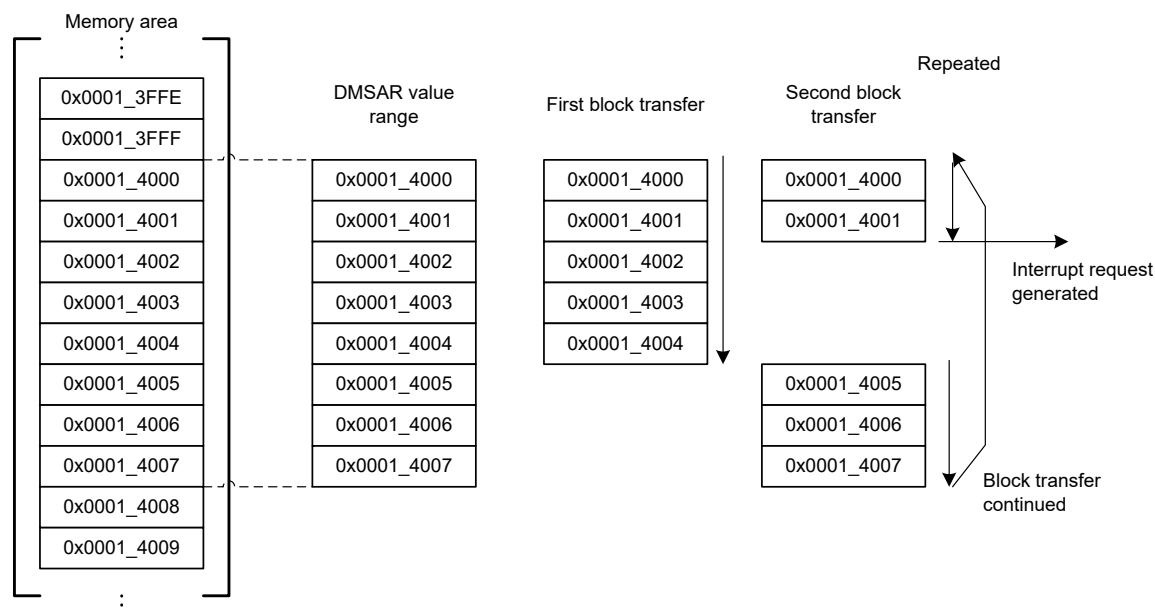


Figure 19.6 Example of Extended Repeat Area Function in Block Transfer Mode

19.3.3 Address Update Function using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. When the offset addition is selected, the offset specified by the DMA offset register (DMOFR) is added to the address every time the DMAC performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas.

Offset subtraction can also be realized by setting a negative value in DMOFR. In this case, the negative value must be 2's complement.

Table 19.6 shows the address update method in each address update mode.

Table 19.6 Address Update Method in Each Address Update Mode

Address Update Mode	Settings of DMAMD.SM[1:0] and DMAMD.DM[1:0] for Address Update Modes	Address Update Method (for Different SZ[1:0] Settings in DMTMD)		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+DMOFR ^{*1}		
Increment	10b	+1	+2	+4
Decrement	11b	-1	-2	-4

Note 1. When setting a negative value in the DMA Offset Register, the value must be in two's complement, obtained by the following formula:

two's complement of a negative offset value = $\sim(\text{offset}) + 1$ ($\sim$ = bit inversion)

19.3.3.1 Basic Transfer Using Offset Addition

Figure 19.7 shows an example of address updating using offset addition.

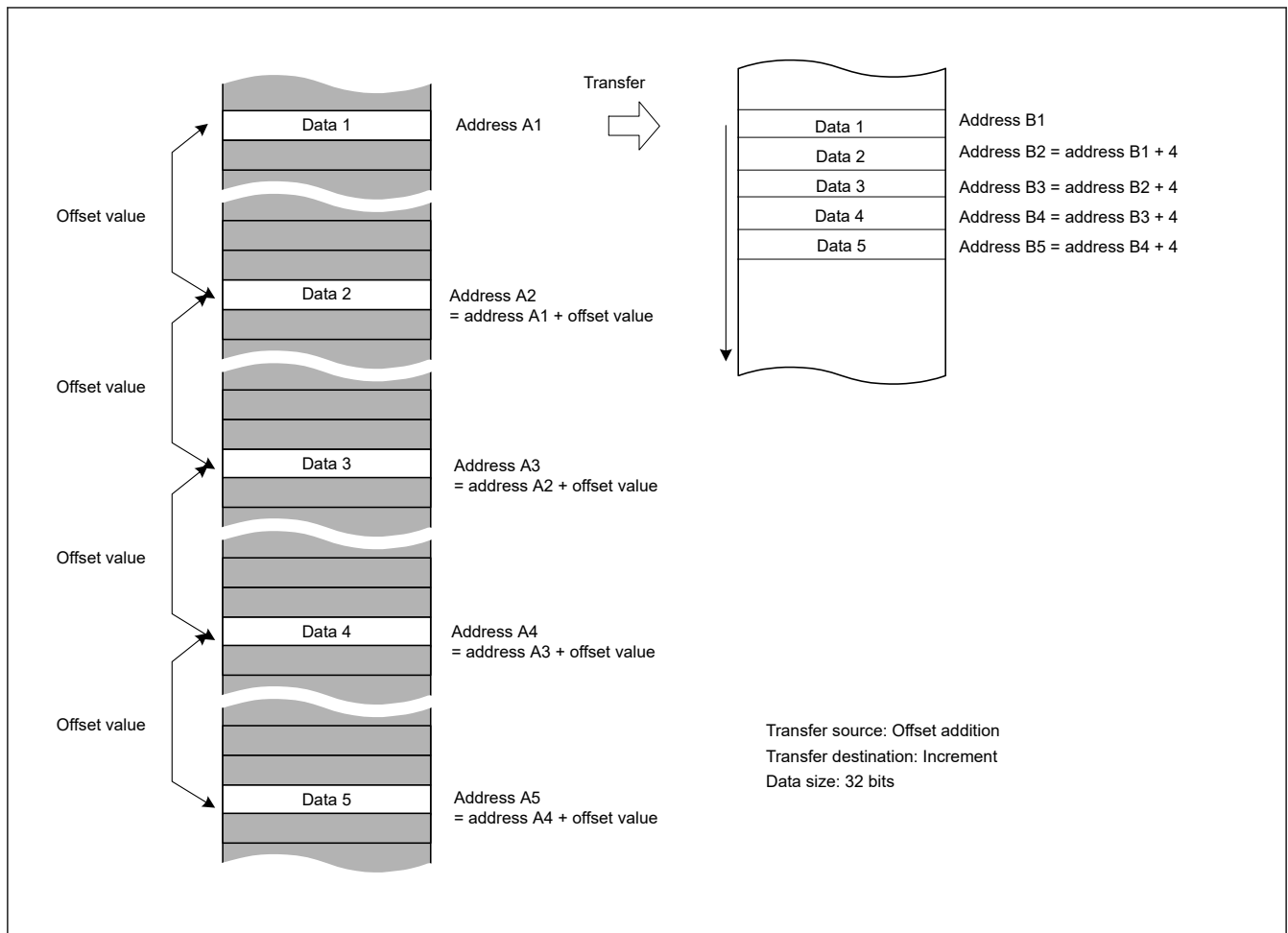


Figure 19.7 Example of Address Updating by Offset Addition

Figure 19.7 shows the setting of the following.

- The transfer data is 32 bits long
- Offset addition is set as the transfer source address update
- Increment is set as the transfer destination address update mode

The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

19.3.3.2 Example of XY Conversion Using Offset Addition

Figure 19.8 shows the XY conversion using offset addition in repeat transfer mode.

Settings are as follows:

- DMAMD.SM — Transfer source address update mode: Offset addition
- DMAMD.DM — Transfer destination address update mode: Destination address is incremented.
- DMTMD.SZ — Transfer data size select: 32 bits
- DMTMD.MD — Transfer mode select: Repeat transfer
- DMTMD.DTS — Repeat area select: The source is specified as the repeat area.
- DMOFR — Offset address: 0x10
- DMCRA — Repeat size: 0x4
- DMINT.RPTIE — The repeat size end interrupt is enabled.

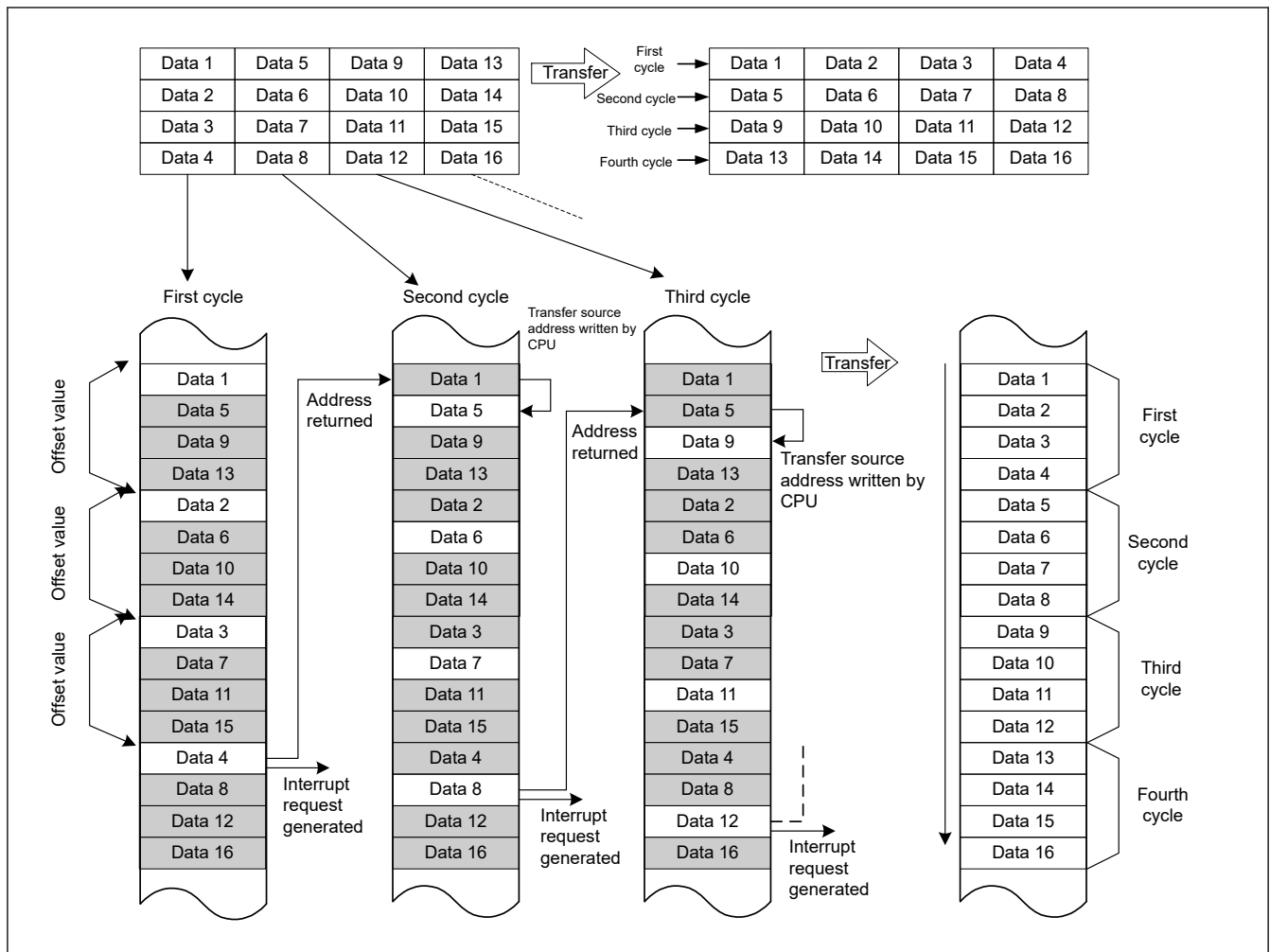


Figure 19.8 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to continuous transfer destination addresses. When data 4 is transferred:

- The repeat size of data transfer is complete.
- The transfer source address returns to the transfer start address (the address of data 1 on the transfer source).
- A repeat size end interrupt is requested.

During the time this interrupt pauses the transfer, the following operations are performed.

- DMSAR — Rewrite the DMA transfer source address to the address of data 5 (with the above example, the data 1 address + 4).
- DMCNT — Set the DTE bit to 1.

The DMA transfer is resumed from the state when the DMA transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion).

Figure 19.9 shows a flowchart of the XY conversion.

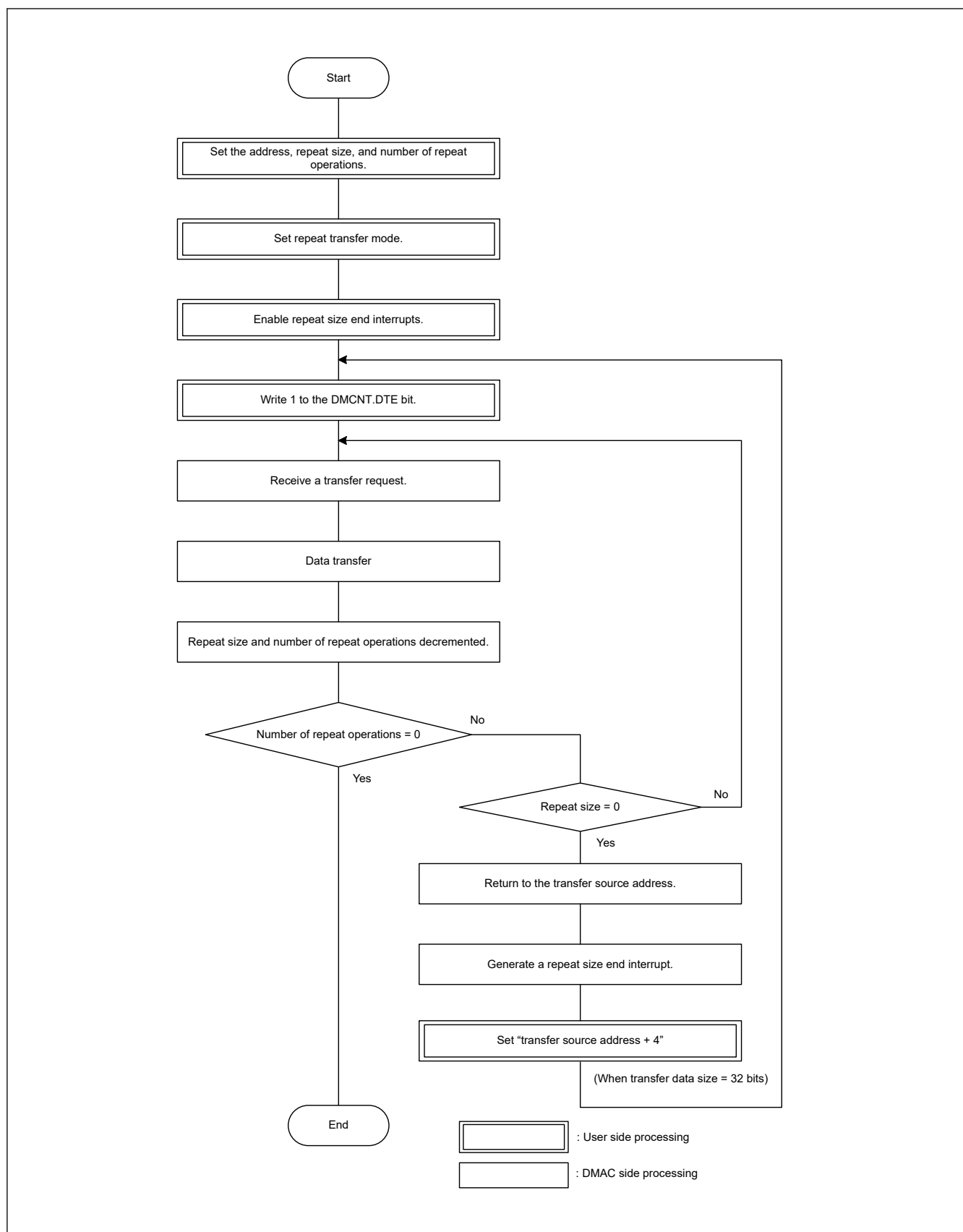


Figure 19.9 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

19.3.4 Activation Sources

Software, interrupt requests from the peripheral modules, and external interrupt requests can all be specified as DMAC activation sources. Set the DMTMD.DCTG[1:0] bits to select the activation source.

19.3.4.1 DMAC Activation by Software

When start DMA transfer by software, follow below procedure.

1. Set the DMTMD.DCTG[1:0] bits to 00b
2. Set the DMCNT.DTE bit to 1 (DMA transfer is enabled)
3. Set the DMAST.DMST bit set to 1 (DMAC activation enabled)
4. Set the DMREQ.SWREQ bit to 1 (DMA requested)

When the DMAC is activated by software while the DMREQ.CLRS bit is 0, the DMREQ.SWREQ bit is cleared to 0 after data transfer is started in response to a DMA transfer request.

When the DMAC is activated by software while the CLRS bit is 1, the SWREQ bit is not cleared to 0 after data transfer is started. In this case, a DMA transfer request is issued again after completion of a transfer.

19.3.4.2 DMAC Activation through Interrupt Requests from On-Chip Peripheral Modules or External Interrupt Requests

You can specify interrupt requests from on-chip peripheral modules and external interrupt requests as DMAC activation sources. The activation sources can be selected individually for each channel in ICU.DELSRn.DELS[7:0] (n = 0 to 3).

To start DMA transfer through an interrupt request from an on-chip peripheral module or an external interrupt request, follow the procedures as indicated below.

1. Set ICU.DELSRn.DELS[7:0] to the event number (select the DMAC event link).
2. Set the DMTMD.DCTG[1:0] bits to 01b (interrupts from the peripheral modules and the external interrupt pins).
3. Set the DMCNT.DTE bit to 1 (enable DMA transfer).
4. Set the DMAST.DMST bit set to 1 (DMAC activation enabled)

For interrupt requests specified as DMAC activation sources, see [Table 16.3](#), in [section 16, Interrupt Controller Unit \(ICU\)](#).

19.3.5 Operation Timing

The following timing charts have indicated the number of execution cycles of the minimum.

[Figure 19.10](#) and [Figure 19.11](#) show DMAC operation timing examples.

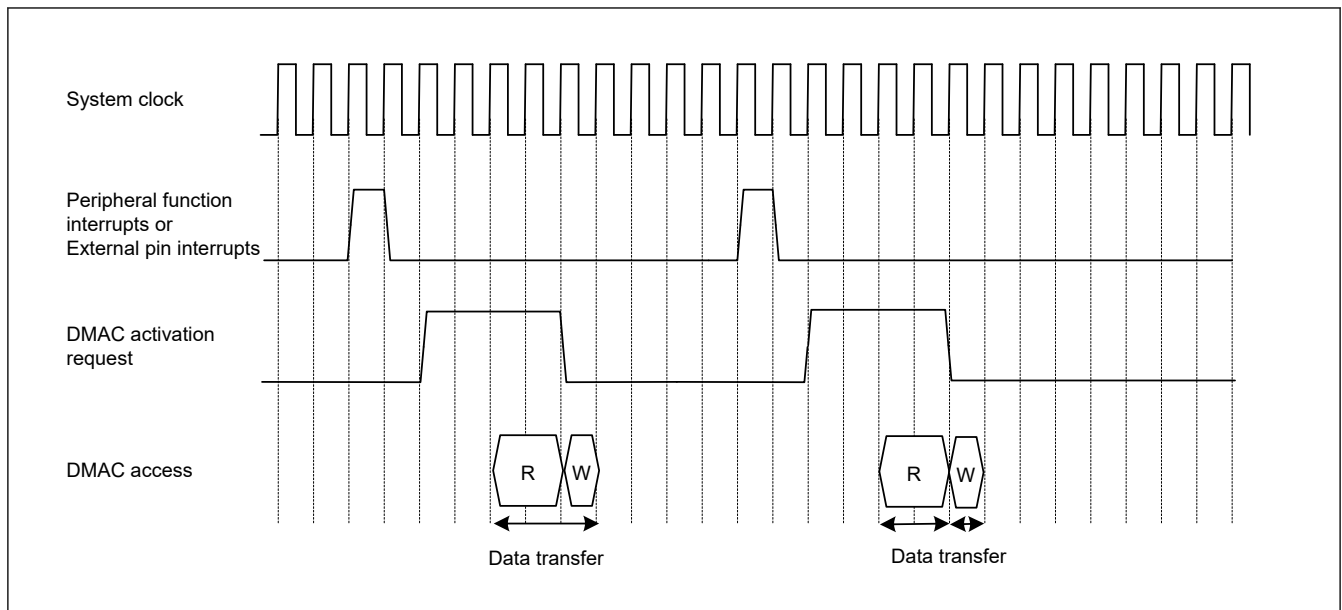


Figure 19.10 DMAC operation timing example 1 with DMAC activation by Interrupt from peripheral module or external interrupt input pin, in normal transfer mode or repeat transfer mode

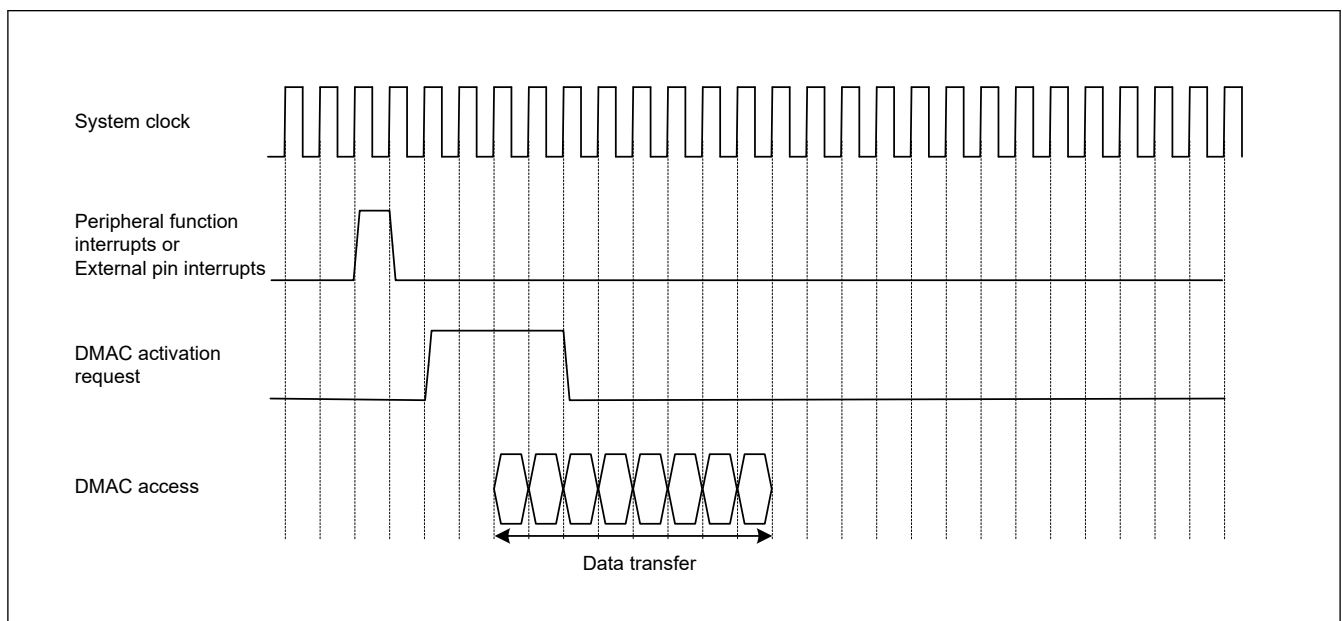


Figure 19.11 DMAC operation timing example 2 with DMAC activation by interrupt from peripheral module or external interrupt input pin, in block transfer mode with block size = 4

19.3.6 DMAC Execution Cycles

Table 19.7 lists execution cycles in one DMAC data transfer operation.

Table 19.7 DMAC Execution Cycles

Transfer Mode	Data Transfer (Read)	Data Transfer (Write)
Normal	Cr+1	Cw
Repeat	Cr+1	Cw
Block ^{*1}	P × Cr	P × Cw

Note: P: Block size (DMCRAH register setting)
 Cr: Data read destination access cycle
 Cw: Data write destination access cycle

Note 1. This is the case when the block size is 2 or more. When the block size is 1, normal transfer cycle is applied.

Cr and Cw depend on the access destination. For the number of cycles for each access destination, see [section 49, SRAM](#), [section 50, Flash Memory](#), [section 17, Buses](#). The frequency ratio of the system clock and the peripheral clock is also taken into consideration.

The unit for +1 in “Data Transfer (Read)” column is one system clock cycle (ICLK). For the operation example, see [section 19.3.5. Operation Timing](#).

The DMAC response time is the time from when the DMAC activation source is detected until the DMA transfer starts. [Table 19.7](#) does not include the time until the DMA data transfer starts after the DMAC activation source becomes active.

19.3.7 Activating the DMAC

[Table 19.8](#) shows the register setting procedure of normal, repeat and block transfer mode.

Table 19.8 Register Setting Procedure of Normal Transfer Mode, Repeat Transfer Mode and Block Transfer Mode (1 of 2)

No.	Step Name	Description
1	Disable the peripheral function as the DMACn request source.	To use peripheral function interrupts as DMAC activation sources. Disable the control register for the peripheral function.
2	Disable the IRQn pin as the DMACn request source.	To use external pin interrupts as DMAC activation sources.
3	Set the DMACn Event Link select (ICU.DELSRn.DELS[7:0]) to 0x00	Disable the DMACn request.
4	Clear the DMCNT.DTE bit to 0	Disable DMA transfer.
5	Set the interrupt request as a DMACn request source in the DMAC Event Link Setting Register (ICU.DELSRn) by using the ICU.	To use internal peripheral interrupts or external pin interrupts as DMAC activation sources. Enable the interrupt bit for the activation source. Set the DMACn activation source.
6	Set the peripheral module as a DMACn request source	To use peripheral function interrupt as a DMAC activation source. Set the control register for the peripheral function without starting it.
7	Set the IRQn pin function by using the ICU.	To use external pin interrupt as a DMAC activation source. Set the IRQn pin function by using the Interrupt Controller Unit.
8	Set the DMAMD.DM[1:0] bits Set the DMAMD.SM[1:0] bits Set the DMAMD.DARA[4:0] bits Set the DMAMD.SARA[4:0] bits	Set the Transfer destination address update mode bits Set the Transfer source address update mode bits Set the Transfer destination address extended repeat area bits Set the Transfer source address extended repeat area bits
9	Set the DMTMD.DCTG[1:0] bits Set the DMTMD.SZ[1:0] bits Set the DMTMD.DTS[1:0] bits Set the DMTMD.MD[1:0] bits	Set the Transfer request select bits Set the Data transfer size bits Set the Repeat area select bits Set the Transfer mode select bits
10	Set the DMSAR register Set the DMDAR register Set the DMCRA register	Set the transfer source start address. Set the transfer destination start address. Set the number of transfer operations.
11	Set the DMCRB register	To use block transfer mode or repeat transfer mode. Set the number of block transfer operations.
12	Set the DMOFR register	To use the address update function with offset. Set the offset value.
13	Set the DMINT.DTIE bit to 1	To use the DMA transfer end interrupts. Enable DMACn transfer end interrupts.
14	Set the DMINT.RPTIE bit Set the DMINT.SARIE bit Set the DMINT.DARIE bit Set the DMINT.ESIE bit to 1	To use the DMA transfer escape end interrupts Set the repeat size end interrupt. Set the transfer source address extended repeat area overflow interrupt. Set the transfer destination address extended repeat area overflow interrupt. Enable the DMA transfer escape end interrupt.
15	Set the DMCNT.DTE bit to 1	Enable DMA transfer.

Table 19.8 Register Setting Procedure of Normal Transfer Mode, Repeat Transfer Mode and Block Transfer Mode (2 of 2)

No.	Step Name	Description
16	Set the DMAST.DMST bit to 1	Enable DMAC operation. *1 Common settings for DMAC
17	Start the peripheral function as a DMACn request source	To use peripheral function interrupt as a DMAC activation source
18	Enable the IRQn pin as a DMACn request source	To use external pin interrupt as a DMAC activation source
19	End of initial settings	For activation by software On completion of the initial settings, writing 1 to the DMA software start bit (DMREQ.SWREQ) starts DMA transfer.

Note: n: DMAC channel (n = 0 to 3)

Note 1. The DMAST.DMST bit setting does not necessarily have to follow the settings for the individual activation sources.

19.3.8 Starting DMA Transfer

To enable the DMA transfer, set the DMCNT.DTE bit to 1 (enable the DMA transfer), and then set the DMAST.DMST bit to 1 (enable the DMAC activation).

New activation requests are not accepted during the transfer of another DMAC channel or DTC. When the preceding transfer is complete, channel arbitration selects the DMA transfer request of the highest priority channel, and the DMA transfer of that channel starts. When the DMA transfer starts, the DMSTS.ACT flag is set to 1 (the DMAC is in the active state).

19.3.9 Registers during DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are DMSAR, DMDAR, DMCRA, DMCRB, DMCNT, and DMSTS.

For details on register update operation in each transfer mode, see [Table 19.3](#) to [Table 19.5](#)

DMA Source Address Register (DMSAR)

When data has been transferred in response to one transfer request, the contents of DMSAR are updated to the address to be accessed by the next transfer request.

DMA Destination Address Register (DMDAR)

When data has been transferred in response to one transfer request, the contents of DMDAR are updated to the address to be accessed by the next transfer request.

DMA Transfer Count Register (DMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

DMA Block Transfer Count Register (DMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

DMA Transfer Enable Bit (DMCNT.DTE)

Although the DMCNT.DTE bit enables or disables data transfer by the register write access, it is automatically cleared to 0 by the DMAC according to the DMA transfer state.

The conditions for clearing this bit by the DMAC are as follows:

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt

Writing to the registers for the channels when the corresponding DMCNT.DTE bit is set to 1 is prohibited (except for DMCNT). In this case, writing must be performed after the bit is cleared to 0.

DMAC Active Flag (DMSTS.ACT)

The DMSTS.ACT flag indicates whether the DMACn is in the idle or active state.

This flag is set to 1 when the DMAC starts data transfer, and is cleared to 0 when data transfer in response to one transfer request is completed.

Even when DMA transfer is stopped by writing 0 to the DMCNT.DTE bit during DMA transfer, this flag remains 1 until DMA transfer is completed.

Transfer End Interrupt Flag (DMSTS.DTIF)

The DMSTS.DTIF flag is set to 1 after DMA transfer of the total transfer size of data is completed.

When both this flag and the DMINT.DTIE bit are set to 1, a transfer end interrupt is requested.

This flag is set to 1 when the DMA transfer bus cycle is completed and the DMSTS.ACT flag is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DMCNT.DTE bit is set to 1 during the interrupt handling.

Transfer Escape End Interrupt Flag (DMSTS.ESIF)

The DMSTS.ESIF flag is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this bit and the DMINT.ESIE bit are set to 1, a transfer escape end interrupt is requested.

This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the DMSTS.ACT flag is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DMCNT.DTE bit is set to 1 during an interrupt handling.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set.

For details, see [section 16, Interrupt Controller Unit \(ICU\)](#).

19.3.10 Channel Priority

When multiple DMA transfer requests are present, the DMAC determines the priority of channels that have DMA transfer requests.

- The channel priority is fixed as follows: Channel 0 > Channel 1 > Channel 2 > Channel 3 (Channel 0: highest).

When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

19.4 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DMCNT.DTE bit and the DMSTS.ACT flag are changed from 1 to 0, indicating that DMA transfer has ended.

19.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations**(1) In Normal Transfer Mode (DMTMD.MD[1:0] = 00b)**

When the value of DMCRAL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(2) In Repeat Transfer Mode (DMTMD.MD[1:0] = 01b)

When the value of DMCRB changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

(3) In Block Transfer Mode (DMTMD.MD[1:0] = 10b)

When the value of DMCRB changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set.

For details, see [section 16, Interrupt Controller Unit \(ICU\)](#).

19.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the DMINT.RPTIE bit is set to 1. When the interrupt is requested to complete DMA transfer, the DMCNT.DTE bit is cleared to 0 and the DMSTS.ESIF flag is set to 1. If the DMINT.ESIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC. Here, the transfer can be resumed by writing 1 to the DMCNT.DTE bit.

A repeat size end interrupt can be requested also in block transfer mode. In block transfer mode, the interrupt is requested in the same way as in repeat transfer mode when transfer of a 1-block size data is completed.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see [section 16, Interrupt Controller Unit \(ICU\)](#).

19.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the DMINT.SARIE or DMINT.DARIE bit is set to 1, an interrupt by an extended repeat area overflow is requested. When the interrupt is requested, the DMA transfer is terminated, the DMCNT.DTE bit is cleared to 0, and the ESIF flag in DMSTS is set to 1. If the DMINT.ESIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

Even if an interrupt by an extended repeat area overflow is requested during a read cycle, the following write cycle is performed.

In block transfer mode, even if an interrupt by an extended repeat area overflow is requested during a 1-block transfer, the remaining data in the block is transferred; transfer is terminated after a block transfer.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see [section 16, Interrupt Controller Unit \(ICU\)](#).

19.5 Interrupts

19.5.1 Transfer End Interrupt

Each DMAC channel can output an interrupt request (DMACn_INT) to the CPU or the DTC after transfer in response to one request is completed.

[Table 19.9](#) lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. [Figure 19.12](#) shows the schematic logic diagram of interrupt outputs (DMACn (n = 0 to 3)). [Figure 19.13](#) shows the DMAC interrupt handling routine to resume/terminate DMA transfer.

Table 19.9 Relation among Interrupt Sources, Interrupt Status Flags, and Interrupt Enable Bits

Interrupt Sources		Interrupt Enable Bits	Interrupt Status Flags	Request Output Enable Bits
Transfer end		—	DMSTS.DTIF	DMINT.DTIE
Escape transfer end	Repeat size end	DMINT.RPTIE	DMSTS.ESIF	DMINT.ESIE
	Source address extended repeat area overflow	DMINT.SARIE		
	Destination address extended repeat area overflow	DMINT.DARIE		

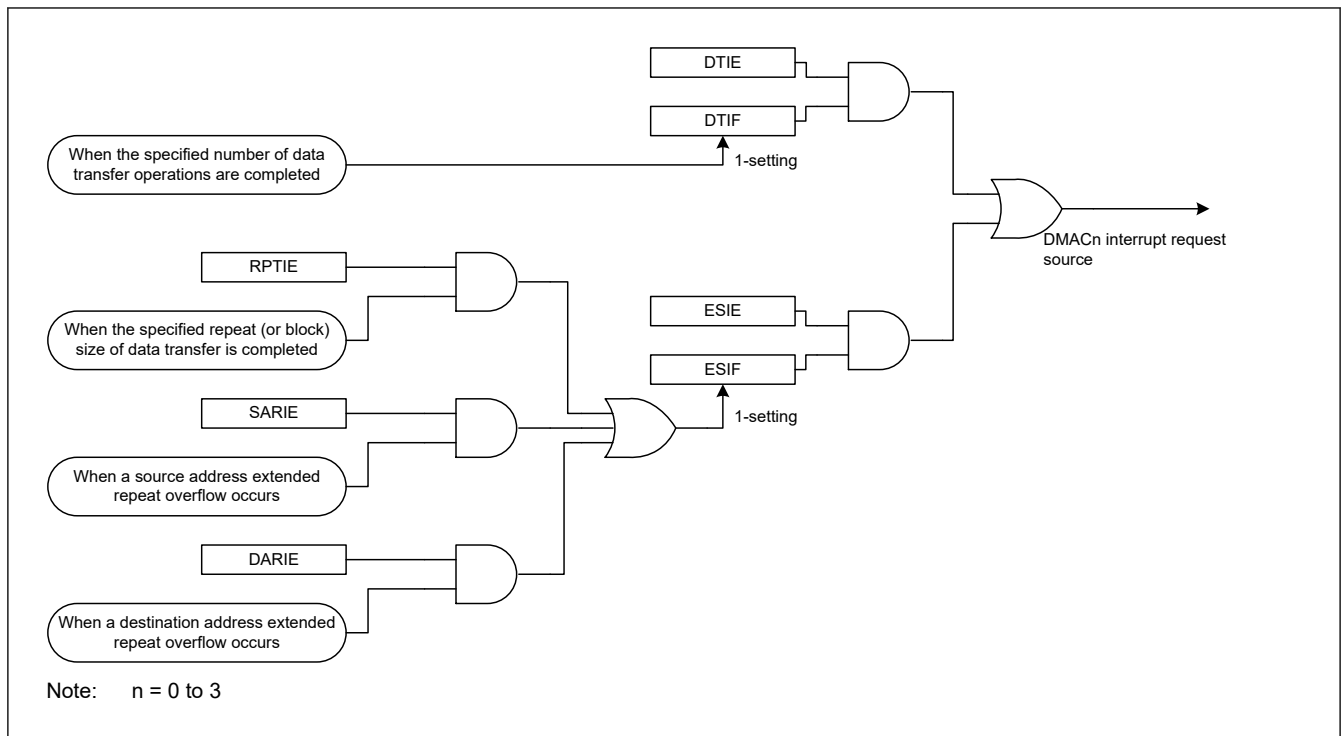


Figure 19.12 Schematic Logic Diagram of Interrupt Output Source (DMACn)

Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases:

- When terminating a DMA transfer
- When continuing a DMA transfer

19.5.1.1 When Terminating a DMA Transfer

Write 0 to the DMSTS.DTIF flag to clear a transfer end interrupt, and to the DMSTS.ESIF flag to clear a repeat size interrupt and an extended repeat area overflow interrupt. The DMACn remains in the stop state. When starting another DMA transfer after that, set the appropriate registers, and set the DMCNT.DTE bit to 1 (DMA transfer enabled).

19.5.1.2 When Continuing a DMA Transfer

Write 1 to the DMCNT.DTE bit. The DMSTS.ESIF flag is automatically cleared to 0 (interrupt source cleared), and DMA transfer is resumed.

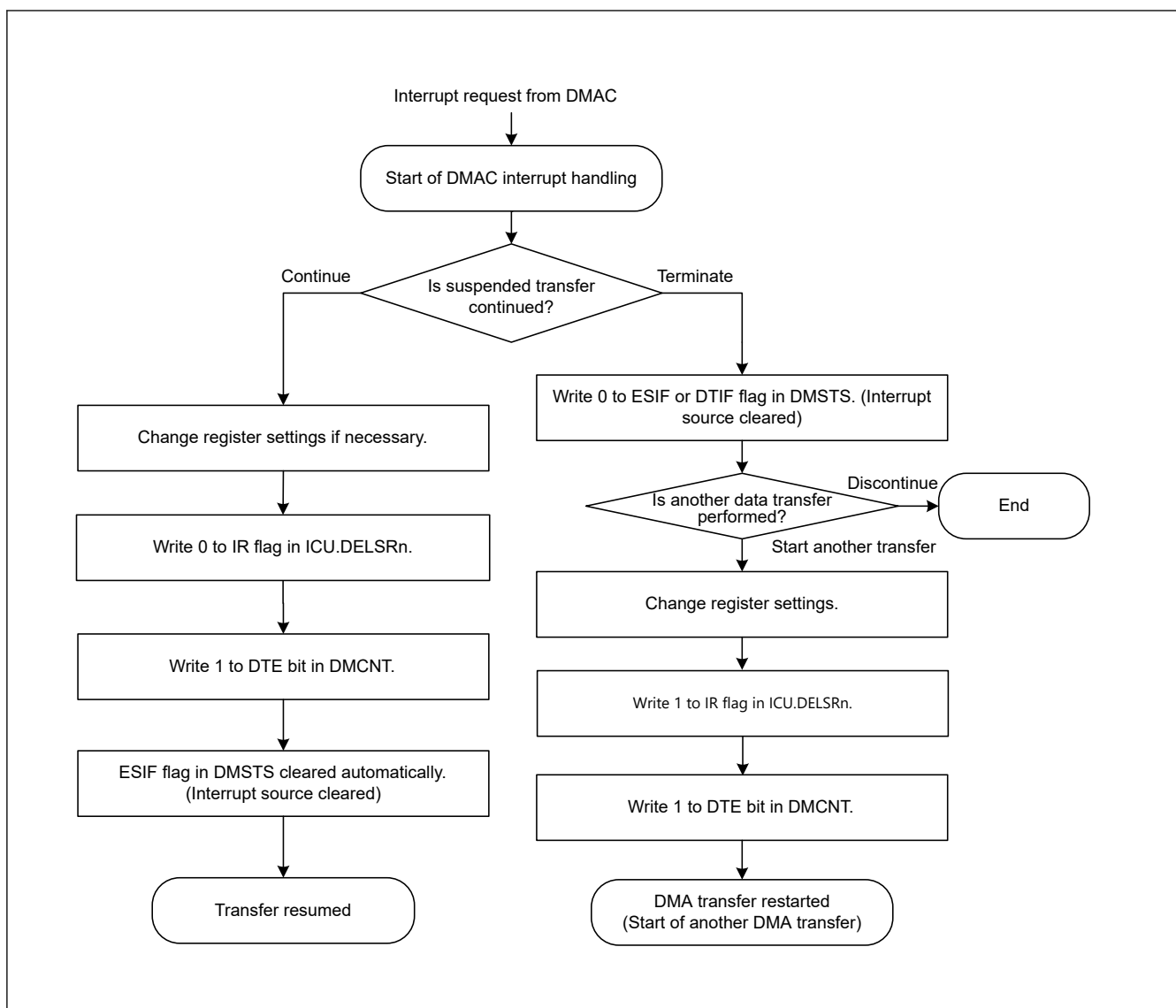


Figure 19.13 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

19.6 Event Link

Each DMAC channel outputs an event link request signal (DMACn_INT) every time it completes a data transfer, or a block transfer in block transfer mode.

For details, see [section 21, Event Link Controller \(ELC\)](#).

If a bus error occurs when writing the last data of transfer, a transfer end event and error response detection interrupt (DMA_TRANSERR) occurs.

19.7 Low-Power Consumption Function

Before entering the module-stop state or Software Standby mode, or Deep Software Standby mode, you must first set the DMAST.DMST bit to 0 (the DMAC module suspended), and use the settings in the sections that follow.

(1) Module-stop function

Writing 1 to the MSTPCRA.MSTPA22 bit enables the module-stop function of the DMAC. If a DMA transfer is in progress when 1 is written to the MSTPA22 bit, the transition to the module-stop state proceeds after the DMA transfer ends. Access to the DMAC registers is prohibited while the MSTPA22 bit is 1. Writing 0 to the MSTPA22 bit releases the DMAC from the module-stop state.

(2) Software Standby mode and Deep Software Standby mode

Use the settings described in [section 13.6.2.1. Transition to Software Standby Mode](#), or in [section 13.6.4.1. Transition to Deep Software Standby Mode](#).

If DMA transfer operations are in progress when the WFI instruction is executed, the DMA transfer completes before the transition to Software Standby mode or Deep Software Standby mode.

(3) Notes on low power consumption function

For information on the WFI instruction and register settings, see [section 13.7.7. Timing of WFI Instruction](#).

To perform a DMA transfer after returning from a low power consumption mode, set the DMAST.DMST 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 DMAC startup request, specify the CPU as the interrupt request destination, as described in [section 16.4.2. Detecting Interrupts](#), and then execute the WFI instruction.

19.8 Usage Notes

19.8.1 Access to the Registers during DMA Transfer

Do not write to the following registers while the DMSTS.ACT flag of the same channel is set to 1 (DMAC active state) or the DMCNT.DTE bit of the same channel is set to 1 (DMA transfer enabled):

- DMSAR
- DMDAR
- DMCRA
- DMCRB
- DMTMD
- DMINT
- DMAMD
- DMOFR

19.8.2 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see [section 4, Address Space](#).

19.8.3 Setting of DMAC Event Link Setting Register of the Interrupt Controller Unit (ICU.DELSRn n = 0 to 3)

The DMAC event link setting register (ICU.DELSRn) should be set while the DMA transfer enable bit (DMCNT.DTE) is cleared to 0 (DMA transfer is disabled). Moreover, the DTC activation enable register (ICU.IELSRn.DTCE (n = 0 to 31)) that corresponds to the same event number that has been set by the ICU.DELSRn register should not be set to 1. For details on the ICU.IELSRn.DTCE and ICU.DELSRn, see [section 16, Interrupt Controller Unit \(ICU\)](#).

19.8.4 Suspending or Restarting DMAC Activation

To suspend a DMAC activation request, write 0x00 to the DMAC Event Link select bits (ICU.DELSRn.DELS[7:0]). To restart the DMA transfer, write the event number to the ICU.DELSRn.DELS[7:0] bits following the settings shown in [section 19.3.7. Activating the DMAC](#).

19.8.5 Precautions for Resuming DMA Transfer

A DMAC activation request might occur in the next request after a DMA transfer completes. If this happens, the DMA transfer starts and the DMAC activation request is held in the DMAC. To prevent this, stop the DMAC activation requests by setting the DELSRn.DELS[7:0] bits in the ICU to 0.

When a DMAC activation request occurs after the last round of the DMA transfer is generated, clear the DMAC activation request with either of the following approaches.

- Clear the DMAC activation request with a DMA dummy transfer.
- Set the DMCNT.DTE bit to 0 and then set the ICU.DELSRn.IR flag to 0.

See [Figure 19.14](#).

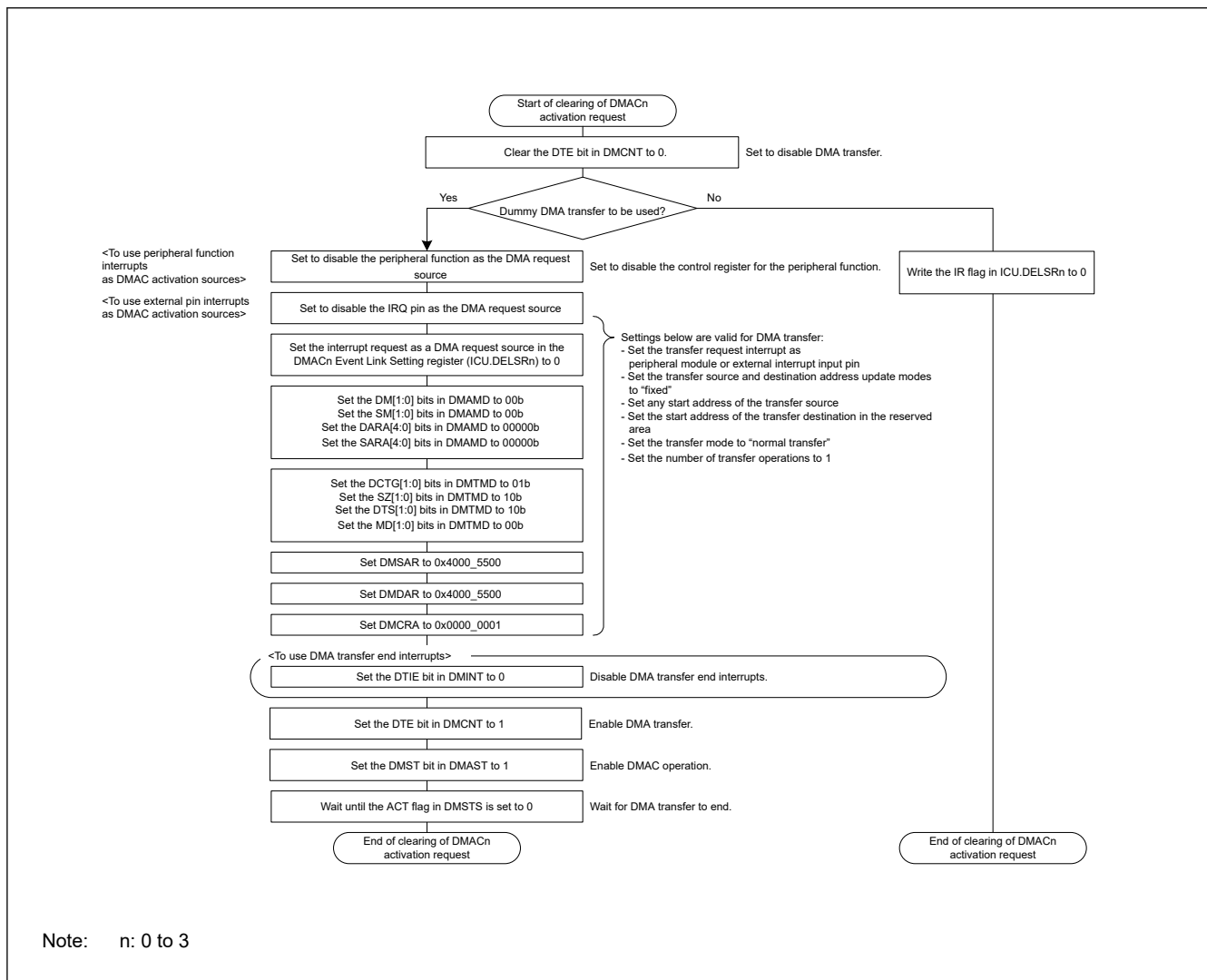


Figure 19.14 Example of register setting procedure to clear the DMAC activation interrupt

20. Data Transfer Controller (DTC)

20.1 Overview

A Data Transfer Controller (DTC) module is provided for transferring data when activated by an interrupt request.

[Table 20.1](#) lists the DTC specifications and [Figure 20.1](#) shows DTC block diagram.

Table 20.1 DTC specifications

Parameter	Description
Transfer modes	- Normal transfer mode A single activation leads to a single data transfer. - Repeat transfer mode A single activation leads to a single data transfer. The transfer address returns to the start address after the number of data transfers reaches the specified repeat size. The maximum number of repeat transfers is 256 and the maximum data transfer size is 256×32 bits (1024 bytes) - Block transfer mode A single activation leads to a transfer of a single block. The maximum block size is 256×32 bits = 1024 bytes.
Transfer channel	- Channel transfer can be associated with the interrupt source (transferred by a DTC activation request from the ICU) - Multiple data units can be transferred on a single activation source (chain transfer) - Chain transfers are selectable to either execute when the counter is 0, or always execute.
Transfer space	4 GB area from 0x0000_0000 to 0xFFFF_FFFF, excluding reserved areas
Data transfer units	- Single data unit: 1 byte (8 bits), 1 halfword (16 bits), 1 word (32 bits) - Single block size: 1 to 256 data units.
CPU interrupt source	- An interrupt request can be generated to the CPU on a DTC activation interrupt - An interrupt request can be generated to the CPU after a single data transfer - An interrupt request can be generated to the CPU after a data transfer of a specified volume.
Event link function	An event link request is generated after one data transfer (for block, after one block transfer)
Read skip	Read of transfer information can be skipped
Write-back skip	When the transfer source or destination address is specified as fixed, a write-back of transfer information can be skipped
Module-stop function	Module-stop state can be set to reduce power consumption

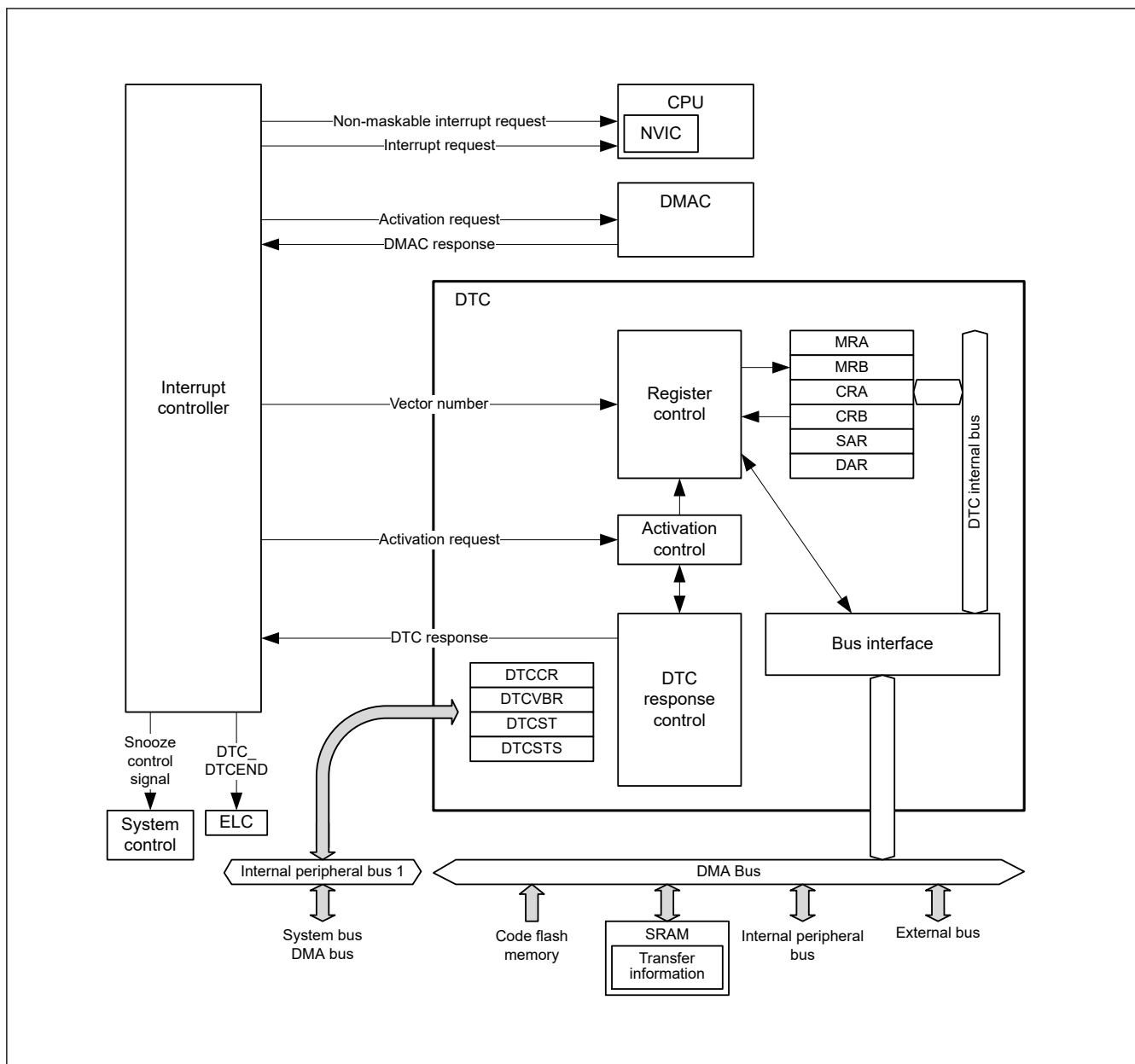


Figure 20.1 DTC block diagram

See [section 16.1. Overview](#) in [section 16, Interrupt Controller Unit \(ICU\)](#) for the connections between the DTC and NVIC in the CPU.

20.2 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.

20.2.1 MRA : DTC Mode Register A

Base address: DTCVBR

Offset address: $0x03 + 0x4 \times \text{Vector number}$

(Inaccessible directly from the CPU. See [section 20.3.1. Allocating Transfer Information and DTC Vector Table](#))

Bit position: 7 6 5 4 3 2 1 0

Bit field:	MD[1:0]	SZ[1:0]	SM[1:0]	—	—

Value after reset: x x x x x x x x

Bit	Symbol	Function	R/W
1:0	—	The read values are undefined. The write value should be 0.	—
3:2	SM[1:0]	Transfer Source Address Addressing Mode 0 0: Address in the SAR register is fixed (write-back to SAR is skipped.) 0 1: Address in the SAR register is fixed (write-back to SAR is skipped.) 1 0: SAR value is incremented after data transfer: +1 when SZ[1:0] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b 1 1: SAR value is decremented after data transfer: -1 when SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b	—
5:4	SZ[1:0]	DTC Data Transfer Size 0 0: Byte (8-bit) transfer 0 1: Halfword (16-bit) transfer 1 0: Word (32-bit) transfer 1 1: Setting prohibited	—
7:6	MD[1:0]	DTC Transfer Mode Select 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited	—

The MRA register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0x03) and DTC transfers it automatically to and from the MRA register. See [section 20.3.1. Allocating Transfer Information and DTC Vector Table](#).

20.2.2 MRB : DTC Mode Register B

Base address: DTCVBR

Offset address: $0x02 + 0x4 \times \text{Vector number}$

(Inaccessible directly from the CPU. See [section 20.3.1. Allocating Transfer Information and DTC Vector Table](#))

Bit position: 7 6 5 4 3 2 1 0

Bit field:	CHNE	CHNS	DISEL	DTS	DM[1:0]	—	—

Value after reset: x x x x x x x x

Bit	Symbol	Function	R/W
1:0	—	The read values are undefined. The write value should be 0.	—

Bit	Symbol	Function	R/W
3:2	DM[1:0]	Transfer Destination Address Addressing Mode 0 0: Address in the DAR register is fixed (write-back to DAR is skipped) 0 1: Address in the DAR register is fixed (write-back to DAR is skipped) 1 0: DAR value is incremented after data transfer: +1 when MRA.SZ[1:0] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b 1 1: DAR value is decremented after data transfer: -1 when MRA.SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b	—
4	DTS	DTC Transfer Mode Select 0: Select transfer destination as repeat or block area 1: Select transfer source as repeat or block area	—
5	DISEL	DTC Interrupt Select 0: Generate an interrupt request to the CPU when specified data transfer is complete 1: Generate an interrupt request to the CPU each time DTC data transfer is performed	—
6	CHNS	DTC Chain Transfer Select 0: Chain transfer is continuous 1: Chain transfer occurs only when the transfer counter changes from 1 to 0 or 1 to CRAH	—
7	CHNE	DTC Chain Transfer Enable 0: Chain transfer is disabled 1: Chain transfer is enabled	—

The MRB register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0x02) and DTC transfers it automatically to and from the MRB register. See [section 20.3.1. Allocating Transfer Information and DTC Vector Table](#).

DM[1:0] bits (Transfer Destination Address Addressing Mode)

The DM[1:0] bits is to fix the address of the DAR register or specify increment / decrement of the DAR register after transfer.

DTS bit (DTC Transfer Mode Select)

The DTS bit specifies whether the transfer source or destination is the repeat or block area in repeat or block transfer mode.

DISEL bit (DTC Interrupt Select)

The DISEL bit specifies the condition for generating an interrupt request to the CPU.

CHNS bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition. When CHNE is 0, the CHNS setting is ignored. For details on the conditions for chain transfer, see [Table 20.3](#).

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the activation source flag is not cleared, and an interrupt request to the CPU is not generated.

CHNE bit (DTC Chain Transfer Enable)

The CHNE bit enables chain transfer. The chain transfer condition is selected by the CHNS bit. For details on chain transfer, see [section 20.4.6. Chain Transfer](#).

20.2.3 SAR : DTC Transfer Source Register

Base address: DTCVBR

Offset address: $0x04 + 0x4 \times \text{Vector number}$
(Inaccessible directly from the CPU. See [section 20.3.1. Allocating Transfer Information and DTC Vector Table](#))

Bit position: 31 0

Bit field:

Value after reset: x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x

The SAR sets the transfer source start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x04) and DTC transfers it automatically to and from the SAR register. See [section 20.3.1. 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.

20.2.4 DAR : DTC Transfer Destination Register

Base address: DTCVBR

Offset address: $0x08 + 0x4 \times \text{Vector number}$
(Inaccessible directly from the CPU. See [section 20.3.1. Allocating Transfer Information and DTC Vector Table](#))

Bit position: 31 0

Bit field:

Value after reset: x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x

The DAR sets the transfer destination start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x08) and DTC transfers it automatically to and from the DAR register. See [section 20.3.1. 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.

20.2.5 CRA : DTC Transfer Count Register A

Base address: DTCVBR

Offset address: $0x0E + 0x4 \times \text{Vector number}$
(Inaccessible directly from the CPU. See [section 20.3.1. Allocating Transfer Information and DTC Vector Table](#))

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:

Value after reset: x x x x x x x x x x x x x x x x x

Bit	Symbol	Function	R/W
7:0	CRAL	Transfer Counter A Lower Register Specify the transfer count.	—
15:8	CRAH	Transfer Counter A Upper Register Specify the transfer count.	—

Note: The function depends on the transfer mode.

Note: Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

The CRA register consists of 16 bits. CRAL is the lower 8 bits and CRAH is the upper 8 bits. CRA is used in normal mode. CRAL and CRAH are used in repeat transfer mode and block transfer mode.

The CRA register cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x0E) and DTC transfers it automatically to and from the CRA register. See [section 20.3.1. Allocating Transfer Information and DTC Vector Table](#).

(1) Normal transfer mode (MRA.MD[1:0] = 00b)

In normal transfer mode, CRA functions as a 16-bit transfer counter. The transfer count is 1, 65535, and 65536 when the set value is 0x0001, 0xFFFF, and 0x0000, respectively. The CRA value is decremented (-1) on each data transfer.

(2) Repeat transfer mode (MRA.MD[1:0] = 01b)

In repeat transfer mode, the CRAH register holds the transfer count and the CRAL register functions as an 8-bit transfer counter. The transfer count is 1, 255, and 256 when the set value is 0x01, 0xFF, and 0x00, respectively. The CRAL value is decremented (-1) on each data transfer. When it reaches 0x00, the CRAH value is transferred to CRAL.

(3) Block transfer mode (MRA.MD[1:0] = 10b)

In block transfer mode, the CRAH register holds the block size and the CRAL register functions as an 8-bit block size counter. The transfer count is 1, 255, and 256 when the set value is 0x01, 0xFF, and 0x00, respectively. The CRAL value is decremented (-1) on each data transfer. When it reaches 0x00, the CRAH value is transferred to CRAL.

20.2.6 CRB : DTC Transfer Count Register B

Base address: DTCVBR

Offset address: 0x0C + 0x4 × Vector number
(Inaccessible directly from the CPU. See [section 20.3.1. Allocating Transfer Information and DTC Vector Table](#))

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

The CRB sets the block transfer count for block transfer mode. The transfer count is 1, 65535, and 65536 when the set value is 0x0001, 0xFFFF, and 0x0000, respectively. The CRB value is decremented (-1) when the final data of a single block size is transferred. When normal transfer mode or repeat transfer mode is selected, this register is not used and the set value is ignored.

The CRB cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x0C) and DTC transfers it automatically to and from the CRB register. See [section 20.3.1. Allocating Transfer Information and DTC Vector Table](#).

20.2.7 DTCCR : DTC Control Register

Base address: DTC = 0x4000_5400

Offset address: 0x00

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

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
4	RRS	DTC Transfer Information Read Skip Enable 0: Transfer information read is not skipped 1: Transfer information read is skipped when vector numbers match	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

RRS bit (DTC Transfer Information Read Skip Enable)

The RRS bit enables skipping of transfer information reads when vector numbers match. The DTC vector number is compared with the vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transfer information. However, when the previous transfer is a chain transfer, the transfer information is read regardless of the RRS bit.

When the transfer counter (CRA register) becomes 0 during the previous normal transfer and when the transfer counter (CRB register) becomes 0 during the previous block transfer, the transfer information is read regardless of the RRS bit value.

20.2.8 DTCVBR : DTC Vector Base Register

Base address: DTC = 0x4000_5400

Offset address: 0x04

Bit position: 31

0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	DTC Vector Base Address Set the DTC vector base address. The lower 10 bits should be 0.	R/W

The DTCVBR sets the base address for calculating the DTC vector table address, which can be set in the range of 0x0000_0000 to 0xFFFF_FFFF (4 GB) in 1-KB units.

20.2.9 DTCST : DTC Module Start Register

Base address: DTC = 0x4000_5400

Offset address: 0x0C

Bit position: 7 6 5 4 3 2 1 0

Bit field:

—	—	—	—	—	—	—	DTCS T

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	DTCST	DTC Module Start 0: DTC module stopped 1: DTC module started	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

DTCST bit (DTC Module Start)

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is set to 0, transfer requests are no longer accepted. If this bit is set to 0 during a data transfer, the accepted transfer request is active until processing completes.

DTCST must be set to 0 before transitioning to one of the following state or mode:

- Module-stop state
- Software Standby mode without Snooze mode transition.
- Deep Software standby mode

For details on these transitions, see [section 20.9. Low Power Consumption Function](#) and [section 13, Power-Saving Functions](#).

20.2.10 DTCSTS : DTC Status Register

Base address: DTC = 0x4000_5400

Offset address: 0x0E

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

Bit	Symbol	Function	R/W
7:0	VECN[7:0]	DTC-Activating Vector Number Monitoring These bits indicate the vector number for the activation source when a DTC transfer is in progress. The value is only valid if a DTC transfer is in progress (ACT flag is 1).	R
14:8	—	These bits are read as 0.	R
15	ACT	DTC Active Flag 0: DTC transfer operation is not in progress 1: DTC transfer operation is in progress	R

VECN[7:0] bits (DTC-Activating Vector Number Monitoring)

While transfer by the DTC is in progress, the VECN[7:0] bits indicate the vector number associated with the activation source for the transfer. The value read from the VECN[7:0] bits is valid if the ACT flag is 1, indicating a DTC transfer in progress, and invalid if the ACT flag is 0, indicating no DTC transfer is in progress.

ACT flag (DTC Active Flag)

The ACT flag indicates the state of the DTC transfer operation.

[Setting condition]

- When the DTC is activated by a transfer request.

[Clearing condition]

- When transfer by the DTC, in response to a transfer request, is complete.

20.3 Activation Sources

The DTC is activated by an interrupt request. Setting the ICU.IELSRn.DTCE bit to 1 enables activation of the DTC by the associated interrupt. The selector output n number set in ICU.IELSRn is defined as the interrupt vector number, where $n = 0$ to 31. For an enabled interrupt, the specific DTC interrupt source associated with each interrupt vector number n is selected in ICU.IELSRn.IELS[4:0] where $n = 0$ to 31, as listed in [section 16.3.2. Event Table](#) in [section 16, Interrupt Controller Unit \(ICU\)](#). For activation by software, see [section 21.2.3. ELSEGRn : Event Link Software Event Generation Register n \(\$n = 0, 1\$ \)](#).

The interrupt vector number is equivalent to the DTC vector table number. After the DTC accepted an activation request, it does not accept another activation request until 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 flag of the activation source is set to 0 at the start of the data transfer.

20.3.1 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 20.2 shows the relationship between the DTC vector table and transfer information. Figure 20.3 shows the allocation of transfer information in the SRAM area.

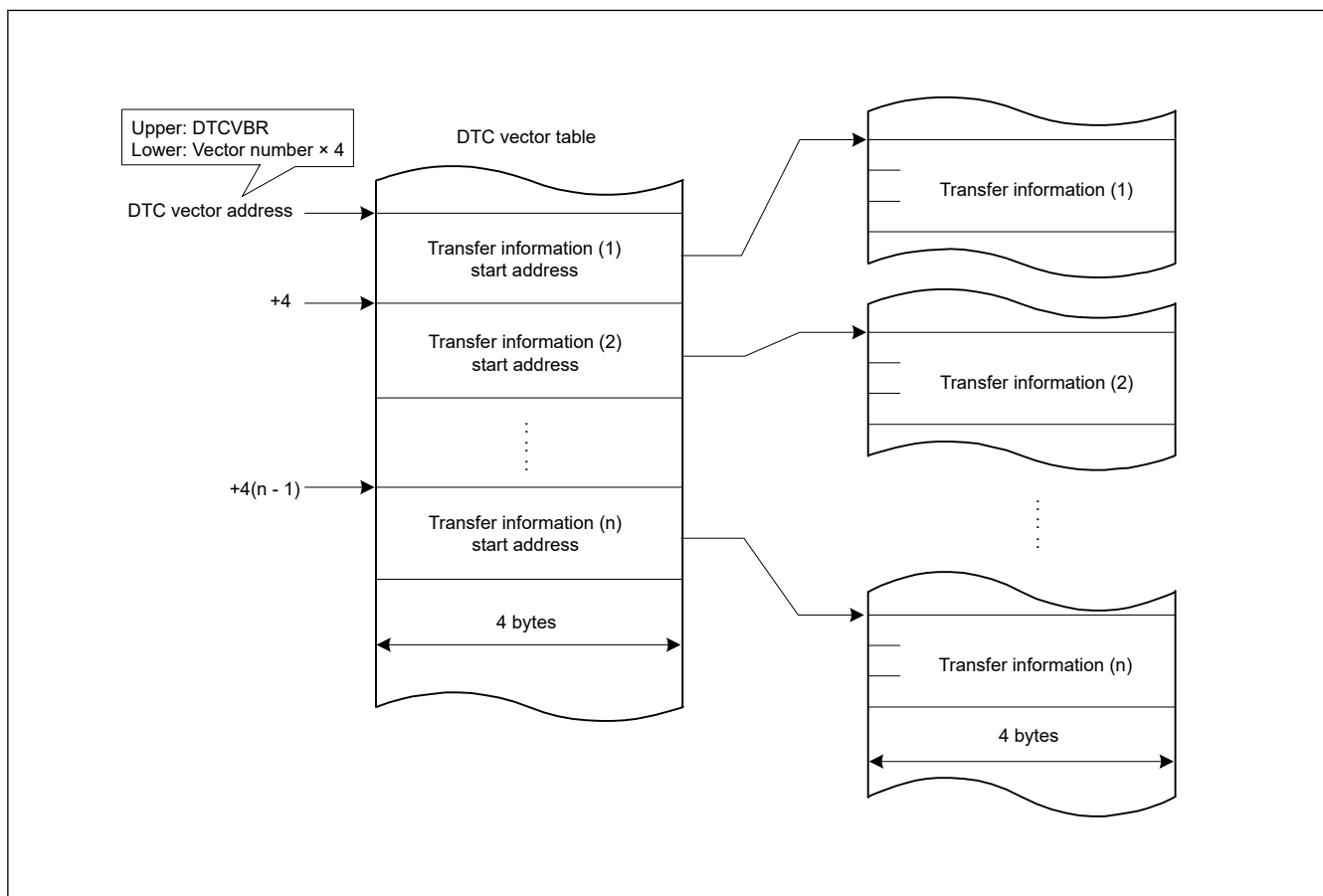


Figure 20.2 DTC vector table and transfer information

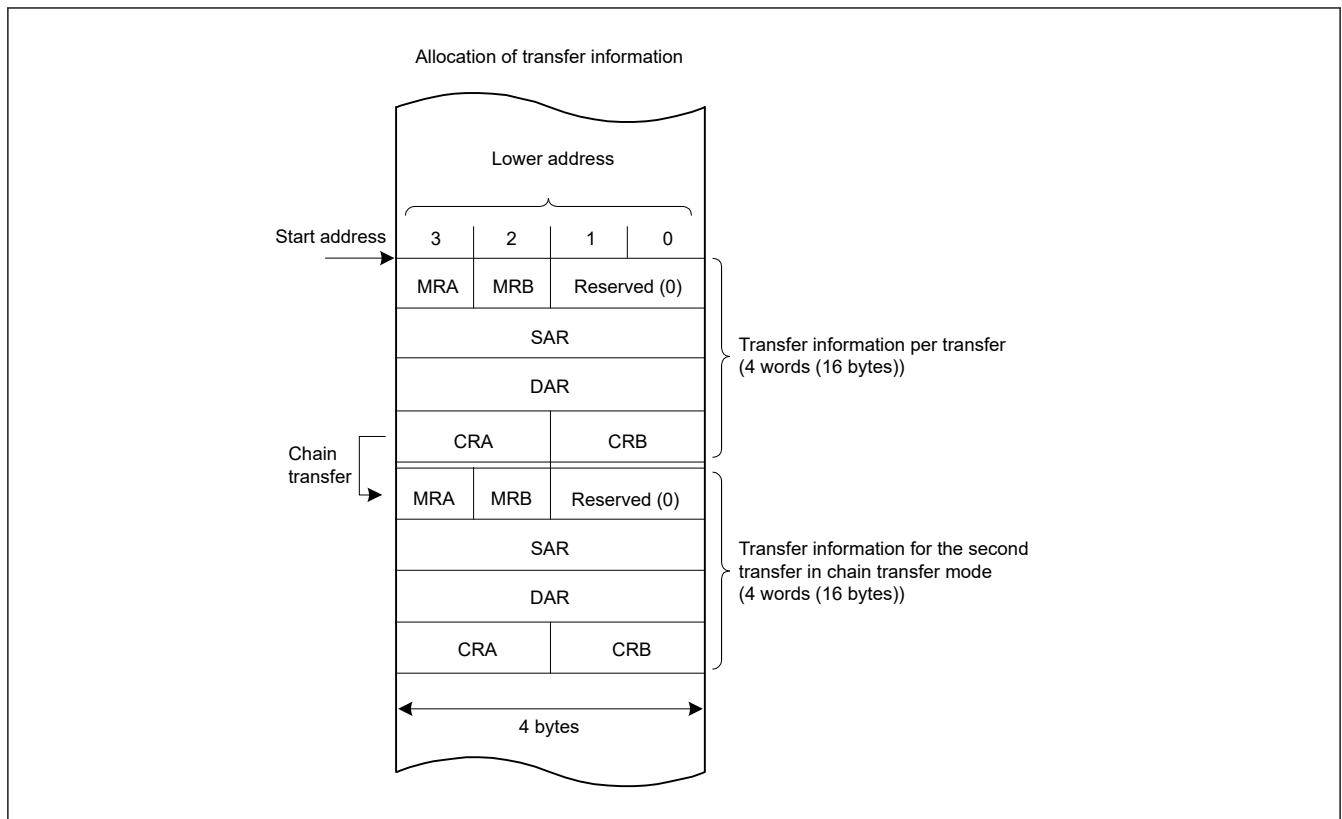


Figure 20.3 Allocation of transfer information in the SRAM area

20.4 Operation

The DTC transfers data according to the transfer information. Storage of the transfer information in the SRAM area is required before a DTC operation. When the DTC is activated, it reads the DTC vector associated with the vector number. The DTC reads the transfer information from the transfer information store address referenced by the DTC vector and transfers the data. After the data transfer, the DTC writes back the transfer information. Storing the transfer information in the SRAM area allows data transfer of any number of channels.

The transfer modes include:

- Normal transfer mode
- Repeat transfer mode
- Block transfer mode.

The DTC specifies a transfer source address in the SAR register and a transfer destination address in the DAR register. The values of these registers are incremented, decremented, or address-fixed independently after the data transfer.

Table 20.2 describes the DTC transfer modes.

Table 20.2 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 20.4 shows the operation flow of the DTC. Table 20.3 lists the chain transfer conditions. The combination of control information for the second and subsequent transfers are omitted in this table.

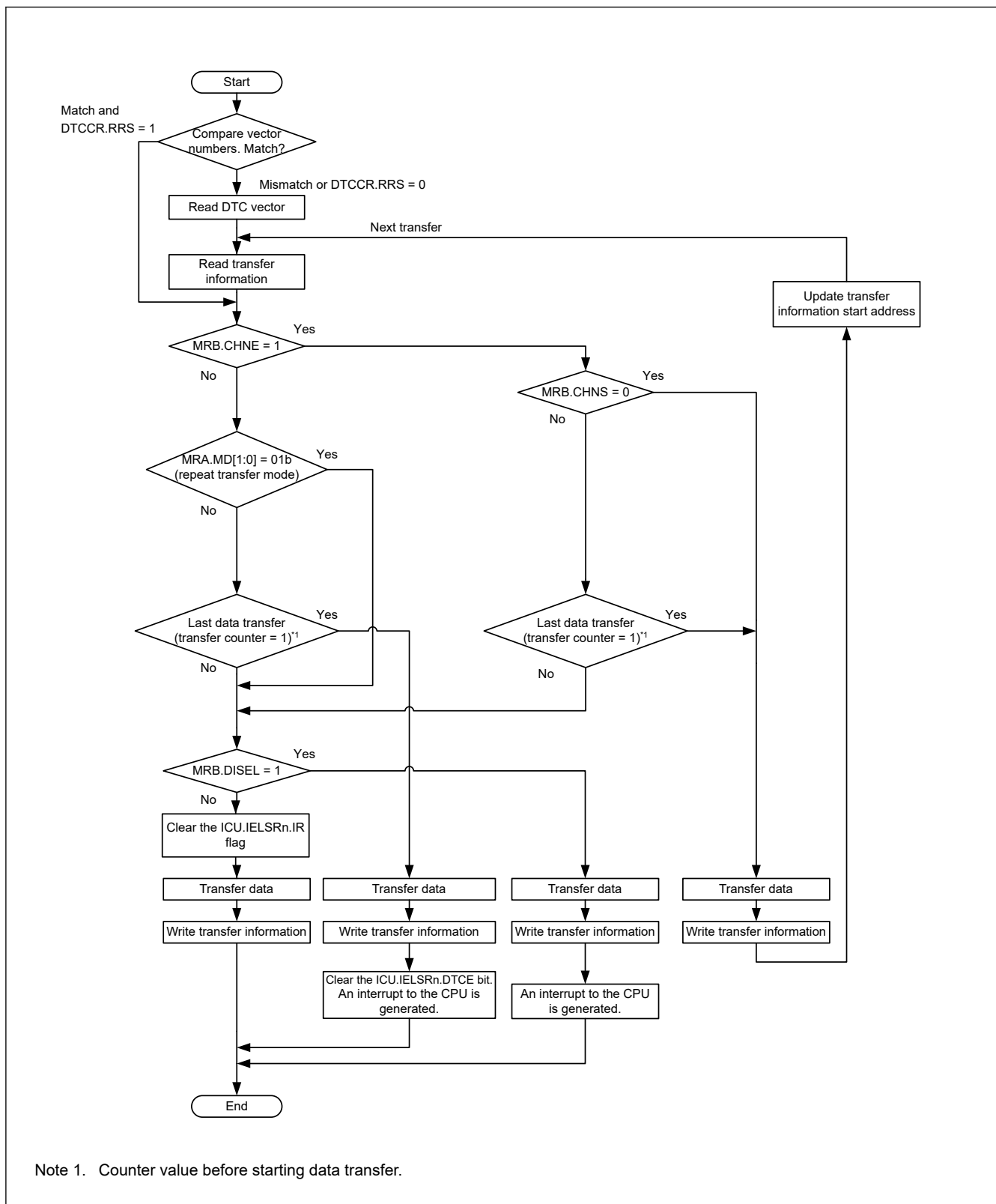


Figure 20.4 DTC operation flow

Table 20.3 Chain transfer conditions

First transfer				Second transfer ^{*3}				DTC transfer
CHNE bit	CHNS bit	DISEL bit	Transfer counter ^{*1 *2}	CHNE bit	CHNS bit	DISEL bit	Transfer counter ^{*1 *2}	
0	—	0	Other than (1 → 0)	—	—	—	—	Ends after the first transfer
0	—	0	(1 → 0)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	0	Other than (1 → *)	—	—	—	—	Ends after the first transfer
1	1	—	(1 → *)	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	1	Other than (1 → *)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU

Note 1. The transfer counter used depends on the transfer modes as follows:

Normal transfer mode — CRA register
Repeat transfer mode — CRAL register
Block transfer mode — CRB register

Note 2. On completion of a data transfer, the counters operate as follows:

1 → 0 in normal and block transfer modes
1 → CRAH in repeat transfer mode

(1 → *) in the table indicates both of these two operations, depending on the mode.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The conditions for the combination of the second transfer and CHNE = 1 is omitted.

20.4.1 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 20.12](#) 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.

20.4.2 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 20.4](#) 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 20.4 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		

20.4.3 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 and destination addresses can be independently set to increment, decrement, or fixed. This mode enables an interrupt request to the CPU to be generated at the end of a specified-count transfer.

[Table 20.5](#) lists register functions in normal transfer mode, and [Figure 20.5](#) shows the memory map of normal transfer mode.

Table 20.5 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.

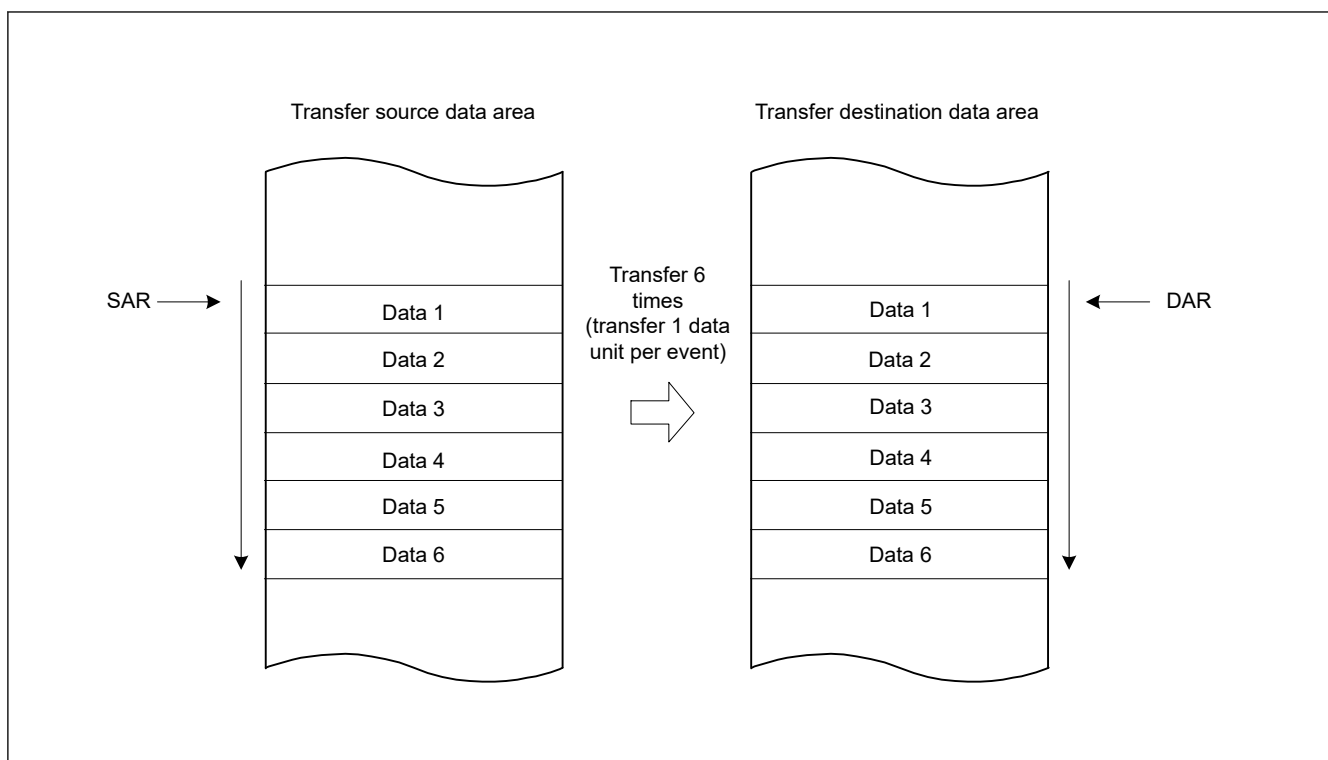


Figure 20.5 Memory map of normal transfer mode (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRA = 0x0006)

20.4.4 Repeat Transfer Mode

The repeat transfer mode allows a 1-byte (8-bit), 1-halfword (16-bit), or 1-word (32-bit) data transfer on a single activation source. Transfer source or transfer destination for the repeat area must be specified in the MRB.DTS bit. The transfer count can be set from 1 to 256. When the specified transfer count is complete, the initial value of the address register specified in the repeat area is restored, the initial value of the transfer counter is restored, and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.

When the transfer counter CRAL decrements to 0x00 in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. As a result, the transfer counter does not clear to 0x00, which disables interrupt requests to the CPU when the MRB.DISEL bit is set to 0. An interrupt request to the CPU is generated when the specified data transfer completes.

Table 20.6 lists the register functions in repeat transfer mode, and Figure 20.6 shows the memory map of repeat transfer mode.

Table 20.6 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.

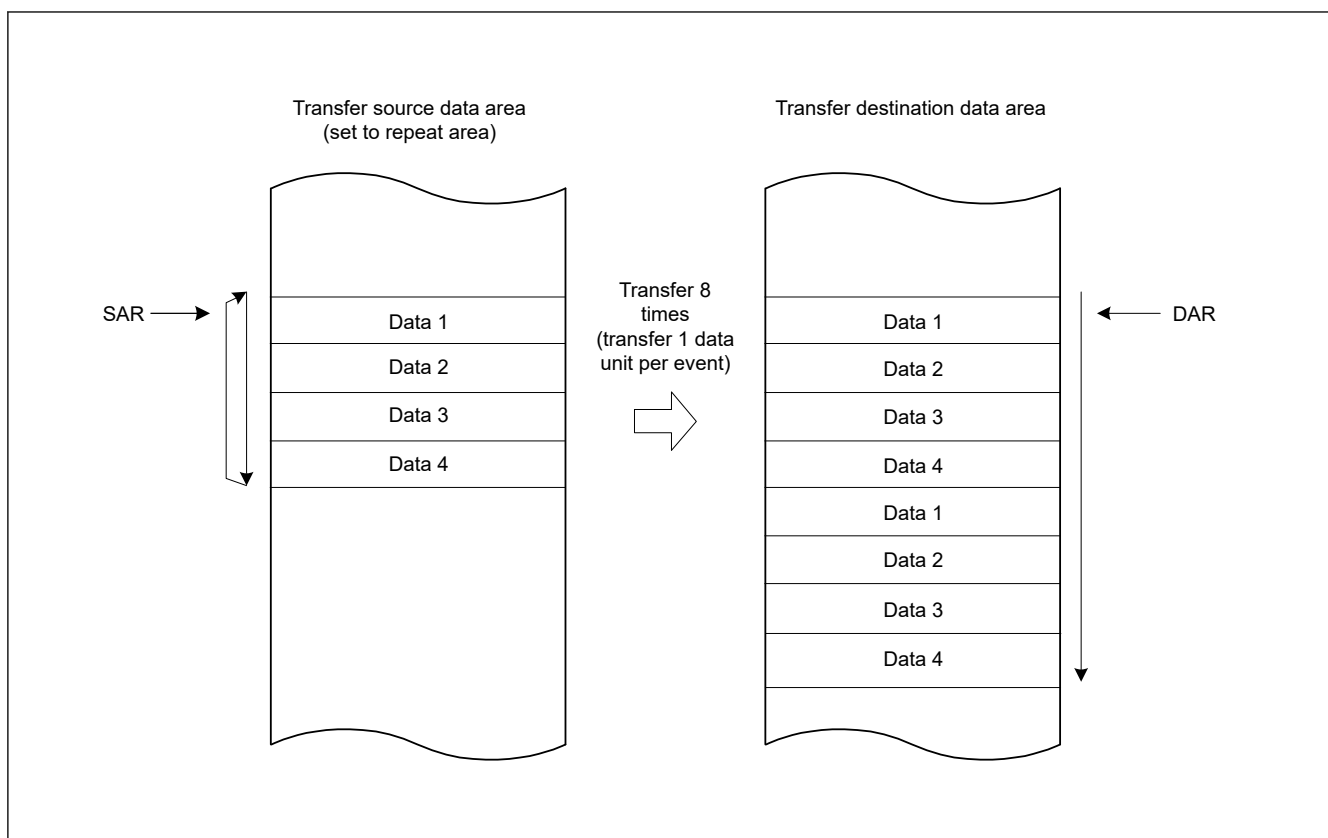


Figure 20.6 Memory map of repeat transfer mode when transfer source is a repeat area (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRAH = 0x04)

20.4.5 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 20.7 lists the register functions in block transfer mode, and Figure 20.7 shows the memory map for block transfer mode.

Table 20.7 Register functions in block transfer mode

Register	Description	Value written back by writing transfer information
SAR	Transfer source address	- When MRB.DTS bit is 0 Increment, decrement, or fixed*1 - When MRB.DTS bit is 1 SAR register initial value.
DAR	Transfer destination address	- When MRB.DTS bit is 0 DAR register initial value - When MRB.DTS bit is 1 Increment, decrement, or fixed*1.
CRAH	Holds block size	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB - 1

Note 1. Write-back is skipped in address-fixed mode.

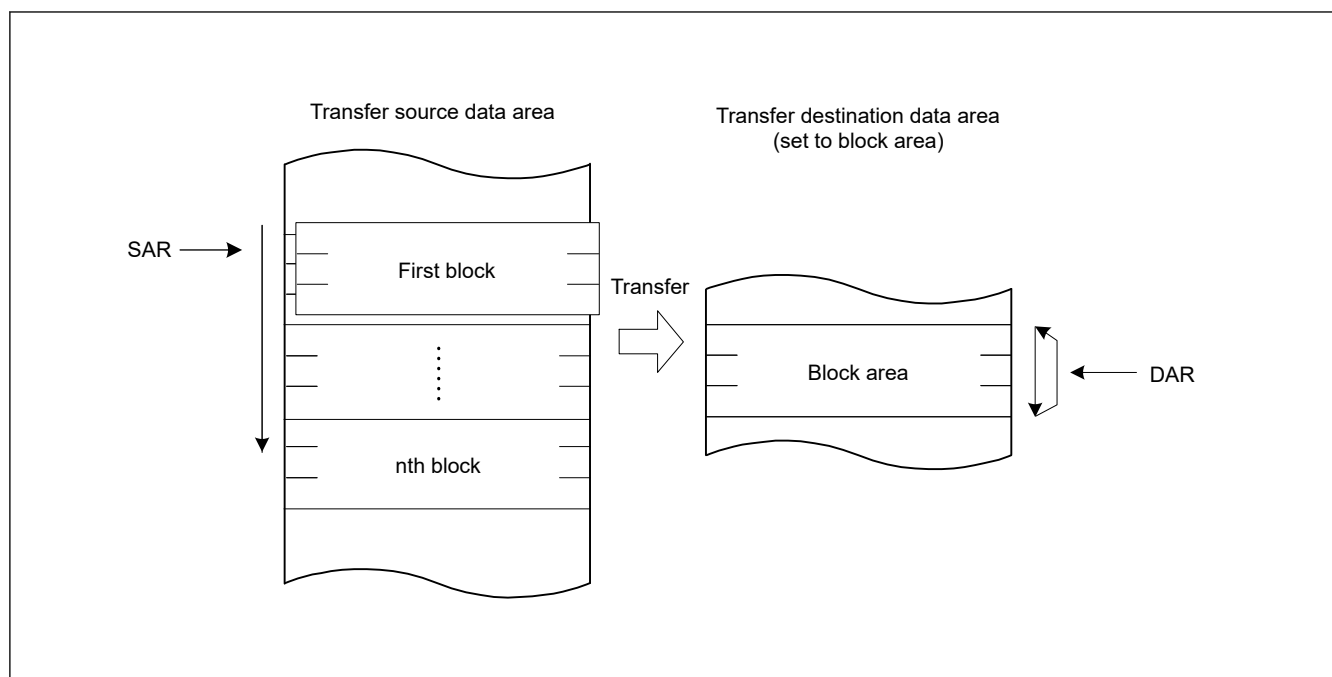


Figure 20.7 Memory map of block transfer mode

20.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single activation source. If the MRB.CHNE is set to 1 and CHNS to 0, an interrupt request to the CPU is not generated on completion of the specified number of rounds of transfer or by setting the MRB.DISEL bit to 1. An interrupt request is sent to the CPU each time DTC data transfer is performed. Data transfer has no effect on the ICU.IELSRn.IR flag of the activation source.

The SAR, DAR, CRA, CRB, MRA, and MRB registers can be set independently of each other to define the data transfer.

Figure 20.8 shows a chain transfer operation.

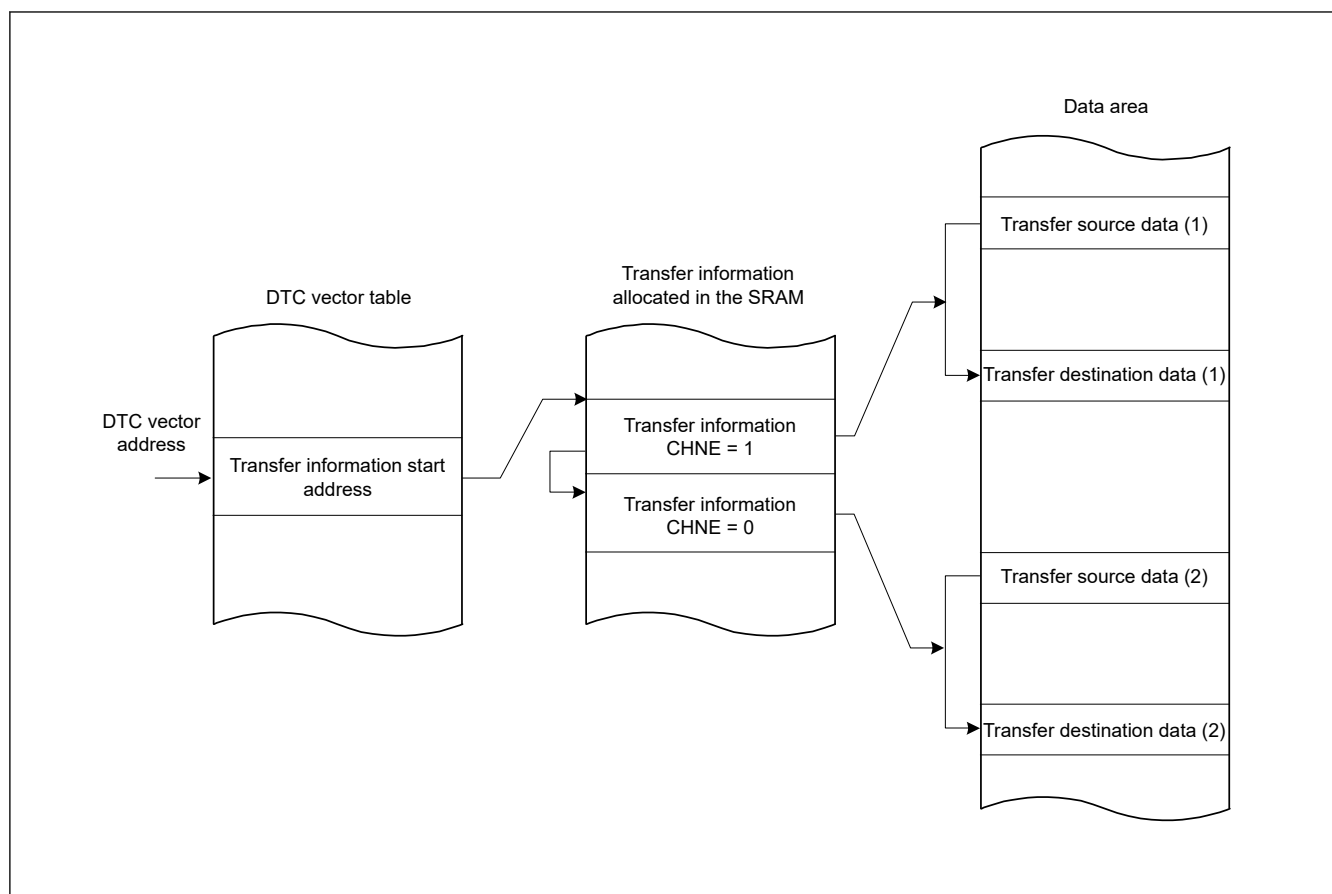


Figure 20.8 Chain transfer operation

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 20.2](#).

20.4.7 Operation Timing

[Figure 20.9](#) to [Figure 20.12](#) are timing diagrams that show the minimum number of execution cycles.

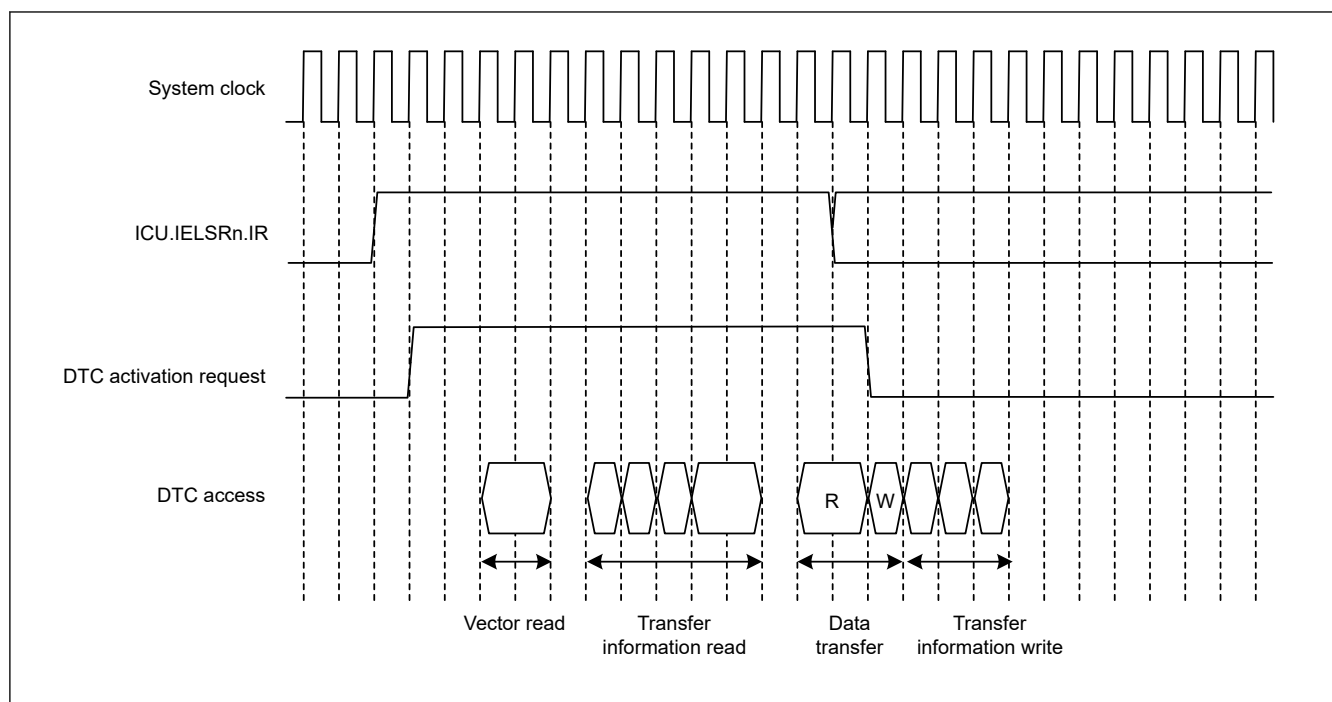


Figure 20.9 Example 1 of DTC operation timing in normal transfer and repeat transfer modes

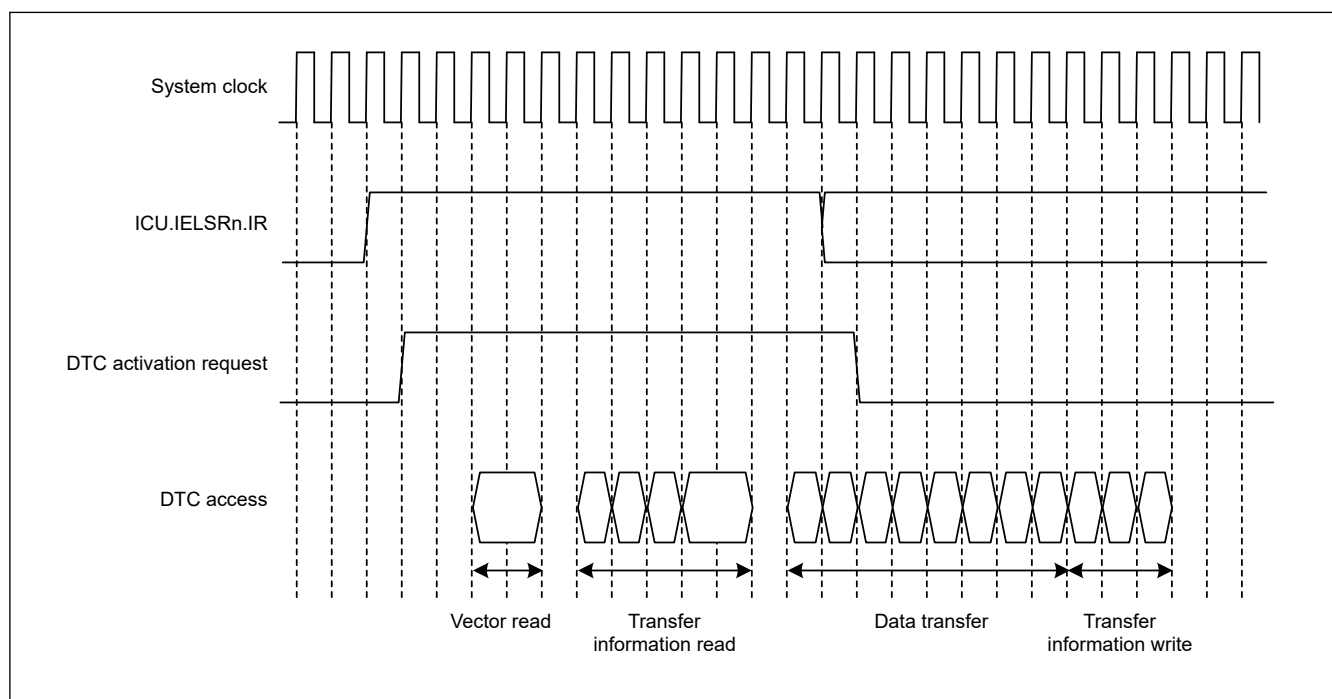


Figure 20.10 Example 2 of DTC operation timing in block transfer mode when the block size = 4

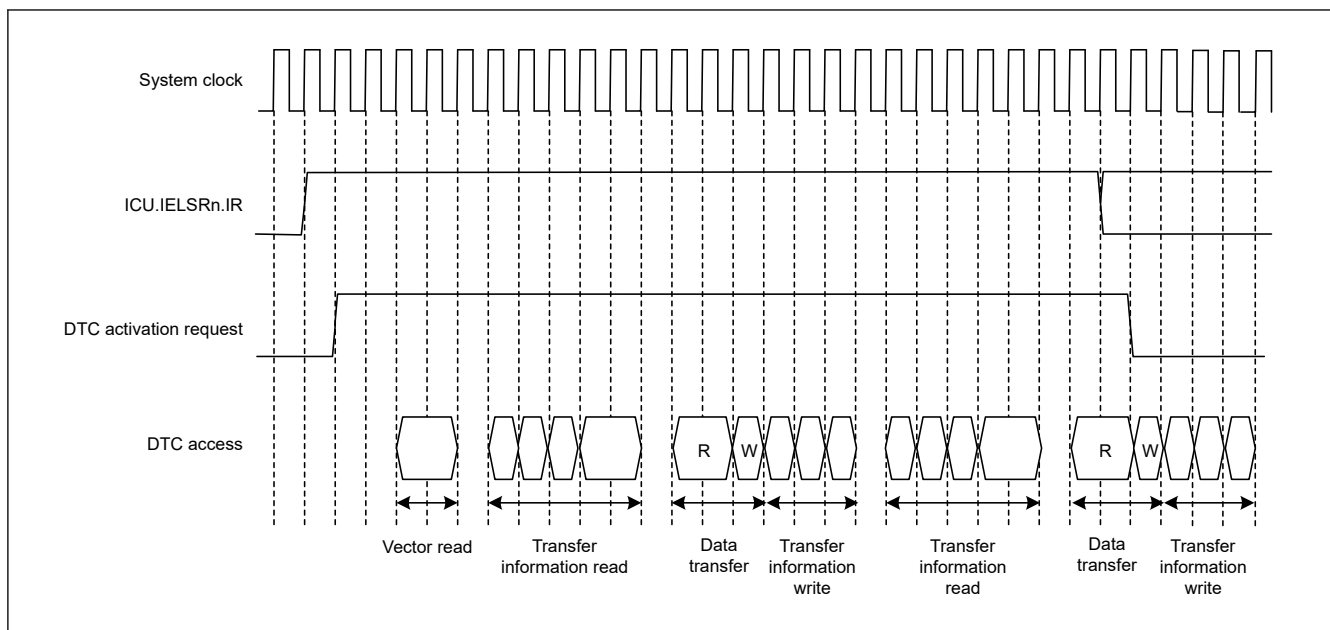


Figure 20.11 Example 3 of DTC operation timing for chain transfer

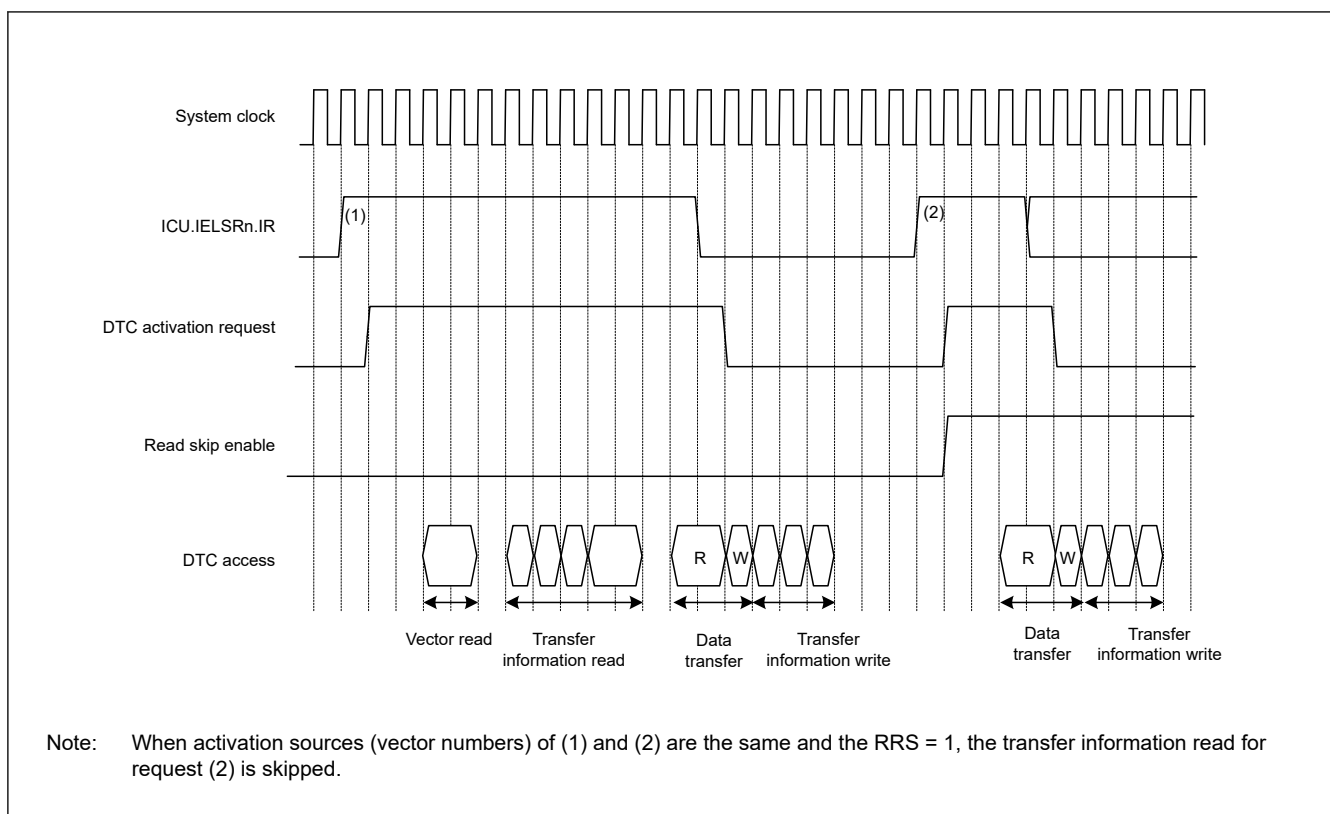


Figure 20.12 Example of operation when a transfer information read is skipped with the vector, transfer information, and transfer destination data on the SRAM, and the transfer source data on the peripheral module

20.4.8 Execution Cycles of DTC

Table 20.8 lists the execution cycles of single data transfer of the DTC. For the order of the execution states, see [section 20.4.7. Operation Timing](#).

Table 20.8 Execution cycles of DTC

P: Block size (initial settings of CRAH and CRAL)

Cv: Cycles for access to vector transfer information storage destination

Ci: Cycles for access to transfer information storage destination address

Cr: Cycles for access to data read destination

Cw: Cycles for access to data write destination

The unit is for system clocks (ICLK) + 1 in the Vector read, Transfer information read, and Data transfer read columns and 2 in the Internal operation column.

Cv, Ci, Cr, and Cw vary depending on the corresponding access destination. For the number of cycles for respective access destinations, see [section 49, SRAM](#), [section 50, Flash Memory](#), and [section 17, Buses](#).

The frequency ratio of the system clock and peripheral clock is also taken into consideration.

The DTC response time is the time from when the DTC activation source is detected until DTC transfer starts.

[Table 20.8](#) does not include the time until DTC data transfer starts after the DTC activation source becomes active.

Transfer mode	Vector read		Transfer information read		Transfer information write			Data transfer		Internal operation	
								Read	Write		
Normal	Cv + 1	0 ^{*1}	4 × Ci + 1	0 ^{*1}	3 × Ci + 1 ^{*2}	2 × Ci + 1 ^{*3}	Ci ^{*4}	Cr + 1	Cw + 1	2	0 ^{*1}
Repeat								Cr + 1	Cw + 1		
Block ^{*5}								P × Cr	P × Cw		

Note 1. When transfer information read is skipped.

Note 2. When neither SAR nor DAR is set to address-fixed mode.

Note 3. When SAR or DAR is set to address-fixed mode.

Note 4. When SAR and DAR are set to address-fixed mode.

Note 5. When the block size is 2 or more. If the block size is 1, the cycle number for normal transfer applies.

20.4.9 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 17, Buses](#).

20.5 DTC Setting Procedure

Before using the DTC, set the DTC Vector Base Register (DTCVBR). Set the ICU.IELSRn.IELS[4:0] bits to 0 to disable the interrupt in the NVIC and follow the procedure in [Table 20.9](#) to set the DTC.

Table 20.9 DTC setting procedure (1 of 2)

No.	Step Name	Description
1	Set the DTCCR.RRS bit to 0	Set the DTCCR.RRS bit to 0 to reset the transfer information read skip flag. After that, the transfer information read is not skipped while the DTC is activated. Be sure to specify this setting when the transfer information is updated.
2	Set transfer information (MRA, MRB, SAR, DAR, CRA, and CRB)	Allocate transfer information (MRA, MRB, SAR, DAR, CRA, and CRB) in the data area. To set transfer information, see section 20.2, Register Descriptions . To allocate transfer information, see section 20.3.1, Allocating Transfer Information and DTC Vector Table .
3	Set transfer information start addresses in the DTC vector table	Set the transfer information start addresses in the DTC vector table. To set the DTC vector table, see section 20.3.1, Allocating Transfer Information and DTC Vector Table .
4	Set the DTCCR.RRS bit to 1	Set the DTCCR.RRS bit to 1 to enable skipping of the second and subsequent transfer information read cycles for continuous DTC activation from the same interrupt source. The RRS bit can be set to 1, but if this is set during DTC transfer, it becomes valid from the next transfer.
5	Set the ICU.IELSRn.DTCE bit to 1. Set the ICU.IELSRn.IELS[4:0] as interrupt source. The interrupt should be enabled in the NVIC.	Set the ICU.IELSRn.DTCE bit to 1. Set ICU.IELSRn.IELS[4:0] as interrupt sources that trigger DTC. The interrupt must be enabled in the NVIC. See section 16.3.2, Event Table in section 16, Interrupt Controller Unit (ICU) .
6	Set the enable bit for an activation source interrupt	Set the enable bit for the activation source interrupts to 1. When a source interrupt is generated, the DTC is activated. To set the interrupt source enable bit, see the settings for the modules that are to be the activation sources.

Table 20.9 DTC setting procedure (2 of 2)

No.	Step Name	Description
7	Set the DTCST.DTCST bit to 1	Set the DTC Module Start bit (DTCST.DTCST) to 1.

Note: The DTCST.DTCST bit can be set even if the setting for each activation source is not completed.

20.6 Examples of DTC Usage

20.6.1 Normal Transfer

This section provides an example of DTC usage and its application when receiving 128 bytes of data from an SCI.

(1) Transfer information settings

In the MRA register, select a fixed source address (MRA.SM[1:0] = 00b), normal transfer mode (MRA.MD[1:0] = 00b), and byte-sized transfer (MRA.SZ[1:0] = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] = 10b) and single data transfer by a single interrupt (MRB.CHNE = 0 and MRB.DISEL = 0). The MRB.DTS bit can be set to any value. Set the RDR register address of the SCI in the SAR register, the start address of the SRAM area for data storage in the DAR register, and 128 (0x0080) in the CRA register. The CRB register can be set to any value.

(2) DTC vector table settings

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

(3) ICU settings and DTC module activation

Set the ICU.IELSRn.DTCE bit to 1 and set ICU.IELSRn.IELS[4:0] as the SCI interrupt. The interrupt must be enabled in the NVIC. Set the DTCST.DTCST bit to 1.

(4) SCI settings

Enable the SCIn_RXI interrupt by setting the SCR.RIE bit in the SCI to 1. If a reception error occurs during the SCI receive operation, reception stops. To manage this, use settings that allow the CPU to accept receive error interrupts.

(5) DTC transfer

Each time a reception of 1 byte by the SCI is complete, an SCIn_RXI interrupt is generated to activate the DTC. The DTC transfers the received byte from the RDR of the SCI to the SRAM, after which the DAR register is incremented and the CRA register is decremented.

(6) Interrupt handling

After 128 rounds of data transfer are complete and the value in the CRA register becomes 0, an SCIn_RXI interrupt request is generated for the CPU. Complete the process in the handling routine for this interrupt.

20.6.2 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 to 321, 162 to 165). For the second transfer, normal transfer mode is specified for transfer to the GPTm.GTCCRE register (m = 320 to 321, 162 to 165). For the third transfer of the chained transfer, normal transfer mode for transfer to the GPTm.GTPBR register (m = 320 to 321, 162 to 165) 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 register 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 register.
6. Set the CRA register to the size of the data table. The CRB register can be set to any value.

(4) Transfer information assignment

Place the transfer information for use in the transfer to the GPT320.GTPBR immediately after the transfer control information for use in the GPT320.GTCCRC and GPT320.GTCCRE registers.

(5) DTC vector table

In the DTC vector table, set the address where the transfer control information for use in transfer to the GPT320.GTCCRC and GPT320.GTCCRE registers starts.

(6) ICU setting and DTC module activation

1. Set the ICU.IELSRn.DTCE bit associated with the GPT320 counter overflow interrupt.
2. Set the ICU.IELSRn.IELS[4:0] bits and specify 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.

20.6.3 Chain Transfer when Counter = 0

The second data transfer is performed only when the transfer counter is set to 0 in the first data transfer, and the first data transfer information is repeatedly changed in the second transfer. Chain transfer enables transfers to be repeated 256 times or more.

The following procedure shows an example of configuring a 1-KB input buffer, where the input buffer is set so that its lower address starts with 0x00. [Figure 20.13](#) shows a chain transfer when the counter = 0.

1. Set the normal transfer mode to input data for the first data transfer. Set the following:
 - (a) Transfer source address = fixed.
 - (b) CRA register = 0x0200 (512) times.
 - (c) MRB.CHNE bit = 1 (chain transfer is enabled).
 - (d) MRB.CHNS bit = 1 (chain transfer is performed only when the transfer counter is 0).
 - (e) MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
2. Prepare the upper 8-bit address of the start address at every 512 times of the transfer destination address for the first data transfer in different area such as the flash. For example, when setting the input buffer to 0x8000 to 0x83FF, prepare 0x82 and 0x80.
3. For the second data transfer:
 - (a) Set the repeat transfer mode (with transfer source and destination address = fixed.) to reset the transfer counter of the first data transfer.
 - (b) Specify the CRA register in the first transfer information area for the transfer destination.
 - (c) Set the MRB.CHNE bit = 1 (chain transfer is enabled).
 - (d) Set the MRB.CHNS bit = 0 (select continuous chain transfer).
 - (e) Set the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
 - (f) CRA register = 0x0101 (The transfer count is 1).
4. For the third data transfer:
 - (a) Set the repeat transfer mode (with the source as the repeat area) to reset the transfer destination address of the first data transfer.
 - (b) Specify the upper 8 bits of the DAR register in the first transfer information area for the transfer destination.
 - (c) Set the MRB.CHNE bit = 0 (chain transfer is disabled).

- (d) Set the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
 - (e) When setting the input buffer to 0x8000 to 0x83FF, also set the transfer counter to 2.
5. The first data transfer is performed by an interrupt 512 times. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the transfer counter of the first data transfer to 0x0200. The lower 8 bits of the transfer destination address and the transfer counter of the first data transfer becomes 0x0200.
 6. The second data transfer is performed by an interrupt 1 times. When the transfer counter of the first data transfer becomes 0, the third data transfer starts. Set the upper 8 bits of the transfer destination address of the first data transfer to 0x82. The lower 8 bits of the transfer destination address becomes 0x00 and the transfer counter of the first data transfer becomes 0x0200.
 7. In succession, the first data transfer is performed by an interrupt 512 times as specified for the first data transfer. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the transfer counter of the first data transfer to 0x0200. The lower 8 bits of the transfer destination address and the transfer counter of the first data transfer becomes 0x0200.
 8. The second data transfer is performed by an interrupt 1 times. When the transfer counter of the first data transfer becomes 0, the third data transfer starts. Set the upper 8 bits of the transfer destination address of the first data transfer to 0x80. The lower 8 bits of the transfer destination address becomes 0x00 and the transfer counter of the first data transfer becomes 0x0200.
 9. Steps 5 to 8 are repeated indefinitely. Because the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

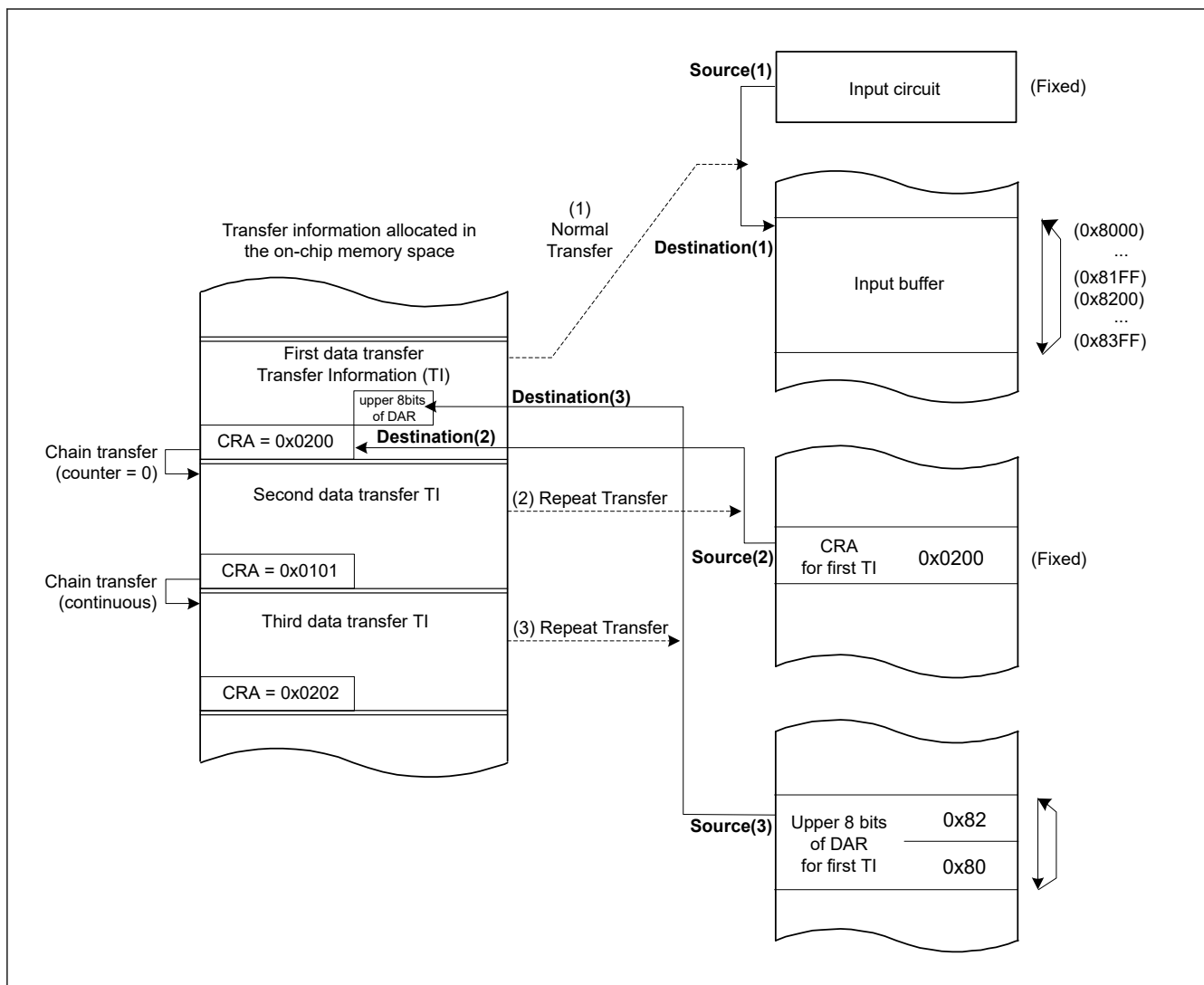


Figure 20.13 Chain transfer when counter = 0

20.7 Interrupt

20.7.1 Interrupt Sources

When the DTC completes data transfer of the specified count or when data transfer with MRB.DISEL set to 1 is complete, a DTC activation source generates an interrupt to the CPU. Two types of interrupt are available: interrupts triggered by a DTC activation (per channel) and an interrupt triggered by the event signal DTC_COMPLETE (common to all channels). Interrupts to the CPU are controlled according to the settings in the NVIC and the ICU.IELSRn.IELS[4:0] bits. See [section 16, 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.

20.8 Event Link

The DTC can produce an event link request on completion of one transfer request.

20.9 Low Power Consumption Function

Before transitioning to the module-stop function, Software Standby mode without Snooze mode transition, deep software standby mode, set the DTCST.DTCST bit to 0, and then perform the operations described in the following sections. The DTC is available in Snooze mode by setting the SYSTEM.SNZCR.SNZDTCEN bit to 1. See [section 13, Power-Saving Functions](#).

(1) Module-stop function

Writing 1 to the MSTPCRA.MSTPA22 bit enables the module-stop function of the DTC. If the DTC transfer is in progress at the time, 1 is written to the MSTPCRA.MSTPA22 bit. The transition to the module-stop state proceeds after DTC transfer ends. While the MSTPCRA.MSTPA22 bit is 1, accessing the DTC registers is prohibited. Writing 0 to the MSTPCRA.MSTPA22 bit releases the DTC from the module-stop state.

(2) Software Standby mode, deep software standby mode

Use the settings described in [section 13.6.2.1. Transition to Software Standby Mode](#) or [section 13.6.4.1. Transition to Deep Software Standby Mode](#).

If DTC transfer operations are in progress when the WFI instruction is executed, the transition to Software Standby mode or deep software standby mode follows the completion of the DTC transfer.

(3) Snooze Mode

When the snooze control circuit receives a snooze request in Software Standby mode, the MCU transfers to Snooze mode. See [section 13.6.3.1. Transition to Snooze Mode](#). DTC operation in Snooze mode can be selected in the SYSTEM.SNZCR.SNZDTCEN bit. If DTC operation is enabled in Snooze mode, before transitioning to Software Standby mode, set the DTCST.DTCST bit to 1. To return to Software Standby mode through DTC, set SYSTEM.SNZEDCR.DTCZRED or SYSTEM.SNZEDCR.DTCNZRED to 1. See [section 13.6.3.3. Return from Snooze Mode 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 when 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 when CRA and CRB are not 0. The DTC activation request from the ICU is stopped during Software Standby mode but not stopped during Snooze mode.

(4) Notes on Low Power Consumption Function

For the WFI instruction and the register setting procedure, see [section 13, Power-Saving Functions](#).

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 16.4.2. Detecting Interrupts](#), then execute the WFI instruction. If DTC operation is enabled in Snooze mode, do not use the module-stop function of the DTC.

20.10 Usage Notes

20.10.1 Transfer Information Start Address

You must set multiples of 4 for the transfer information start addresses in the vector table. Otherwise, such addresses are accessed with their lowest 2 bits regarded as 00b.

21. Event Link Controller (ELC)

21.1 Overview

The Event Link Controller (ELC) uses the interrupt requests generated by various peripheral modules as event signals to interconnect (link) modules, allowing direct link between modules without CPU intervention. Event signals can be output regardless of the setting of the associated interrupt request enable bit.

Table 21.1 lists the ELC specifications and Figure 21.1 shows a block diagram.

Table 21.1 ELC specifications

Parameter	Specifications
Event link function	124 types of event signals can be directly connected to modules. - The operation of timer modules can be selected when an event is input to the timer module. - Generates ELC event signals and events to wakeup the DTC.
Function for reducing power consumption	Module stop state can be set

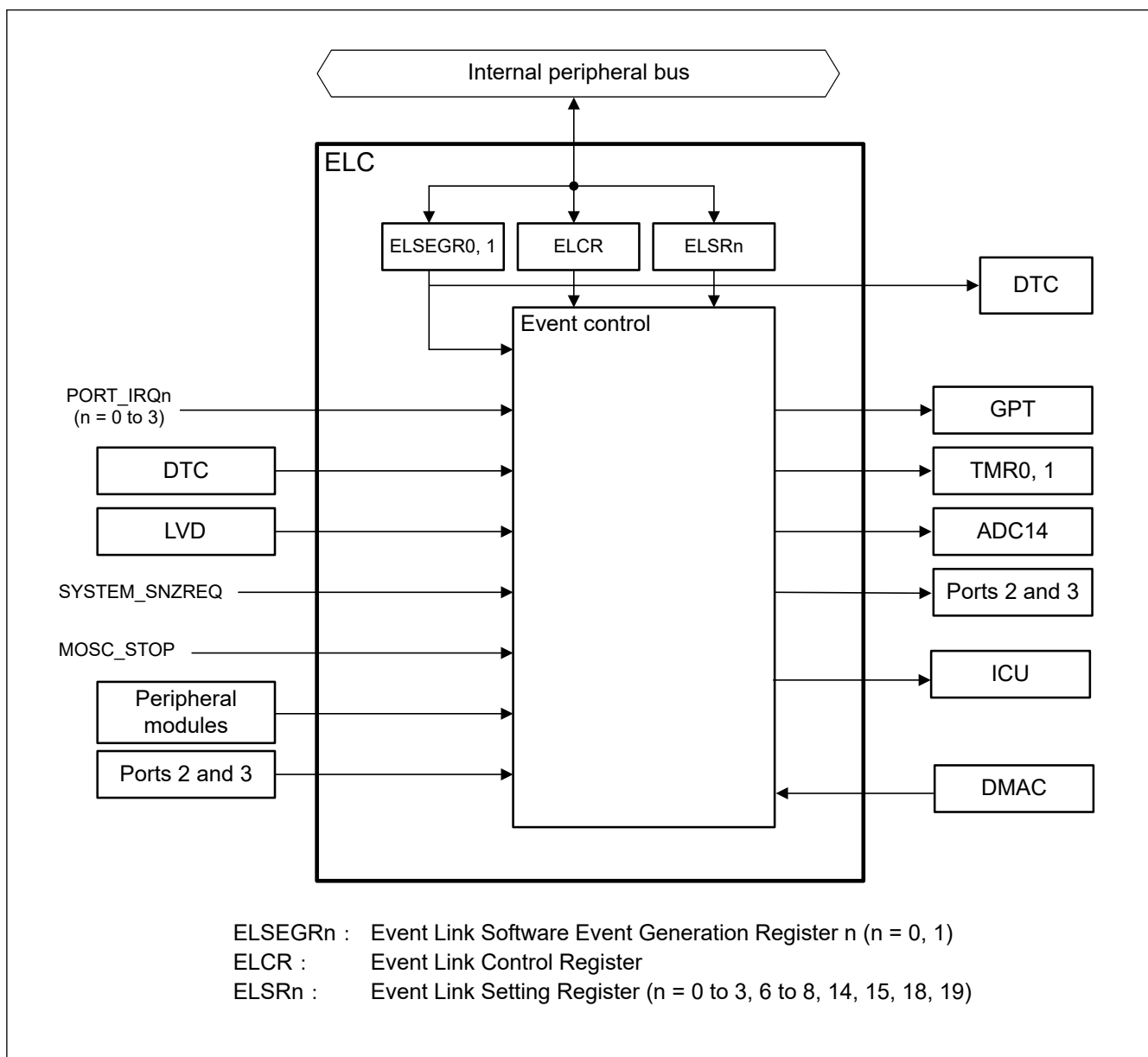


Figure 21.1 ELC block diagram

21.2 Register Descriptions

21.2.1 ELCR : Event Link Control Registers

Base address: ELC = 0x4004_1000

Offset address: 0x00

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	ELCON	All Event Link Enable 0: Disable ELC function 1: Enable ELC function	R/W

The ELCR register controls the ELC operation.

21.2.2 ELSRn : Event Link Setting Register n (n = 0 to 3, 6 to 8, 14, 15, 18, 19)

Base address: ELC = 0x4004_1000

Offset address: 0x10 + 0x04 × n (n = 0 to 3, 6 to 8, 14, 15, 18, 19)

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

Bit	Symbol	Function	R/W
7:0	ELS[7:0]	Event Link Select 0x00: Disable event output for the associated peripheral module 0x01: Number setting for the event signal to be linked ⋮ 0xAB: Number setting for the event signal to be linked Others: Settings prohibited	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

The ELSRn register specifies an event signal to be linked to for each peripheral module. [Table 21.2](#) shows the associations between the ELSRn registers and the peripheral modules. [Table 21.3](#) shows the associations between the event signal names and the signal numbers set in the ELSRn register.

Table 21.2 Associations between ELSRn registers and peripheral modules (1 of 2)

Register name	Peripheral module
ELSR0	GPT (ELCA)
ELSR1	GPT (ELCB)
ELSR2	GPT (ELCC)
ELSR3	GPT (ELCD)
ELSR6	TMR0
ELSR7	TMR1
ELSR8	ADC14 (ELC_ADC14)
ELSR14	PORT3

Table 21.2 Associations between ELSRn registers and peripheral modules (2 of 2)

Register name	Peripheral module
ELSR15	PORT2
ELSR18	ICU (ELC_INT0: event number 0x4E)
ELSR19	ICU (ELC_INT1: event number 0x4F)

Table 21.3 Associations between event signal names and signal numbers set in ELSRn.ELS[7:0] bits (1 of 2)

ELS[7:0] bits value	Interrupt request source	Event signal name	Event description
0x01	Ports	PORT_IRQ0* ¹	External pin interrupt 0
0x02		PORT_IRQ1* ¹	External pin interrupt 1
0x03		PORT_IRQ2* ¹	External pin interrupt 2
0x04		PORT_IRQ3* ¹	External pin interrupt 3
0x09	DMA controller	DMAC0_INT	DMAC0 transfer end
0x0A	DTC	DTC_COMPLETE* ²	DTC transfer end
0x0D	LVD	LVD_LVD1	Voltage monitor 1 interrupt
0x0F	MOSC	MOSC_STOP	Main clock oscillation stop
0x10	Low power consumption mode	SYSTEM_SNZREQ* ^{2,3}	Snooze entry
0x11	AGT0	AGT0_AGTI	AGT0 interrupt
0x12		AGT0_AGTCMAI	AGT0 compare match A
0x13		AGT0_AGTCMBI	AGT0 compare match B
0x14	AGTW0	AGTW0_AGTI	AGTW0 interrupt
0x15		AGTW0_AGTCMAI	AGTW0 compare match A
0x16		AGTW0_AGTCMBI	AGTW0 compare match B
0x17	IWDT	IWDT_NMIUNDF	IWDT underflow
0x18	WDT	WDT_NMIUNDF	WDT underflow
0x1A	RTC	RTC_PRD	RTC periodic interrupt
0x1C	ADC14	ADC140_ADI	A/D scan conversion end interrupt
0x20		ADC140_WCMPPM* ²	A/D window A/B compare match
0x21		ADC140_WCMPUM* ²	A/D window A/B compare mismatch
0x22	TMR0	TMR_CMIA0	TMR compare match A
0x23		TMR_CMIB0	TMR compare match B
0x24		TMR_OVF0	TMR overflow
0x29	IIC0	IIC0_RXI	IIC0 receive data full
0x2A		IIC0_TXI	IIC0 receive data empty
0x2B		IIC0_TEI	IIC0 transmit end
0x2C		IIC0_EEI	IIC0 communication error
0x2E	IIC1	IIC1_RXI	IIC1 receive data full
0x2F		IIC1_TXI	IIC1 receive data empty
0x30		IIC1_TEI	IIC1 transmit end
0x31		IIC1_EEI	IIC1 communication error
0x32	CCC	CCC_PRD	CCC periodic interrupt
0x33		CCC_CUP	CCC overflow
0x36	DOC	DOC_DOPCI* ²	DOC data operation circuit interrupt

Table 21.3 Associations between event signal names and signal numbers set in ELSRn.ELS[7:0] bits (2 of 2)

ELS[7:0] bits value	Interrupt request source	Event signal name	Event description
0x37	AGTW1	AGTW1_AGTI	AGTW1 interrupt
0x38		AGTW1_AGTCMAI	AGTW1 compare match A
0x39		AGTW1_AGTCMBI	AGTW1 compare match B
0x3F	I/O ports	IOPORT_GROUP3	I/O port 3 event
0x40		IOPORT_GROUP2	I/O port 2 event
0x41	ELC	ELC_SWEVT0	ELC software event 0
0x42		ELC_SWEVT1	ELC software event 1
0x8E	SCI2	SCI2_RXI	SCI2 receive data full
0x8F		SCI2_TXI	SCI2 transmit data empty
0x90		SCI2_TEI	SCI2 transmit end
0x91		SCI2_ERI	SCI2 errors (receive error, error signal detection)
0x94	SPI0	SPI0_SPRI	SPI0 receive data full
0x95		SPI0_SPTI	SPI0 transmit data empty
0x96		SPI0_SPII	SPI0 idle
0x97		SPI0_SPEI	SPI0 errors (mode fault, overrun, underrun, parity error)
0x98		SPI0_SPTEND	SPI0 transmit end
0x9C	GPT	GPT_UVWEDGE	GPT UVW edge event
0x9E	MLCD	MLCD_TEI	MLCD transmit end interrupt
0x9F		MLCD_TEMI	MLCD data register empty interrupt
0xA0	GPT320	GPT0_CCMPA	GPT0 compare match A
0xA1		GPT0_CCMPB	GPT0 compare match B
0xA2		GPT0_CMPC	GPT0 compare match C
0xA3		GPT0_CMPD	GPT0 compare match D
0xA4		GPT0_OVF	GPT0 overflow
0xA5		GPT0_UDF	GPT0 underflow
0xA6	GPT164	GPT4_CCMPA	GPT4 compare match A
0xA7		GPT4_CCMPB	GPT4 compare match B
0xA8		GPT4_CMPC	GPT4 compare match C
0xA9		GPT4_CMPD	GPT4 compare match D
0xAA		GPT4_OVF	GPT4 overflow
0xAB		GPT4_UDF	GPT4 underflow
Settings other than the above are prohibited.			

Note 1. Only pulse (edge detection) is supported.

Note 2. This event can occur even in Snooze mode.

Note 3. This event cannot be selected by the ELSRn (n = 0 to 3, 6, 7) registers.

21.2.3 ELSEGRn : Event Link Software Event Generation Register n (n = 0, 1)

Base address: ELC = 0x4004_1000

Offset address: 0x02 + 0x02 × n

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

Bit	Symbol	Function	R/W
0	SEG	Software Event Generation 0: Normal operation 1: Software event is generated	W
5:1	—	These bits are read as 0. The write value should be 0.	R/W
6	WE	SEG Bit Write Enable 0: Disable writes to the SEG bit 1: Enable writes to the SEG bit	R/W
7	WI	ELSEGRn Register Write Disable 0: Enable writes to the ELSEGRn register 1: Disable writes to the ELSEGRn register	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. It is not set to 1 even on writing 1 to it.

A software event generates an event for DTC.

WE bit (SEG Bit Write Enable)

The SEG bit can be written to only when the WE bit is 1.

[Setting condition]

- When 1 is written to this bit while WI bit is 0.

[Clearing condition]

- When 0 is written to this bit while WI bit is 0.

WI bit (ELSEGRn Register Write Disable)

The ELSEGRn register can be written to only when the write value to the WI bit is 0.

This bit is read as 1.

21.2.4 ELOPA : Event Link Option Setting Register A

Base address: ELC = 0x4004_1000

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	TMR1MD[1:0]	TMR0MD[1:0]	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TMR0MD[1:0]	TMR0 Operation Select Bit 0 0: Disable event 0 1: Start count 1 0: Restart counter 1 1: Event counter	R/W
3:2	TMR1MD[1:0]	TMR1 Operation Select Bit 0 0: Disable event 0 1: Start count 1 0: Restart counter 1 1: Event counter	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

21.3 Operation

21.3.1 Relation between Interrupt Handling and Event Linking

The peripheral modules included in this MCU has interrupt request status flags and enabling bits that enable or disable the associated interrupts. An interrupt request generated by a peripheral module sets the associated interrupt request status flag to 1. At this time, if the associated interrupt request has been enabled, the interrupt is requested to CPU.

On the other hand, the ELC uses the interrupt requests generated by various peripheral modules as event signals to interconnect (link) peripheral modules, allowing direct link between peripheral modules without CPU intervention. Output of event signals is possible regardless of the settings of the associated interrupt request enable bit. [Figure 21.2](#) shows the Associations between interrupt handling and ELC.

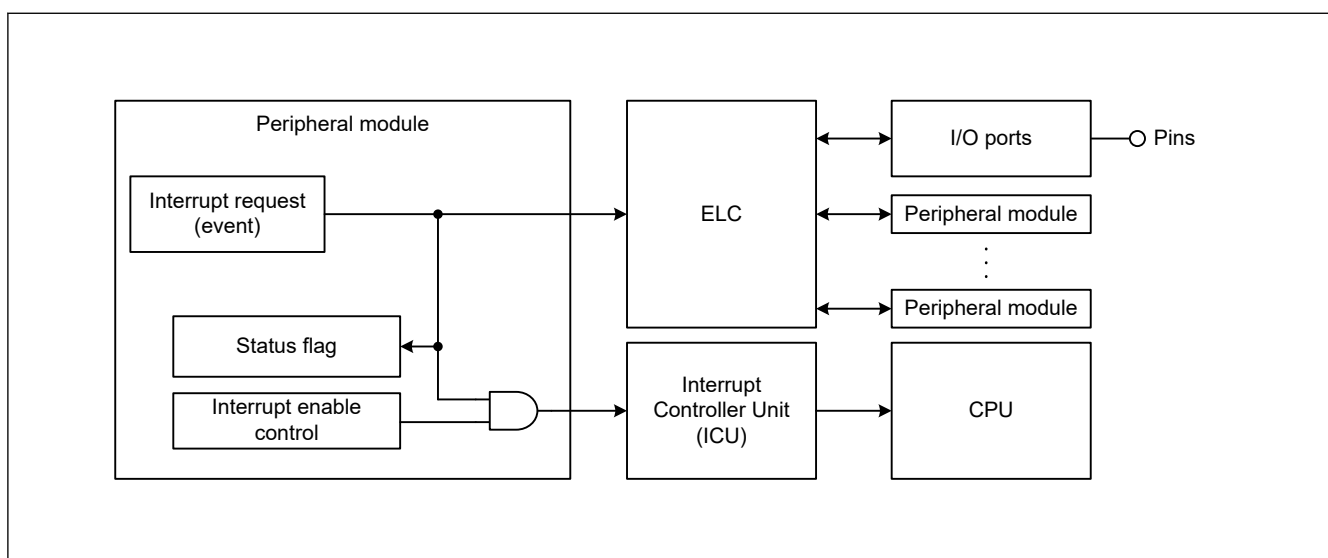


Figure 21.2 Associations between interrupt handling and ELC

21.3.2 Linking Events

When an event occurs and that event is already set as a trigger in the ELSRn register, the associated peripheral module is activated. The number of events to be linked to a peripheral module is only one at a time. When the ELC activates the peripheral module, the operation of the peripheral module must be set up in advance. [Table 21.4](#) lists the operations of peripheral modules when an event output from the ELC is received.

Table 21.4 Peripheral module operations when an event is input

Peripheral module	Peripheral module operations
GPT	• Start counting • Stop counting • Clear the counts • Counting upward • Counting downward • Input capture
TMR	Operations by the setting of the ELOPA register are as follows: • Start count by input of an event signal • Restart count by input of an event signal • Count the number of an input event
ADC14	Start A/D conversion when the ADCSR.ADST bit is set to 1
I/O ports	• Change pin output based on the EORR register (reset) or EOSR register (set) • Latch pin state to the EIDR register • The following ports can be used for the ELC: PORT 2 PORT 3
ICU	Make an interrupt request to CPU, start DMAC data transfer, or start DTC data transfer.

21.3.3 Example of Operation Setting Procedure for Linking Events

To link events:

1. Set the operation of the peripheral module to which an event is to be linked.
2. Set the number of the event signal to be linked to the ELSRn register for the peripheral module to which an event is to be linked.
3. Set the ELOPA register when TMR is the peripheral module to which an event is to be linked.
4. Set the ELCR.ELCON bit to 1. This step enables event link operation for all peripheral modules with the event link specified.
5. Set the operation of the peripheral module from which an event is output, and activate the peripheral module. The event link destination peripheral module starts the operation set in advance triggered by the event output from the module.

To stop event link operation by peripheral module, set 0x00 to the ELSRn.ELS[7:0] bits associated with the modules.

To stop event link operation for all peripheral modules, set the ELCR.ELCON bit to 0.

Note: When you use the event output from the RTC, set the ELC after setting the RTC (for example, for initialization and time settings). Unintended events can be output if the RTC settings are made after the ELC settings.

Note: When you use the event output from the LVD, set the ELC after setting the LVD. To disable the LVD, do so after setting 00h to the associated ELSRn register.

21.4 Usage Notes

21.4.1 Linking DMAC or DTC Transfer End Signals as Events

When linking the DMAC or DTC transfer end signals as events, do not set the same peripheral module as the DMAC or DTC transfer destination and event link destination. If set, the peripheral module might be started before the DMAC or DTC transfer to the peripheral module is complete.

21.4.2 Setting Clocks

To link events, you must enable the operations of the ELC and the target modules in addition to the ELC setting. The modules cannot operate if the target modules are in the module-stop state or low power consumption state in which the module is stopped (Software Standby mode or Deep Software Standby mode).

21.4.3 Notes on event settings

Do not set the same peripheral module for the event source and the event destination from ELC.

21.4.4 Settings for the Module-Stop Function

The Module Stop Control Register D (MSTPCRD) can enable or disable VREF operation. The VREF module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

22. I/O Ports

22.1 Overview

The pins of the I/O ports operate as general-purpose I/O port pins, I/O pins for peripheral modules, interrupt input pins, analog I/O pins, or for the port-group function of the ELC. All pins operate as input pins immediately after a reset and pin functions are switched by register settings. You can specify the associated I/O port or peripheral module for each pin in the registers. The same function can be specified for all pins of a port at the same time*1. Figure 22.1 is a connection diagram for the I/O port registers.

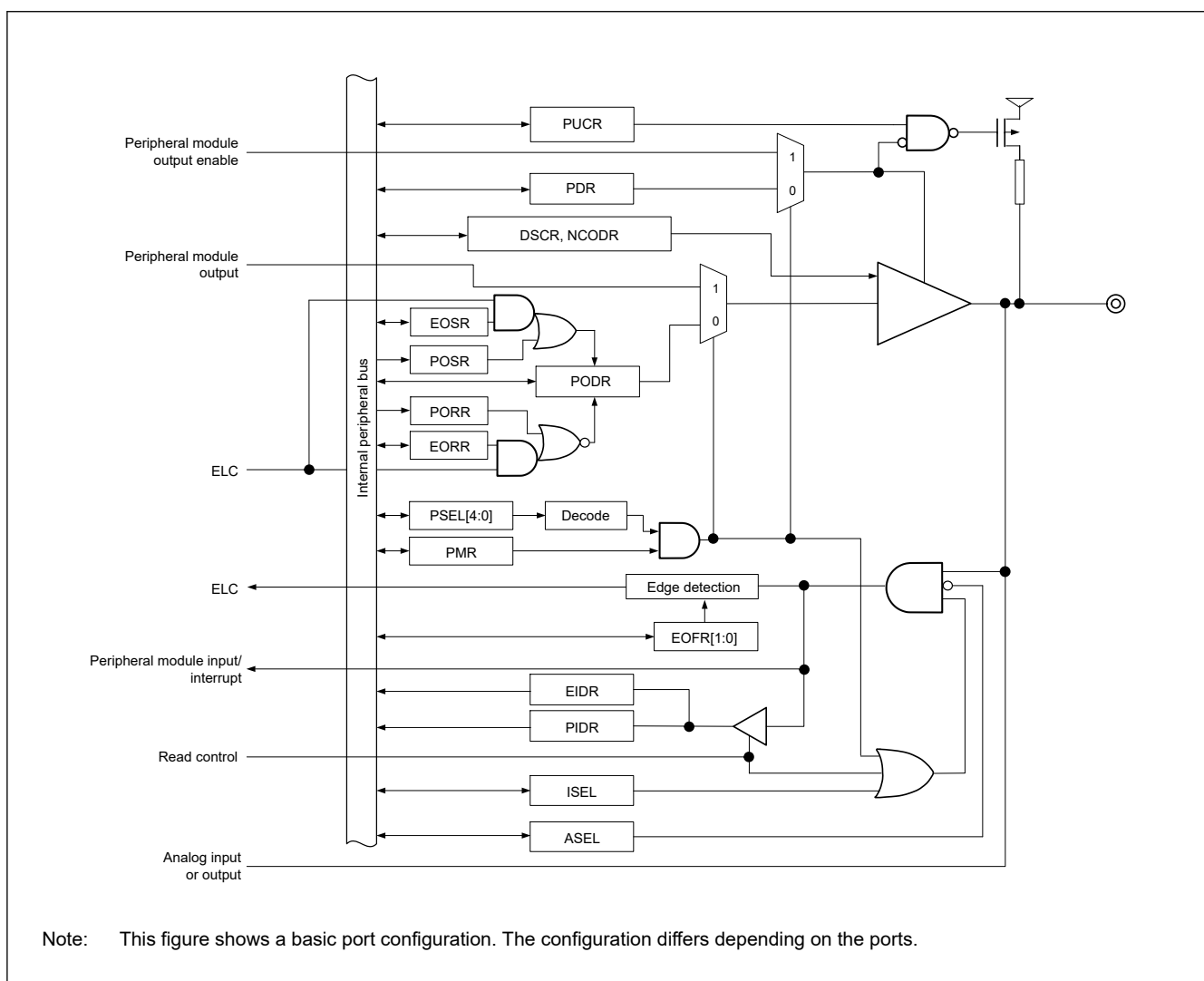


Figure 22.1 I/O port registers connection diagram

Note 1. If you want to set a function for each port, see [section 22.3.4. Peripheral Select Settings for Each Product](#).

The configuration of the I/O ports depends on the package. [Table 22.1](#) shows I/O port specifications, and [Table 22.2](#) lists the port functions. [Table 22.3](#) lists pins of which initial values are output and the output values.

Table 22.1 I/O port specifications (1 of 2)

Port symbol	Package		Package		Package		Package	
	100-pin LFQFP/BGA	Number of pins	72-pin WLPGA	Number of pins	64-pin LFQFP	Number of pins	56-pin QFN	Number of pins
PORT0	P000 to P007, P010 to P015	14	P000 to P007, P012 to P015	12	P000 to P007, P014 to P015	10	P000 to P006, P012 to P013	9

Table 22.1 I/O port specifications (2 of 2)

Port symbol	Package		Package		Package		Package	
	100-pin LQFP/BGA	Number of pins	72-pin WLPGA	Number of pins	64-pin LQFP	Number of pins	56-pin QFN	Number of pins
PORT1	P100 to P113	14	P100 to P113	14	P100 to P113	14	P100 to P113	14
PORT2	P200 to P205, P207 to P210	10	P200 to P205, P207 to P210	10	P200 to P204, P207 to P210	9	P200 to P204, P207 to P209	8
PORT3	P300 to P302, P314 to P315	5	—	0	—	0	—	0
PORT4	P409 to P413	5	P410 to P413	4	P411 to P413	3	P411 to P413	3
PORT5	P500 to P501, P508 to P511	6	—	0	—	0	—	0
PORT6	P600 to P604	5	—	0	—	0	—	0
PORT7	P700 to P704	5	P700 to P701	2	P700 to P701	2	—	0
PORT8	P806 to P815	10	P807 to P808	2	—	0	—	0
	Total number of ports	74	Total number of ports	44	Total number of ports	38	Total number of ports	34

Table 22.2 I/O port functions

Port symbol	Port name	Input pull-up function	Channel N open drain output function	Drive capability switching function
PORT0	P000 to P007, P010 to P015	✓	✓	High/standard
PORT1	P100 to P113	✓	✓	High/standard
PORT2	P200 to P201	✓	✓	Standard
	P202 to P205, P207	✓	✓	High/standard
	P208 to P210	×	✓	Standard
PORT3	P300 to P302, P314 to P315	✓	✓	High/standard
PORT4	P409 to P410	✓	✓	Standard
	P411 to P413	✓	✓	High/standard
PORT5	P500 to P501, P508 to P511	✓	✓	High/standard
PORT6	P600 to P604	✓	✓	High/standard
PORT7	P700 to P704	✓	✓	High/standard
PORT8	P806 to P815	✓	✓	High/standard

Table 22.3 Pins of which initial values are output

Pin name	Initial value
P208	Low-level output
P209	Low-level output
P210	High-level output
P409	Low-level output
P410	High-level output

22.2 Register Descriptions

22.2.1 PCNTR1/PODR/PDR : Port Control Register 1

Base address: $\text{PORTm} = 0x4004\_0000 + 0x0020 \times m$ ($m = 0$ to 8)

Offset address: $0x000$ (PCNTR1/PODR)
 $0x002$ (PDR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	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
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PDR1 5	PDR1 4	PDR1 3	PDR1 2	PDR11	PDR1 0	PDR0 9	PDR0 8	PDR0 7	PDR0 6	PDR0 5	PDR0 4	PDR0 3	PDR0 2	PDR0 1	PDR0 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PDR15 to PDR00	Pmn Direction Control 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W
31:16	PODR15 to PODR00	Pmn Output Data Store 0: Low output 1: High output	R/W

Note: $m = 0$ to 8 ; $n = 00$ to 15

Note: The following registers have different values after reset:

PORT2.PCNTR1: $0x0400\ 0700$
 PORT4.PCNTR1: $0x0400\ 0600$

The Port Control Register 1 (PCNTR1/PODR/PDR) is a 32-bit or 16-bit read/write register for controlling the direction and output data of ports.

PCNTR1 register specifies the port direction and the output data, and is accessed in 32-bit units. PODR register (bits [31:16] in PCNTR1) and PDR register (bits [15:0] in PCNTR1) respectively, are accessed in 16-bit units.

PDRn bits (Pmn Direction Control)

The PDR 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 PmnPFS.PDR bit. The I/O direction can be specified in 1-bit units. The bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

PODRn bits (Pmn Output Data Store)

PODR is a bit that holds data to be output from the pins used for general I/O.

22.2.2 PCNTR2/EIDR/PIDR : Port Control Register 2

Base address: $\text{PORTm} = 0x4004\_0000 + 0x0020 \times m$ ($m = 0$ to 8)

Offset address: $0x004$ (PCNTR2/EIDR)
 $0x044 + 0x020 \times m$ ($m = 0, 1$) (EIDR)
 $0x006 + 0x020 \times m$ ($m = 0$ to 8) (PIDR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EIDR1 5	EIDR1 4	EIDR1 3	EIDR1 2	EIDR1 1	EIDR1 0	EIDR0 9	EIDR0 8	EIDR0 7	EIDR0 6	EIDR0 5	EIDR0 4	EIDR0 3	EIDR0 2	EIDR0 1	EIDR0 0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PIDR1 5	PIDR1 4	PIDR1 3	PIDR1 2	PIDR1 1	PIDR1 0	PIDR0 9	PIDR0 8	PIDR0 7	PIDR0 6	PIDR0 5	PIDR0 4	PIDR0 3	PIDR0 2	PIDR0 1	PIDR0 0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
15:0	PIDR15 to PIDR00	Pmn Input Data 0: Low input 1: High input	R
31:16	EIDR15 to EIDR00	Pmn Event Input Data* ¹ When the ELC_PORTx occurs: 0: Low input 1: High input	R

Note: $m = 0$ to 8 ; $n = 00$ to 15 ; $x = 2, 3$

Note 1. Supported for PORT2 and PORT3.

The Port Control Register 2 (PCNTR2/EIDR/PIDR) allows read access to the port input data and the port event input data by 32-bit and 16-bit access.

PCNTR2 specifies the port input data and the port event input data, and is accessed in 32-bit units.

EIDR register (bits [31:16] in PCNTR2) and PIDR register (bits [15:0] in PCNTR2) respectively, are accessed in 16-bit units.

The PIDRn bit reflects the pin status of each port when the port is set for input. If the port is not set to input, the read value is undefined. The bits associated with nonexistent pins are reserved. A reserved bit is read as undefined. These cannot be modified. The NMI pin state is read from P200.

The EIDRn bits latch the pin state when an ELC_PORTx signal occurs. The pin states can be input into EIDRn when $\text{PmnPFS.PMR}^{*1} = 0$ and $\text{PmnPFS.PDR} = 0$ only. When $\text{PmnPFS.PMR}^{*1} = 1$ or $\text{PmnPFS.PDR} = 1$, the read data is don't care.

The bits associated with non-existent pins are reserved. A reserved bit is read as undefined. The NMI pin state is read from P200.

Note 1. See [section 22.2.6. PmnPFS : Port mn Pin Function Select Register](#) ($m = 0$ to 8 , $n = 00$ to 15).

22.2.3 PCNTR3/PORR/POSR : Port Control Register 3

Base address: $\text{PORTm} = 0x4004\_0000 + 0x0020 \times m$ ($m = 0$ to 8)

Offset address: $0x008$ (PCNTR3/PORR)
 $0x00A$ (POSR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PORR 15	PORR 14	PORR 13	PORR 12	PORR 11	PORR 10	PORR 09	PORR 08	PORR 07	PORR 06	PORR 05	PORR 04	PORR 03	PORR 02	PORR 01	PORR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	POSR 15	POSR 14	POSR 13	POSR 12	POSR 11	POSR 10	POSR 09	POSR 08	POSR 07	POSR 06	POSR 05	POSR 04	POSR 03	POSR 02	POSR 01	POSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	POSR15 to POSR00	Pmn Output High Set 0: No affect on output 1: High output	W
31:16	PORR15 to PORR00	Pmn Output Low Set 0: No affect on output 1: Low output	W

Note: $m = 0$ to 8 ; $n = 00$ to 15

Note: When EORRn and EOSRn are set, writes to PODRn, PORRn, and POSRn are prohibited.

Note: Do not set PORRn and POSRn at the same time.

The PCNTR3 register contains bits that are write-only. The read values of these write-only bits are reset values regardless of the previous settings. Therefore, the result of the logical operation on the read value should not be directly written to this register.

Setting example: `PORTn -> PCNTR3 = 'setting value'; // (n = 0 to 8)`

The Port Control Register 3 (PCNTR3/PORR/POSR) is a 32-bit or 16-bit write register for controlling the high output or low output of port output data.

PCNTR3 register controls the high output or low output of port output data that is set in 32-bit units.

PORR register (bits [31:16] in PCNTR3) and POSR register (bits [15:0] in PCNTR3) respectively, are accessed in 16-bit units.

POSRn bits (Pmn Output High Set)

POSR changes PODRn when set by a software write. For example, for P100, when `PORT1.POSR00 = 1`, `PORT1.PODR00` outputs 1. The bits associated with non-existent pins are reserved. The write value must always be 0.

PORRn bits (Pmn Output Low Set)

PORR changes PODRn when reset by a software write. For example, for P100, when `PORT1.PORR00 = 1`, `PORT1.PODR00` outputs 0. The bits associated with non-existent pins are reserved. The write value must always be 0.

22.2.4 PCNTR4/EORR/EOSR : Port Control Register 4

Base address: $\text{PORTm} = 0x4004\_0000 + 0x0020 \times m$ ($m = 2, 3$)

Offset address: $0x00C$ (PCNTR4/EORR)
 $0x00E$ (EOSR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EORR 15	EORR 14	EORR 13	EORR 12	EORR 11	EORR 10	EORR 09	EORR 08	EORR 07	EORR 06	EORR 05	EORR 04	EORR 03	EORR 02	EORR 01	EORR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EOSR 15	EOSR 14	EOSR 13	EOSR 12	EOSR 11	EOSR 10	EOSR 09	EOSR 08	EOSR 07	EOSR 06	EOSR 05	EOSR 04	EOSR 03	EOSR 02	EOSR 01	EOSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	EORR15 to EOSR00	Pmn Event High Output Set When the ELC_PORTx occurs: 0: No affect on output 1: High output	R/W
31:16	EORR15 to EORR00	Pmn Event Low Output Set When the ELC_PORTx occurs: 0: No affect on output 1: Low output	R/W

Note: $m = 2, 3$; $n = 00$ to 15 ; $x = 2, 3$

Note: When EORR and EOSR are set, writes to PODRn, PORRn, and POSRn are prohibited.

Note: Do not set EORR and EOSR at the same time.

The Port Control Register 4 is a 32-bit or 16-bit read/write register for controlling the high output or low output of port output data caused by event input from the ELC. PCNTR4 register controls the high output or low output of the port output data by event input from the ELC, which is set by 32-bit units.

EORR register (bits [31:16] in PCNTR4) and EOSR register (bits [15:0] in PCNTR4) are accessed in 16-bit units.

EOSRn bits (Pmn Event High Output Set)

EOSR register changes PODRn when set because an ELC_PORTx signal occurs. For example, for P100, if PORT1.EOSR00 is set to 1 when ELC_PORTx occurs, PORT1.PODR00 outputs 1. The bits associated with non-existent pins are reserved. The write value must always be 0. P200 is input only. For this reason, PORT2.PCNTR4.b0 is reserved.

EORRn bits (Pmn Event Low Output Set)

EORR changes PODRn when reset because an ELC_PORTx signal occurs. For example, for P100, if PORT1.EORR00 is set to 1 when ELC_PORTx occurs, PORT1.PODR00 outputs 0. The bits associated with non-existent pins are reserved. The write value must always be 0. P200 is input only. For this reason, PORT2.PCNTR4.b16 is reserved.

22.2.5 PWPR : Write-Protect Register

Base address: $\text{PFS} = 0x4004\_0800$

Offset address: $0x503$

Bit position:	7	6	5	4	3	2	1	0
Bit field:	B0WI	PFSW E	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6	PFSWE	PmnPFS Register Write Enable 0: Writing to the PmnPFS register is disabled 1: Writing to the PmnPFS register is enabled	R/W
7	B0WI	PFSWE Bit Write Disable 0: Writing to the PFSWE bit is enabled 1: Writing to the PFSWE bit is disabled	R/W

PFSWE bit (PmnPFS Register Write Enable)

Writing to the PmnPFS register is enabled only when the PFSWE bit is set to 1. You must first write 0 to the B0WI bit before setting the PFSWE bit.

B0WI bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

22.2.6 PmnPFS : Port mn Pin Function Select Register (m = 0 to 8, n = 00 to 15)

Base address: PFS = 0x4004_0800

Offset address: 0x000 + 0x040 × m + 0x004 × n (m = 0 to 8, n = 00 to 15)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	PSEL[4:0]					—	—	—	—	—	—	—	PMR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 ²
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ASEL	ISEL	EOFR[1:0]	—	DSCR	—	—	—	NCOD R	—	PUCR	—	PDR	PIDR	PODR	
Value after reset:	0	0	0	0	0	1 ²	0	0	0	0	0	0 ²	0	0 ²	x	0 ²

Bit	Symbol	Function	R/W
0	PODR	Port Output Data ^{*3} 0: Low output 1: High output	R/W
1	PIDR	Port Input Data ^{*3} 0: Low input 1: High input	R
2	PDR	Port Direction ^{*3} 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	PUCR	Pull-Up Control 0: Disables input pull-up 1: Enables input pull-up ^{*4}	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	NCODR	N-Channel Open-Drain Control 0: CMOS output 1: NMOS open-drain output	R/W
9:7	—	These bits are read as 0. The write value should be 0.	R/W
10	DSCR	Port Drive Capability ^{*5} 0: Standard drive 1: High drive	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
13:12	EOFR[1:0]	Edge detection*1 0 0: Disable edge detection 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges	R/W
14	ISEL	IRQ Input Enable 0: Do not use as an IRQn input pin 1: Use as an IRQn input pin	R/W
15	ASEL	Analog Input Enable 0: Do not use as an analog pin 1: Use as an analog pin	R/W
16	PMR	Port Mode Control 0: Use the pin as a general I/O port 1: Use the pin as a peripheral module	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
28:24	PSEL[4:0]	Peripheral Function Select Sets a peripheral function. For details on selecting a peripheral function for each pin, see section 22.3.4. Peripheral Select Settings for Each Product .	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Supported for PORT2 and PORT3.

Note 2. For the Port mn Pin Function Select Register (PmnPFS), the value after reset other than the following ports is 0x00000400.

P200: 0x00000000

P201: 0x00000010

P207 and P411: 0x00010410

P208, P209, and P409: 0x00000004

P210 and P410: 0x00000005

Note 3. When the PORTn.PCINTR1 register or PORTn.PCINTR2 register is used, the batch setting instead of the setting per port is available.

Note 4. To enable this function, set PDR = 0 and PMR = 0.

Note 5. When P200, P201, P208, P209, P210, P409, or P410 is used as a port function or alternative function of the port, setting this bit to 1 is prohibited.

22.3 Operation

22.3.1 General I/O Ports

All pins other than the pins listed in [Table 22.3](#) 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 pin with the Pin Function Select Registers. For details on these registers, see [section 22.2. Register Descriptions](#).

Each port has the following registers:

- Pmn Direction Control Bit (PDR), which selects input or output direction.
- Pmn Output Data Store Bit (PODRn), which sets low or high output value.
- Pmn Input Data Bit (PIDRn), which indicates the pin input states (low or high).
- Pmn Event Input Data Bit (EIDRn), which indicates the pin states when an ELC_PORTx signal occurs.
- Pmn Output High Set Bit (POSRn), which outputs 1 from the associated pin when 1 is written to.
- Pmn Output Low Set Bit (PORRn), which outputs 0 from the associated pin when 1 is written to.
- Pmn Event Output High Set Bit (EOSRn), which outputs 1 from the associated pin when 1 is written to and the event from an ELC_PORTx is input.
- Pmn Event Output Low Set Bit (EORRn), which outputs 0 from the associated pin when 1 is written to and the event from an ELC_PORTx is input.

22.3.2 Port Function Select

The following port functions are available for configuring each pin:

- I/O configuration: Complementary or open-drain output, pull-up control, and drive strength
- General I/O port: Port direction, output data setting, and reading input data
- Alternate function: Configured function mapping to the pin

Each pin is associated with a Pin Function Select Register (PmnPFS), which includes the associated PODR, PIDR, and PDR bits. In addition, the PmnPFS Register includes:

- PUCR: Pull-up resistor control bit that turns the input pull-up MOS on or off
- NCODR: N-channel open-drain control bit that selects the output type for each pin
- DSCR: Drive capability control bit that selects the drive capability
- EOFR[0]: Event on rising bit used to detect rising edges on the port input
- EOFR[1]: Event on falling bit used to detect falling edges on the port input
- ISEL: IRQ input enable bit to specify an IRQn 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 bit to select the associated peripheral function

These configurations can be made by a single-register access to the Pin Function Select Register. For details, see [section 22.2.6. PmnPFS : Port mn Pin Function Select Register \(m = 0 to 8, n = 00 to 15\)](#).

22.3.3 Port Group Function for ELC

In the MCU, ports 1 and 2 are assigned for the port group function.

22.3.3.1 Behavior When ELC_PORTx is Input from ELC

The MCU supports two functions when an ELC_PORT2 or 3 signal comes from the ELC.

(1) Input to EIDR register

For the general port input function ($PDR = 0$ and $PMR^{*1} = 0$ in the PmnPFS Register), when an ELC_PORTx signal comes from the ELC, the input enable of the I/O cell is asserted, and then data from the external pins are read into the EIDR bits.

For the general port output function ($PDR = 1$ and $PMR^{*1} = 1$ in the PmnPFS Register), 0 is input to the EIDR bits from the external pins. [Figure 22.2](#) shows event ports input data.

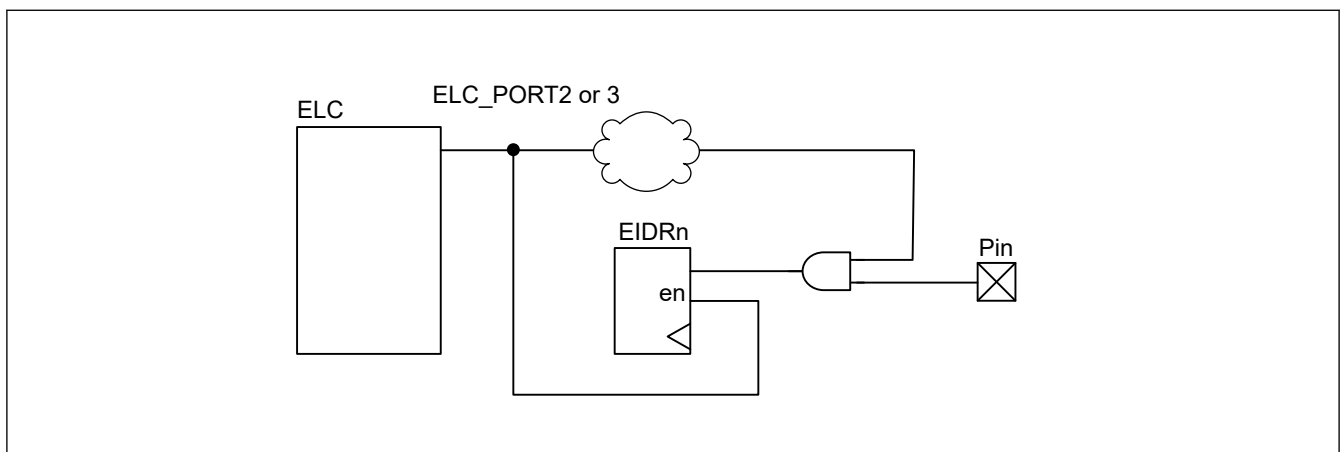


Figure 22.2 Event ports input data

Note 1. See [section 22.2.6. PmnPFS : Port mn Pin Function Select Register \(m = 0 to 8, n = 00 to 15\)](#).

(2) Output from PODRn bit by EOSRn/EORRn bit

When an ELC_PORT2 or 3 signal occurs, the data is output from PODRn bit to the external pin based on the EOSRn/EORRn bit settings as follows:

- If EOSRn is set to 1, when an ELC_PORT2 or 3 signal occurs, the PODRn bit outputs 1 to the external pin. Otherwise, when EOSRn = 0, the PODRn bit value is kept.
- If EORRn is set to 1, when an ELC_PORT2 or 3 signal occurs, the PODRn bit outputs 0 to the external pin. Otherwise, when EORRn = 0, the PODRn bit value is kept.

Figure 22.3 shows event ports output data.

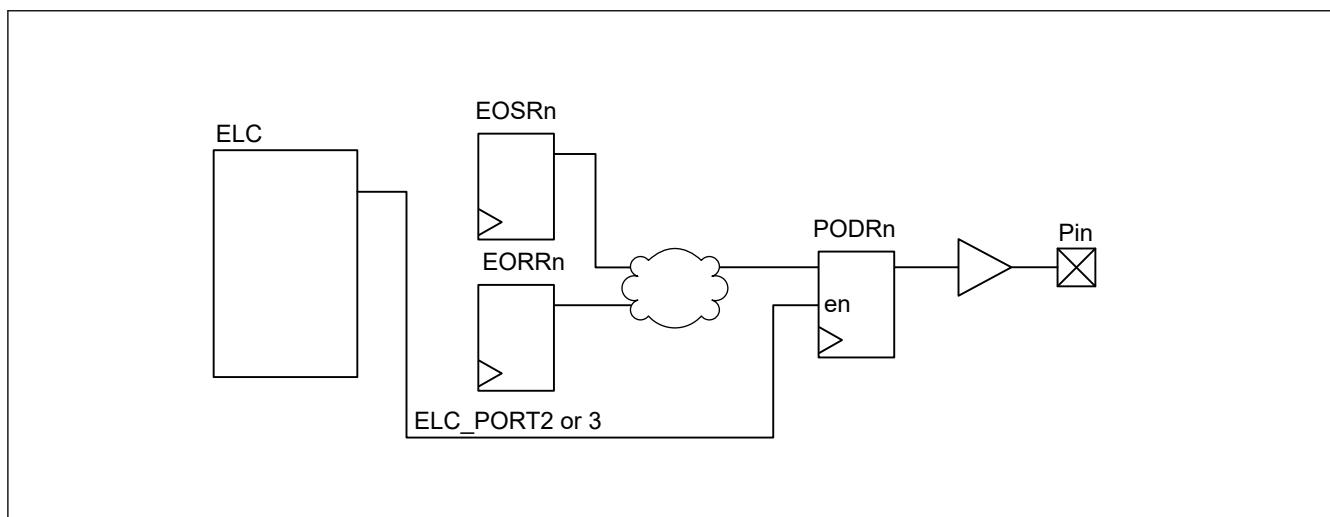


Figure 22.3 Event ports output data

22.3.3.2 ELC Event Pulse Input from Pins

To output the event pulse from the external pins to the ELC, set the PmnPFS.EOFR[1:0] bits. For details, see [section 22.2.6. PmnPFS : Port mn Pin Function Select Register \(m = 0 to 8, n = 00 to 15\)](#).

The EOFR[1:0] bits enable the event input from pins and set the edge detection direction. Figure 22.4 shows event pulse input.

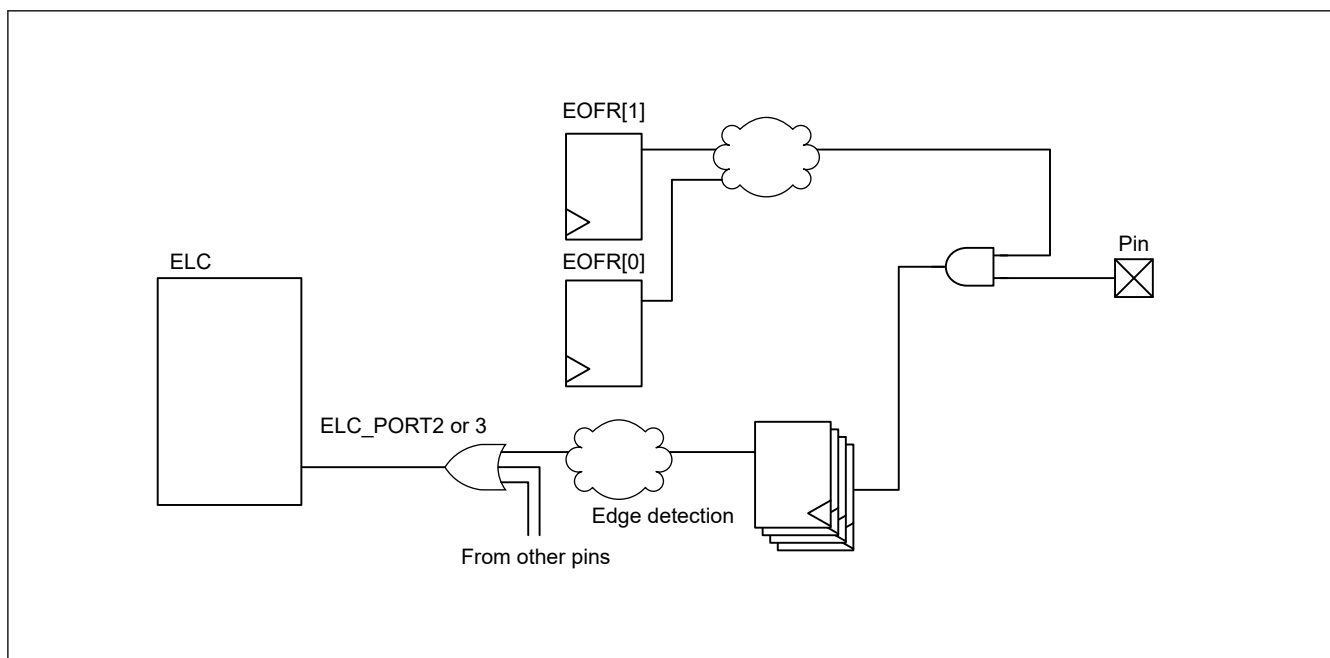


Figure 22.4 Event pulse input

22.3.4 Peripheral Select Settings for Each Product

Table 22.4 to Table 22.18 provide the detail of the pin function select configuration by the PmnPFS register. Several pin names have added _A, _B, and _C suffixes. When assigning the IIC and SPI functions, select the functional pins with the same suffix^{*1}. The other pins can be selected regardless of the suffix. Assigning the same function to two or more pins simultaneously is prohibited.

Note 1. Some signals have the same name and _A, _B, _C and _D suffixes. These suffixes show the group of timing adjusting, and the signals in the different group cannot be used at the same time. For exception, when using "RSPCKA_C, MOSIA_C" added by the SPI, select the pair of RSPCKA_B and RSPCKA_C and the pair of MOSIA_B and MOSIA_C. When using "SSLB0_D" added by SPI, select the pair of SSLB0_D and SSLB0_B.

Table 22.4 Register settings for input/output pin function (Port 0) (1)

PSEL[4:0]	Function	Pin							
Bit settings		P000	P001	P002	P003	P004	P005	P006	P007
00000b (initial value)									
ASEL bit		AN000	AN001	AN002	AN003	AN004	AN005	AN006	AN007
ISEL bit		x	x	x	x	x	x	x	x
NCODR bit		✓	✓	✓	✓	✓	✓	✓	✓
PUCR bit		✓	✓	✓	✓	✓	✓	✓	✓
DSCR bit		✓	✓	✓	✓	✓	✓	✓	✓
100-pin LFQFP/BGA product		✓	✓	✓	✓	✓	✓	✓	✓
72-pin WLPGA product		✓	✓	✓	✓	✓	✓	✓	✓
64-pin LFQFP product		✓	✓	✓	✓	✓	✓	✓	✓
56-pin QFN product		✓	✓	✓	✓	✓	✓	✓	x

Note: ✓: Available
x: Not available

Table 22.5 Register settings for input/output pin function (Port 0) (2)

PSEL[4:0]	Function	Pin					
Bit settings		P010	P011	P012	P013	P014	P015
00000b (initial value)							
00011b	GPT	x	x	x	x	GTIOC3B_A	GTIOC3A_A
00101b	SCI3	x	x	TXD3_B	SCK3_B	x	x
00110b	SPI0/1	MOSIA_B	RSPCKA_B	x	x	SSLA0_B	SSLA1_B
00111b	IIC0/1	x	x	SDA0	SCL0	x	x
11001b	CLKOUT	x	x	x	x	x	CLKOUT
ASEL bit		x	x	x	x	x	x
ISEL bit		x	x	x	x	IRQ2_A_DS	IRQ5_A
NCODR bit		✓	✓	✓	✓	✓	✓
PUCR bit		✓	✓	✓	✓	✓	✓
DSCR bit		✓	✓	✓	✓	✓	✓
100-pin LFQFP/BGA product		✓	✓	✓	✓	✓	✓
72-pin WLBGA product		x	x	✓	✓	✓	✓
64-pin LFQFP product		x	x	x	x	✓	✓
56-pin QFN product		x	x	✓	✓	x	x

Note: ✓: Available
x: Not available

Table 22.6 Register settings for input/output pin function (Port 1) (1)

PSEL[4:0]	Function	Pin							
Bit settings		P100	P101	P102	P103	P104	P105	P106	P107
00000b (initial value)									
00001b	AGT0/1	x	x	AGTIO0_A	AGTEE1_A	AGTIO1_A	AGTO1_A	AGTOA1_A	AGTOB1_A
00010b	GPT-COM	x	x	x	x	x	x	GTETRGB_A	GTETRGA_A
00011b	GPT	GTIOC0B_C	GTIOC0A_C	GTIOC5B_A	GTIOC5A_A	GTIOC4B_A	GTIOC4A_A	GTIOC1B_A	GTIOC1A_A
00100b	SCI0/2	SCK2_A	RXD2_A	TXD2_A	CTS2_A	SCK0_A	RXD0_A	TXD0_A	CTS0_A
00101b	SCI1	SCK1_A	RXD1_A /IRRXD1_A	TXD1_A /IRTXD1_A	CTS1_A	x	x	x	x
00110b	SPI0/1	SSLA3_A	SSLA2_A	SSLA1_A	SSLA0_A	MOSIA_A	MISOA_A	SSLB0_A	RSPCKA_A
01010b	CAC/ ADC(Digital)	CACREF_B	ADTRG0_B	x	x	x	x	x	x
01110b	MLCD	MLCD_SI7	MLCD_SI6	MLCD_SI5	MLCD_SI4	MLCD_SI3	MLCD_SI2	MLCD_SI1	MLCD_SI0
10000b	KINT	KRM00_A	KRM01_A	KRM02_A	KRM03_A	KRM04_A	KRM05_A	KRM06_A	KRM07_A
ASEL bit		x	x	x	x	x	x	x	x
ISEL bit		x	x	x	x	IRQ4_B	IRQ8_A	IRQ3_B	IRQ7_A
NCODR bit		✓	✓	✓	✓	✓	✓	✓	✓
PUCR bit		✓	✓	✓	✓	✓	✓	✓	✓
DSCR bit		✓	✓	✓	✓	✓	✓	✓	✓
100-pin LFQFP/BGA product		✓	✓	✓	✓	✓	✓	✓	✓
72-pin WLBGA product		✓	✓	✓	✓	✓	✓	✓	✓
64-pin LFQFP product		✓	✓	✓	✓	✓	✓	✓	✓
56-pin QFN product		✓	✓	✓	✓	✓	✓	✓	✓

Note: ✓: Available
x: Not available

Table 22.7 Register settings for input/output pin function (Port 1) (2) (1 of 2)

PSEL[4:0]	Function	Pin					
Bit settings		P108	P109	P110	P111	P112	P113
00000b (initial value)							
00001b	AGT0/1	AGTIO0_B	AGTOB0_B	AGTOA0_B	AGTO0_B	AGTEE0_B	AGTEE0_A
00010b	GPT-COM	GTOVLO_A	GTOVUP_A	GTOULO_A	GTOUUP_A	GTOWLO_A	GTOWUP_A
00011b	GPT	×	×	GTIOC2B_A	GTIOC2A_A	×	×
00100b	SCI4/9	SCK4_A	CTS9_A	SCK9_A	CTS4_A	RXD4_A	TXD4_A
00101b	SCI4/5	TXD5_A	CTS5_A	SCK5_A	RXD5_A	×	×
00110b	SPI0/1	RSPCKB_A	MISOB_A	MOSIB_A	SSLB1_A	SSLB3_A	SSLB2_A
01000b	TMR	×	×	×	×	×	TMC11
01110b	MLCD	MLCD_ENBG	MLCD_ENBS	MLCD_DEN	MLCD_SCLK	MLCD_XRST	MLCD_VCOM
10001b	QSPI	QSSL_B	QSPCLK_B	QIO3_B	QIO2_B	QIO1_B	QIO0_B
11000b	AGTW0/1	AGTWIO0_A	AGTWOB0_A	AGTWOA0_A	AGTWO0_A	AGTWEE0_A	×
ASEL bit		×	×	×	×	×	×
ISEL bit		×	×	×	×	IRQ8_B	IRQ3_A_DS
NCODR bit		✓	✓	✓	✓	✓	✓

Table 22.7 Register settings for input/output pin function (Port 1) (2) (2 of 2)

PSEL[4:0]	Function	Pin					
Bit settings		P108	P109	P110	P111	P112	P113
PUCR bit		✓	✓	✓	✓	✓	✓
DSCR bit		✓	✓	✓	✓	✓	✓
100-pin LFQFP/BGA product		✓	✓	✓	✓	✓	✓
72-pin WLBGA product		✓	✓	✓	✓	✓	✓
64-pin LFQFP product		✓	✓	✓	✓	✓	✓
56-pin QFN product		✓	✓	✓	✓	✓	✓

Note: ✓: Available
 ×: Not available

Table 22.8 Register settings for input/output pin function (Port 2) (1)

PSEL[4:0]	Function	Pin						
Bit settings		P200	P201	P202	P203	P204	P205	P207
00000b (initial value)								
00001b	AGT0/1	×	×	AGTOB0_A	AGTOA0_A	AGTO0_A	×	×
00010b	GPT-COM	×	×	GTIW_A	GTIV_A	GTIU_A	×	×
00011b	GPT	×	×	×	×	×	×	GTIOC0A_B
00100b	SCI4/9	×	×	TXD4_B	RXD4_B	SCK4_B	CTS4_B	RXD9_A
00101b	SCI3	×	×	×	×	×	×	CTS3_A
01000b	TMR	×	×	TMO0_A	TMRI0_A	TMCIO_A	×	×
01001b	RTC	×	×	RTCCOUT_A	×	×	×	×
01010b	CAC/ADC(Digital)	×	×	CACREF_A	×	ADTRG0_A	×	×
11000b	AGTW0/1	×	×	×	×	×	AGTWO0_B	AGTWO1_A
RTCCR.TCEN = 1 (RTCIC input enabled)		×	×	×	RTCIC1_A	RTCIC0_A	×	×
R128CTRL.CRTOE = 1 (CCCOU output enabled)		×	×	CCOUT_A	×	×	×	×
ASEL bit		×	×	×	×	×	×	×
ISEL bit		NMI	×	IRQ4_A	×	IRQ7_B	IRQ8_C	IRQ1_A_DS
NCODR bit		✓	✓	✓	✓	✓	✓	✓
PUCR bit		✓	✓	✓	✓	✓	✓	✓
DSCR bit		×	×	✓	✓	✓	✓	✓
100-pin LFQFP/BGA product		✓	✓	✓	✓	✓	✓	✓
72-pin WLBGA product		✓	✓	✓	✓	✓	✓	✓
64-pin LFQFP product		✓	✓	✓	✓	✓	×	✓
56-pin QFN product		✓	✓	✓	✓	✓	×	✓

Note: ✓: Available
 ×: Not available

Table 22.9 Register settings for input/output pin function (Port 2) (2) (1 of 2)

PSEL[4:0]	Function	Pin		
Bit settings		P208	P209	P210
00000b (initial value)				
11000b	AGTW0/1	AGTWIO1_A	AGTWOB1_A	AGTWOA1_A

Table 22.9 Register settings for input/output pin function (Port 2) (2) (2 of 2)

PSEL[4:0]	Function	Pin		
Bit settings		P208	P209	P210
TCR.TMOE = 1 (TMWO output permission)		TMWO	×	×
ASEL bit		✓	✓	✓
ISEL bit		✓	✓	✓
NCODR bit		✓	✓	✓
PUCR bit		✓	✓	✓
DSCR bit		×	×	×
100-pin LFQFP/BGA product		✓	✓	✓
72-pin WLBGA product		✓	✓	✓
64-pin LFQFP product		✓	✓	✓
56-pin QFN product		✓	✓	×

Note: ✓: Available
 ×: Not available

Table 22.10 Register settings for input/output pin function (Port 3) (1)

PSEL[4:0]	Function	Pin		
Bit settings		P300	P301	P302
00000b (initial value)				
00010b	GPT-COM	GTIW_B	GTIV_B	GTIU_B
00011b	GPT	×	GTIOC2B_B	GTIOC2A_B
00101b	SCI5	×	SCK5_B	CTS5_B
01000b	TMR	TMO0_B	TMRI0_B	TMCi0_B
01001b	RTC	×	RTCOUT_B	×
R128CTRL.CRTOE = 1 (CCCOUT output enabled)		×	CCCOUT_B	×
ASEL bit		×	×	×
ISEL bit		×	×	×
NCODR bit		✓	✓	✓
PUCR bit		✓	✓	✓
DSCR bit		✓	✓	✓
100-pin LFQFP/BGA product		✓	✓	✓
72-pin WLBGA product		×	×	×
64-pin LFQFP product		×	×	×
56-pin QFN product		×	×	×

Note: ✓: Available
 ×: Not available

Table 22.11 Register settings for input/output pin function (Port 3) (2) (1 of 2)

PSEL[4:0]	Function	Pin	
Bit settings		P314	P315
00000b (initial value)			
00011b	GPT	GTIOC4B_B	GTIOC4A_B
00101b	SCI5	RXD5_B	TXD5_B

Table 22.11 Register settings for input/output pin function (Port 3) (2) (2 of 2)

PSEL[4:0]	Function	Pin	
Bit settings		P314	P315
11000b	AGTW0/1	x	AGTWIO0_B
ASEL bit		x	x
ISEL bit		x	x
NCODR bit		✓	✓
PUCR bit		✓	✓
DSCR bit		✓	✓
100-pin LFQFP/BGA product		✓	✓
72-pin WLBGA product		x	x
64-pin LFQFP product		x	x
56-pin QFN product		x	x

Note: ✓: Available
x: Not available

Table 22.12 Register settings for input/output pin function (Port 4)

PSEL[4:0]	Function	Pin				
Bit settings		P409	P410	P411	P412	P413
00000b (initial value)						
00011b	GPT	x	x	GTIOC0B_B	GTIOC0B_A	GTIOC0A_A
00100b	SCI9	x	x	TXD9_A	x	x
00101b	SCI3	x	x	SCK3_A	RXD3_A	TXD3_A
11000b	AGTW0/1	x	x	AGTWEE1_A	x	x
11001b	CLKOUT	CLKOUT32_B	x	CLKOUT32_A	x	x
ASEL bit		x	x	x	x	x
ISEL bit		IRQ9_B	IRQ9_A	IRQ0_A_DS	x	x
NCODR bit		✓	✓	✓	✓	✓
PUCR bit		✓	✓	✓	✓	✓
DSCR bit		x	x	✓	✓	✓
100-pin LFQFP/BGA product		✓	✓	✓	✓	✓
72-pin WLBGA product		x	✓	✓	✓	✓
64-pin LFQFP product		x	x	✓	✓	✓
56-pin QFN product		x	x	✓	✓	✓

Note: ✓: Available
x: Not available

Table 22.13 Register settings for input/output pin function (Port 5) (1) (1 of 2)

PSEL[4:0]	Function	Pin	
Bit settings		P500	P501
00000b (initial value)			
00010b	GPT-COM	GTOWUP_B	x
00100b	SCI0	CTS0_B	x
ASEL bit		x	AN016
ISEL bit		x	x

Table 22.13 Register settings for input/output pin function (Port 5) (1) (2 of 2)

PSEL[4:0]	Function	Pin	
Bit settings		P500	P501
NCODR bit		✓	✓
PUCR bit		✓	✓
DSCR bit		✓	✓
100-pin LFQFP/BGA product		✓	✓
72-pin WLBGA product		x	x
64-pin LFQFP product		x	x
56-pin QFN product		x	x

Note: ✓: Available
x: Not available

Table 22.14 Register settings for input/output pin function (Port 5) (2)

PSEL[4:0]	Function	Pin			
Bit settings		P508	P509	P510	P511
00000b (initial value)					
00010b	GPT-COM	x	x	GTOVLO_B	GTOVUP_B
00011b	GPT	x	x	GTIOC1A_B	GTIOC1B_B
00100b	SCI0	x	TXD0_B	RXD0_B	SCK0_B
10000b	KINT	x	KRM01_B	KRM02_B	KRM03_B
ASEL bit		AN017	AN020	AN021	x
ISEL bit		IRQ4_C	x	x	x
NCODR bit		✓	✓	✓	✓
PUCR bit		✓	✓	✓	✓
DSCR bit		✓	✓	✓	✓
100-pin LFQFP/BGA product		✓	✓	✓	✓
72-pin WLBGA product		x	x	x	x
64-pin LFQFP product		x	x	x	x
56-pin QFN product		x	x	x	x

Note: ✓: Available
x: Not available

Table 22.15 Register settings for input/output pin function (Port 6) (1 of 2)

PSEL[4:0]	Function	Pin				
Bit settings		P600	P601	P602	P603	P604
00000b (initial value)						
00010b	GPT-COM	GTETRG_A_B	GTOULO_B	GTOUUP_B	GTETRGB_B	GTOWLO_B
00011b	GPT	x	x	x	GTIOC5A_B	GTIOC5B_B
00100b	SCI9	x	CTS9_B	SCK9_B	RXD9_B	TXD9_B
RTCCR.TCEN = 1 (RTCIC input enabled)		x	x	RTCIC2_B	RTCIC1_B	RTCIC0_B
ASEL bit		x	x	x	x	x
ISEL bit		x	x	x	x	IRQ3_C
NCODR bit		✓	✓	✓	✓	✓

Table 22.15 Register settings for input/output pin function (Port 6) (2 of 2)

PSEL[4:0]	Function	Pin				
Bit settings		P600	P601	P602	P603	P604
PUCR bit		✓	✓	✓	✓	✓
DSCR bit		✓	✓	✓	✓	✓
100-pin LFQFP/BGA product		✓	✓	✓	✓	✓
72-pin WLBGA product		x	x	x	x	x
64-pin LFQFP product		x	x	x	x	x
56-pin QFN product		x	x	x	x	x

Note: ✓: Available
x: Not available

Table 22.16 Register settings for input/output pin function (Port 7)

PSEL[4:0]	Function	Pin				
Bit settings		P700	P701	P702	P703	P704
00000b (initial value)						
00111b	IIC0/1	SDA1	SCL1	x	x	x
01000b	TMR	TMO1	TMRI1	x	x	x
00100b	SCI0	SCK0_C	x	RXD0_C	TXD0_C	CTS0_C
11000b	AGTW0/1	x	x	AGTWEE0_B	AGTWOB0_B	AGTWOA0_B
RTCCR.TCEN = 1 (RTCIC input enabled)		x	RTCIC2_A	x	x	x
ASEL bit		x	x	x	x	x
ISEL bit		x	x	x	x	x
NCODR bit		✓	✓	✓	✓	✓
PUCR bit		✓	✓	✓	✓	✓
DSCR bit		✓	✓	✓	✓	✓
100-pin LFQFP/BGA product		✓	✓	✓	✓	✓
72-pin WLBGA product		✓	✓	x	x	x
64-pin LFQFP product		✓	✓	x	x	x
56-pin QFN product		x	x	x	x	x

Note: ✓: Available
x: Not available

Table 22.17 Register settings for input/output pin function (Port 8) (1) (1 of 2)

PSEL[4:0]	Function	Pin	
Bit settings		P806	P807
00000b (initial value)			
00001b	AGT0/1	AGTOB1_B	AGTOA1_B
00101b	SCI3	x	CTS3_B
10001b	QSPI	x	QSSL_A
ASEL bit		x	x
ISEL bit		x	IRQ6_A
NCODR bit		✓	✓
PUCR bit		✓	✓

Table 22.17 Register settings for input/output pin function (Port 8) (1) (2 of 2)

PSEL[4:0]	Function	Pin	
Bit settings		P806	P807
DSCR bit		✓	✓
100-pin LFQFP/BGA product		✓	✓
72-pin WLBGA product		x	✓
64-pin LFQFP product		x	x
56-pin QFN product		x	x

Note: ✓: Available
x: Not available

Table 22.18 Register settings for input/output pin function (Port 8) (2)

PSEL[4:0]	Function	Pin							
Bit settings		P808	P809	P810	P811	P812	P813	P814	P815
00000b (initial value)									
00001b	AGT0/1	AGT01_B	AGTEE1_B	AGTIO1_B	x	x	x	x	x
00011b	GPT	x	GTIOC3B_B	GTIOC3A_B	x	x	x	x	x
00100b	SCI4	x	x	x	x	TXD4_C	RXD4_C	SCK4_C	CTS4_C
00101b	SCI3	RXD3_B	x	x	x	x	x	x	x
00110b	SPI0/1	x	x	x	x	x	SSLA3_B	SSLA2_B	MISOA_B
10001b	QSPI	QIO3_A	QIO2_A	QIO1_A	QIO0_A	QSPCLK_A	x	x	x
11000b	AGTW0/1	x	x	x	AGTWIO1_B	AGTWEE1_B	AGTWO1_B	AGTWOA1_B	AGTWOB1_B
ASEL bit		x	x	x	x	x	x	x	x
ISEL bit		IRQ2_B	IRQ6_B	IRQ5_B	x	x	x	x	x
NCODR bit		✓	✓	✓	✓	✓	✓	✓	✓
PUCR bit		✓	✓	✓	✓	✓	✓	✓	✓
DSCR bit		✓	✓	✓	✓	✓	✓	✓	✓
100-pin LFQFP/BGA product		✓	✓	✓	✓	✓	✓	✓	✓
72-pin WLBGA product		x	x	x	x	x	x	x	x
64-pin LFQFP product		x	x	x	x	x	x	x	x
56-pin QFN product		x	x	x	x	x	x	x	x

Note: ✓: Available
x: Not available

22.4 Handling of Unused Pins

Table 22.19 shows how to handle unused pins.

Table 22.19 Handling of unused pins (1 of 2)

Pin name	Description
P201/MD	(Use this as a mode pin.)
EHMD	Connect this pin to VSS.
RES#	Connect this pin to VCC via a resistor (pull-up).
P200/NMI	Connect this pin to VCC via a resistor (pull-up).

Table 22.19 Handling of unused pins (2 of 2)

Pin name	Description
P412/EXTAL	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P412). When this pin is not used as port P412, connect it to VSS through a resistor (pull-down).
P413/XTAL	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P413). When this pin is not used as port P413, release the pin.
Ports 0 to 8	If the direction setting is for input (PCNTR1.PDRn = 0), connect the associated pin to VCC through a resistor (pull-up) or to VSS through a resistor (pull-down). ^{*1} If the direction setting is for output (PCNTR1.PDRn = 1), release the pin. ^{*1 *2}
VREFH0	Connect this pin to AVCC0. This can be left open when AVCC0 is not supplied power.
VREFL0	Connect this pin to AVSS0. This can be left open when AVCC0 is not supplied power.
BSCANP	When the boundary scan function is not used, connect it to VSS.

Note 1. Clear the PmnPFS.PMR bit, the PmnPFS.ISEL bit, and the PmnPFS.ASEL bit to 0. For details, see [section 22.2.6. PmnPFS : Port mn Pin Function Select Register \(m = 0 to 8, n = 00 to 15\)](#).

Note 2. When a pin is set to output and released, the port becomes an input from the time the reset is cleared to the time the pin becomes an output. Because the voltage on the pin is undefined while the port is an input, this might lead to an increase in the current drawn.

22.5 Usage Notes

22.5.1 Port Output Data Register (PODR) Summary

PODR register outputs data as follows:

1. Outputs 0 if PCNTR4.EORRn is set to 1 when the ELC_PORT1 or 2 occurs from the ELC.
2. Outputs 1 if PCNTR4.EOSRn is set to 1 when the ELC_PORT1 or 2 occurs from the ELC.
3. Outputs 0 if PCNTR3.PORRn is set to 1.
4. Outputs 1 if PCNTR3.POSRn is set to 1.
5. Outputs 0 or 1 if PCNTR1.PODRn is set.
6. Outputs 0 or 1 if PmnPFS.PODRn is set.

Numbers in this list correspond to the priority for writing the PODR register. For example, if 1. and 5. from the list above occur at same time, the higher priority number 1. is executed.

22.5.2 Procedure for Specifying Input/Output Pin Function

Use the following procedure to specify the input/output pin functions.

1. Write 0 to the PWPR.BOWI bit to enable writing to the PFSWE bit in the PWPR register.
2. Write 1 to the PWPR.PFSWE bit to enable writing to the PmnPFS register.
3. Set the Port Mode Control bit (PMR) to 0 for the target pin to set the general I/O port function.
4. For the peripheral module, set the input/output signal to be assigned to the target pin. However, it is prohibited to set to assign the same function to multiple terminals.
5. Specify the input/output function for the pin through the PSEL[4:0] bits in the PmnPFS register.
6. Set the Port Mode Control bit (PMR) to 1 as required to switch to the selected input/output function for the pin.
7. Set the PFSWE bit in the PWPR register to 0 to disable writing to the PmnPFS register.
8. Set the BOWI bit in the PWPR register to 1 to disable writing to the PWPR.PFSWE bit.

22.5.3 Procedure for Using Port Group Input

To use the port group input (Ports 2 and 3):

1. Set the ELSRn.ELS[7:0] bits to 00000000b to ignore the unexpected pulse. For details, see [section 21, Event Link Controller \(ELC\)](#).

2. Set the EOFR[1:0] bits of the PmnPFS register to specify the rising, falling, or both edge detections.
3. Execute a dummy read or wait for a short time, for example 100 ns.
4. Set the ELSRn.ELS[7:0] bits to enable the event signals.

22.5.4 Notes on Using of Analog Functions

To use an analog function for the associated pin, set the Port Mode Control bit (PMR) in the Port mn Pin Function Select register (PmnPFS) to 0 and set the Pmn Direction Control bit (PDRn) in the Port Control register 1 (PDR) to 0 so that the associated pin acts as a general input port. After that, set the Analog Input Enable bit (ASEL) in the Port mn Pin Function Select register (PmnPFS) to 1.

23. Port Output Enable for GPT (POE)

23.1 Overview

The Port Output Enable (POE) 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 GTETR_{Gn} (n = A, B) pins
- Output-disable request from the GPT
- Oscillation stop detection of the clock generation circuit
- Register settings

The GTETR_{Gn} (n = A, B) pins can be used as GPT external trigger input pins.

Table 23.1 lists the POE specifications, Figure 23.1 shows a block diagram, and Table 23.2 lists the input pins.

Table 23.1 POE specifications

Parameter	Specifications
Output-disable control through input level detection	• The GPT output pins can be disabled when a GTETR_{Gn} rising edge or high level is sampled after polarity and filter selection.
Output-disable request from the GPT	• When the GTIOCxA pin and the GTIOCxB pin are driven to an active level simultaneously, the GPT generates an output-disable request to the POE. Through reception of these requests, the POE can control whether the GTIOCxA and GTIOCxB pins are output-disabled.
Output-disable control through oscillation stop detection	• The GPT output pins can be disabled when oscillation of the clock generation circuit stops.
Output-disable control by software (registers)	• The GPT output pins can be disabled by modifying the register settings.
Interrupt	• Interrupts can be generated by detecting the input level of external trigger input pins (GTETR_{Gn} pins). • Interrupts can be generated when all GPT output pins are driven to an active level simultaneously.
External trigger output to the GPT	• The GTETR_{Gn} signals can be output to the GPT after polarity and filter selection. (count start, count stop, count clear, up-count, down-count, or input capture function)
Noise filtering	• For input from the GTETR_{Gn} pins, PCLKB/1, PCLKB/8, PCLKB/32, or PCLKB/128 can be selected as the noise filtering clock. (Filtering is performed by sampling the input signals three times using the selected clock.) • Positive or negative polarity can be selected for any of the GTETR_{Gn} input pins. • Signal state after polarity and filter selection can be monitored.

Note: n = A, B, x = 0 to 5

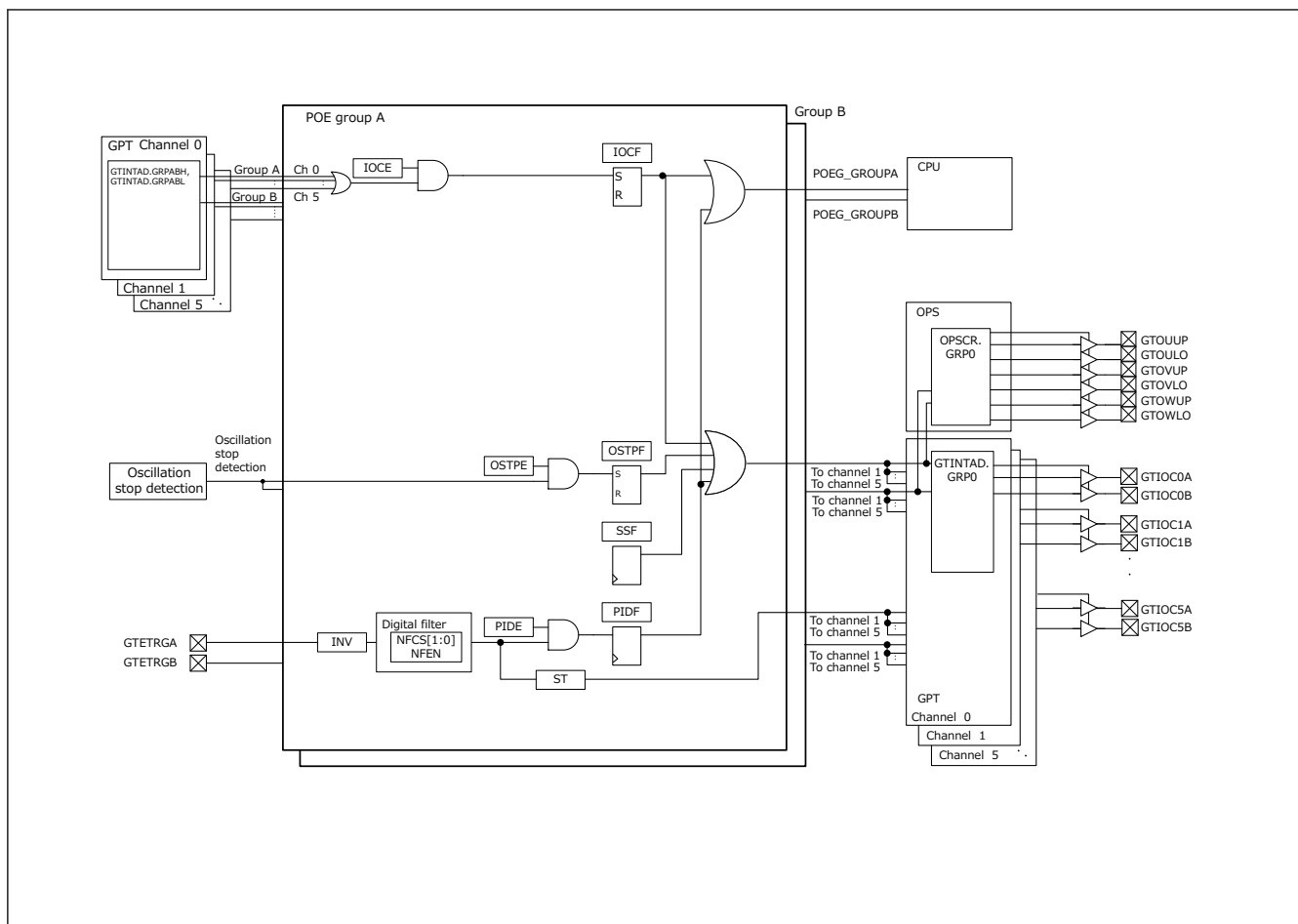


Figure 23.1 POE block diagram

Table 23.2 POE input pins

Pin name	I/O	Description
GTETRGA	Input	GPT output pin output-disable request signal or GPT external trigger input pin A
GTETRGB	Input	GPT output pin output-disable request signal or GPT external trigger input pin B

23.2 Register Descriptions

23.2.1 POEGGn : POE Group n Setting Register (n = A, B)

Base address: POE = 0x4004_2000

Offset address: 0x000 (POEGGA)
0x100 (POEGGB)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFC[1:0]		NFEN	INV	—	—	—	—	—	—	—	—	—	—	—	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	OSTP E	IOCE	PIDE	SSF	OSTP F	IOCF	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIDF	Port Input Detection Flag 0: No output-disable request from the GTETRn pin occurred 1: Output-disable request from the GTETRn pin occurred.	R/W ^{*1}
1	IOCF	Detection Flag for GPT Output-Disable Request 0: No output-disable request from GPT occurred. 1: Output-disable request from GPT occurred.	R/W ^{*1}
2	OSTPF	Oscillation Stop Detection Flag 0: No output-disable request from oscillation stop detection occurred 1: Output-disable request from oscillation stop detection occurred	R/W ^{*1}
3	SSF	Software Stop Flag 0: No output-disable request from software occurred 1: Output-disable request from software occurred	R/W
4	PIDE	Port Input Detection Enable 0: Disable output-disable requests from the GTETRn pins 1: Enable output-disable requests from the GTETRn pins	R/W ^{*2}
5	IOCE	Enable for GPT Output-Disable Request 0: Disable output-disable requests from GPT 1: Enable output-disable requests from GPT	R/W ^{*2}
6	OSTPE	Oscillation Stop Detection Enable 0: Disable output-disable requests from oscillation stop detection 1: Enable output-disable requests from oscillation stop detection	R/W ^{*2}
15:7	—	These bits are read as 0. The write value should be 0.	R/W
16	ST	GTETRn Input Status Flag 0: GTETRn input after filtering was 0 1: GTETRn input after filtering was 1	R
27:17	—	These bits are read as 0. The write value should be 0.	R/W
28	INV	GTETRn Input Reverse 0: Input GTETRn as-is 1: Input GTETRn in reverse	R/W
29	NFEN	Noise Filter Enable 0: Disable noise filtering 1: Enable noise filtering	R/W
31:30	NFCS[1:0]	Noise Filter Clock Select 0 0: Sample GTETRn pin input level three times every PCLKB 0 1: Sample GTETRn pin input level three times every PCLKB/8 1 0: Sample GTETRn pin input level three times every PCLKB/32 1 1: Sample GTETRn pin input level 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 POEGn (n = A, B) registers control the output-disable state of the GPT pins, interrupts, and the external trigger input to the GPT.

In the descriptions, POEGn represents the POEGn (n = A, B) registers.

23.3 Output-Disable Control Operation

If any of the following conditions is satisfied, the GTIOCxA, GTIOCxB, and the 3-phase PWM output for BLDC motor control pins can be set to output-disable:

- Input level or edge detection of the GTETRn pins
When POEGn.PIDE is 1, the active level set in POEGn.INV is input and the POEGn.PIDF flag is set to 1.
- Output-disable request from the GPT
When POEGn.IOCE is 1, the POEGn.IOCF flag is set to 1 if the disable request is enabled by GTINTAD. The GTINTAD.GRPABH and GTINTAD.GRPABL settings apply to the group selected by the GPT register GTINTAD.GRP0 or OPSCR.GRP0.
- Oscillation stop detection for the clock generation circuit

While POEGGn.OSTPE is 1, the halt status of the main clock oscillator is detected and the POEGGn.OSTPF flag is set to 1.

- SSF bit setting

When POEGGn.SSF is set to 1.

The output-disable state is controlled in the GPT module. The output-disable of the GTIOCxA and GTIOCxB pins is set in the GTINTAD.GRP0, GTIOR.OADF[1:0], and GTIOR.OBDF[1:0] bits in GPTx. The output-disable of the 3-phase PWM output for BLDC motor control pins is set in the OPSCR.GRP0 bit and OPSCR.GODF bit in GPT_OPS.

23.3.1 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 GTETRn pins, the GPT output pins are output-disabled.

23.3.1.1 Digital Filter

Figure 23.2 shows high-level detection by the digital filter. When a high level associated with the POEGGn.INV polarity setting is detected three times consecutively with the sampling clock selected in POEGGn.NFCS[1:0] and POEGGn.NFEN, the detected level is recognized as high, and the GPT output pins are output-disabled. If even one low level is detected during this interval, the detected level is not recognized as high. In addition, in an interval where the sampling clock is not output, changes of the levels on the GTETRn pins are ignored.

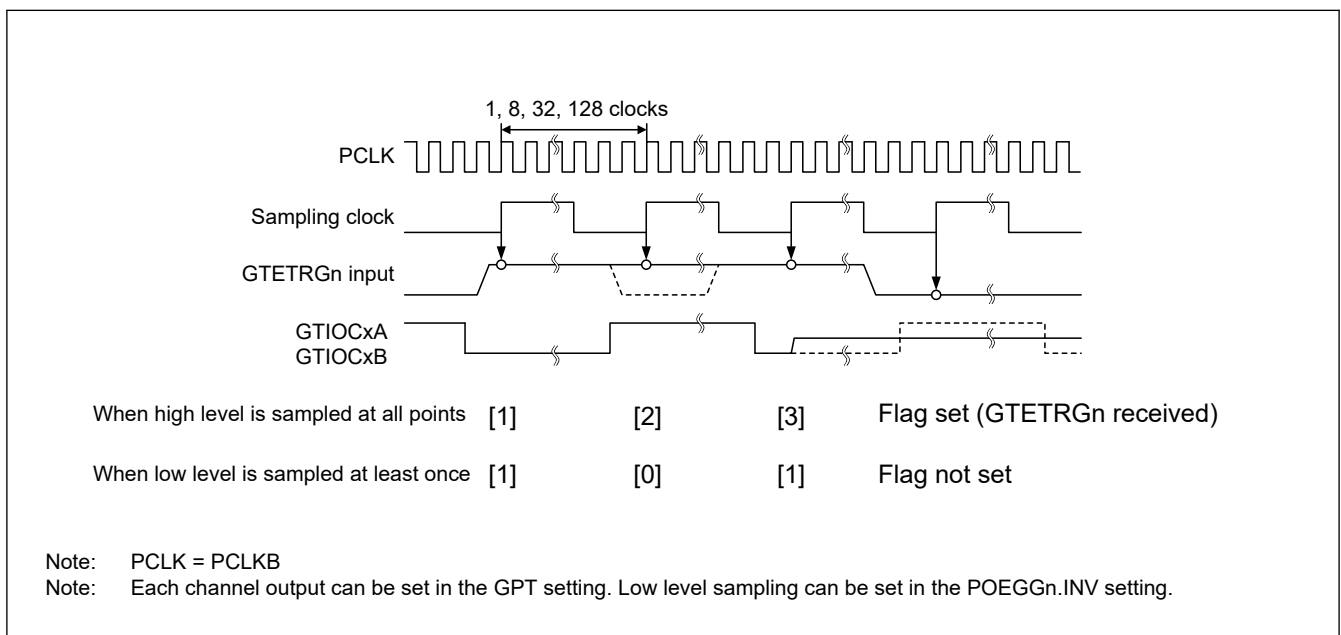


Figure 23.2 Example of digital filter operation

23.3.2 Output-Disable Requests from the GPT

For details on the operation, see [section 24.7.3. GTIOCnm Pin Output Negate Control \(n = 0 to 5, m = A, B\)](#).

23.3.3 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.

23.3.4 Output-Disable Control Using Registers

The GPT output pins can be directly controlled by writing 1 to the Software Stop flag, POEGGn.SSF.

23.3.5 Release from Output-Disable

To release the GPT output pins placed in the output-disable state, either return them to their initial state with a reset or clear all of the following 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, GTETR_{Gn} 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 the 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 23.3 shows the release 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.

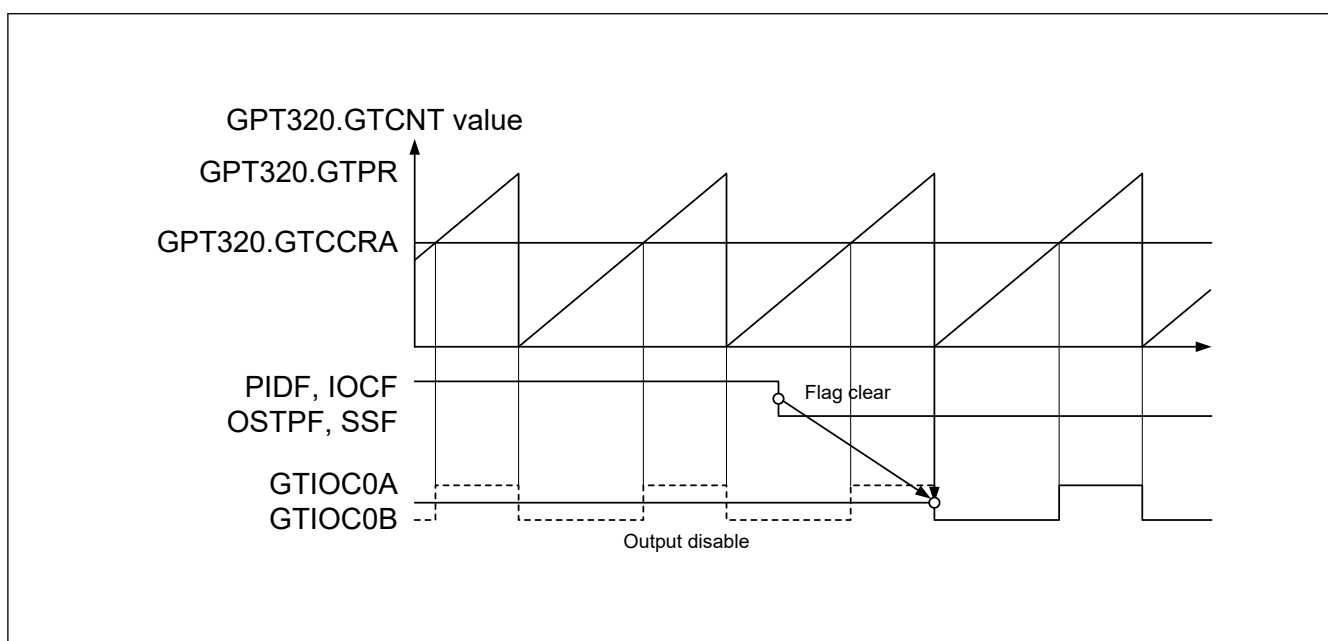


Figure 23.3 Output-disable release timing for GPT pin outputs

23.4 Interrupt Sources

The POE generates an interrupt request when triggered by these sources:

- Output-disable control by the input level detection
- Output-disable request from the GPT

Table 23.3 lists the conditions for interrupt requests.

Table 23.3 Interrupt sources and conditions (1 of 2)

Interrupt source	Symbol	Associated flag	Trigger conditions
POE group A interrupt	POEG_GROUPA	POEGGA.IOCF	An output-disable request from a GPT disable request occurred
		POEGGA.PIDF	An output-disable request from the GTETRGA pin occurred

Table 23.3 Interrupt sources and conditions (2 of 2)

Interrupt source	Symbol	Associated flag	Trigger conditions
POE group B interrupt	POEG_GROUPB	POEGGB.IOCF	An output-disable request from a GPT disable request occurred
		POEGGB.PIDF	An output-disable request from the GTETRGB pin occurred

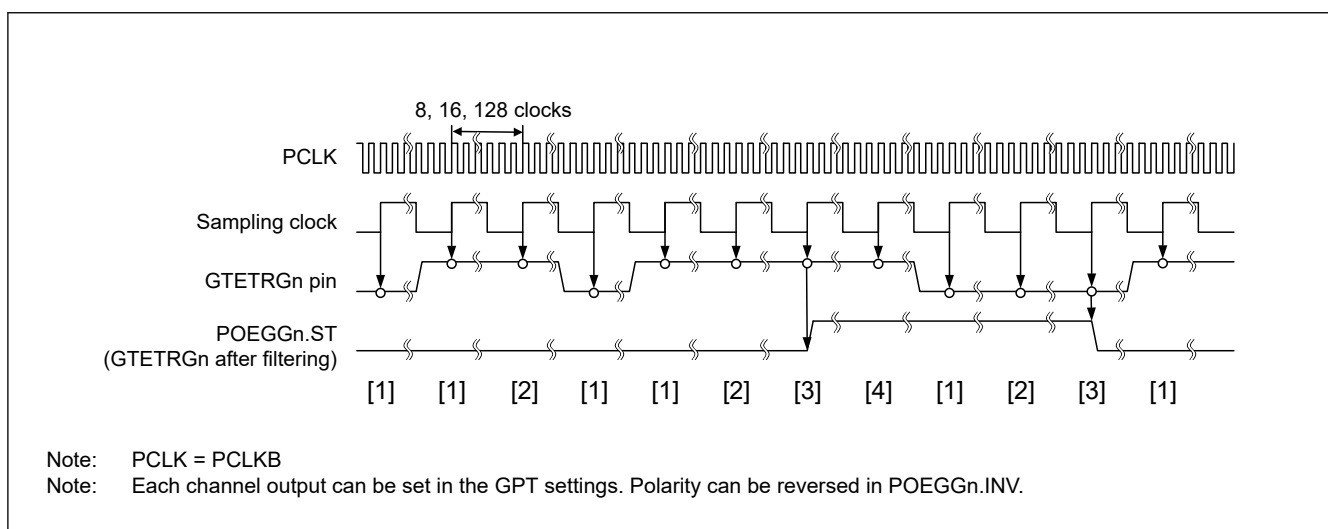
23.5 External Trigger Output to the GPT

The POE outputs signals generated by filtering and level detection of GTETR_{Gn} pins input signals as the GPT operation trigger signal for the following:

- Count start
- Count stop
- Count clear
- Up-count
- Down-count
- Input capture

For the POEG_{Gn}.INV polarity setting signal, when the same level is input three times continuously with the sampling clock selected in POEG_{Gn}.NFCS[1:0] and POEG_{Gn}.NFEN, that value is output. Set the control registers the same as for the input level detection operation described in [section 23.3.1. Pin Input Level Detection Operation](#). The state after filtering can be monitored in POEG_{Gn}.ST.

Figure 23.4 shows the output timing of an external trigger to the GPT.

**Figure 23.4** Output timing of external trigger to the GPT

23.6 Usage Notes

23.6.1 Transition to Software Standby Mode

When using the POE, do not invoke Software Standby mode. In this mode, the POE stops and therefore output disable of the pins cannot be controlled.

23.6.2 Specifying Pins Associated with the GPT

The POE controls output-disable only when a pin is associated with the GPT in the PmnPFS.PMR and PmnPFS.PSEL settings. When the pin is specified as a general I/O pin, the POE does not perform output-disable control.

23.6.3 Settings for the Module-Stop Function

Operation of the POE can be disabled or enabled by the corresponding bit in the Module Stop Control Register D (MSTPCRD). The POE operation is stopped by the initial setting after reset. Releasing the module-stop state enables access to the registers. For more information, see [section 13, Power-Saving Functions](#).

24. General PWM Timer (GPT)

24.1 Overview

The General PWM Timer (GPT) is a 32-bit timer with $GPT32 \times 2$ channels and a 16-bit timer with $GPT16 \times 4$ channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. In addition, PWM waveforms can be generated for controlling brushless DC motors. The GPT can also be used as a general-purpose timer.

Table 24.1 lists the GPT specifications, Table 24.2 shows the GPT functions, and Figure 24.1 shows a block diagram.

Table 24.1 GPT specifications

Parameter	Description
Functions	• 32 bits $\times$ 2 channels (GPT32n (n = 0 to 1)) • 16 bits $\times$ 4 channels (GPT16m (m = 2 to 5)) • Up-counting or down-counting (saw waves) or up/down-counting (triangle waves) for each counter • Clock sources independently selectable for each channel • Two input/output pins per channel • Two output compare/input capture registers per channel • For the two output compare/input capture registers of each channel, four registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use • In output compare operation, buffer switching can be at crests or troughs, enabling the generation of laterally asymmetric PWM waveforms • Registers for setting up frame cycles in each channel with capability for generating interrupts at overflow or underflow • Generation of dead times in PWM operation • Synchronous starting, stopping and clearing counters for arbitrary channels • Count start, count stop, count clear, up-count, down-count, or input capture operation in response to a maximum of 4 ELC events • Count start, count stop, count clear, up-count, down-count, or input capture operation in response to the status of two input pins • Count start, count stop, count clear, up-count, down-count, or input capture operation in response to a maximum of 2 external triggers • Output pin disable function by 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. • Bus clock: PCLKB, Core clock: PCLKA

Table 24.2 GPT functions (1 of 2)

Parameter	Description
Count clock	PCLKA PCLKA/4 PCLKA/16 PCLKA/64 PCLKA/256 PCLKA/1024
Output compare/input capture registers (GTCCR)	GTCCRA GTCCRB
Compare/buffer registers	GTCCRC GTCCRD GTCCRE GTCCRF
Cycle setting register	GTPR
Cycle setting buffer register	GTPBR
I/O pins	GTIOCnA GTIOCnB (n = 0 to 5)
External trigger input pin ^{*1}	GTETRGA GTETRGB

Table 24.2 GPT functions (2 of 2)

Parameter		Description
Counter clear sources		GTPR register compare match, input capture, input pin status, ELC event input, and GTETRGn (n = A, B) pin input
Compare match output	Low output	Available
	High output	Available
	Toggle output	Available
Input capture function		Available
Automatic addition of dead time		Available (no dead time buffer)
PWM mode		Available
Phase count function		Available
Buffer operation		Double buffer
One-shot operation		Available
DTC activation		All the interrupt sources
Brushless DC motor control function		Available
Interrupt sources		6 sources (n = 0 to 5) • GTCCRA comare match/input capture(GPTn_CCMPA) • GTCCRB comare match/input capture(GPTn_CCMPB) • GTCCRC comare match(GPTn_CMPC) • GTCCRD comare match(GPTn_CMPD) • GTCNT overflow (GTPR compare match) (GPTn_OVF) • GTCNT underflow (GPTn_UDF)
Event linking (ELC) function		Available ^{*2}
Noise filtering function		Available

Note 1. GTRETRGn connects to GPT through the POEG module. Therefore, to use the GPT function, supply the POEG clock by clearing the MSTPCRD.MSTPD14 bit.

Note 2. see [section 24.5. Operations Linked by ELC](#)

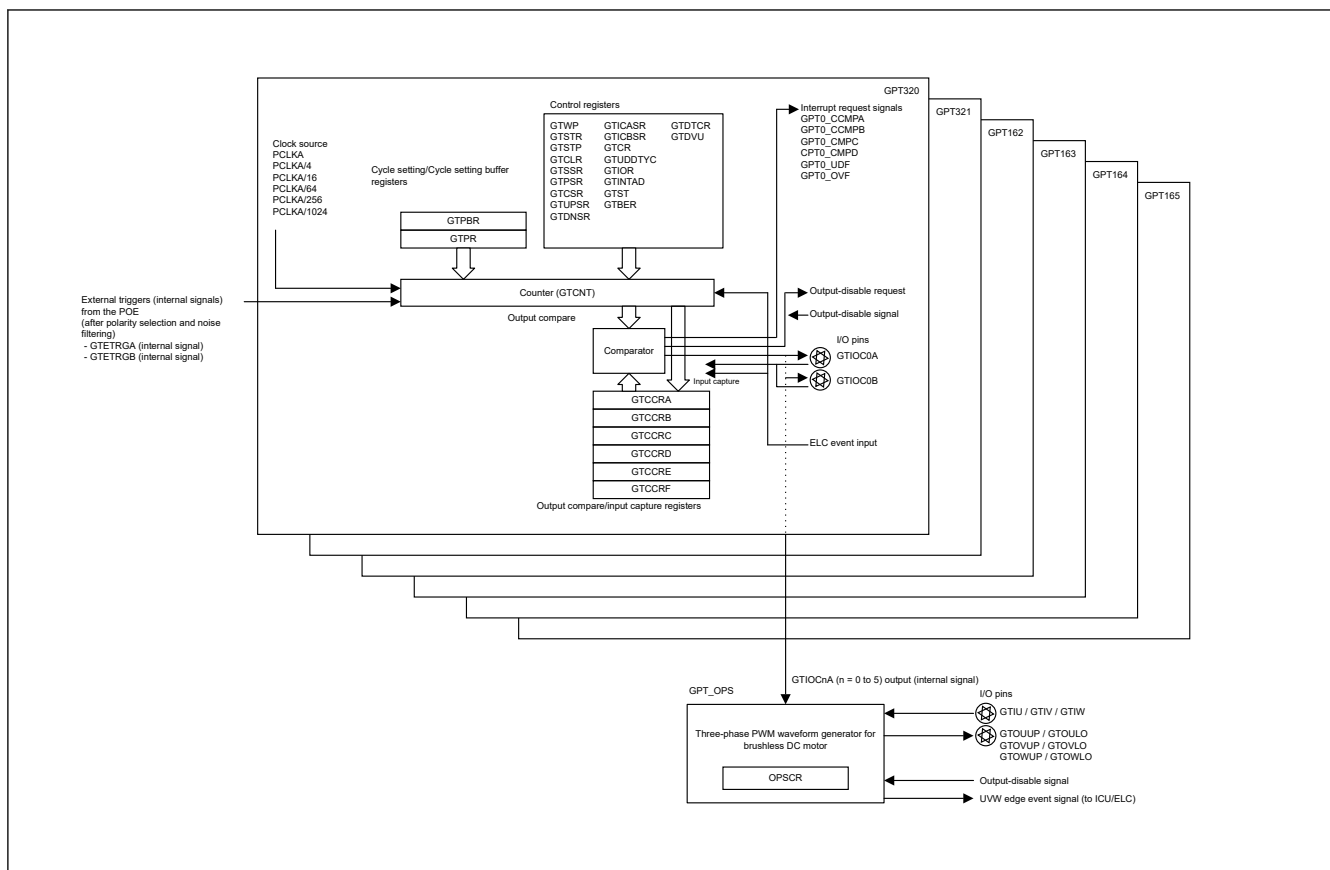


Figure 24.1 GPT block diagram

Figure 24.2 shows an example using multiple GPTs.

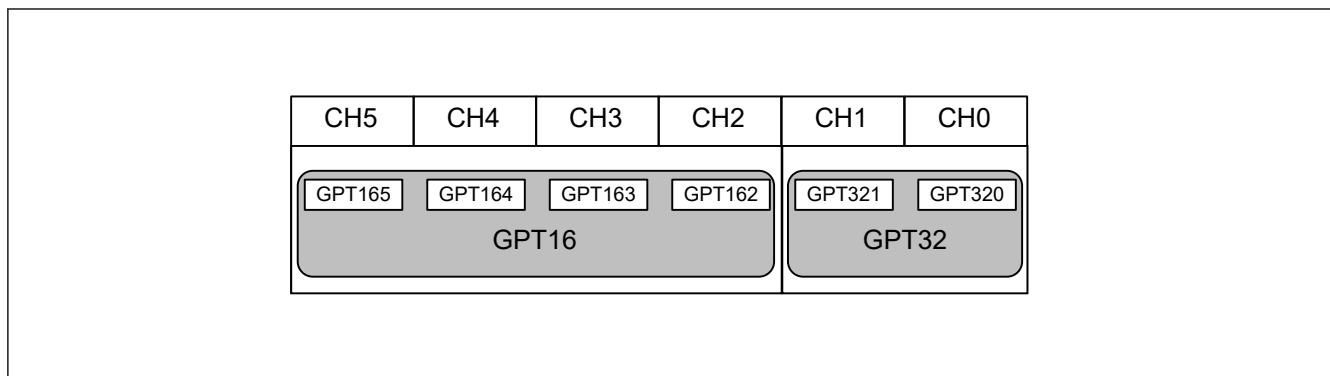


Figure 24.2 Association between GPT channels and module names

Table 24.3 lists the I/O pins.

Table 24.3 GPT I/O pins (1 of 2)

Channel	Pin name	I/O	Function
Common	GTETRGx	Input	External trigger input pin x (input through the POEG)
GPT32n	GTIOCnA	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOCnB	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT16m	GTIOCmA	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOCmB	I/O	GTCCRB register input capture input/output compare output/PWM output pin

Table 24.3 GPT I/O pins (2 of 2)

Channel	Pin name	I/O	Function
GPT_OPS	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U-phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U-phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V-phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V-phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W-phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W-phase)

Note: x: A, B
n: 0 to 1
m: 2 to 5

24.2 Register Descriptions

24.2.1 GTWP : General PWM Timer Write-Protection Register

Base address: $GPT32n = 0x4005\_5000 + 0x0100 \times n$ ($n = 0$ to 1)
 $GPT16m = 0x4005\_5000 + 0x0100 \times m$ ($m = 2$ to 5)

Offset address: $0x00$

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								—	—	—	—	—	—	—	WP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WP	Register Write Disable 0: Write to the register enabled 1: Write to the register disabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	GTWP Key Code When 0xA5 is written to these bits, writing to the WP bit is permitted. These bits are read as 0.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

GTWP enables or disables writing to registers to prevent accidental modification. Protection by the GTWP register is only for the writes by the CPU. GTWP does not protect registers from updates that occur in association with CPU writes.

WP bit (Register Write Disable)

The following is a list of write enabled or disabled registers:

GTSSR, GTPSR, GTCR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTDTCR, GTDVU.

PRKEY[7:0] bit (GTWP Key Code)

This bit controls whether the WP bit can be overwritten.

24.2.2 GTSTR : General PWM Timer Software Start Register

Base address: GPT32n = 0x4005_5000 + 0x0100 × n (n = 0 to 1)
GPT16m = 0x4005_5000 + 0x0100 × m (m = 2 to 5)

Offset address: 0x04

Bit position: 31

0

Bit field:

CSTRT31 to CSTRT0

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CSTRT0 to CSTRT31 ^{*1}	Channel n GTCNT Count Start (n : the same as bit position value) 0: GTCNT counter not start 1: GTCNT counter start	R/W

Note 1. The bits that can be used vary depending on the product. The n of CSTRTn is the same as the GPT channel number. For this product, n is 0 to 5.

The GTSTR starts the GTCNT counter operation for each channel n, where n = 0 to 5.

The GTSTR bit number represents the channel number. The GTSTR register of each channel is shared by all of the channels. The GTCNT counter starts for the channel associated with the GTSTR bit number where 1 is written. Writing 0 has no effect on the status of GTCNT counter and the value of GTSTR register.

For the association between GTSTR bit number and a channel number, see [Figure 24.2](#).

CSTRTn bits (Channel n GTCNT Count Start (n = 0 to 5))

The CSTRTn bits start channel n of the GTCNT counter operation. Writing to the GTSTR.CSTRTn bit (n = 0 to 5) has no effect unless the GTSSR.CSTRT bit is set to 1.

The read data shows the counter status of each channel (GTCR.CST bit). Zero means the counter is stopped and 1 means the counter is running.

24.2.3 GTSTP : General PWM Timer Software Stop Register

Base address: GPT32n = 0x4005_5000 + 0x0100 × n (n = 0 to 1)
GPT16m = 0x4005_5000 + 0x0100 × m (m = 2 to 5)

Offset address: 0x08

Bit position: 31

0

Bit field:

CSTOP31 to CSTOP0

Value after reset: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	CSTOP0 to CSTOP31 ^{*1}	Channel n GTCNT Count Stop (n : the same as bit position value) 0: GTCNT counter not stop 1: GTCNT counter stop	R/W

Note 1. The bits that can be used vary depending on the product. The n of CSTOPn is the same as the GPT channel number. For this product, n is 0 to 5.

The GTSTP stops the GTCNT counter operation for each channel n, where n = 0 to 5.

The GTSTP bit number represents the channel number. The GTSTP register of each channel is shared by all the channels. The GTCNT counter stops for the channel associated with the GTSTP bit number where 1 is written. Writing 0 has no effect on the status of the GTCNT counter and the value of GTSTP register.

For the association between GTSTP bit number and a channel number, see [Figure 24.2](#).

CSTOPn bits (Channel n GTCNT Count Stop (n = 0 to 5))

The CSTOPn bits stop channel n of the GTCNT counter operation. Writing to the GTSTP.CSTOPn bit (n = 0 to 5) has no effect unless the GTPSR.CSTOPn bit is set to 1. 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 is stopped.

24.2.4 GTCLR : General PWM Timer Software Clear Register

Base address: GPT32n = 0x4005_5000 + 0x0100 × n (n = 0 to 1)
GPT16m = 0x4005_5000 + 0x0100 × m (m = 2 to 5)

Offset address: 0x0C

Bit position: 31

0

Bit field:

CCLR31 to CCLR0

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CCLR0 to CCLR31 ^{*1}	Channel n GTCNT Count Clear (n : the same as bit position value) 0: GTCNT counter is not cleared 1: GTCNT counter is cleared	W

Note 1. The bits that can be used vary depending on the product. The n of CCLRn is the same as the GPT channel number. For this product, n is 0 to 5.

The GTCLR is a write-only register that clears the GTCNT counter operation for each channel n, where n = 0 to 5.

The GTCLR bit number represents the channel number. The GTCLR register of each channel is shared by all the channels. The GTCNT counter is cleared for the channel associated with the GTCLR bit number where 1 is written. Writing 0 has no effect on the status of GTCNT counter.

For the association between GTCLR bit number and a channel number, see [Figure 24.2](#).

CCLRn bits (Channel n GTCNT Count Clear (n = 0 to 5))

Channel n of the GTCNT counter value is cleared on writing 1 to this bit. This bit is read as 0.

24.2.5 GTSSR : General PWM Timer Start Source Select Register

Base address: GPT32n = 0x4005_5000 + 0x0100 × n (n = 0 to 1)
GPT16m = 0x4005_5000 + 0x0100 × m (m = 2 to 5)

Offset address: 0x10

Bit position:

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

Bit field:

CSTR T	—	—	—	—	—	—	—	—	—	—	—	SSEL CD	SSEL CC	SSEL CB	SSEL CA

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit position:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:

SSCB FAH	SSCB FAL	SSCB RAH	SSCB RAL	SSCA FBH	SSCA FBL	SSCA RBH	SSCA RBL	—	—	—	—	SSGT RBF	SSGT RBR	SSGT RGAF	SSGT RGAR

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	SSGTRGAR	GTETRGA Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGA input 1: Counter start enabled on the rising edge of GTETRGA input	R/W
1	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

Bit	Symbol	Function	R/W
2	SSGTRGBR	GTETRGB Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGB input 1: Counter start enabled on the rising edge of GTETRGB input	R/W
3	SSGTRGBF	GTETRGB Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGB input 1: Counter start enabled on the falling edge of GTETRGB input	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	SSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter start enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	SSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter start enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	SSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter start enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	SSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter start enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	SSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter start enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	SSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter start enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	SSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter start enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	SSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter start enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	SSELCA	ELC_GPTA Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTA input 1: Counter start enabled at the ELC_GPTA input	R/W
17	SSELCB	ELC_GPTB Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTB input 1: Counter start enabled at the ELC_GPTB input	R/W
18	SSELCC	ELC_GPTC Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTC input 1: Counter start enabled at the ELC_GPTC input	R/W

Bit	Symbol	Function	R/W
19	SSELCD	ELC_GPTD Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTD input 1: Counter start enabled at the ELC_GPTD input	R/W
30:20	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTRT	Software Source Counter Start Enable 0: Counter start disabled by the GTSTR register 1: Counter start enabled by the GTSTR register	R/W

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 (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Start Enable)

The SSCARBL bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

SSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Start Enable)

The SSCARBH bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

SSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Start Enable)

The SSCAFBL bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

SSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Start Enable)

The SSCAFBH bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

SSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Start Enable)

The SSCBRAL bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

SSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Start Enable)

The SSCBRAH bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

SSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Start Enable)

The SSCBFAL bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

SSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Start Enable)

The SSCBFAH bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

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 GTSTR register.

24.2.6 GTPSR : General PWM Timer Stop Source Select Register

Base address: $GPT32n = 0x4005\_5000 + 0x0100 \times n$ (n = 0 to 1)
 $GPT16m = 0x4005\_5000 + 0x0100 \times m$ (m = 2 to 5)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSTO P	—	—	—	—	—	—	—	—	—	—	—	PSEL CD	PSEL CC	PSEL CB	PSEL CA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PSCB FAH	PSCB FAL	PSCB RAH	PSCB RAL	PSCA FBH	PSCA FBL	PSCA RBH	PSCA RBL	—	—	—	—	PSGT RGBF	PSGT RGBR	PSGT RGAF	PSGT RGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PSGTRGAR	GTETRGA Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGA input 1: Counter stop enabled on the rising edge of GTETRGA input	R/W
1	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
2	PSGTRGBR	GTETRGB Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGB input 1: Counter stop enabled on the rising edge of GTETRGB input	R/W
3	PSGTRGBF	GTETRGB Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGB input 1: Counter stop enabled on the falling edge of GTETRGB input	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	PSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter stop enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	PSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter stop enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	PSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter stop enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	PSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter stop enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W

Bit	Symbol	Function	R/W
12	PSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter stop enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	PSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter stop enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	PSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter stop enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	PSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter stop enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	PSELCA	ELC_GPTA Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTA input 1: Counter stop enabled at the ELC_GPTA input	R/W
17	PSELCB	ELC_GPTB Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTB input 1: Counter stop enabled at the ELC_GPTB input	R/W
18	PSELCC	ELC_GPTC Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTC input 1: Counter stop enabled at the ELC_GPTC input	R/W
19	PSELCD	ELC_GPTD Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTD input 1: Counter stop enabled at the ELC_GPTD input	R/W
30:20	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTOP	Software Source Counter Stop Enable 0: Counter stop disabled by the GTSTP register 1: Counter stop enabled by the GTSTP register	R/W

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 (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Stop Enable)

The PSCARBL bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

PSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Stop Enable)

This bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

PSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Stop Enable)

The PSCAFBL bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

PSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Stop Enable)

The PSCAFBH bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

PSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Stop Enable)

The PSCBRAL bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

PSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Stop Enable)

The PSCBRAH bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

PSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Stop Enable)

The PSCBFAL bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

PSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Stop Enable)

The PSCBFAH bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

PSELCm bit (ELCm 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.

24.2.7 GTCSR : General PWM Timer Clear Source Select Register

Base address: GPT32n = 0x4005_5000 + 0x0100 × n (n = 0 to 1)
GPT16m = 0x4005_5000 + 0x0100 × m (m = 2 to 5)

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CCLR	—	—	—	—	—	—	—	—	—	—	—	CSEL CD	CSEL CC	CSEL CB	CSEL CA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CSCB FAH	CSCB FAL	CSCB RAH	CSCB RAL	CSCA FBH	CSCA FBL	CSCA RBH	CSCA RBL	—	—	—	—	CSGT RGBF	CSGT RGBR	CSGT RGAF	CSGT RGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CSGTRGAR	GTETRGA Pin Rising Input Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTETRGA input 1: Counter clear enabled on the rising edge of GTETRGA input	R/W

Bit	Symbol	Function	R/W
1	CSGTRGAF	GTETRGA Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGA input 1: Counter clear enabled on the falling edge of GTETRGA input	R/W
2	CSGTRGBR	GTETRGB Pin Rising Input Source Counter Clear Enable 0: Disable counter clear on the rising edge of GTETRGB input 1: Enable counter clear on the rising edge of GTETRGB input	R/W
3	CSGTRGBF	GTETRGB Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGB input 1: Counter clear enabled on the falling edge of GTETRGB input	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	CSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter clear enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	CSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter clear enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	CSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter clear enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	CSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter clear enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	CSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter clear enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	CSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter clear enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	CSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter clear enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	CSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter clear enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	CSELCA	ELC_GPTA Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTA input 1: Counter clear enabled at the ELC_GPTA input	R/W
17	CSELCB	ELC_GPTB Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTB input 1: Counter clear enabled at the ELC_GPTB input	R/W

Bit	Symbol	Function	R/W
18	CSELCC	ELC_GPTC Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTC input 1: Counter clear enabled at the ELC_GPTC input	R/W
19	CSELCD	ELC_GPTD Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTD input 1: Counter clear enabled at the ELC_GPTD input	R/W
30:20	—	These bits are read as 0. The write value should be 0.	R/W
31	CCLR	Software Source Counter Clear Enable 0: Counter clear disabled by the GTCLR register 1: Counter clear enabled by the GTCLR register	R/W

The GTCSCR 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 (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Clear Enable)

The CSCARBL bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

CSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Clear Enable)

The CSCARBH bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

CSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Clear Enable)

The CSCAFBL bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

CSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Clear Enable)

The CSCAFBH bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

CSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Clear Enable)

The CSCBRAL bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

CSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Clear Enable)

The CSCBRAH bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

CSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Clear Enable)

The CSCBFAL bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

CSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Clear Enable)

The CSCBFAH bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

CSELCm bit (ELCm 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.

24.2.8 GTUPSR : General PWM Timer Up Count Source Select Register

Base address: $GPT32n = 0x4005\_5000 + 0x0100 \times n$ ($n = 0$ to 1)
 $GPT16m = 0x4005\_5000 + 0x0100 \times m$ ($m = 2$ to 5)

Offset address: $0x1C$

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	USEL CD	USEL CC	USEL CB	USEL CA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	USCB FAH	USCB FAL	USCB RAH	USCB RAL	USCA FBH	USCA FBL	USCA RBH	USCA RBL	—	—	—	—	USGT RBF	USGT RBR	USGT RGAF	USGT RGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	USGTRGAR	GTETRGA Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGA input 1: Counter count up enabled on the rising edge of GTETRGA input	R/W
1	USGTRGAF	GTETRGA Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGA input 1: Counter count up enabled on the falling edge of GTETRGA input	R/W
2	USGTRGBR	GTETRGB Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGB input 1: Counter count up enabled on the rising edge of GTETRGB input	R/W
3	USGTRGBF	GTETRGB Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGB input 1: Counter count up enabled on the falling edge of GTETRGB input	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	USCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count up enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	USCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count up enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	USCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count up enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W

Bit	Symbol	Function	R/W
11	USCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count up enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	USCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count up enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	USCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count up enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	USCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count up enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	USCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count up enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	USELCA	ELC_GPTA Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTA input 1: Counter count up enabled at the ELC_GPTA input	R/W
17	USELCB	ELC_GPTB Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTB input 1: Counter count up enabled at the ELC_GPTB input	R/W
18	USELCC	ELC_GPTC Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTC input 1: Counter count up enabled at the ELC_GPTC input	R/W
19	USELCD	ELC_GPTD Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTD input 1: Counter count up enabled at the ELC_GPTD input	R/W
31:20	—	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 (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Up Enable)

The USCARBL bit enables or disables GTCNT counter count up on the rising edge of GTIOCnA pin input, when GTIOCnB input is 0.

USCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Up Enable)

The USCARBH bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

USCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Up Enable)

The USCAFBL bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

USAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Up Enable)

The USAFBH bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

USCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Up Enable)

The USCBRAL bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

USCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Up Enable)

The USCBRAH bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnB pin input, when the GTIOCnA input is 1.

USCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Up Enable)

The USCBFAL bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnB pin input, when the GTIOCnA input is 0.

USCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Up Enable)

The USCBFAH bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnB pin input, when the GTIOCnA input is 1.

USELCm bit (ELC_GPTm Event Source Counter Count Up Enable) (m = A to D)

The USELCm bit enables or disables the GTCNT counter count up at the ELC_GPTm event input.

24.2.9 GTDNSR : General PWM Timer Down Count Source Select Register

Base address: $GPT32n = 0x4005\_5000 + 0x0100 \times n$ ($n = 0$ to 1)
 $GPT16m = 0x4005\_5000 + 0x0100 \times m$ ($m = 2$ to 5)

Offset address: $0x20$

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	DSEL CD	DSEL CC	DSEL CB	DSEL CA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DSCB FAH	DSCB FAL	DSCB RAH	DSCB RAL	DSCA FBH	DSCA FBL	DSCA RBH	DSCA RBL	—	—	—	—	DSGT RGBF	DSGT RGBR	DSGT RGAF	DSGT RGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DSGTRGAR	GTETRGA Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGA input 1: Counter count down enabled on the rising edge of GTETRGA input	R/W

Bit	Symbol	Function	R/W
1	DSGTRGAF	GTETRGA Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGA input 1: Counter count down enabled on the falling edge of GTETRGA input	R/W
2	DSGTRGBR	GTETRGB Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGB input 1: Counter count down enabled on the rising edge of GTETRGB input	R/W
3	DSGTRGBF	GTETRGB Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGB input 1: Counter count down enabled on the falling edge of GTETRGB input	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	DSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count down enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	DSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count down enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	DSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count down enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	DSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count down enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	DSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count down enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	DSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count down enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	DSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count down enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	DSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count down enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W

Bit	Symbol	Function	R/W
16	DSELCA	ELC_GPTA Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTA input 1: Counter count down enabled at the ELC_GPTA input	R/W
17	DSELCB	ELC_GPTB Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTB input 1: Counter count down enabled at the ELC_GPTB input	R/W
18	DSELCC	ELC_GPTC Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTC input 1: Counter count down enabled at the ELC_GPTC input	R/W
19	DSELCD	ELC_GPTD Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTD input 1: Counter count down enabled at the ELC_GPTD input	R/W
31:20	—	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 (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Down Enable)

The DSCARBL bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnA pin input, when the GTIOCnB input is 0.

DSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Down Enable)

The DSCARBH bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

DSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Down Enable)

The DSCAFBL bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

DSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Down Enable)

The DSCAFBH bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

DSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Down Enable)

The DSCBRAL bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

DSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Down Enable)

The DSCBRAH bit enables or disables the GTCNT counter count down on the rising edge of GTIOCnB pin input, when GTIOCnA input is 1.

DSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Down Enable)

The DSCBFAL bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

DSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Down Enable)

The DSCBFAH bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

DSELCm bit (ELC_GPTm Event Source Counter Count Down Enable) (m = A to D)

The DSELCm bit enables or disables the GTCNT counter count down at the ELC_GPTm event input.

24.2.10 GTICASR : General PWM Timer Input Capture Source Select Register A

Base address: $GPT32n = 0x4005\_5000 + 0x0100 \times n$ ($n = 0$ to 1)
 $GPT16m = 0x4005\_5000 + 0x0100 \times m$ ($m = 2$ to 5)

Offset address: $0x24$

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	ASEL CD	ASEL CC	ASEL CB	ASEL CA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ASCB FAH	ASCB FAL	ASCB RAH	ASCB RAL	ASCA FBH	ASCA FBL	ASCA RBH	ASCA RBL	—	—	—	—	ASGT RGBF	ASGT RGBR	ASGT RGAF	ASGT RGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ASGTRGAR	GTETRGA Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGA input 1: GTCCRA input capture enabled on the rising edge of GTETRGA input	R/W
1	ASGTRGAF	GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGA input 1: GTCCRA input capture enabled on the falling edge of GTETRGA input	R/W
2	ASGTRGBR	GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGB input 1: GTCCRA input capture enabled on the rising edge of GTETRGB input	R/W
3	ASGTRGBF	GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGB input 1: GTCCRA input capture enabled on the falling edge of GTETRGB input	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	ASCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRA input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W

Bit	Symbol	Function	R/W
9	ASCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRA input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	ASCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRA input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	ASCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRA input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	ASCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRA input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	ASCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRA input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	ASCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRA input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	ASCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRA input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	ASELCA	ELC_GPTA Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTA input 1: GTCCRA input capture enabled at the ELC_GPTA input	R/W
17	ASELCB	ELC_GPTB Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTB input 1: GTCCRA input capture enabled at the ELC_GPTB input	R/W
18	ASELCC	ELC_GPTC Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTC input 1: GTCCRA input capture enabled at the ELC_GPTC input	R/W
19	ASELCD	ELC_GPTD Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTD input 1: GTCCRA input capture enabled at the ELC_GPTD input	R/W
31:20	—	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 (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable)

The ASCARBL bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

ASCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRA Input Capture Enable)

The ASCARBH bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

ASCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable)

The ASCAFBL bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

ASCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRA Input Capture Enable)

The ASCAFBH bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnA pin input, when the GTIOCnB input is 1.

ASCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable)

The ASCBRAL bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnB pin input, when the GTIOCnA input is 0.

ASCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRA Input Capture Enable)

The ASCBRAH bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

ASCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable)

The ASCBFAL bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

ASCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRA Input Capture Enable)

The ASCBFAH bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

ASELCm bit (ELC_GPTm Event Source Counter GTCCRA Input Capture Enable) (m = A to D)

The ASELCm bit enables or disables the input capture for GTCCRA at the ELC_GPTm event input.

24.2.11 GTICBSR : General PWM Timer Input Capture Source Select Register B

Base address: GPT32n = 0x4005_5000 + 0x0100 × n (n = 0 to 1)
GPT16m = 0x4005_5000 + 0x0100 × m (m = 2 to 5)

Offset address: 0x28

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSCB FAH	BSCB FAL	BSCB RAH	BSCB RAL	BSCA FBH	BSCA FBL	BSCA RBH	BSCA RBL	—	—	—	—	BSGT RGBF	BSGT RGBR	BSGT RGAF	BSGT RGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BSGTRGAR	GTETRGA Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGA input 1: GTCCRB input capture enabled on the rising edge of GTETRGA input	R/W
1	BSGTRGAF	GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGA input 1: GTCCRB input capture enabled on the falling edge of GTETRGA input	R/W
2	BSGTRGBR	GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGB input 1: GTCCRB input capture enabled on the rising edge of GTETRGB input	R/W
3	BSGTRGBF	GTETRGB Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGB input 1: GTCCRB input capture enabled on the falling edge of GTETRGB input	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	BSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRB input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	BSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRB input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	BSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRB input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	BSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRB input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	BSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRB input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W

Bit	Symbol	Function	R/W
13	BSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRB input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	BSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRB input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	BSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRB input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	BSELCA	ELC_GPTA Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTA input 1: GTCCRB input capture enabled at the ELC_GPTA input	R/W
17	BSELCB	ELC_GPTB Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTB input 1: GTCCRB input capture enabled at the ELC_GPTB input	R/W
18	BSELCC	ELC_GPTC Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTC input 1: GTCCRB input capture enabled at the ELC_GPTC input	R/W
19	BSELCD	ELC_GPTD Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTD input 1: GTCCRB input capture enabled at the ELC_GPTD input	R/W
31:20	—	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 (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable)

The BSCARBL bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnA pin input, when the GTIOCnB input is 0.

BSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRB Input Capture Enable)

The BSCARBH bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

BSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable)

The BSCAFBL bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

BSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRB Input Capture Enable)

The BSCAFBH bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

BSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable)

The BSCBRAL bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

BSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRB Input Capture Enable)

The BSCBRAH bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

BSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable)

The BSCBFAL bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

BSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRB Input Capture Enable)

The BSCBFAH bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

BSELCm bit (ELC_GPTm Event Source Counter GTCCRB Input Capture Enable) (m = A to D)

The BSELCm bit enables or disables the input capture for GTCCRB at the ELC_GPTm event input.

24.2.12 GTCR : General PWM Timer Control Register

Base address: GPT32n = 0x4005_5000 + 0x0100 × n (n = 0 to 1)
GPT16m = 0x4005_5000 + 0x0100 × m (m = 2 to 5)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	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
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CST	Count Start 0: Count operation is stopped 1: Count operation is performed	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
18:16	MD[2:0]	Mode Select 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
23:19	—	These bits are read as 0. The write value should be 0.	R/W
26:24	TPCS[2:0]	Timer Prescaler Select 0 0 0: PCLKA/1 0 0 1: PCLKA/4 0 1 0: PCLKA/16 0 1 1: PCLKA/64 1 0 0: PCLKA/256 1 0 1: PCLKA/1024 Others: Setting prohibited	R/W
31:27	—	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, the external trigger, or the GTIOCNnA/GTIOCNnB input that are enabled by GTSSR for the starting counter source, occurs (n = 0 to 5)
- 1 is written by software directly.

[Clearing conditions]

- The GTSTP value where the channel number associated with the bit number is set to 1 with the GTPSR.CSTOP bit at 1
- The ELC event input, the external trigger, or the GTIOCNnA/GTIOCNnB input enabled by GTSSR as the counter stop source, occurs (n = 0 to 5)
- 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.

24.2.13 GTUDDTYC : General PWM Timer Count Direction and Duty Setting Register

Base address: GPT32n = 0x4005_5000 + 0x0100 × n (n = 0 to 1)
 GPT16m = 0x4005_5000 + 0x0100 × m (m = 2 to 5)

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	OBDT YR	OBDT YF	OBDTY[1:0]	—	—	—	—	OADT YR	OADT YF	OADTY[1:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	UD	Count Direction Setting 0: GTCNT counts down 1: GTCNT counts up	R/W
1	UDF	Forcible Count Direction Setting 0: Not forcibly set 1: Forcibly set	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
17:16	OADTY[1:0]	GTIOCN _A Output Duty Setting 0 0: GTIOCN _A pin duty depends on the compare match 0 1: GTIOCN _A pin duty depends on the compare match 1 0: GTIOCN _A pin duty 0% 1 1: GTIOCN _A pin duty 100%	R/W
18	OADTYF	Forcible GTIOCN _A Output Duty Setting 0: Not forcibly set 1: Forcibly set	R/W
19	OADTYR	GTIOCN _A 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
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	OBDTY[1:0]	GTIOCN _B Output Duty Setting 0 0: GTIOCN _B pin duty depends on the compare match 0 1: GTIOCN _B pin duty depends on the compare match 1 0: GTIOCN _B pin duty 0% 1 1: GTIOCN _B pin duty 100%	R/W
26	OBDTYF	Forcible GTIOCN _B Output Duty Setting 0: Not forcibly set 1: Forcibly set	R/W
27	OBDTYR	GTIOCN _B 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
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The GTUDDTYC sets the direction in which the GTCNT counts (up-counting or down-counting), and sets the duty of the GTIOCN_A/GTIOCN_B 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 this bit while counting stops, return this bit to 0 before counting starts.

Output duty

- In saw-wave mode.

When the OADTY/OBDTY value changes during up-counting, the duty is reflected at an overflow (GTCNT = GTPR). When the OADTY/OBDTY value is changed during down-counting, the duty is reflected at an underflow (GTCNT = 0).

When the OADTY/OBDTY value changes to 1 with the OADTYF/OBDTYF bit being 0 and while counting stops the output duty is not reflected at the starting counter operation. When the count direction is up, the output duty is reflected at an overflow (GTCNT = GTPR). When the count direction is down, the output duty is reflected at an underflow (GTCNT = 0).

When the OADTY/OBDTY value changes to 0 with the OADTYF/OBDTYF bit being 1 and while counting 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 to 1 with the OADTYF/OBDTYF bit being 0 and while counting stops, the output duty is not reflected at the starting counter operation. The output duty is reflected at an underflow.

When the OADTY/OBDTY value changes to 0 with the OADTYF/OBDTYF bit being 1 and while counting stops, the output duty is reflected at 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 (Forcible GTIOCm 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% or 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).

The GPT internally continues to perform compare match operation while 0% or 100% duty operation is running. When the OmDTYR bit is set to 1, the value of the compare match at cycle end is applied to the GTIOR.GTIOM [3:2] bits.

24.2.14 GTIOR : General PWM Timer I/O Control Register

Base address: $GPT32n = 0x4005\_5000 + 0x0100 \times n$ ($n = 0$ to 1)
 $GPT16m = 0x4005\_5000 + 0x0100 \times m$ ($m = 2$ to 5)

Offset address: $0x34$

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFC SB[1:0]		NFBE N	—	—	OBDF[1:0]		OBE	OBHLD	OBDFLT	—	GTIOB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NFC SA[1:0]		NFAEN	—	—	OADF[1:0]		OAE	OAHL D	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	Function	R/W
4:0	GTIOA[4:0]	GTIOCnA Pin Function Select See Table 24.4 .	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	OADFLT	GTIOCnA Pin Output Value Setting at the Count Stop 0: The GTIOCnA pin outputs low when counting stops 1: The GTIOCnA pin outputs high when counting stops	R/W
7	OAHL D	GTIOCnA Pin Output Setting at the Start/Stop Count 0: The GTIOCnA pin output level at the start or stop of counting depends on the register setting 1: The GTIOCnA pin output level is retained at the start or stop of counting	R/W
8	OAE	GTIOCnA Pin Output Enable 0: Output is disabled 1: Output is enabled	R/W
10:9	OADF[1:0]	GTIOCnA Pin Disable Value Setting 0 0: Output disable is prohibited 0 1: GTIOCnA pin is set to Hi-Z on output disable 1 0: GTIOCnA pin is set to 0 on output disable 1 1: GTIOCnA pin is set to 1 on output disable	R/W
12:11	—	These bits are read as 0. The write value should be 0.	R/W
13	NFAEN	Noise Filter A Enable 0: The noise filter for the GTIOCnA pin is disabled 1: The noise filter for the GTIOCnA pin is enabled	R/W
15:14	NFC SA[1:0]	Noise Filter A Sampling Clock Select 0 0: PCLKA/1 0 1: PCLKA/4 1 0: PCLKA/16 1 1: PCLKA/64	R/W
20:16	GTIOB[4:0]	GTIOCnB Pin Function Select See Table 24.4 .	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W
22	OBDFLT	GTIOCnB Pin Output Value Setting at the Count Stop 0: The GTIOCnB pin outputs low when counting stops 1: The GTIOCnB pin outputs high when counting stops	R/W
23	OBHLD	GTIOCnB Pin Output Setting at the Start/Stop Count 0: The GTIOCnB pin output level at the start/stop of counting depends on the register setting 1: The GTIOCnB pin output level is retained at the start/stop of counting	R/W
24	OBE	GTIOCnB Pin Output Enable 0: Output is disabled 1: Output is enabled	R/W

Bit	Symbol	Function	R/W
26:25	OBDF[1:0]	GTIOCnB Pin Disable Value Setting 0 0: Output disable is prohibited 0 1: GTIOCnB pin is set to Hi-Z on output disable 1 0: GTIOCnB pin is set to 0 on output disable 1 1: GTIOCnB pin is set to 1 on output disable	R/W
28:27	—	These bits are read as 0. The write value should be 0.	R/W
29	NFBEN	Noise Filter B Enable 0: The noise filter for the GTIOCnB pin is disabled 1: The noise filter for the GTIOCnB pin is enabled	R/W
31:30	NFCSB[1:0]	Noise Filter B Sampling Clock Select 0 0: PCLKA/1 0 1: PCLKA/4 1 0: PCLKA/16 1 1: PCLKA/64	R/W

The GTIOR sets the functions of the GTIOCnA and GTIOCnB pins.

GTIOA[4:0] bits (GTIOCnA Pin Function Select)

The GTIOA[4:0] bits select the GTIOCnA pin function. For details, see [Table 24.4](#).

OADFLT bit (GTIOCnA Pin Output Value Setting at the Count Stop)

The OADFLT bit sets whether the GTIOCnA pin outputs high or low when counting stops.

OAHLDBit (GTIOCnA Pin Output Setting at the Start/Stop Count)

The OAHLDBit specifies whether the GTIOCnA pin output level is retained or the level at the start or stop of counting depends on the register setting.

When the OAHLDBit 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 OAHLDBit is set to 1:

- The output is retained when counting starts or stops.

OAE bit (GTIOCnA Pin Output Enable)

The OAE bit disables or enables the GTIOCnA pin output.

When GTCCRA register is used as the input capture register (at least one bit in the GTICASR register is set to 1), the GTIOCnA pin does not output regardless of the OAE bit value.

OADF[1:0] bits (GTIOCnA Pin Disable Value Setting)

The OADF[1:0] bits select the output value of the GTIOCnA pin when an output disable request occurs.

NFAEN bit (Noise Filter A Enable)

The NFAEN bit disables or enables the noise filter for input from the GTIOCnA pin. Because changing the value of the bit might lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before doing so.

NFCSA[1:0] bits (Noise Filter A Sampling Clock Select)

The NFCSA[1:0] bits set the sampling interval for the noise filter of the GTIOCnA pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input capture function.

GTIOB[4:0] bits (GTIOCnB Pin Function Select)

The GTIOB[4:0] bits select the GTIOCnB pin function. For details, see [Table 24.4](#).

OBDFLT bit (GTIOCNB Pin Output Value Setting at the Count Stop)

The OBDFLT bit sets whether the GTIOCNB pin outputs high or low when counting stops.

OBHLD bit (GTIOCNB Pin Output Setting at the Start/Stop Count)

The OBHLD bit specifies whether the GTIOCNB pin output level is retained or the level at the start or stop of counting depends on the register setting.

When the OBHLD bit is set to 0:

- The value specified in bit [4] of the GTIOB[4:0] bits is output when counting starts
- The value specified in the OBDFLT bit is output when counting stops
- If the OBDFLT bit is modified while counting stops, the new value is immediately reflected in the output.

When the OBHLD bit is set to 1:

- The output is retained when counting starts or stops.

OBE bit (GTIOCNB Pin Output Enable)

The OBE bit disables or enables the GTIOCNB pin output.

When GTCCRB register is used as the input capture register (at least one bit in the GTICBSR register is set to 1), the GTIOCNB pin does not output regardless of the OBE bit value.

OBDF[1:0] bits (GTIOCNB Pin Disable Value Setting)

The OBDF[1:0] bits select the output value of the GTIOCNB pin, when an output disable request occurs.

NFBEN bit (Noise Filter B Enable)

The NFBEN bit disables or enables the noise filter for input from the GTIOCNB pin. Because changing the value of the bit might lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before doing so.

NFCSB[1:0] bits (Noise Filter B Sampling Clock Select)

The NFCSB[1:0] bits set the sampling interval for the noise filter of the GTIOCNB pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input capture function.

Table 24.4 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) or underflow (GTCNT changes from 0 to GTPR in down-counting). The GTCNT counter is cleared for saw waves and for the trough (GTCNT changes from 0 to 1) for triangle waves.

Note 2. When the timing of a cycle end and the timing of a GTCCRA/GTCCRB compare match are the same in a compare-match operation, the b3 and b2 settings are given priority in saw-wave PWM mode, and the b1 and b0 settings are given priority in any other mode.

Note 3. In event count operation where at least one bit in GTUPSR or GTDNSR is set to 1, the setting of b3 and b2 is ignored.

24.2.15 GTINTAD : General PWM Timer Interrupt Output Setting Register

Base address: GPT32n = 0x4005_5000 + 0x0100 × n (n = 0 to 1)
 GPT16m = 0x4005_5000 + 0x0100 × m (m = 2 to 5)

Offset address: 0x38

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

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

Bit	Symbol	Function	R/W
23:0	—	These bits are read as 0. The write value should be 0.	R/W
25:24	GRP[1:0]	Output Disable Source Select 0 0: Group A output disable request is selected 0 1: Group B output disable request is selected Others: Setting prohibited	R/W
28:26	—	These bits are read as 0. The write value should be 0.	R/W
29	GRPABH	Same Time Output Level High Disable Request Enable 0: Same time output level high disable request disabled 1: Same time output level high disable request enabled	R/W
30	GRPABL	Same Time Output Level Low Disable Request Enable 0: Same time output level low disable request disabled 1: Same time output level low disable request enabled	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

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 GTIOCnA pin and GTIOCnB pin output disable sources.

GTST.ODF shows the request of the output disable source group that is selected with the GRP[1:0] bits. Set the GRP[1:0] bits when both GTIOR.OAE and GTIOR.OBE bits are 0.

GRPABH bit (Same Time Output Level High Disable Request Enable)

The GRPABH bit enables or disables the output disable request when the GTIOCnA pin and GTIOCnB pin output 1 at the same time.

GRPABL bit (Same Time Output Level Low Disable Request Enable)

The GRPABL bit enables or disables the output disable request when the GTIOCnA pin and GTIOCnB pin output 0 at the same time.

24.2.16 GTST : General PWM Timer Status Register

Base address: GPT32n = 0x4005_5000 + 0x0100 × n (n = 0 to 1)
 GPT16m = 0x4005_5000 + 0x0100 × m (m = 2 to 5)

Offset address: 0x3C

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TUCF	—	—	—	—	—	—	—	TCFPU	TCFPO	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCFA	Input Capture/Compare Match Flag A 0: No input capture/compare match of GTCCRA is generated 1: An input capture/compare match of GTCCRA is generated	R/(W) ¹
1	TCFB	Input Capture/Compare Match Flag B 0: No input capture/compare match of GTCCRB is generated 1: An input capture/compare match of GTCCRB is generated	R/(W) ¹
2	TCFC	Input Compare Match Flag C 0: No compare match of GTCCRC is generated 1: A compare match of GTCCRC is generated	R/(W) ¹
3	TCFD	Input Compare Match Flag D 0: No compare match of GTCCRD is generated 1: A compare match of GTCCRD is generated	R/(W) ¹
4	TCFE	Input Compare Match Flag E 0: No compare match of GTCCRE is generated 1: A compare match of GTCCRE is generated	R/(W) ¹
5	TCFF	Input Compare Match Flag F 0: No compare match of GTCCRF is generated 1: A compare match of GTCCRF is generated	R/(W) ¹
6	TCFPO	Overflow Flag 0: No overflow (crest) occurred 1: An overflow (crest) occurred	R/(W) ¹
7	TCFPU	Underflow Flag 0: No underflow (trough) occurred 1: An underflow (trough) occurred	R/(W) ¹
14:8	—	These bits are read as 0. The write value should be 0.	R/W
15	TUCF	Count Direction Flag 0: GTCNT counter counts downward 1: GTCNT counter counts upward	R
23:16	—	These bits are read as 0. The write value should be 0.	R/W
24	ODF	Output Disable Flag 0: No output disable request is generated 1: An output disable request is generated	R
28:25	—	These bits are read as 0. The write value should be 0.	R/W
29	OABHF	Same Time Output Level High Flag 0: GTIOCnA pin and GTIOCnB pin do not output 1 at the same time 1: GTIOCnA pin and GTIOCnB pin output 1 at the same time	R
30	OABLF	Same Time Output Level Low Flag 0: GTIOCnA pin and GTIOCnB pin do not output 0 at the same time 1: GTIOCnA pin and GTIOCnB pin output 0 at the same time	R

Bit	Symbol	Function	R/W
31	—	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 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 to 0 in down-counting.

ODF flag (Output Disable Flag)

The ODF flag shows the request of the output disable source group that is selected in the GRP[1:0] bits.

When output is disabled, an output disable control is not released within the same cycle in which an output disable request is negated. It is released in the next cycle.

OABHF flag (Same Time Output Level High Flag)

The OABHF flag indicates that the GTIOCnA pin and GTIOCnB pin output 1 at the same time.

When the GTIOCnA or GTIOCnB pin outputs 0, this flag returns to 0. This flag is read only. Writing 0 to clear the flag is prohibited.

When an interrupt by the OABHF flag is enabled (GTINTAD.GRPABH = 1), the OABHF flag is output to POEG as an output disable request.

[Setting condition]

- The GTIOCnA and GTIOCnB pins output 1 at the same time when both OAE and OBE bits are set to 1.

[Clearing conditions]

- The GTIOCnA pin output value is different from the GTIOCnB pin output value when both OAE and OBE bits are set to 1
- The GTIOCnA and GTIOCnB pins output 0 at the same time when both OAE and OBE bits are set to 1
- Either the OAE bit or OBE bit is set to 0.

OABLF flag (Same Time Output Level Low Flag)

The OABLF flag indicates that the GTIOCnA and GTIOCnB pins output 0 at the same time.

When the GTIOCnA pin or GTIOCnB pin outputs 1, this flag returns to 0. This flag is read only. Writing 0 to clear the flag is prohibited.

When an interrupt by the OABLF flag is enabled (GTINTAD.GRPABL = 1), the OABLF flag is output to POEG as an output disable request.

[Setting condition]

- The GTIOCnA and GTIOCnB pins output 0 at the same time when both OAE and OBE bits are set to 1.

[Clearing conditions]

- The GTIOCnA pin output value is different from the GTIOCnB pin output value when both OAE and OBE bits are set to 1
- The GTIOCnA and GTIOCnB pins output 1 at the same time when both OAE and OBE bits are set to 1
- Either the OAE bit or the OBE bit is set to 0.

The compare-target signals to generate the OABHF/OABLF flag are the compare match outputs (PWM outputs) signals before they are masked by the output disable function. 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.

24.2.17 GTBER : General PWM Timer Buffer Enable Register

Base address: $GPT32n = 0x4005\_5000 + 0x0100 \times n$ ($n = 0$ to 1)
 $GPT16m = 0x4005\_5000 + 0x0100 \times m$ ($m = 2$ to 5)

Offset address: $0x40$

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BD1	BD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BD0	GTCCR Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
1	BD1	GTPR Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
17:16	CCRA[1:0]	GTCCRA Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTCCRA ↔ GTCCRC) Others: Double buffer operation (GTCCRA ↔ GTCCRC ↔ GTCCRD)	R/W
19:18	CCRB[1:0]	GTCCRB Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTCCRB ↔ GTCCRE) Others: Double buffer operation (GTCCRB ↔ GTCCRE ↔ GTCCRF)	R/W
21:20	PR[1:0]	GTPR Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTPBR → GTPR) Others: Setting prohibited	R/W
22	CCRSWT	GTCCRA and GTCCRB Forcible Buffer Operation Writing 1 to this bit forces a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after 1 is written. This bit is read as 0.	R/W
31:23	—	These bits are read as 0. The write value should be 0.	R/W

The GTBER register provides settings for the buffer operation and must be set while the GTCNT operation stops.

BD0 bit (GTCCR Buffer Operation Disable)

The BD0 bit disables the buffer operation using GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, and GTCCRF combined.

When GTDTCR.TDE is 1 and when BD0 is set to 0, GTCCRB does not perform buffer operation. The GTCCRB register is automatically set to a compare match value for negative-phase waveform with dead time.

BD1 bit (GTPR Buffer Operation Disable)

The BD1 bit disables the buffer operation using GTPR and GTPBR combined.

CCRA[1:0] bits (GTCCRA Buffer Operation)

The CCRA[1:0] bits set the buffer operation with GTCCRA, GTCCRC, and GTCCRD combined. When the buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.

The buffer operation mode is fixed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough).

CCRB[1:0] bits (GTCCRB Buffer Operation)

The CCRB[1:0] bits set the buffer operation using GTCCRB, GTCCRE, and GTCCRF combined. When the buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.

The buffer operation mode is fixed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough).

PR[1:0] bits (GTPR Buffer Operation)

The PR[1:0] bits set the buffer operation with 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 1 is written. This bit is read as 0, and is valid only when counting is stopped with a compare match operation specified.

24.2.18 GTCNT : General PWM Timer Counter

Base address: $GPT32n = 0x4005\_5000 + 0x0100 \times n$ ($n = 0$ to 1)
 $GPT16m = 0x4005\_5000 + 0x0100 \times m$ ($m = 2$ to 5)

Offset address: 0x48

Bit position: 31

0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	GTCNT is a 32-bit read/write counter for GPT32n ($n = 0$ to 1). For GPT16m ($m = 2$ to 5), GTCNT is a 16-bit register. GTCNT can only be written to after counting stops. For GPT16m ($m = 2$ to 5), the upper 16 bits for access in a 32-bit unit are always read as 0x0000, and writing to these bits is ignored. GTCNT must be set within the range of $0 \leq GTCNT \leq GTPR$.	R/W

24.2.19 GTCCRn : General PWM Timer Compare Capture Register n ($n = A$ to F)

Base address: $GPT32n = 0x4005\_5000 + 0x0100 \times n$ ($n = 0$ to 1)
 $GPT16m = 0x4005\_5000 + 0x0100 \times m$ ($m = 2$ to 5)

Offset address: 0x4C (GTCCRA)
 0x50 (GTCCRB)
 0x54 (GTCCRC)
 0x58 (GTCCRE)
 0x5C (GTCCRD)
 0x60 (GTCCRF)

Bit position: 31

0

Bit field:

Value after reset:¹ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	n/a	GTCCRn registers are read/write registers. The effective size of GTCCRn is the same as GTCNT (16- or 32-bit). If the effective size of GTCCRn is 16 bits, the upper 16 bits for access in a 32-bit unit are always read as 0x0000, and writing to these bits is ignored. GTCCRA and GTCCRB are registers used for both output compare and input capture. GTCCRC and GTCCRE are compare match registers 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).	R/W

Note 1. For GPT16m ($m = 2$ to 5), the value of the upper 16 bits after reset is 0x0000.

24.2.20 GTPR : General PWM Timer Cycle Setting Register

Base address: GPT32n = 0x4005_5000 + 0x0100 × n (n = 0 to 1)
GPT16m = 0x4005_5000 + 0x0100 × m (m = 2 to 5)

Offset address: 0x64

Bit position: 31 0

Bit field:

Value after reset: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	n/a	GTPR is a read/write register that sets the maximum count value of GTCNT. The effective size of GTPR is the same as GTCNT (16- or 32-bit). If the effective size of GTPR is 16 bits, the upper 16 bits for access in a 32-bit unit are always read as 0x0000, 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.	R/W

Note 1. For GPT16m (m = 2 to 5), the value of the upper 16 bits after reset is 0x0000.

24.2.21 GTPBR : General PWM Timer Cycle Setting Buffer Register

Base address: GPT32n = 0x4005_5000 + 0x0100 × n (n = 0 to 1)
GPT16m = 0x4005_5000 + 0x0100 × m (m = 2 to 5)

Offset address: 0x68

Bit position: 31 0

Bit field:

Value after reset: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	n/a	GTPBR is a read/write register that functions as a buffer register for GTPR. The effective size of GTPBR is the same as GTCNT (16- or 32-bit). If the effective size of GTPBR is 16 bits, the upper 16 bits for access in a 32-bit unit are always read as 0x0000, and writing to these bits is ignored.	R/W

Note 1. For GPT16m (m = 2 to 5), the value of the upper 16 bits after reset is 0x0000.

24.2.22 GTDTCR : General PWM Timer Dead Time Control Register

Base address: GPT32n = 0x4005_5000 + 0x0100 × n (n = 0 to 1)
GPT16m = 0x4005_5000 + 0x0100 × m (m = 2 to 5)

Offset address: 0x88

Bit position: 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	TDE	Negative-Phase Waveform Setting 0: GTCCRB is set without using GTDVU 1: GTDVU sets the compare match value for negative-phase waveform with automatic dead time in GTCCRB	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

GTDTCCR 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.

24.2.23 GTDVU : General PWM Timer Dead Time Value Register U

Base address: $GPT32n = 0x4005\_5000 + 0x0100 \times n$ ($n = 0$ to 1)
 $GPT16m = 0x4005\_5000 + 0x0100 \times m$ ($m = 2$ to 5)

Offset address: 0x8C

[illegible]

Bit	Symbol	Function	R/W
31:0	n/a	GTDVU is a read/write register that sets the dead time for generating PWM waveforms with dead time. The effective size of GTDVU is the same as GTCNT (16 or 32 bits). If the effective size of GTDVU is 16 bits, the upper 16 bits for access in a 32-bit unit are always read as 0x0000, and writing to these bits is ignored. Setting a GTDVU value greater than or equal to GTPR is prohibited. When using the automatic dead time setting function, do not set a value that makes a change point of the waveform exceeding the count period. 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.	R/W

Note 1. For GPT16m (m = 2 to 5), the value of the upper 16 bits after reset is 0x0000.

24.2.24 OPSCR : Output Phase Switching Control Register

Base address: GPT_OPS = 0x4005_5FF0

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCS[1:0]	NFEN	—	—	GODF	GRP[1:0]	—	—	ALIGN	RV	INV	N	P	FB		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	EN	—	W	V	U	—	WF	VF	UF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	UF	Input Phase Soft Setting	R/W
1	VF	These bits set the input phase from software settings. Setting these bits is valid when OPSCR.FB = 1.	R/W
2	WF		R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	U	Input U-Phase Monitor This bit monitors the state of the input phase. OPSCR.FB = 0 : External input that are synchronized by PCLKA OPSCR.FB = 1 : Software settings (UF)	R
5	V	Input V-Phase Monitor This bit monitors the state of the input phase. OPSCR.FB = 0 : External input that are synchronized by PCLKA OPSCR.FB = 1 : Software settings (VF)	R
6	W	Input W-Phase Monitor This bit monitors the state of the input phase. OPSCR.FB = 0 : External input that are synchronized by PCLKA OPSCR.FB = 1 : Software settings (WF)	R
7	—	This bit is read as 0. The write value should be 0.	R/W
8	EN	Enable-Phase Output Control 0: Do not output (Hi-Z external pin) 1: Output ^{*1}	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	FB	External Feedback Signal Enable This bit selects the input phase from software settings and external input. 0: Select the external input 1: Select the soft setting (OPSCR.UF, VF, WF)	R/W
17	P	Positive-Phase Output (P) Control 0: Level signal output 1: PWM signal output (PWM of GPT162)	R/W
18	N	Negative-Phase Output (N) Control 0: Level signal output 1: PWM signal output (PWM of GPT162)	R/W
19	INV	Invert-Phase Output Control 0: Positive logic (active-high) output 1: Negative logic (active-low) output	R/W
20	RV	Output Phase Rotation Direction Reversal Control 0: Positive rotation 1: Reverse rotation	R/W
21	ALIGN	Input Phase Alignment 0: Input phase aligned to PCLKA 1: Input phase aligned to PWM	R/W

Bit	Symbol	Function	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
25:24	GRP[1:0]	Output Disabled Source Selection 0 0: Select group A output disable source 0 1: Select group B output disable source Others: Setting prohibited	R/W
26	GODF	Group Output Disable Function 0: This bit function is ignored 1: Group disable clears the OPSCR.EN bit*1	R/W
28:27	—	These bits are read as 0. The write value should be 0.	R/W
29	NFEN	External Input Noise Filter Enable 0: Do not use a noise filter on the external input 1: Use a noise filter on the external input	R/W
31:30	NFCS[1:0]	External Input Noise Filter Clock Selection Noise filter sampling clock setting of the external input. 0 0: PCLKA/1 0 1: PCLKA/4 1 0: PCLKA/16 1 1: PCLKA/64	R/W

Note 1. When OPSCR.GODF = 1 and the signal value selected by the OPSCR.GRP[1:0] bit is high, the OPSCR.EN bit is set to 0.

The OPSCR register sets the output of the signal waveform required for brushless DC motor control.

UF , VF , WF bits (Input Phase Soft Setting)

The UF , VF , WF bits set the input phase from the software settings. When OPSCR.FB bit is 1, these bits are valid. The set value of the UF /VF /WF takes the place of the U/V/W external input.

U, V, W bits (Input Phase Monitor)

When the OPSCR.FB bit is 0, external inputs that are synchronized by PCLKA 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[1:0], OPSCR.GODF, OPSCR.NFEN, OPSCR.NFCS. Then, set the bit to 1. Also when OPSCR.GODF is 1 and the signal value selected in the OPSCR.GRP[1:0] bit is high, the OPSCR.EN bit is set to 0.

FB bit (External Feedback Signal Enable)

The FB bit selects the input phase from the software settings (OPSCR.UF, VF, WF) and external input such as a Hall element.

P bit (Positive-Phase Output (P) Control)

The P bit selects one of the level signal output (PWM of GPT162) or PWM signal output for the positive-phase output (GTOUUP pin, GTOVUP pin, GTOWUP pin).

N bit (Negative-Phase Output (N) Control)

The N bit selects one of the level signal output (PWM of GPT162) or PWM signal output for the negative-phase output (GTOULO pin, GTOVLO pin, GTOWLO pin).

INV bit (Invert-Phase Output Control)

The INV bit selects one of the positive logic (active-high) output or negative logic (active-low) output for the output phase.

RV bit (Output Phase Rotation Direction Reversal Control)

The RV bit reverses the direction of rotation of the motor by inverting the input phase.

ALIGN bit (Input Phase Alignment)

The ALIGN bit selects the PCLKA 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 PCLKA.

Note: When PWM output is selected (OPSCR.P/N is 1) and the PCLKA input phase is aligned, the PWM pulse can be short-pulsed.

Note: When OPSCR.ALIGN bit is 1, input phase is aligned with PWM output.

GRP[1:0] bit (Output Disabled Source Selection)

The GRP[1:0] bit selects the output disable source.

GODF bit (Group Output Disable Function)

When OPSCR.GODF is 1 and the signal value selected by the OPSCR.GRP[1:0] bit is high, the OPSCR.EN bit is set to 0.

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 for the external input is not used.

Note: When this bit is switched because of an unintentional internal edge, set the OPSCR.EN bit to 0.

NFCS[1:0] bits (External Input Noise Filter Clock Selection)

The NFCS[1:0] bits select the clock for the external input noise filter. When the OPSCR.NFEN bit is 1, noise filter sampling clock setting of the external input is enabled.

1. Set the NFCS[1:0].
2. Wait for 2 cycles.
3. Set the OPSCR.EN bit to 1.

24.3 Operation

24.3.1 Basic Operation

Each channel has a 32-bit and 16-bit timer that performs a periodic count operation using the count clock and hardware sources. The count function provides both up-counting and down-counting. The GTPR controls the count cycle.

When the GTCNT counter value matches the value in GTCCRA or GTCCRB, the output from the associated GTIOCnA or GTIOCnB can be changed (n = 0 to 5). 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.

24.3.1.1 Counter operation

(1) Counter start and stop

The counter of each channel starts the count operation when GTCR.CST is set to 1. The GTCR.CST bit value is changed by the following sources:

- Writing to GTCR register
- Writing 1 to the bit in GTSTR associated with the GPT channel number when the GTSSR.CSTRT bit 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 0x00000000. 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 0x00000000.

Figure 24.3 shows an example of a periodic count operation in up-counting by the count clock.

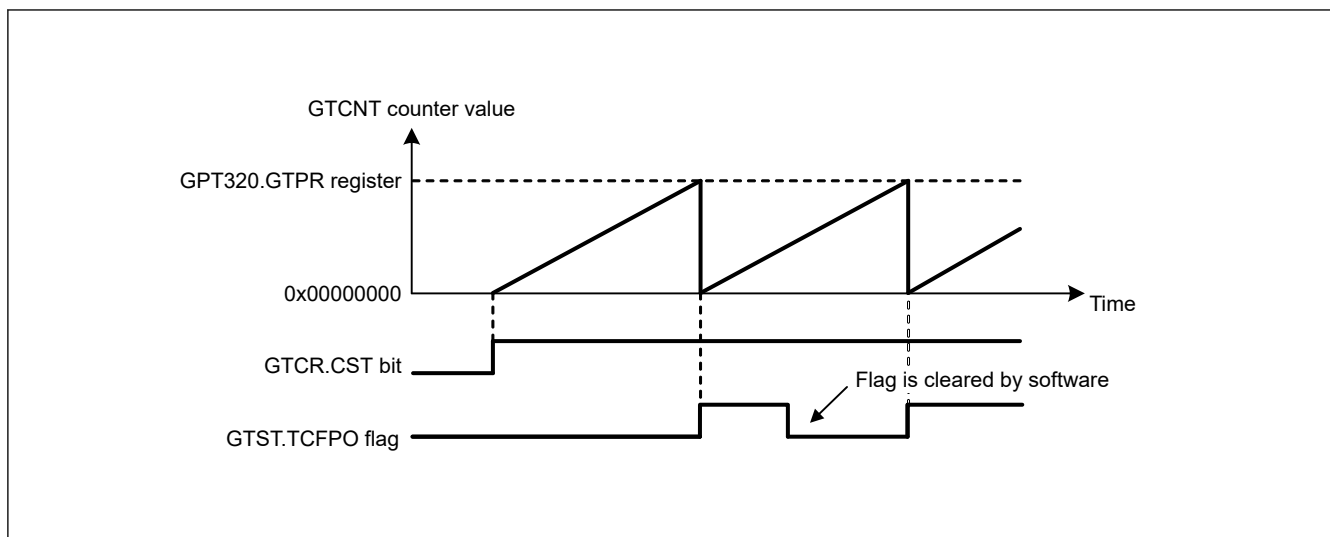


Figure 24.3 Example of periodic count operation in up-counting by the count clock

Table 24.5 shows an example for setting periodic count operation in up-counting.

Table 24.5 Example for setting a periodic count operation in up-counting by the count clock

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.3, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction with the GTUDDTYC register. In Figure 24.3, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 24.3, 0x00000000 is set.
6	Start count operation	Set GTCR.CST to 1 to start count operation.

(3) Periodic count operation in down-counting by count clock

The GTCNT counter in each channel can perform down-counting by setting GTUDDTYC.UD with GTUPSR and GTDNSR registers set to 0x00000000. When GTCNT changes from 0 to the GTPR value (underflow), GTST.TCFPU is set to 1. After the GTCNT counter underflows, down-counting resumes from the GTPR value.

Figure 24.4 shows an example of periodic count operation in down-counting by the count clock.

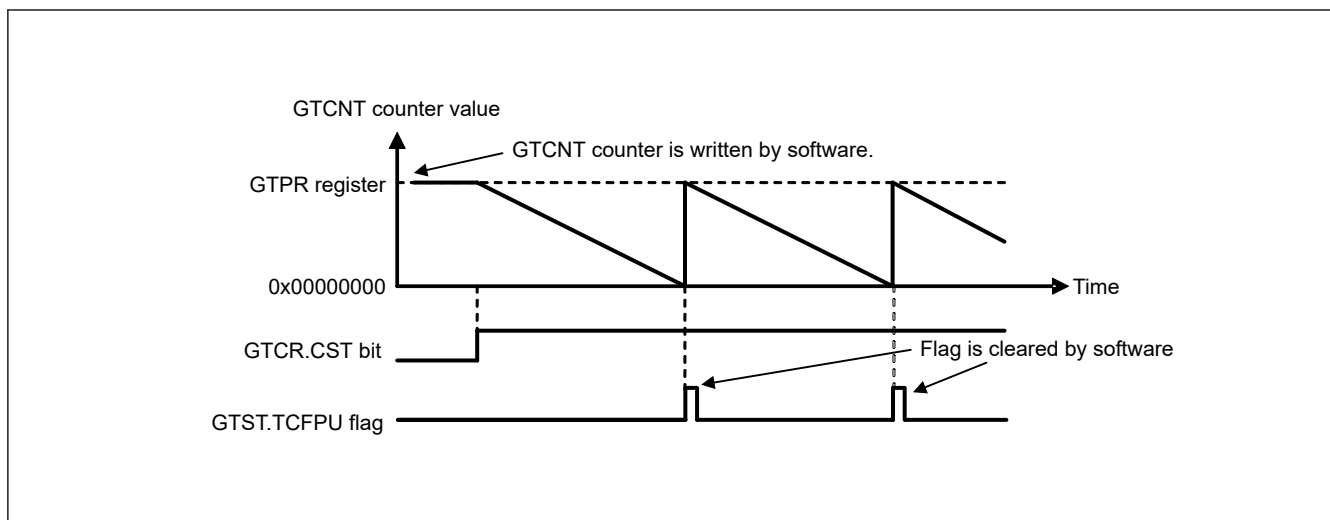


Figure 24.4 Example of periodic count operation in down-counting by the count clock

Table 24.6 shows an example for setting periodic count operation in down-counting by the count clock.

Table 24.6 Example for setting periodic count operation in down-counting by count clock

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.4, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction with the GTUDDTYC register. In Figure 24.4, after 10b is set in GTUDDTYC[1:0], 00b is set in GTUDDTYC[1:0] (down-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 24.4, the GTPR value is set.
6	Start count operation	Set GTCR.CST to 1 to start count operation.

(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.

When GTCR.CST bit is set to 1 to count up using hardware sources, the count operation is enabled. After GTCR.CST is set to 1, the counter cannot count up for 1 clock cycle as specified in GTCR.TPCS[2:0] because the count operation is synchronized by the count clock selected in GTCR.TPCS[2:0]. Set GTCR.TPCS[2:0] to 000b to count up with a 1 PCLKA delay after GTCR.CST is set to 1.

Figure 24.5 shows an example of an event count operation in up-counting by a hardware resource (rising edge of GTETRGA pin).

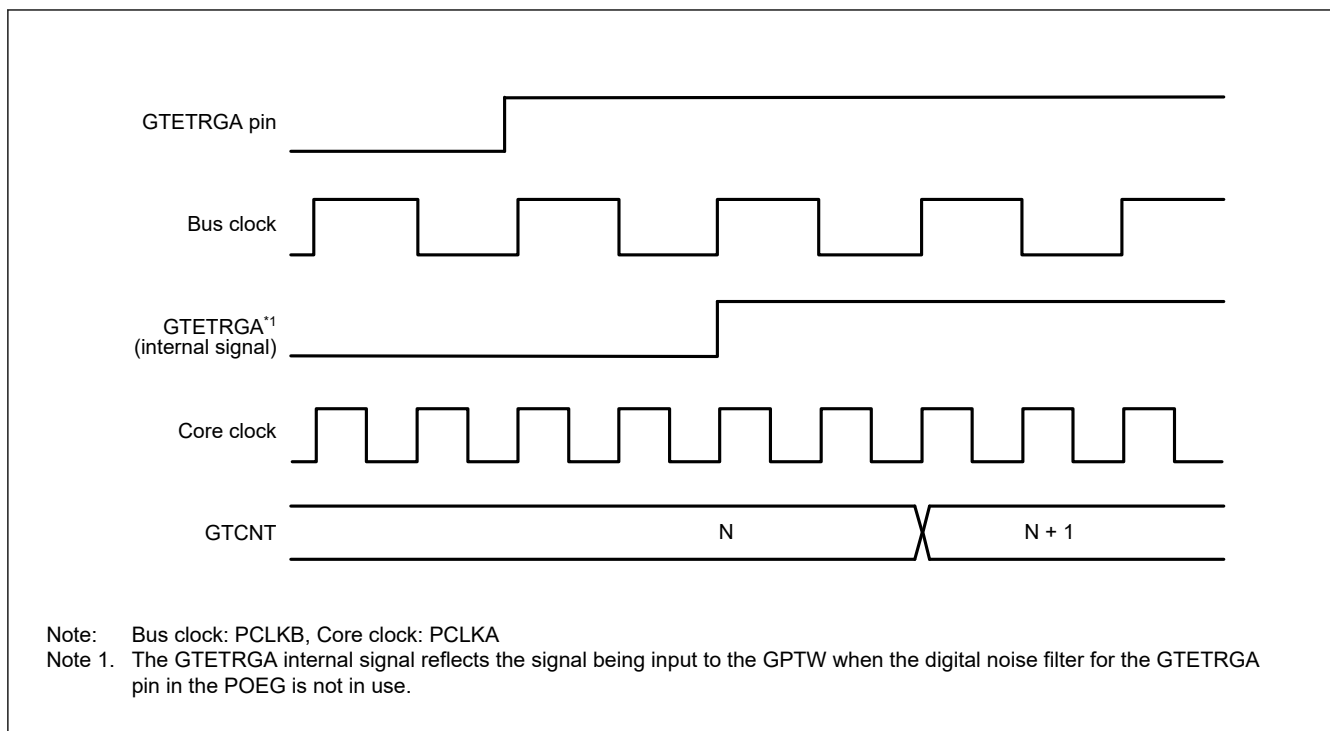


Figure 24.5 Example of event count operation in up-counting using hardware sources

Table 24.7 shows an example for setting event count operation in up-counting by the count clock.

Table 24.7 Example for setting an event count operation in up-counting using hardware sources

No.	Step Name	Description
1	Set count source	Select the counting-up source with the GTUPSR register.
2	Set cycle	Set the cycle in GTPR.
3	Set initial value for counter	Set the initial value in the GTCNT counter.
4	Start count operation	Set GTCR.CST to 1 to start count operation.

(5) Event count operation in down-counting using hardware sources

The GTCNT counter in each channel can perform down-counting using hardware sources set in the GTDNSR.

When GTDNSR is set to enable, the count clock selected in GTCR.TPCS[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 when down-counting using hardware sources is the same as when down-counting by the count clock.

When GTCR.CST bit is set to 1 to count down using hardware sources, the count operation is enabled. After GTCR.CST is set to 1, the counter cannot count down for 1 clock cycle as specified in GTCR.TPCS[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 PCLKA delay after GTCR.CST is set to 1.

Figure 24.6 shows an example of a event count operation in down-counting by a hardware resource (rising edge of GTETRGA pin).

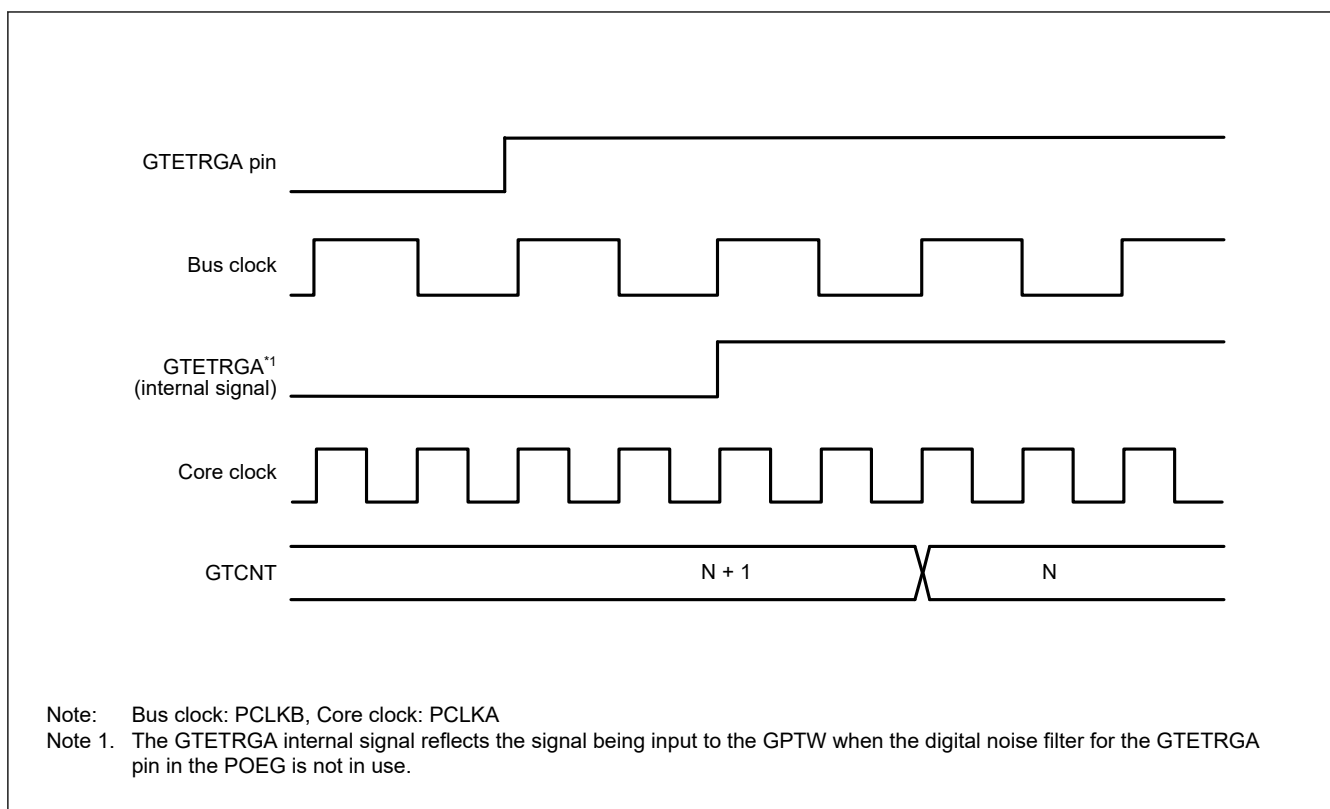


Figure 24.6 Example of event count operation in down-counting using hardware sources

Table 24.8 shows an example for setting a periodic count operation in down-counting using a hardware resource.

Table 24.8 Example for setting an event count operation in down-counting using hardware sources

No.	Step Name	Description
1	Set count source	Select the counting-down source with the GTDNSR register.
2	Set cycle	Set the cycle in GTPR.
3	Set initial value for counter	Set the initial value in the GTCNT counter.
4	Start count operation	Set GTCR.CST to 1 to start count operation.

(6) Counter clear operation

The counter of each channel is cleared by following sources:

- Writing 0 to GTCNT register
- Writing 1 to the bit in GTCLR associated with the GPT channel number when the GTCSR.CCLR bit set to 1
- The hardware source selected in GTCSR register.

Writing to the GTCNT register is prohibited during count operation. 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 and when clearing by hardware sources are performed.

When not in saw waves mode and down-counting, the GTCNT register is set to 0 when writing 1 to the GTCLR register and when clearing by hardware sources are performed.

In event count operation when at least 1 bit in 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 PCLKA. If other settings are used, clear is synchronized with the counter clock selected in GTCR.TPCS[2:0].

24.3.1.2 Waveform output by compare match

Compare match means that the GTCNT counter value matches the value of GTCCRA or GTCCRB. When a compare match occurs, the compare match flag is generated synchronously with the count clock, including the event count. At the same time, the GPT can output low, high, or toggled output from the associated GTIOCnA or GTIOCnB output pin (n = 0 to 5). In addition, the GTIOCnA or GTIOCnB pin output can be low, high, or toggled at the cycle end which is determined by GTPR (n = 0 to 5).

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 24.7 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 GTIOC0B pin by a GPT320.GTCCRB compare match. The pin level does not change when the specified level and pin level match.

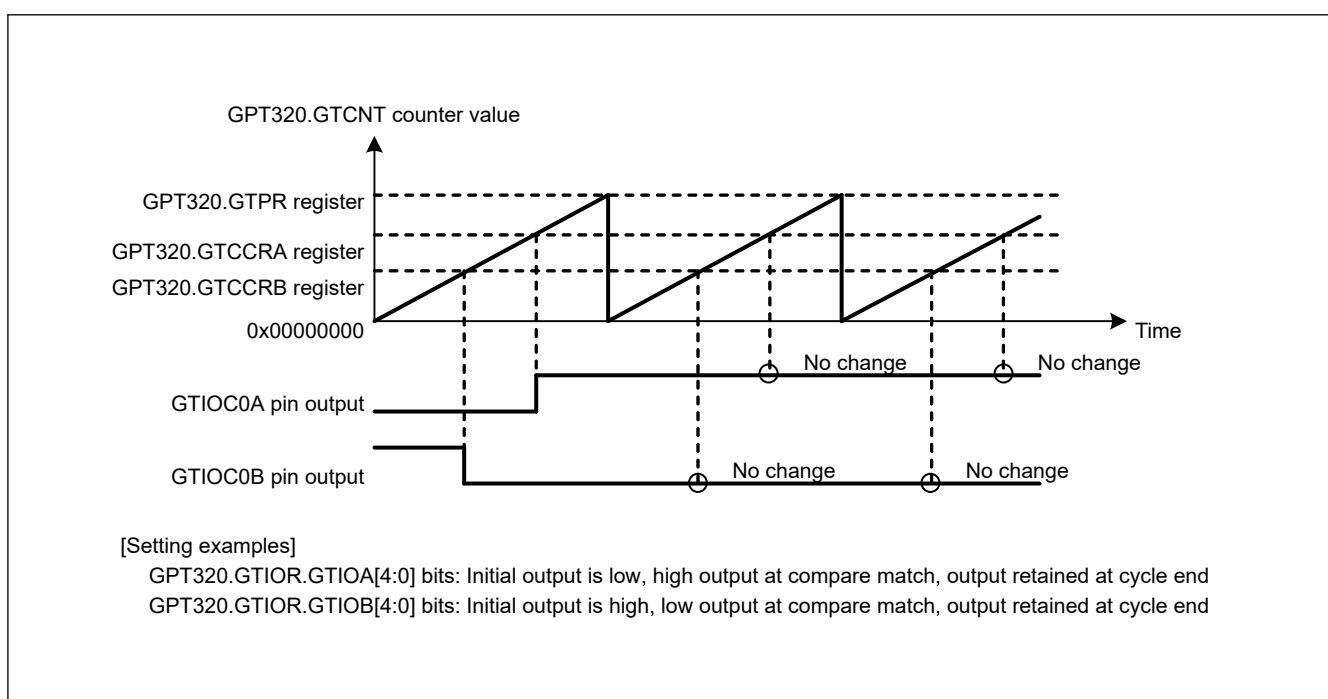


Figure 24.7 Example of low output and high output operation

Table 24.9 shows an example for setting low output and high output operation.

Table 24.9 Example for setting low output and high output operation (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.7, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 24.7, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter.

Table 24.9 Example for setting low output and high output operation (2 of 2)

No.	Step Name	Description
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure Figure 24.7 , GTIOA[4:0] = 00010b, GTIOB[4:0] = 10001b.
7	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with OAE and OBE in GTIOR.
8	Set compare match value	Set compare match values in the GTCCRA and GTCCRB registers.
9	Start count operation	Set GTCR.CST to 1 to start count operation.

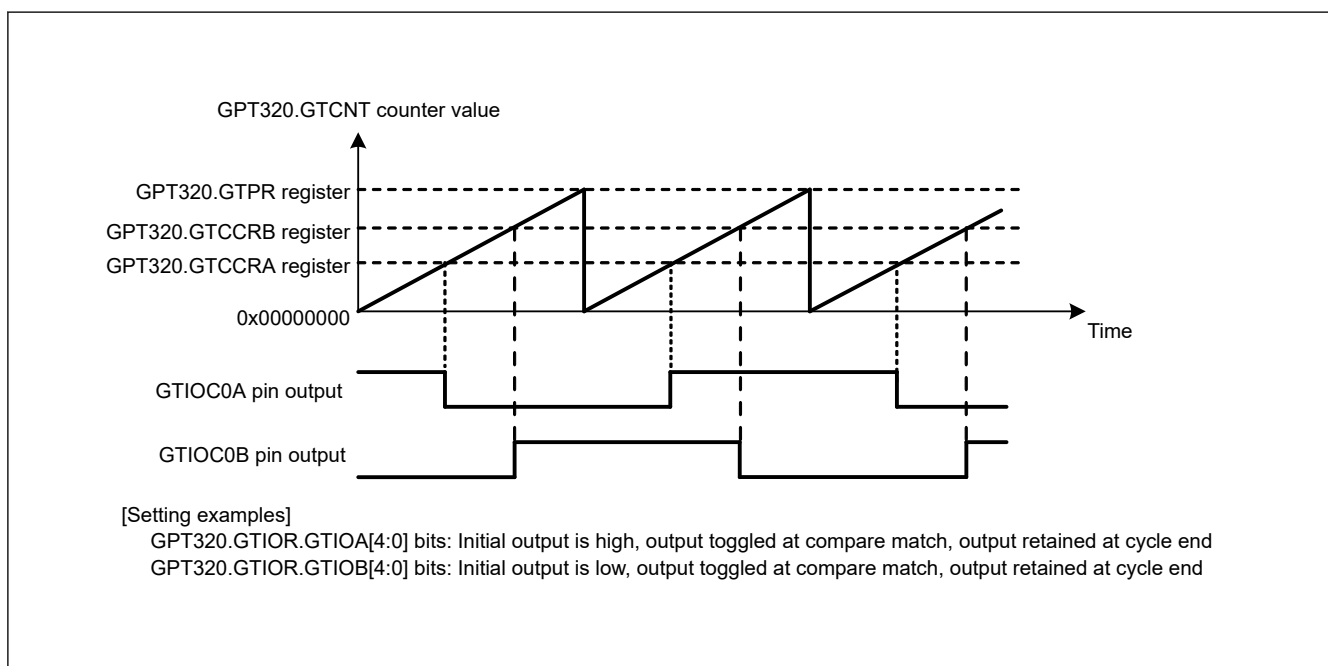
Note: n: 0 to 5
m: A, B

(2) Toggled output

[Figure 24.8](#) and [Figure 24.9](#) show examples of toggled output operation by compare matches of GTCCRA and GTCCRB.

In [Figure 24.8](#), 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 24.9](#), the GPT320.GTCNT counter performs up-counting, and settings are made so that a GPT320.GTCCRA compare match toggles the GTIOC0A pin output level and a cycle end toggles the GTIOC0B pin output level.

**Figure 24.8 Example of toggled output operation (1)**

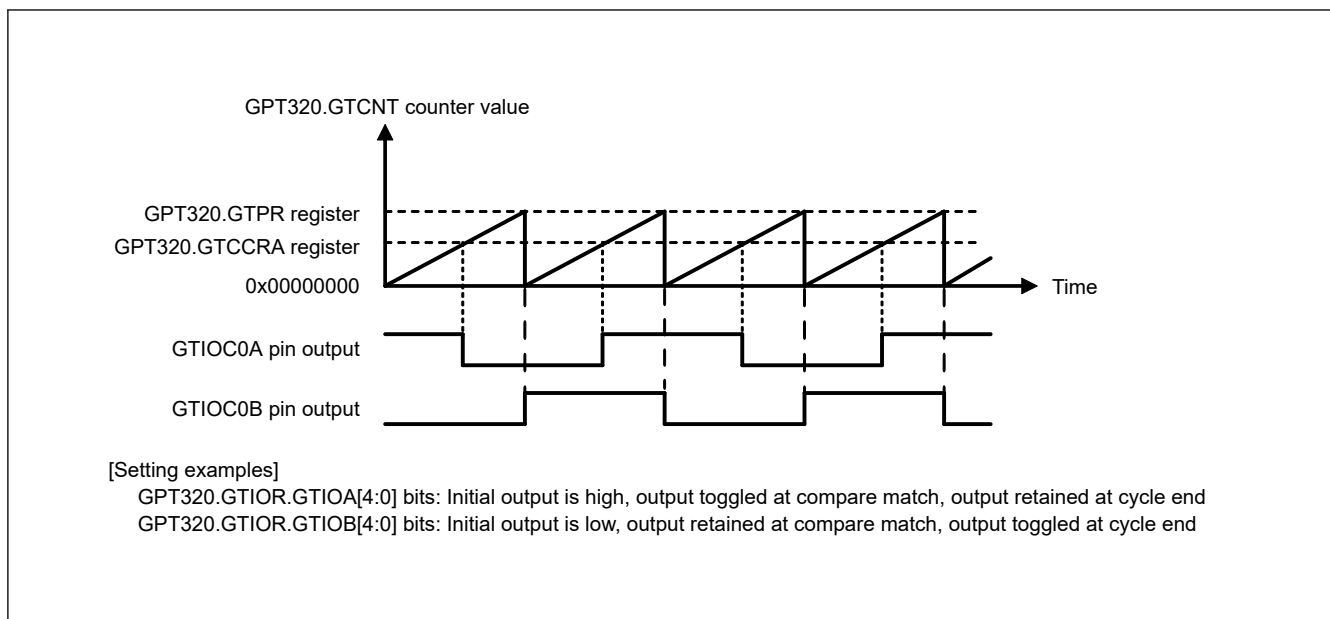
**Figure 24.9 Example of toggled output operation (2)**

Figure 24.15 shows an example for setting toggled output operation.

Table 24.10 Example for setting toggled output operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.8 and Figure 24.9, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 24.8 and Figure 24.9, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 24.8, GTIOA[4:0] = 10011b, GTIOB[4:0] = 00011b in Figure 24.9, GTIOA[4:0] = 10011b, GTIOB[4:0] = 01100b.
7	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with OAE and OBE in GTIOR.
8	Set compare match value	Set compare match values in the GTCCRA and GTCCRB registers.
9	Start count operation	Set GTCR.CST to 1 to start count operation.

Note: n: 0 to 5
 m: A, B

24.3.1.3 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 24.10 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 GTICCRB at both edges of the GTIOC0A input pin and to GTICCRB on the rising edge of the GTIOC0B input pin.

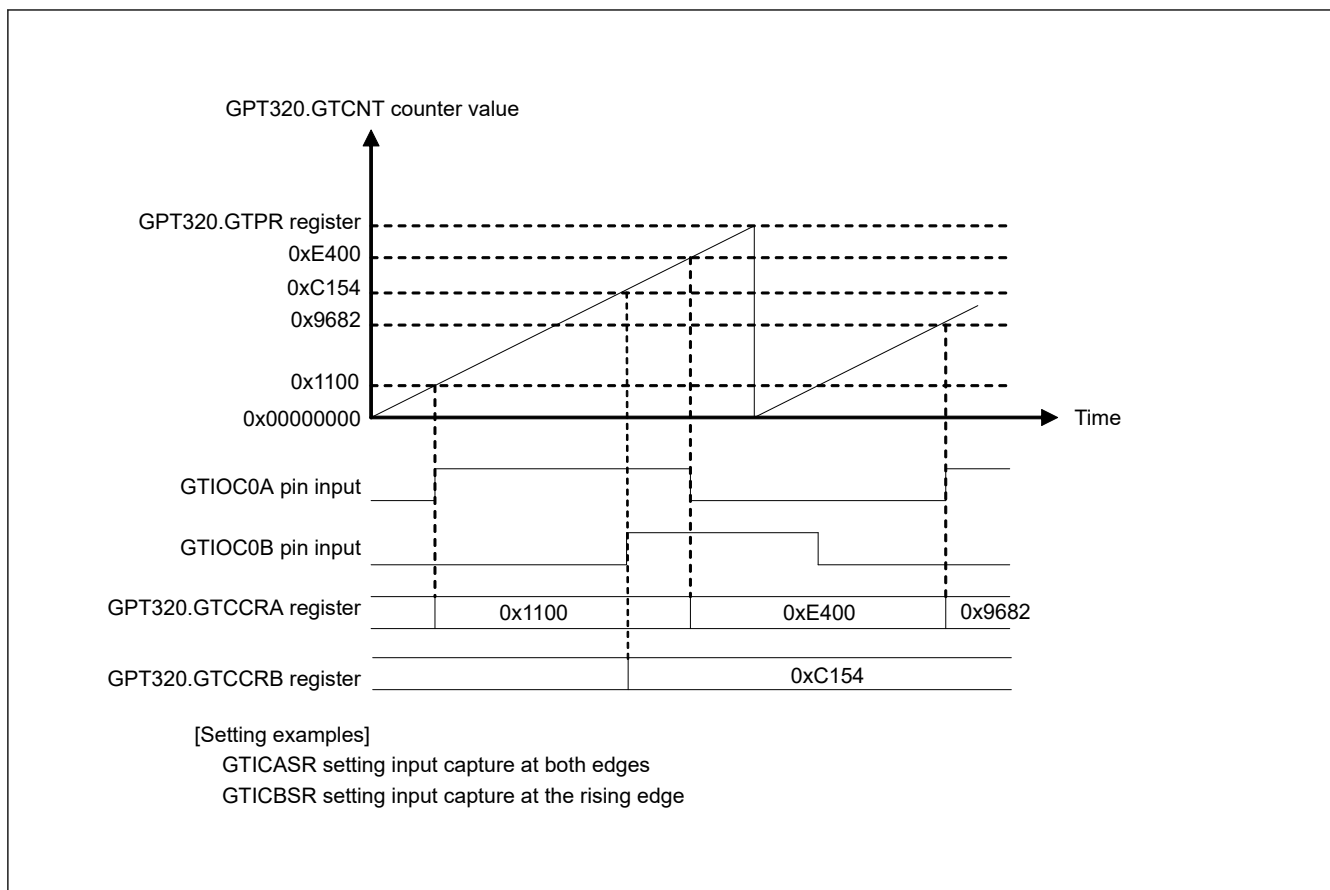


Figure 24.10 Example of input capture operation

Table 24.11 shows an example for setting an input capture operation with count operation by the count clock.

Table 24.11 Example for setting input capture operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.10, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 24.10, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Select input capture source	Select the input capture source in GTICASR and GTICBSR. In Figure 24.10, GTICASR = 0x00000F00, GTICBSR = 0x00003000.
7	Start count operation	Set GTCR.CST to 1 to start count operation.

24.3.2 Buffer Operation

The following buffer operations can be set with GTBER:

- GTPR and GTPBR
- GTCCRA, GTCCRC, and GTCCRD
- GTCCRB, GTCCRE, and GTCCRF.

24.3.2.1 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 CTCSR register)
- Clear by software (when GTCSR.CCLR bit is 1 and GTCLR.CTCCRn bit is set to 1, n = 0 to 5).

To set GTPR to function as a buffer, set the GTBER.PR bit to 1. To set GTPR not to function as a buffer, set the GTBER.PR bit to 0.

Figure 24.11 to Figure 24.13 show examples of GTPR buffer operation and Table 24.12 shows an example for setting GTPR buffer operation.

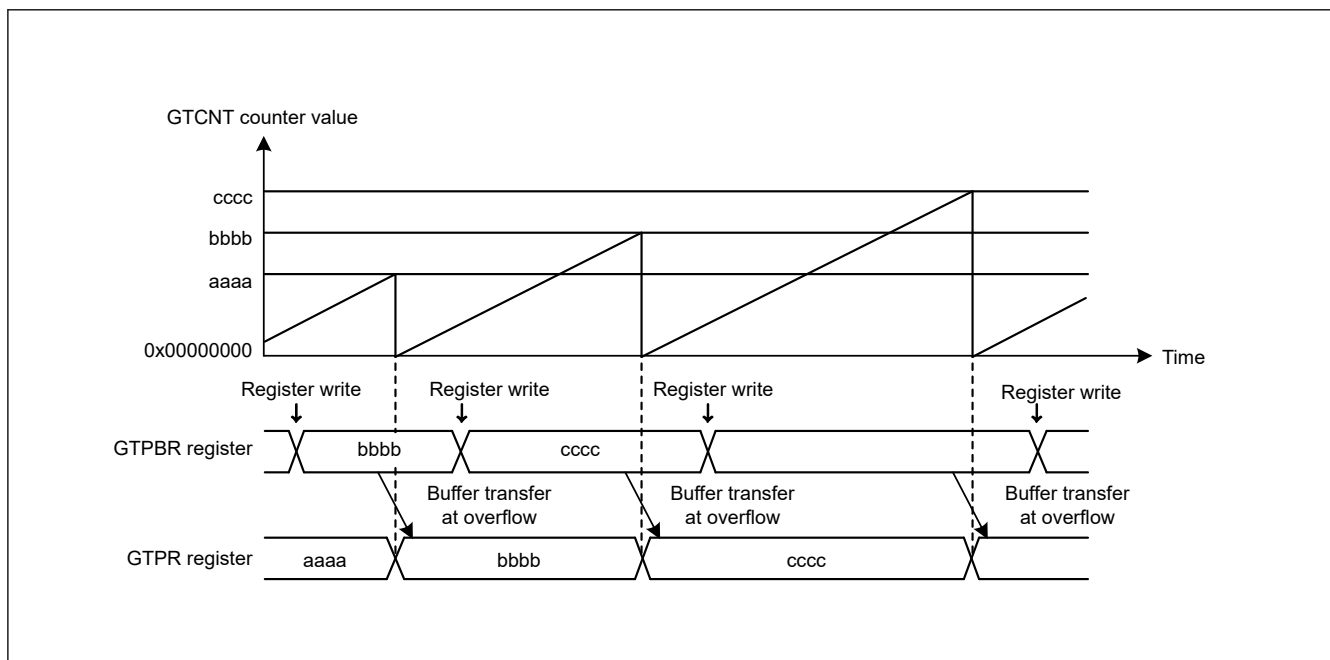


Figure 24.11 Example of GTPR buffer operation with saw waves in up-counting

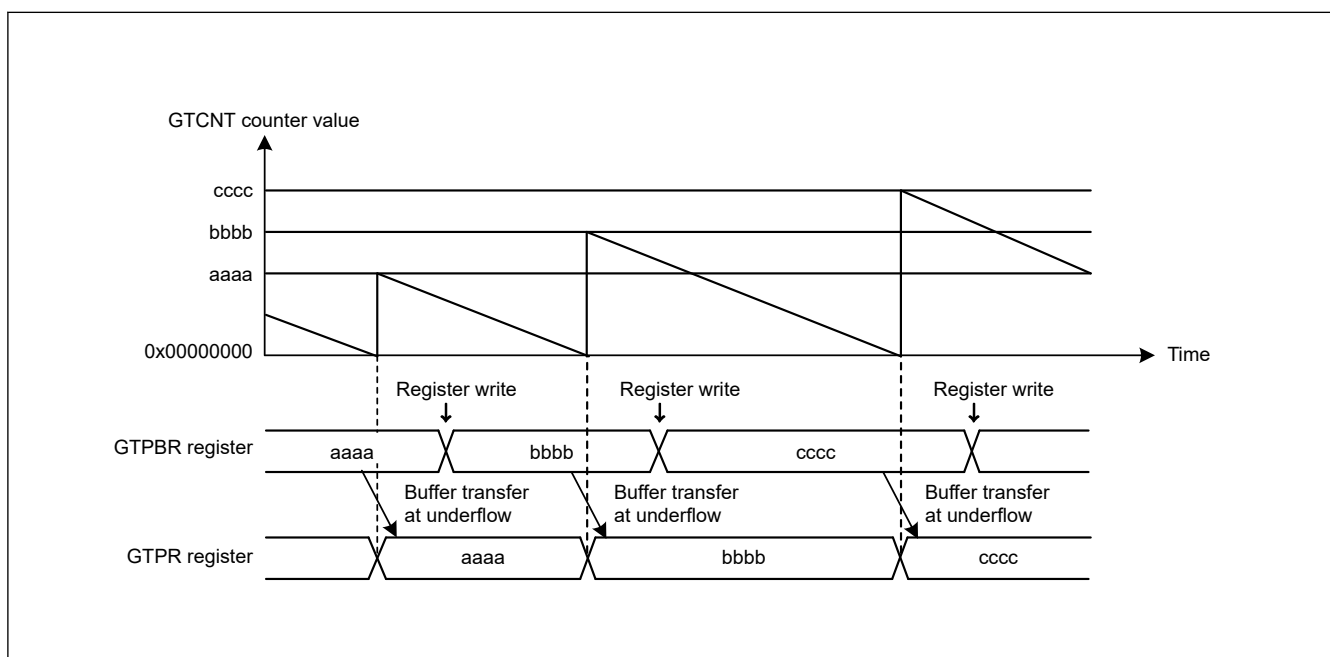


Figure 24.12 Example of GTPR buffer operation with saw waves in down-counting

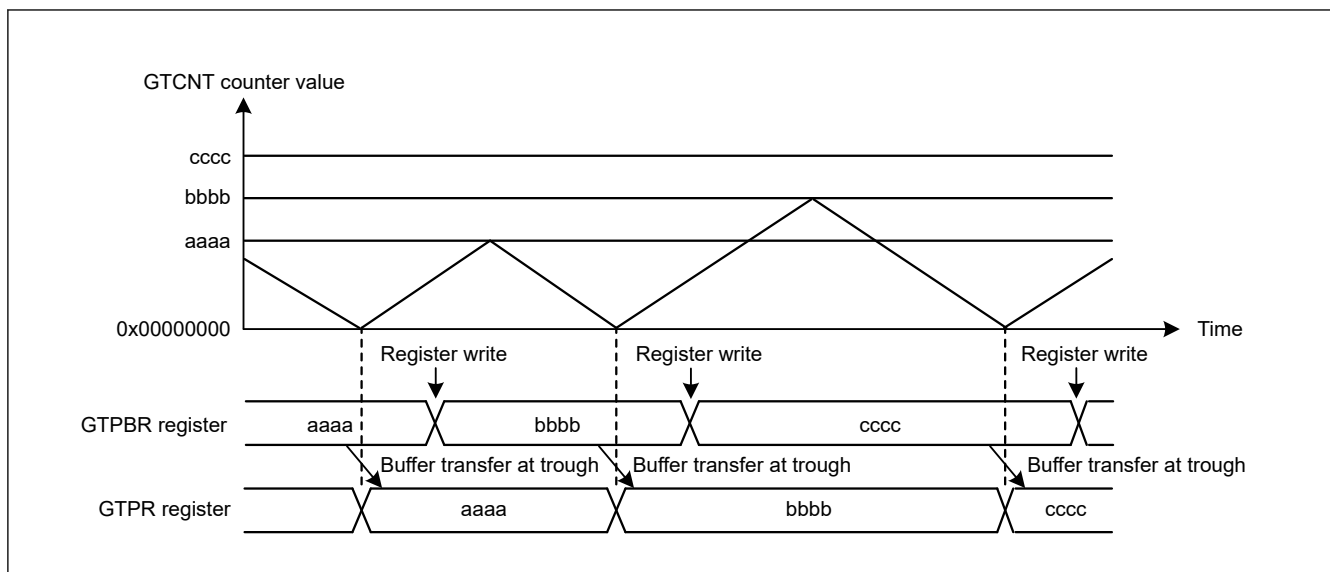


Figure 24.13 Example of GTPR buffer operation with triangle waves

Table 24.12 Example for setting GTPR buffer operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.11 and Figure 24.12, 000b (saw-wave PWM mode) is set, and in Figure 24.13, 100b (triangle-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 24.11, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting). In Figure 24.12, after 10b is set in GTUDDTYC[1:0], 00b is set in GTUDDTYC[1:0] (down-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set buffer operation	Set buffer operation with GTBER.PR[1:0]. In Figure 24.11, Figure 24.12, and Figure 24.13, PR[1:0] = 01b.
7	Set buffer value	For buffer operation, set a value in one cycle after the current cycle in GTPBR.
8	Start count operation	Set GTCR.CST to 1 to start count operation.
9	Set buffer value for each cycle	For buffer operation, set a value in one cycle after the current cycle in GTPBR.

24.3.2.2 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 24.3.2.1. 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 transfer are performed forcibly in saw-wave mode, in event count operation and in triangle-wave mode.

Additionally buffer transfer from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B are performed in saw-wave 1 shot pulse mode or triangle-wave PWM mode 3.

[Figure 24.14](#) to [Figure 24.16](#) show examples of GTCCRA and GTCCRB buffer operation and [Table 24.13](#) shows an example for setting GTCCRA and GTCCRB buffer operation.

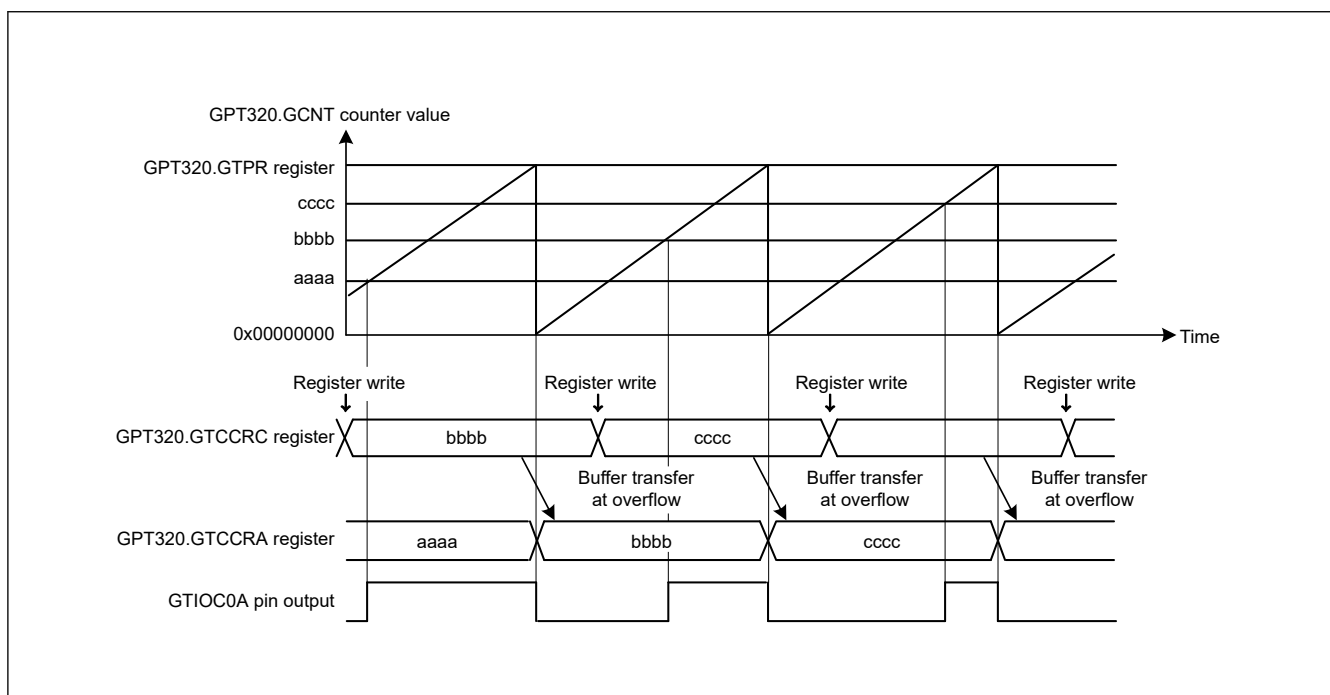


Figure 24.14 Example of GTCCRA and GTCCRB buffer operation with output compare, saw waves in up-counting, high output at GTCCRA compare match, and low output at cycle end

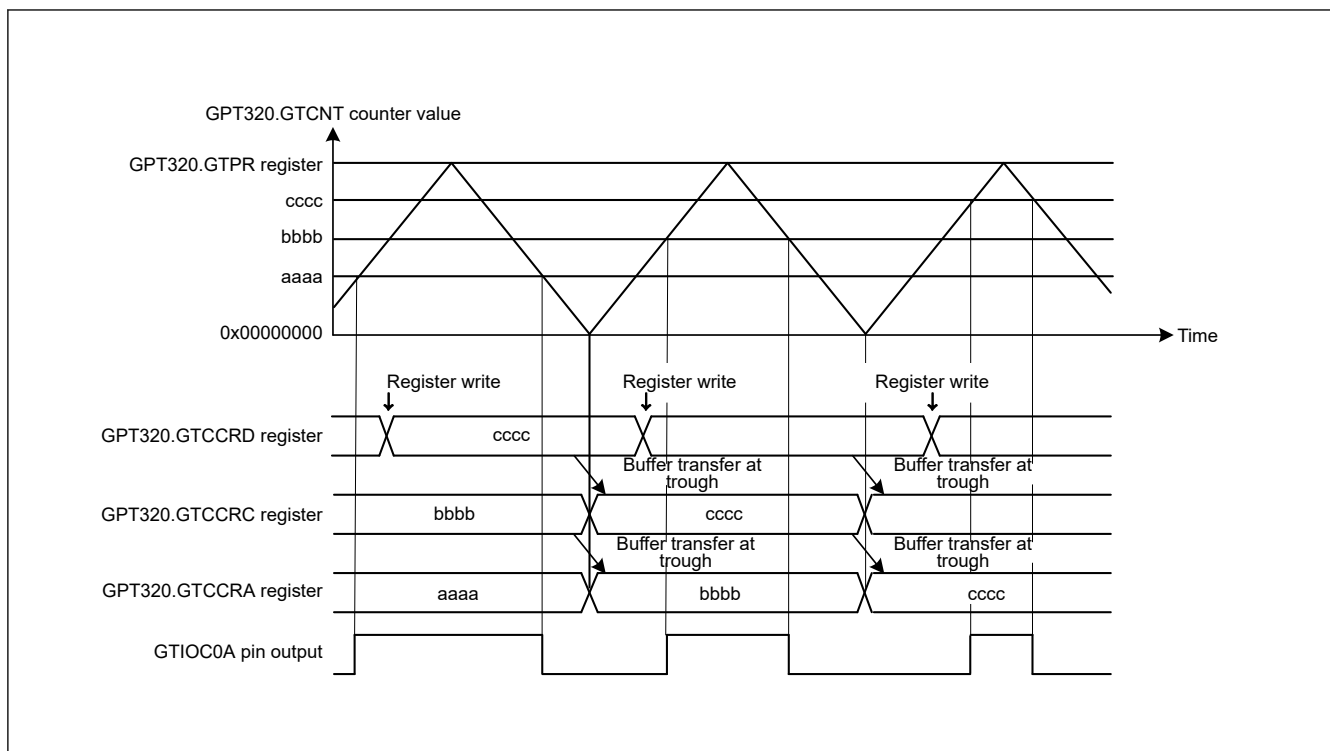


Figure 24.15 Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at trough, output toggled at GTCCRA compare match, and output retained at cycle end

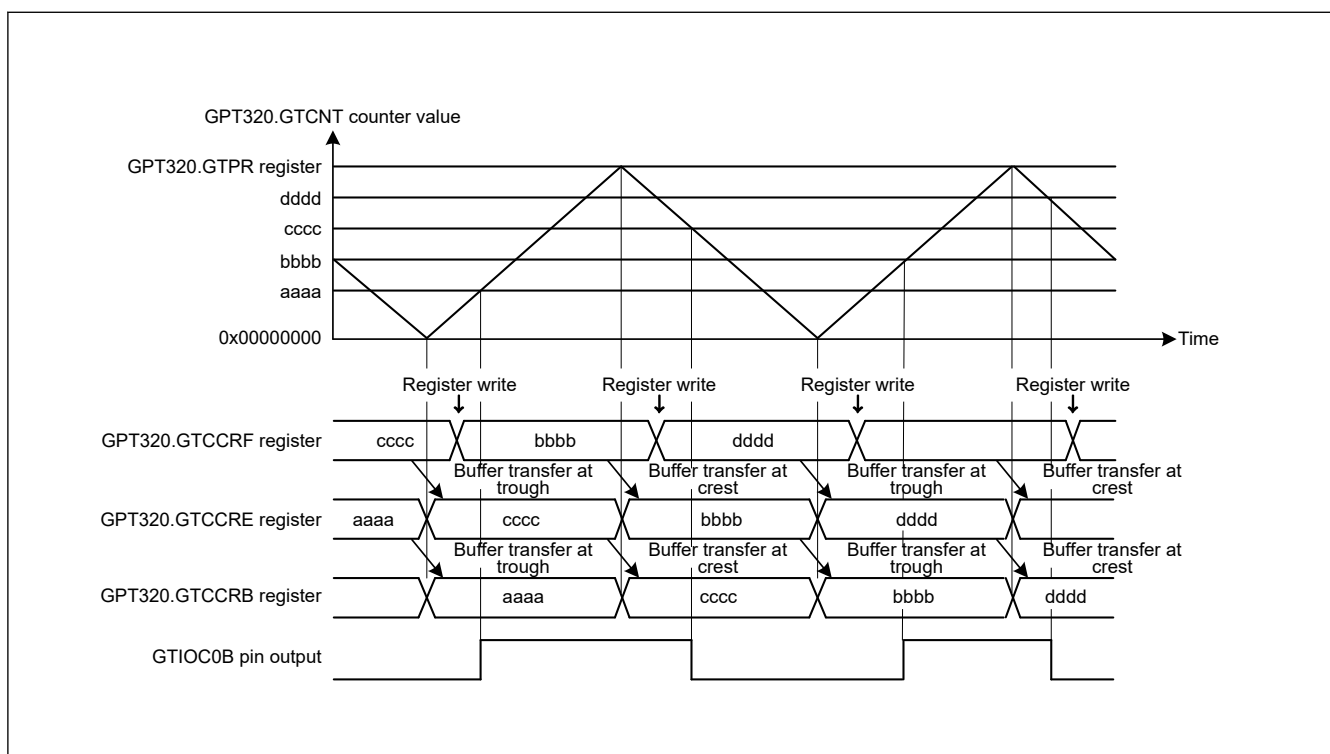


Figure 24.16 Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at both troughs and crests, output toggled at GTCCRB compare match, and output retained at cycle end

Table 24.13 Example for setting GTCCRA and GTCCRB buffer operation for output compare

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.14 , 000b (saw-wave PWM mode) is set, in Figure 24.15 , 100b (triangle-wave PWM mode 1) is set, and in Figure 24.16 , 101b (triangle-wave PWM mode 2) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 24.14 , after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCNm pin function	Set the GTIOCNm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 24.14 , GTIOA[4:0] = 00110b, in Figure 24.15 , GTIOA[4:0] = 00011b, and in Figure 24.16 , GTIOB[4:0] = 00011b.
7	Enable GTIOCNm pin output	Set to enable the GTIOCNm pin output with OAE and OBE in GTIOR.
8	Set buffer operation	Set buffer operation with CCRA and CCRB in GTBER. In Figure 24.14 , CCRA[1:0] = 01b, in Figure 24.15 , CCRA[1:0] = 1xb, and in Figure 24.16 , CCRB[1:0] = 1xb.
9	Set compare match value	Set the GTIOCNm pin transition in GTCCRA and GTIOCNm pin transition in the GTCCRB.
10	Set buffer value	For buffer operation, set the GTIOCNm pin and GTIOCNm pin transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTCCRC and GTCCRE, respectively. For double buffer operation, also set the GTIOCNm pin and GTIOCNm pin transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTCCRD and GTCCRF, respectively.
11	Start count operation	Set GTCR.CST to 1 to start count operation.
12	Set buffer value for each cycle	For buffer operation, set the GTIOCNm pin and GTIOCNm pin transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTCCRC and GTCCRE, respectively. For double buffer operation, also set the GTIOCNm pin and GTIOCNm pin transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTCCRD and GTCCRF, respectively.

Note: n: 0 to 5
m: A, B

(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 24.17](#) and [Figure 24.18](#) show examples of GTCCRA and GTCCRB buffer operation and [Table 24.14](#) shows an example for setting GTCCRA and GTCCRB buffer operation.

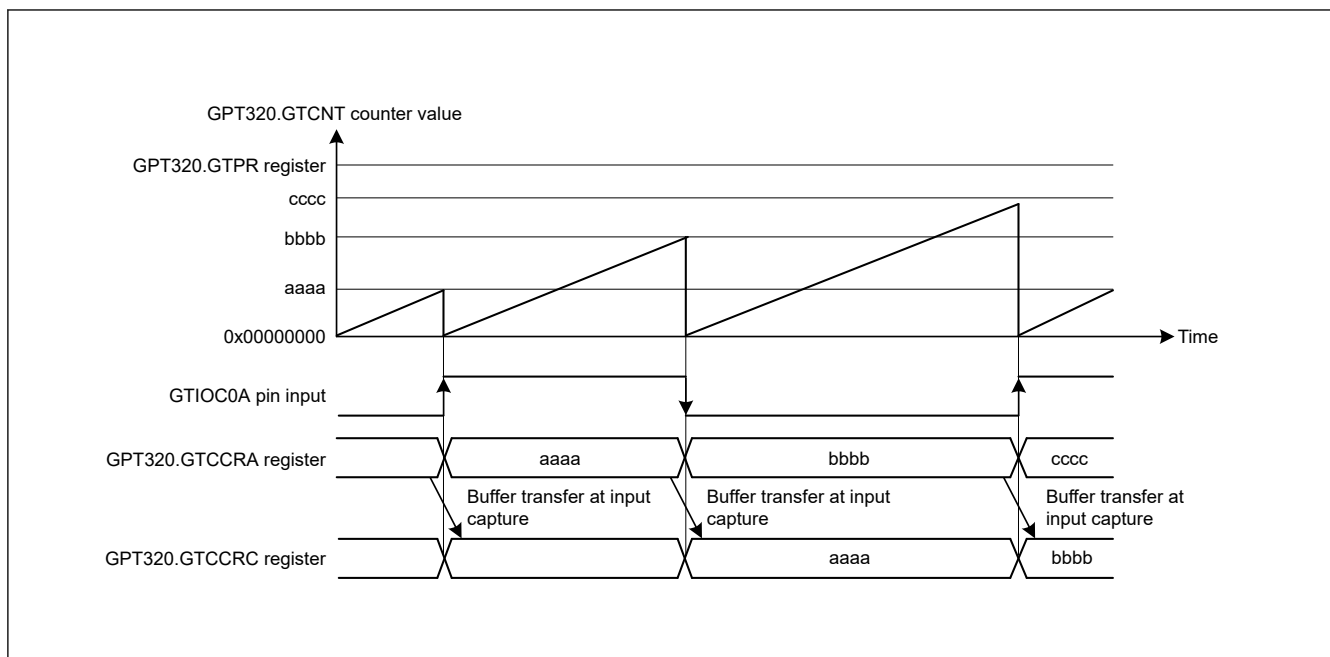


Figure 24.17 Example of GTCCRA and GTCCRC buffer operation with input capture at both edges of GTIOC0A input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOC0A input

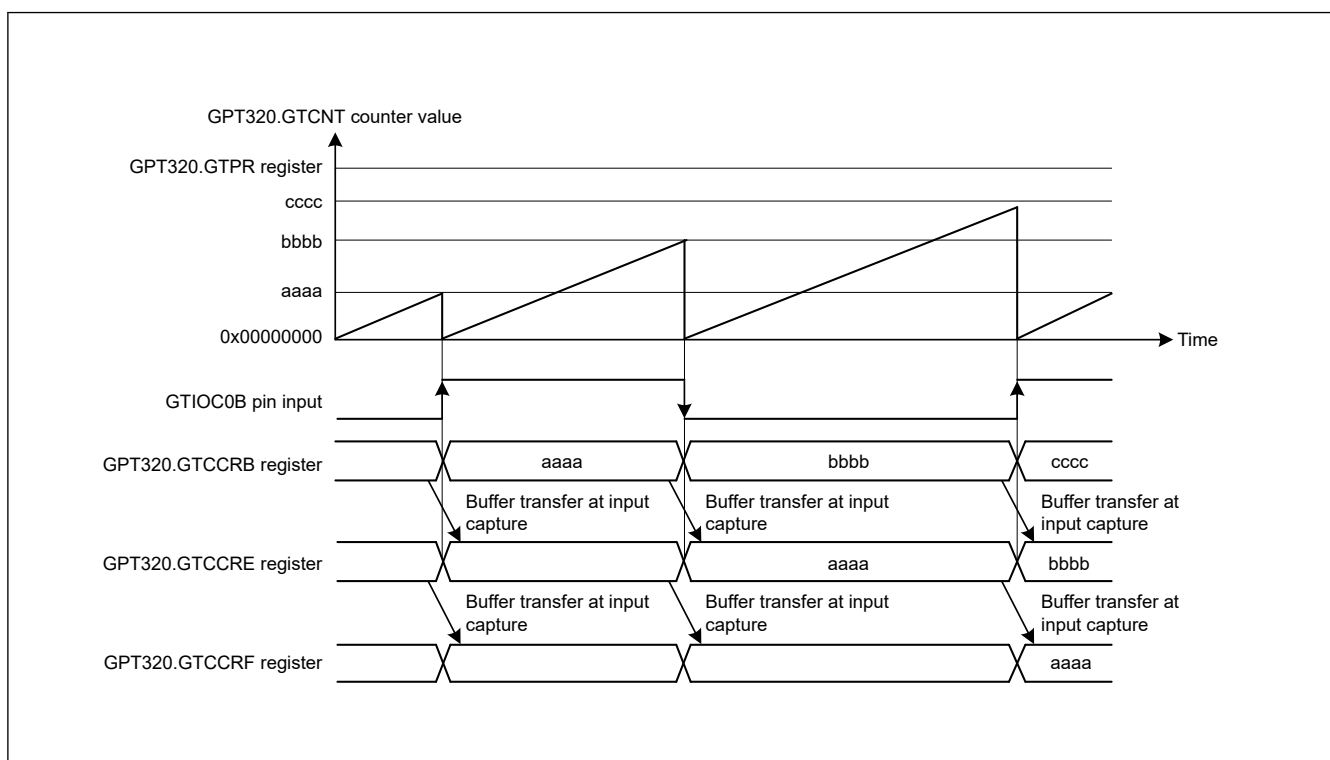


Figure 24.18 Example of GTCCRA and GTCCRB double buffer operation with input capture at both edges of GTIOC0B input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOC0B input

Table 24.14 Example for setting GTCCRA and GTCCRB buffer operation for input capture (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] and count clear source with GTCSCR. In Figure 24.17 , MD[2:0] = 000b (saw-wave PWM mode) and GTCSCR = 0x00000F00, and in Figure 24.18 , MD[2:0] = 000b (saw-wave PWM mode) and GTCSCR = 0x0000F000.

Table 24.14 Example for setting GTCCRA and GTCCRB buffer operation for input capture (2 of 2)

No.	Step Name	Description
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 24.17 , after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Select input capture source	Select input capture source in the GTICASR register and GTICBSR register. In Figure 24.17 , GTICASR = 0x00000F00, and in Figure 24.18 , GTICBSR = 0x0000F000.
7	Set buffer operation	Set buffer operation with CCRA and CCRB in GTBER. In Figure 24.17 , CCRA[1:0] = 01b, and in Figure 24.18 , CCRB = 1xb.
8	Start count operation	Set GTCR.CST to 1 to start count operation.

24.3.3 PWM Output Operating Mode

The GPT can output PWM waveforms to the GTIOCnA pin or GTIOCnB pin (n = 0 to 5) 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.

24.3.3.1 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 GTIOCnA pin or GTIOCnB pin (n = 0 to 5) 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 24.19](#) shows an example of saw-wave PWM mode operation, and [Table 24.15](#) shows an example for setting saw-wave PWM mode.

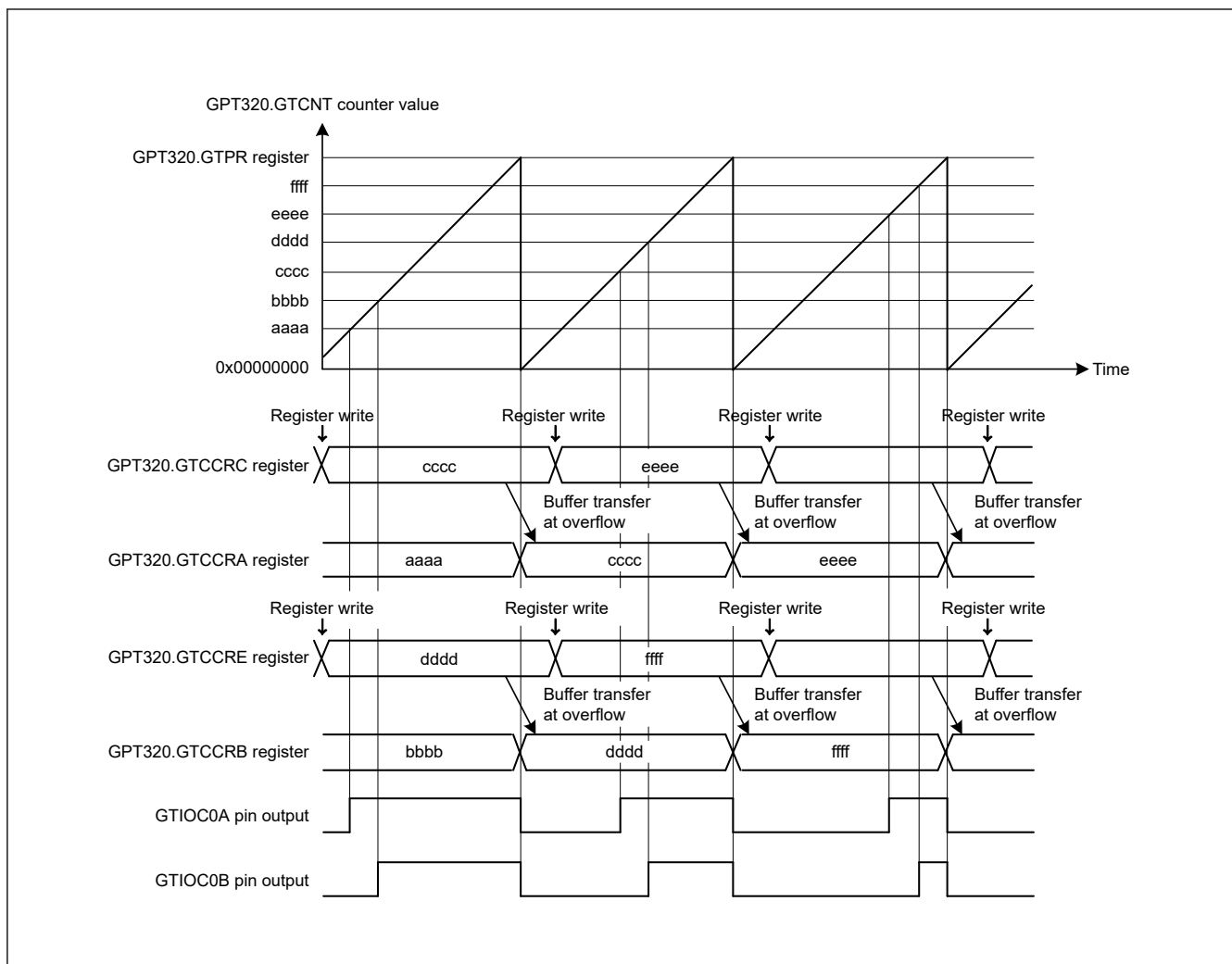


Figure 24.19 Example of saw-wave PWM mode operation with up-counting, buffer operation, high output at GTCCRA/GTCCRB compare match, and low output at cycle end

Table 24.15 Example for setting saw-wave PWM mode (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.19, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 24.19, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 24.19, GTIOA[4:0] = 00110b and GTIOB[4:0] = 00110b.
7	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with OAE and OBE in GTIOR.
8	Set buffer operation	Set buffer operation with CCRA and CCRB in GTBER. In Figure 24.19, CCRA[1:0] = 01b and CCRB[1:0] = 01b.
9	Set compare match value	Set the GTIOCnA pin transition in GTCCRA and GTIOCnB pin transition in GTCCRB.
10	Set buffer value	For buffer operation, set the GTIOCnA pin and GTIOCnB pin transitions in 1 cycle after the current cycle in GTCCRC and GTCCRE, respectively. For double buffer operation, also set the GTIOCnA pin and GTIOCnB pin transitions in 2 cycles after the current cycle in GTCCRD and GTCCRF, respectively.
11	Start count operation	Set GTCR.CST to 1 to start count operation.

Table 24.15 Example for setting saw-wave PWM mode (2 of 2)

No.	Step Name	Description
12	Set buffer value for each cycle	For buffer operation, set the GTIOCnA pin and GTIOCnB pin transitions in 1 cycle after the current cycle in GTCCRC and GTCCRE, respectively. For double buffer operation, also set the GTIOCnA pin and GTIOCnB pin transitions in 2 cycles after the current cycle in GTCCRD and GTCCRF, respectively.

Note: n: 0 to 5
m: A, B

24.3.3.2 Saw-Wave One-Shot Pulse Mode

The saw-wave one-shot pulse mode is a mode in which the cycle is set in GTPR, the GTCNT counter performs saw-wave (half-wave) operation and a PWM waveform is output to the GTIOCnA pin or GTIOCnB pin ($n = 0$ to 5) at a compare match of GTCCRA or GTCCRB with buffer operation fixed.

Buffer operation in saw-wave one-shot pulse mode is different from the usual buffer operation. Buffer transfer is performed from:

- GTCCRC to GTCCRA at the cycle end
- GTCCRE to GTCCRB at the cycle end
- GTCCRD to temporary register A at the cycle end
- GTCCRF to temporary register B at the cycle end
- Temporary register A to GTCCRA at a GTCCRA compare match
- Temporary register B to GTCCRB at a GTCCRB compare match.

The pin output value can be selected from low output, high output, or toggled output separately for a compare match and the cycle end according to the GTIOR setting. When the GTBER.CCRSWT bit is set to 1 while count operation is stopped, the buffer is transferred forcibly from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B. By setting GTDTCR and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 24.20 shows an example of saw-wave one-shot pulse mode operation, and Table 24.16 shows an example for setting saw-wave one-shot pulse mode.

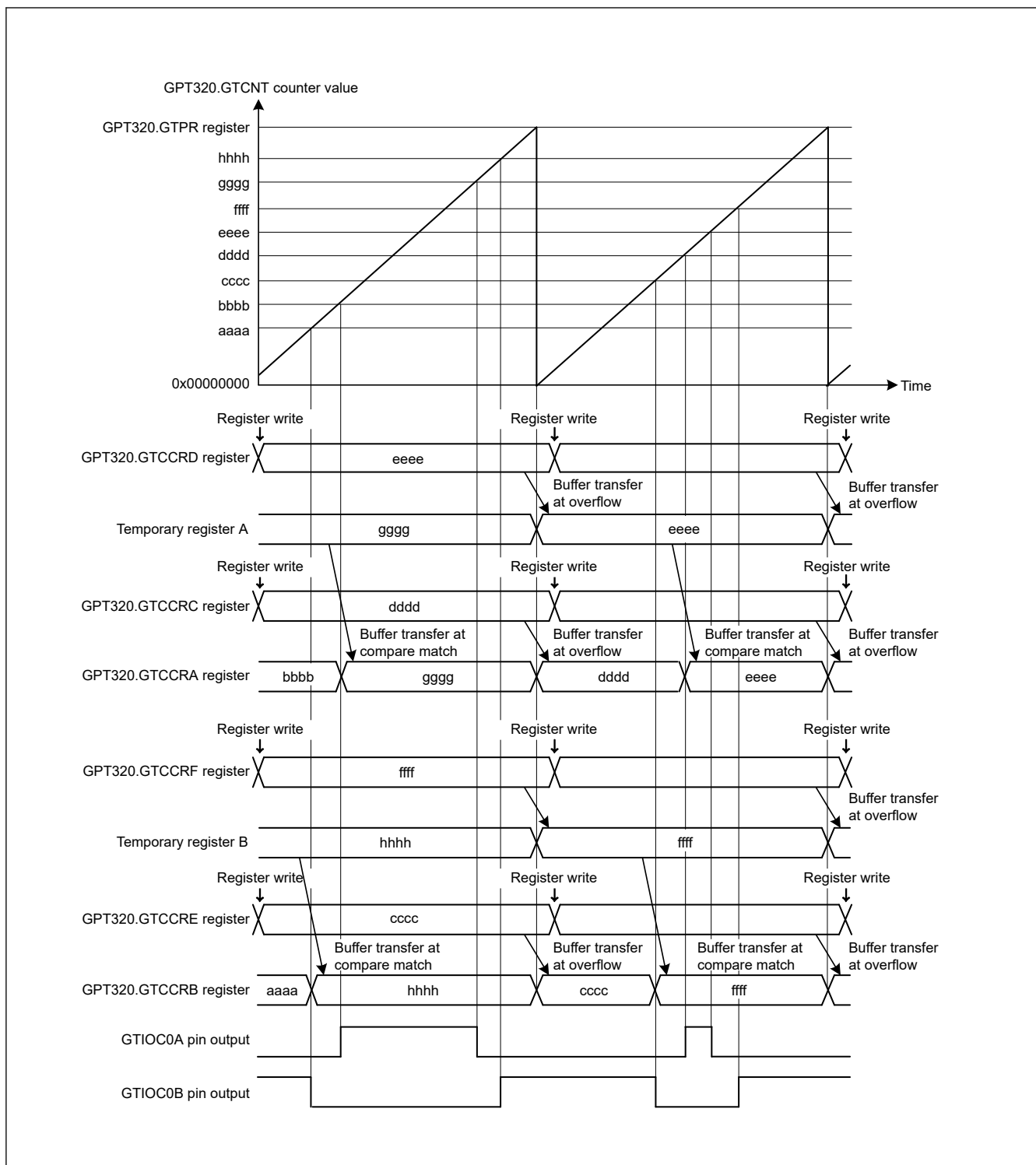


Figure 24.20 Example of saw-wave one-shot pulse mode operation with up-counting, low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/ GTCCRB compare match, and output retained at cycle end

Table 24.16 Example setting for saw-wave one-shot pulse mode (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.20 , 001b (saw-wave one-shot pulse mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 24.20 , after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).

Table 24.16 Example setting for saw-wave one-shot pulse mode (2 of 2)

No.	Step Name	Description
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCNm pin function	Set the GTIOCNm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 24.20 , GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
7	Enable GTIOCNm pin output	Set to enable the GTIOCNm pin output with OAE and OBE in GTIOR.
8	Set buffer value	Set the GTIOCNm pin transition immediately after the count start in GTCCRC and GTCCRD and the GTIOCNB pin transition in GTCCRE and GTCCRF.
9	Set forcible buffer transfer	Set GTBER.CCRSWT to 1 to transfer buffer register data forcibly.
10	Set buffer value	Set the GTIOCNm pin transition in one cycle after the current cycle in GTCCRC and GTCCRD and the GTIOCNB pin transition in GTCCRE and GTCCRF.
11	Start count operation	Set GTCR.CST to 1 to start count operation.
12	Set buffer value for each cycle	Set the GTIOCNm pin transition in one cycle after the current cycle in GTCCRC and GTCCRD and the GTIOCNB pin transition in GTCCRE and GTCCRF.

Note: n: 0 to 5
m: A, B

24.3.3.3 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 GTIOCNm pin or GTIOCNB pin (n = 0 to 5) when a GTCCRA or GTCCRB compare match occurs. Buffer transfer is performed at the trough. The pin output value can be selected from low output, high output, or toggled output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

[Figure 24.21](#) shows an example of a triangle-wave PWM mode 1 operation, and [Table 24.17](#) shows an example for setting a triangle-wave PWM mode 1.

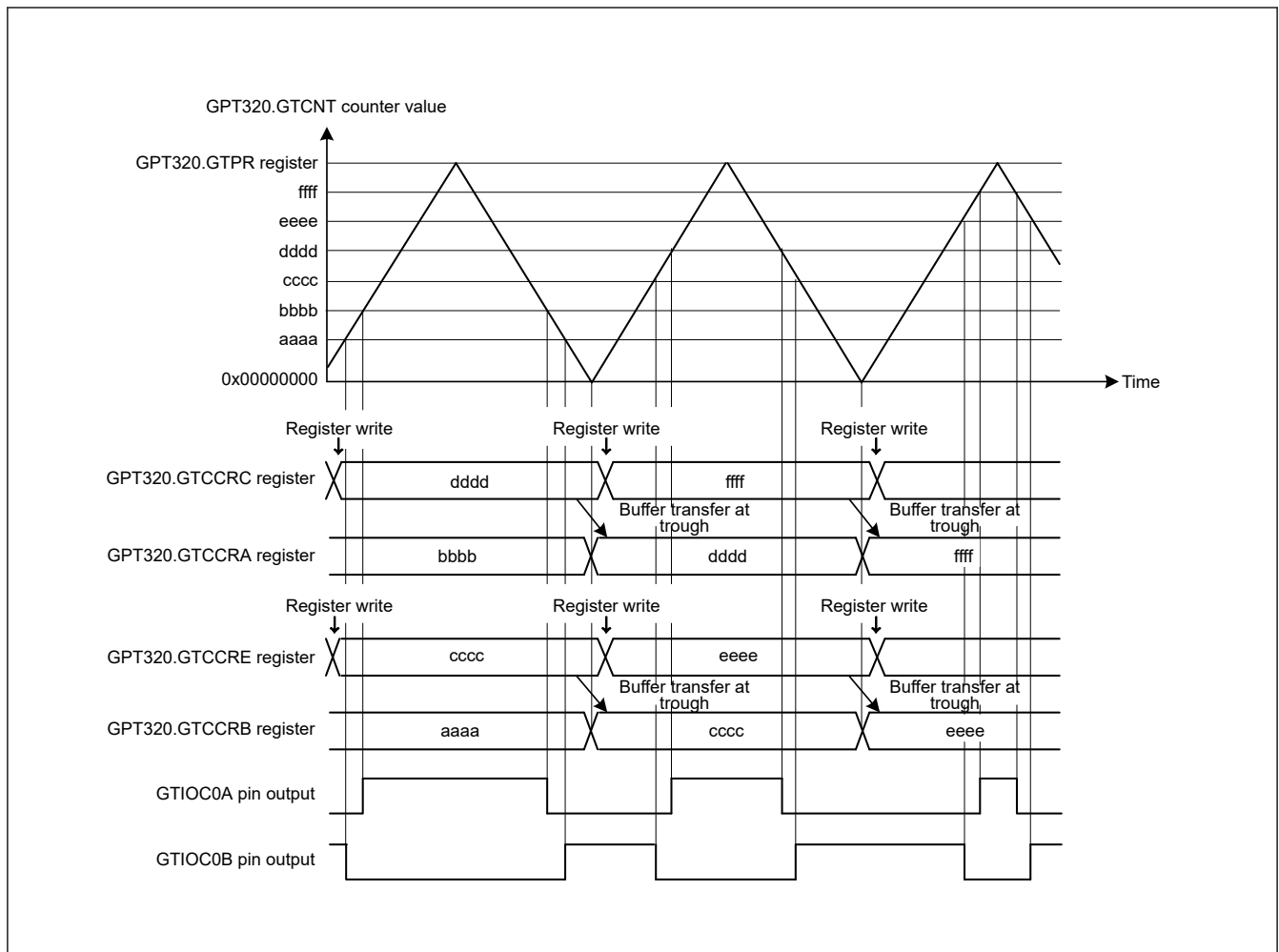


Figure 24.21 Example of triangle-wave PWM mode 1 operation with buffer operation, low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/ GTCCRB register compare match, and output retained at cycle end

Table 24.17 Example setting for triangle-wave PWM mode 1 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.21, 100b (triangle-wave PWM mode 1) is set.
2	Select count clock	Select the count clock with GTCR.TPCS[2:0].
3	Set cycle	Set the cycle in GTPR.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 24.21, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with OAE and OBE in GTIOR.
7	Set buffer operation	Set buffer operation with CCRA and CCRB in GTBER. In Figure 24.21, CCRA[1:0] = 01b and CCRB[1:0] = 01b.
8	Set compare match value	Set the GTIOCnA pin and GTIOCnB pin transitions in GTCCRA and GTCCRB, respectively.
9	Set buffer value	For buffer operation, set the GTIOCnA pin and GTIOCnB pin transitions in 1 cycle after the current cycle in GTCCRC and GTCCRE, respectively. For double buffer operation, also set the GTIOCnA pin and GTIOCnB pin transitions in 2 cycles after the current cycle in GTCCRD and GTCCRF, respectively.
10	Start count operation	Set GTCR.CST to 1 to start count operation.

Table 24.17 Example setting for triangle-wave PWM mode 1 (2 of 2)

No.	Step Name	Description
11	Set buffer value for each cycle	For buffer operation, set the GTIOCnA pin and GTIOCnB pin transitions in 1 cycle after the current cycle in GTCCRC and GTCCRE, respectively. For double buffer operation, also set the GTIOCnA pin and GTIOCnB pin transitions in 2 cycles after the current cycle in GTCCRD and GTCCRF, respectively.

Note: n: 0 to 5
m: A, B

24.3.3.4 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 GTIOCnA pin or GTIOCnB pin (n = 0 to 5) 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 24.22 shows an example of triangle-wave PWM mode 2 operation, and Table 24.18 shows an example for setting triangle-wave PWM mode 2.

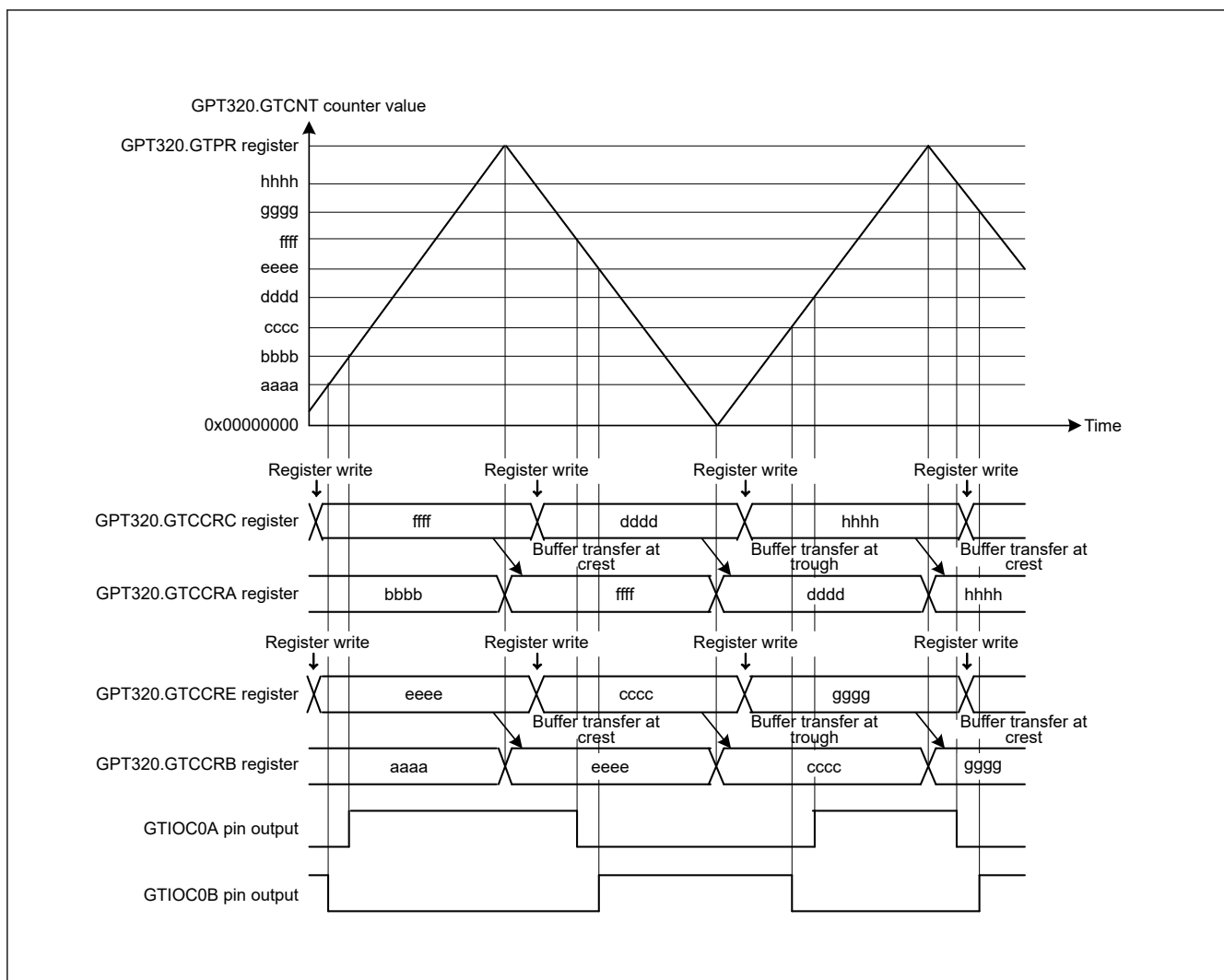


Figure 24.22 Example of triangle-wave PWM mode 2 operation with buffer operation, low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/ GTCCRB compare match, and output retained at cycle end

Table 24.18 Example for setting triangle-wave PWM mode 2

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.22 , 101b (triangle-wave PWM mode 2) is set.
2	Select count clock	Select the count clock with GTCR.TPCS[2:0].
3	Set cycle	Set the cycle in GTPR.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCNm pin function	Set the GTIOCNm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 24.22 , GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTIOCNm pin output	Set to enable the GTIOCNm pin output with OAE and OBE in GTIOR.
7	Set buffer operation	Set buffer operation with CCRA and CCRB in GTBER. In Figure 24.22 , CCRA[1:0] = 01b and CCRB[1:0] = 01b.
8	Set compare match value	Set the GTIOCNm pin and GTIOCNm pin transitions in GTCCRA and GTCCRB, respectively.
9	Set buffer value	For buffer operation, set the GTIOCNm pin and GTIOCNm pin transitions in 1 cycle after the current cycle in GTCCRC and GTCCRE, respectively. For double buffer operation, also set the GTIOCNm pin and GTIOCNm pin transitions in 1 cycle after the current cycle in GTCCRD and GTCCRF, respectively.
10	Start count operation	Set GTCR.CST to 1 to start count operation.
11	Set buffer value for each cycle	For buffer operation, set the GTIOCNm pin and GTIOCNm pin transitions in half cycle after the current cycle in GTCCRC and GTCCRE, respectively. For double buffer operation, also set the GTIOCNm pin and GTIOCNm pin transitions in 1 cycle after the current cycle in GTCCRD and GTCCRF, respectively.

Note: n: 0 to 5
m: A, B

24.3.3.5 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 GTIOCNm pin or GTIOCNm pin (n = 0 to 5) at a compare match of GTCCRA or GTCCRB with buffer operation fixed. Buffer operation in triangle-wave PWM mode 3 is different from the usual buffer operation. Buffer transfer is performed from the following:

- GTCCRC to GTCCRA at the trough
- GTCCRE to GTCCRB at the trough
- GTCCRD to temporary register A at the trough
- GTCCRF to temporary register B at the trough
- Temporary register A to GTCCRA at the crest
- Temporary register B to GTCCRB at the crest.

The pin output value can be selected from low output, high output, or toggled output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

[Figure 24.23](#) shows an example of triangle-wave PWM mode 3 operation, and [Table 24.19](#) shows an example for setting triangle-wave PWM mode 3.

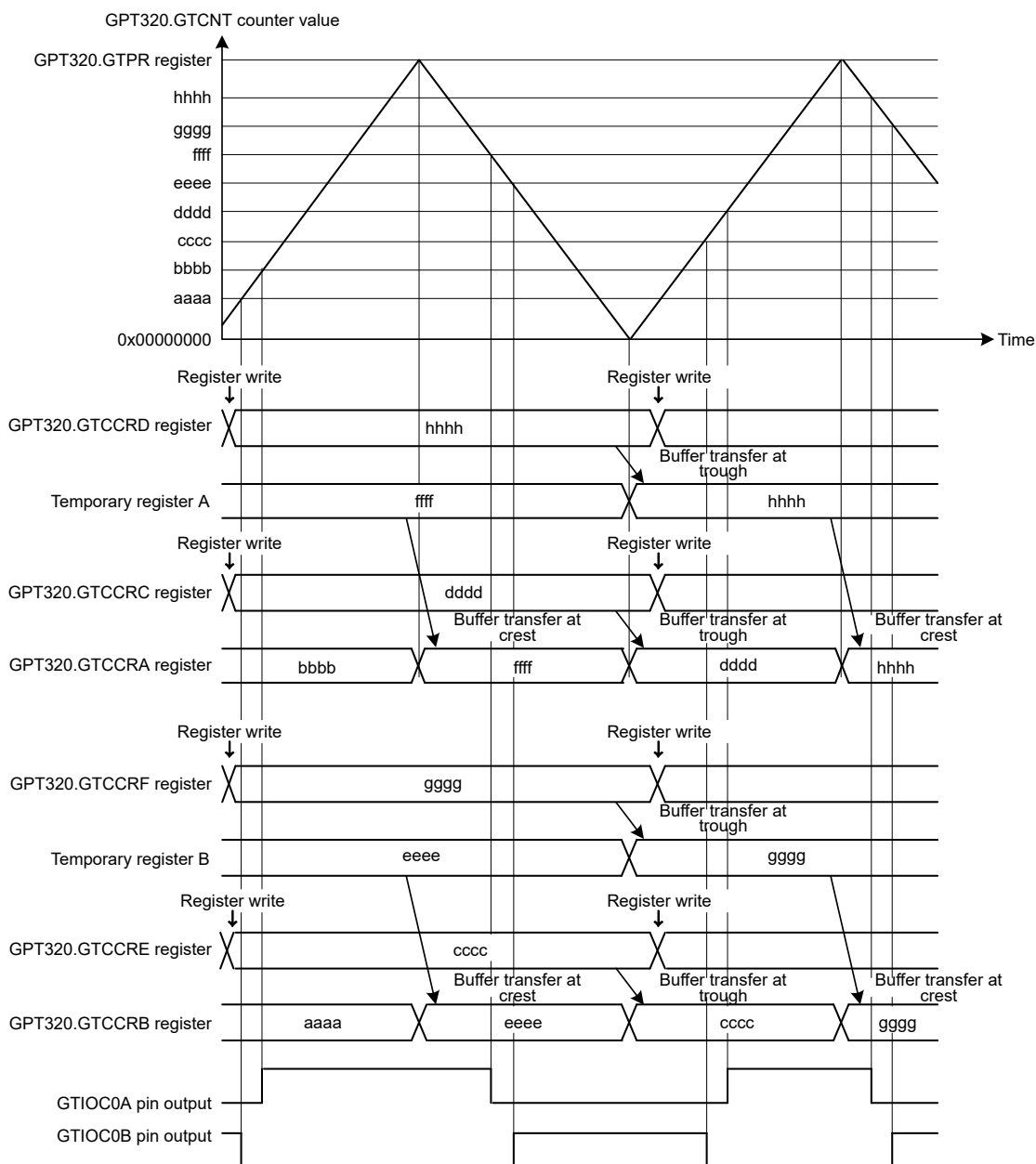


Figure 24.23 Example of triangle-wave PWM mode 3 operation with low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/GTCCRB compare match, and output retained at cycle end

Table 24.19 Example setting for triangle-wave PWM mode 3 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.23 , 110b (triangle-wave PWM mode 3) is set.
2	Select count clock	Select the count clock with GTCR.TPCS[2:0].
3	Set cycle	Set the cycle in GTPR.
4	Set initial value for counter	Set the initial value in the GTCNT counter.

Table 24.19 Example setting for triangle-wave PWM mode 3 (2 of 2)

No.	Step Name	Description
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 24.23 , GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with OAE and OBE in GTIOR.
7	Set buffer value	Set the GTIOCnA pin transition immediately after the count start in GTCCRC and GTCCRD and the GTIOCnB pin transition in GTCCRE and GTCCRF.
8	Set forcible buffer transfer	Set GTBER.CCRSWT to 1 to transfer buffer register data forcibly.
9	Set buffer value	Set the GTIOCnA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD and the GTIOCnB pin transition in GTCCRE and GTCCRF.
10	Start count operation	Set GTCR.CST to 1 to start count operation.
11	Set buffer value for each cycle	Set the GTIOCnA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD and the GTIOCnB pin transition in GTCCRE and GTCCRF.

Note: n: 0 to 5
m: A, B

24.3.4 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. In triangle-wave mode, when the dead time is beyond the cycle by setting the value $GTCCRA = 0$ or $GTCCRA \geq GTPR$ for GTCCRA, the output protection function keeps the level of output. For details, see [section 24.7.3. GTIOCnm Pin Output Negate Control \(n = 0 to 5, m = A, B\)](#). The automatic dead time value setting to GTCCRB is performed at the next clock cycle count when registers that are used for calculating the automatic dead time value are updated.

When a dead time error occurs, the compare match values for positive and negative waveforms are adjusted to generate the waveforms with the dead time as shown in [Table 24.20](#).

The adjusted value for the negative waveform is set for GTCCRB automatically.

The adjusted value for the positive waveform is used as internal signal and not set for GTCCRA.

Table 24.20 Adjustment of the Waveform Change Point When a Dead-Time Error Occurs

Mode	Count Direction	Period	Condition for Dead Time Error	Change Point of the Positive-Phase Waveform after Adjustment	Change Point of the Negative-Phase Waveform after Adjustment
Sawtooth-wave one-shot pulse mode	Up-counting	First half	$GTCCRA - GTDVU < 0$	GTDVU	0
		Second half	$GTCCRA + GTDVU > GTPR$	$GTPR - GTDVU$	GTPR
	Down-counting	First half	$GTCCRA + GTDVU > GTPR$	$GTPR - GTDVU$	GTPR
		Second half	$GTCCRA - GTDVU < 0$	GTDVU	0
Triangle-wave PWM mode 1/2/3	Up-counting	(First half)	$GTCCRA - GTDVU \leq 0$	$GTDVU + 1$	1
	Down-counting	(Second half)	$GTCCRA - GTDVU < 0$	GTDVU	0

[Figure 24.24](#) to [Figure 24.27](#) show examples of automatic dead time setting function operation. [Table 24.21](#) and [Table 24.22](#) show the setting examples.

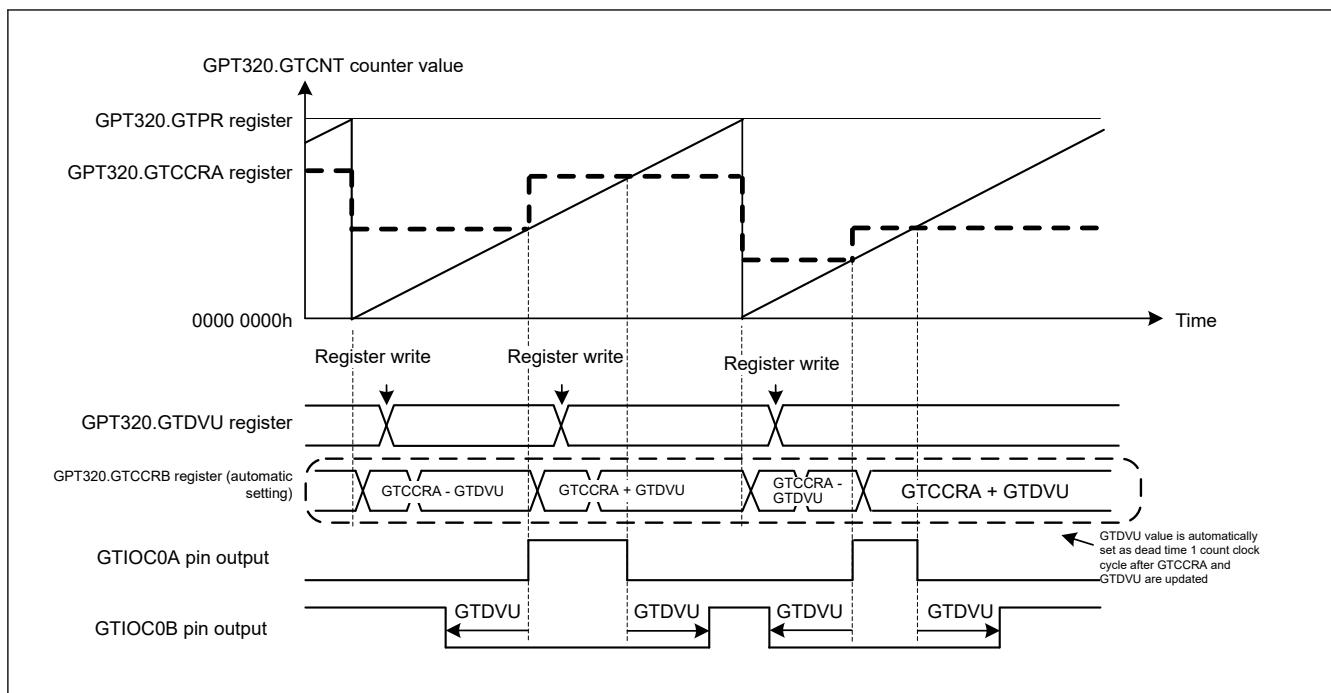


Figure 24.24 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, up-counting, and active-high

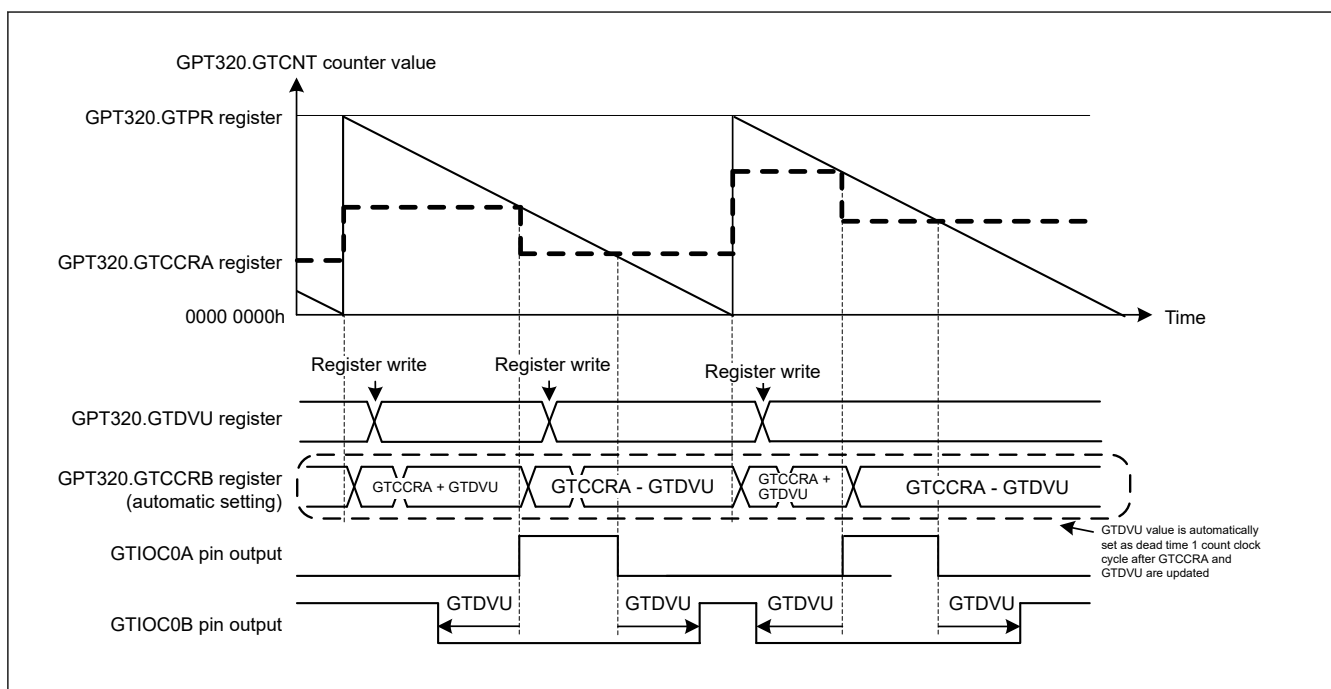


Figure 24.25 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, down-counting, and active-high

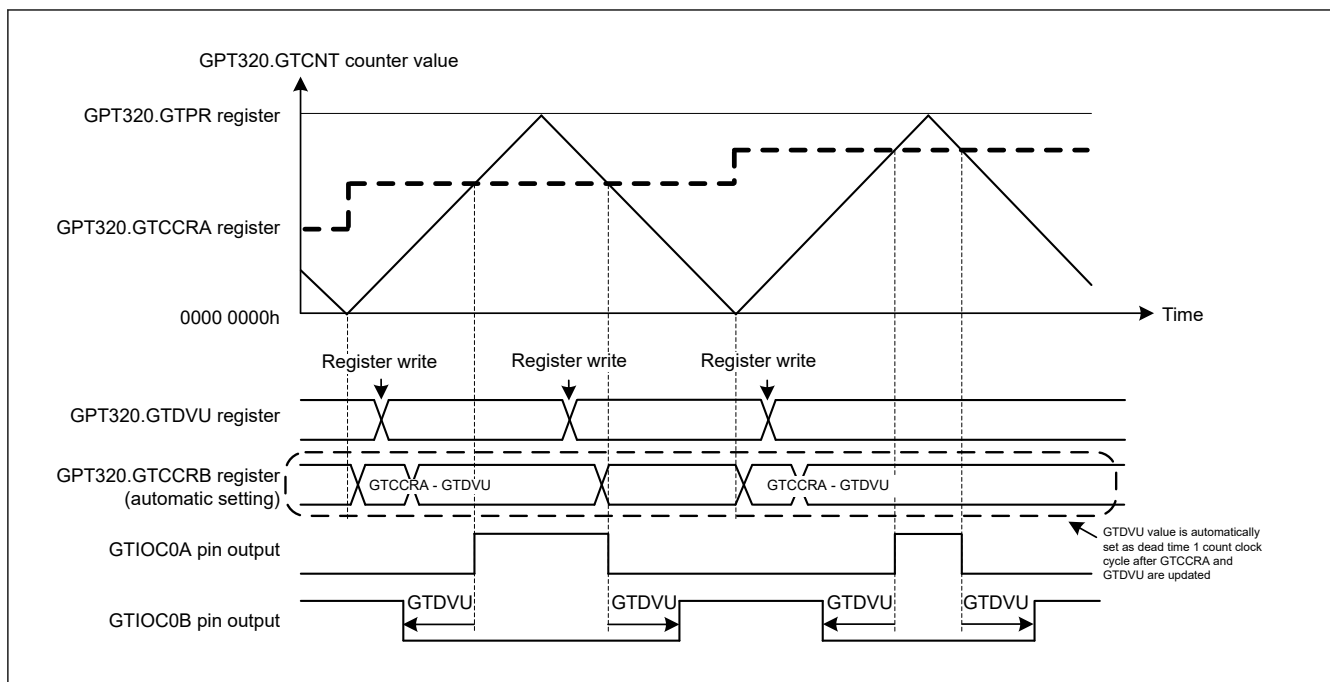


Figure 24.26 Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 1, and active-high

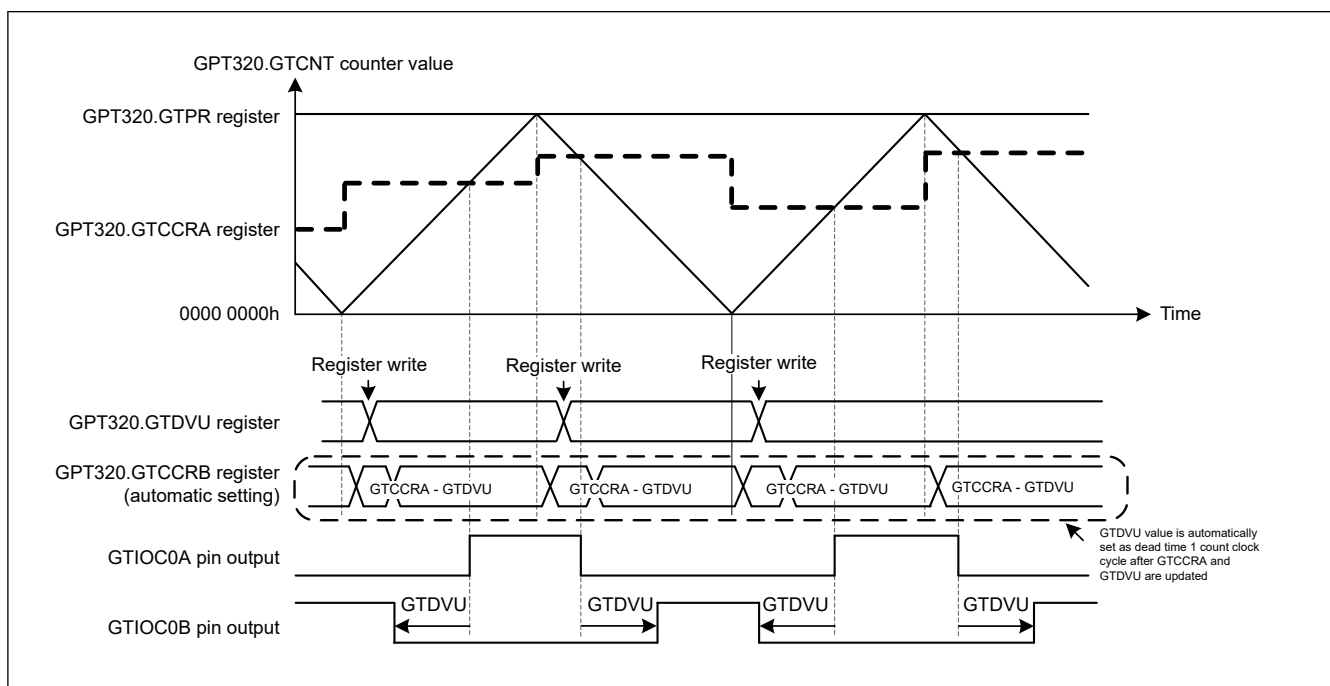


Figure 24.27 Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 2 or 3, and active-high

Table 24.21 Example setting for automatic dead time setting function in saw-wave one-shot pulse mode, and triangle-wave PWM mode 3 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.24 and Figure 24.25, 001b (saw-wave one-shot pulse mode) is set. In Figure 24.27, 110b (triangle-wave PWM mode 3) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 24.24, 01b is set after 11b is set in GTUDDTYC[1:0] (up count). In Figure 24.25, 00b is set after 10b is set in GTUDDTYC[1:0] (down count).

Table 24.21 Example setting for automatic dead time setting function in saw-wave one-shot pulse mode, and triangle-wave PWM mode 3 (2 of 2)

No.	Step Name	Description
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCNm pin function	Set the GTIOCNm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 24.24 , Figure 24.26 , and Figure 24.27 , GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
7	Enable GTIOCNm pin output	Set to enable the GTIOCNm pin output with OAE and OBE in GTIOR.
8	Set buffer value for compare match	Set the GTIOCNm pin transition immediately after the count start in GTCCRC and GTCCRD.
9	Set forcible buffer transfer for compare match	Set GTBER.CCRSWT to 1 to transfer buffer register data forcibly to GTCCRA.
10	Set buffer value for compare match	Set the GTIOCNm pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD.
11	Set automatic dead time setting function	Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.
12	Set dead time value	Set the dead time value in GTDVU.
13	Start count operation	Set GTCR.CST to 1 to start count operation.
14	Set buffer value for each cycle	Set the GTIOCNm pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD.

Note: n:
0 to 5
m: A, B

Table 24.22 Example setting for automatic dead time setting function in triangle-wave PWM mode 1 or 2

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.26 , 100b (triangle-wave PWM mode 1) is set. In Figure 24.27 , 101b (triangle-wave PWM mode 2) is set.
2	Select count clock	Select the count clock with GTCR.TPCS[2:0].
3	Set cycle	Set the cycle in GTPR.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCNm pin function	Set the GTIOCNm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 24.26 and Figure 24.27 , GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTIOCNm pin output	Set to enable the GTIOCNm pin output with OAE and OBE in GTIOR.
7	Set buffer operation for compare match	Set buffer operation with CCRA in GTBER.
8	Set compare match value	Set the GTIOCNm pin transition in GTCCRA.
9	Set buffer value for compare match	For buffer operation, set the GTIOCNm pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRC. For double buffer operation, also set the GTIOCNm pin transition in 2 cycles after the current cycle (in triangle-wave PWM mode 1) or 1 cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRD.
10	Set automatic dead time setting function	Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.
11	Set dead time value	Set the dead time value in GTDVU.
12	Start count operation	Set GTCR.CST to 1 to start count operation.
13	Set buffer value for each cycle	When the compare match register is used for buffer operation, set the GTIOCNm pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRC.

Note: n: 0 to 5
m: A, B

24.3.5 Count Direction Changing Function

The count direction of the GTCNT counter can be changed by modifying the UD bit in GTUDDTYC.

In saw-wave mode, if the UD bit in GTUDDTYC is modified during count operation, the count direction is changed at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the GTUDDTYC.UD bit is modified while the count operation stops and the GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit modification is not reflected at the start of counting and the count direction is changed at an overflow or an underflow. If the UDF bit is set to 1 while the count operation stops, the GTUDDTYC.UD bit value at that time is reflected at the start of counting.

In triangle-wave mode, the count direction does not change even though the UD bit in GTUDDTYC is modified during the count operation. Similarly, even though the GTUDDTYC.UD bit is modified while the count operation stops and GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit value is not reflected to the count operation. If the GTUDDTYC.UDF bit is set to 1 while the count operation is stopped, the GTUDDTYC.UD bit value at that time is reflected at the start of counting.

If the count direction changes during a saw-wave count operation, the GTPR value after the start of up-counting is reflected in the count cycle during up-counting and the GTPR value before the start of down-counting is reflected during down-counting.

Figure 24.28 shows an example of count direction changing function operation.

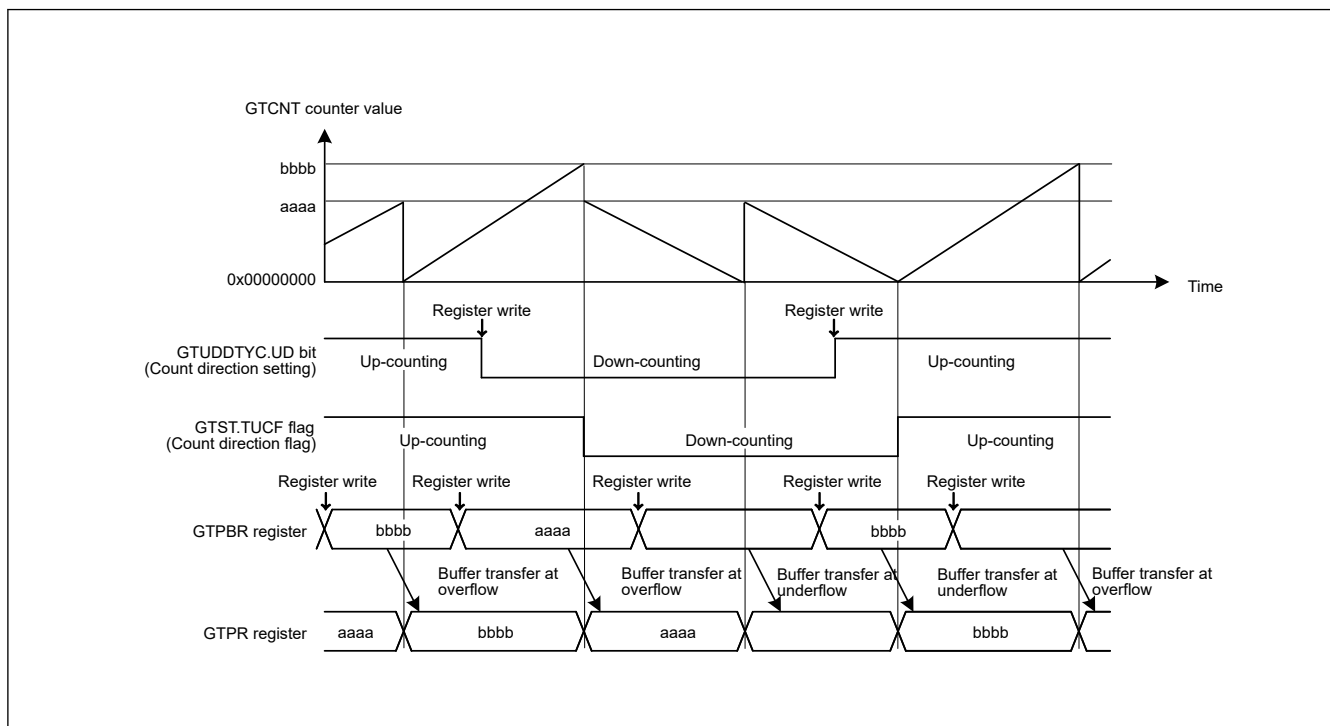


Figure 24.28 Example of a count direction changing function operation during buffer operation

24.3.6 Function of Output Duty 0% and 100%

The output duty of the GTIOCnA pin and the GTIOCnB pin (n = 0 to 5) 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 stops and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 1, the output duty modification is reflected at the start of counting.

In performing 0% or 100% duty operation, GPT internally continues to:

- Perform compare match operation
- Set compare match flag
- Output interrupt
- Perform buffer operation.

When the control is changed from 0% or 100% duty setting to compare match, the output value of GTIOCnA pin at cycle end is decided by GTIOR.GTIOA[3:2] and GTUDDTYC.OADTYR. The output value of GTIOCnB pin at cycle end is decided by GTIOR.GTIOB[3:2] and GTUDDTYC.OBDTYR.

When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 01b, the output pins output low at cycle end. When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 10b, the output pins output high at cycle end.

GTUDDTYC.OADTYR selects the value that is the object of output retained/toggled at cycle end, when GTIOR.GTIOm[3:2] are set to 00b (output retained at cycle end) or when GTIOR.GTIOm[3:2] are set to 11b (output toggled at cycle end). Table 24.23 shows the values of GTIOCnA and GTIOCnB pin output at cycle end.

Table 24.23 Output values after releasing 0% or 100% duty setting (m = A, B)

GTIOR.GTIOm[3:2]	Compare match value at cycle end masked by 0% or 100% duty setting	GTUDDTYC.OmDTYR in duty 0% setting		GTUDDTYC.OmDTYR in duty 100% setting	
		0	1	0	1
00 (output retained at cycle end)	0	0	0	1	0
	1	0	1	1	1
01 (low output at cycle end)	—	0	0	0	0
10 (high output at cycle end)	—	1	1	1	1
11 (output toggled at cycle end)	0	1	1	0	1
	1	1	0	0	0

Figure 24.29 shows an example of output duty 0% and 100% function.

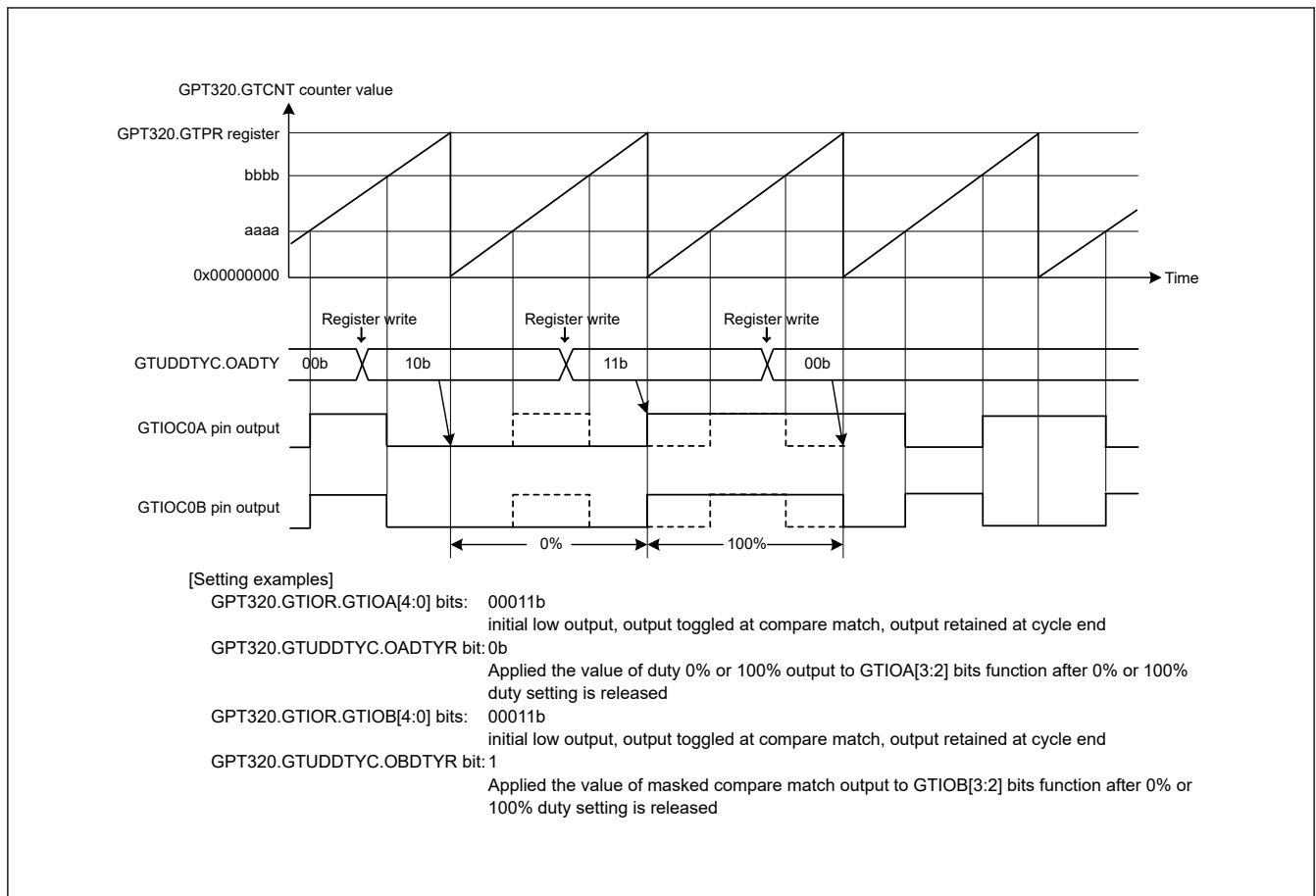


Figure 24.29 Example of output duty 0% and 100% function

24.3.7 Hardware Count Start/Count Stop and Clear Operation

The GTCNT counter can be started, stopped, or cleared by the following hardware sources:

- External trigger input
- ELC event input
- GTIOCnA and GTIOCnB pin input (n = 0 to 5).

24.3.7.1 Hardware Start Operation

The GTCNT counter can be started by selecting a hardware source using GTSSR.

Figure 24.30 shows an example of a count start operation by a hardware source. Table 24.24 shows the setting example.

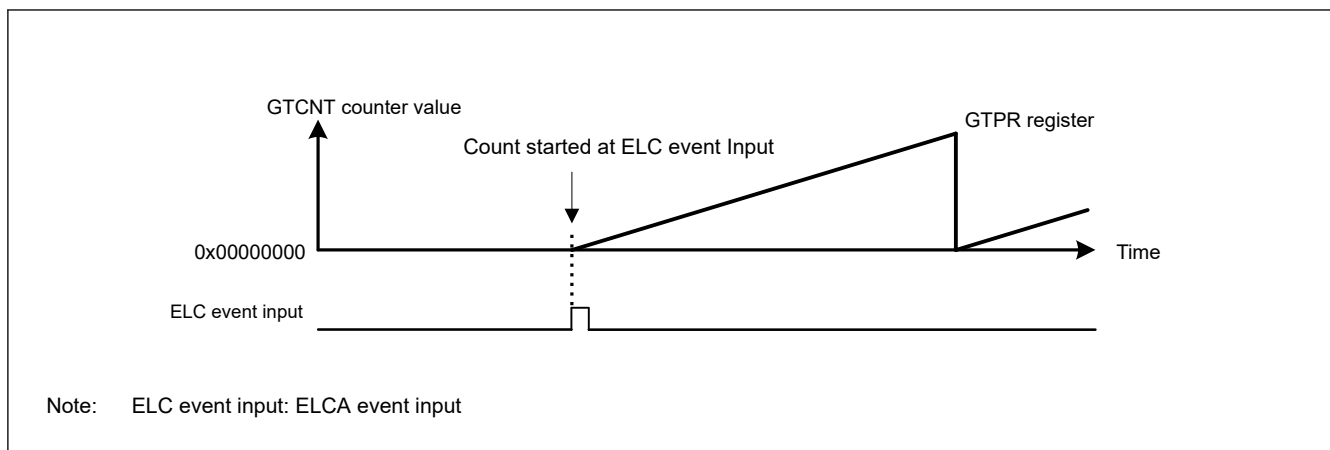


Figure 24.30 Example of count start operation by a hardware source started at the input of the signal from the ELCA event

Table 24.24 Example setting for count start operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.30 , 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 24.30 , after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 24.30 , 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation in GTSSR register. In Figure 24.30 , GTSSR.SSELCA = 1
7	Set hardware source operation	Set operation of the hardware source selected by GTSSR register and start counting. In Figure 24.30 , the ELCA event input operation is set.

24.3.7.2 Hardware stop operation

The GTCNT counter can be stopped by selecting a hardware source using GTPSR.

[Figure 24.31](#) shows an example of a count stop operation by a hardware source. [Table 24.25](#) shows the setting example. In this example, the count operation stops at the edge of the ELCA event input and restarts at the edges of the ELCB event input.

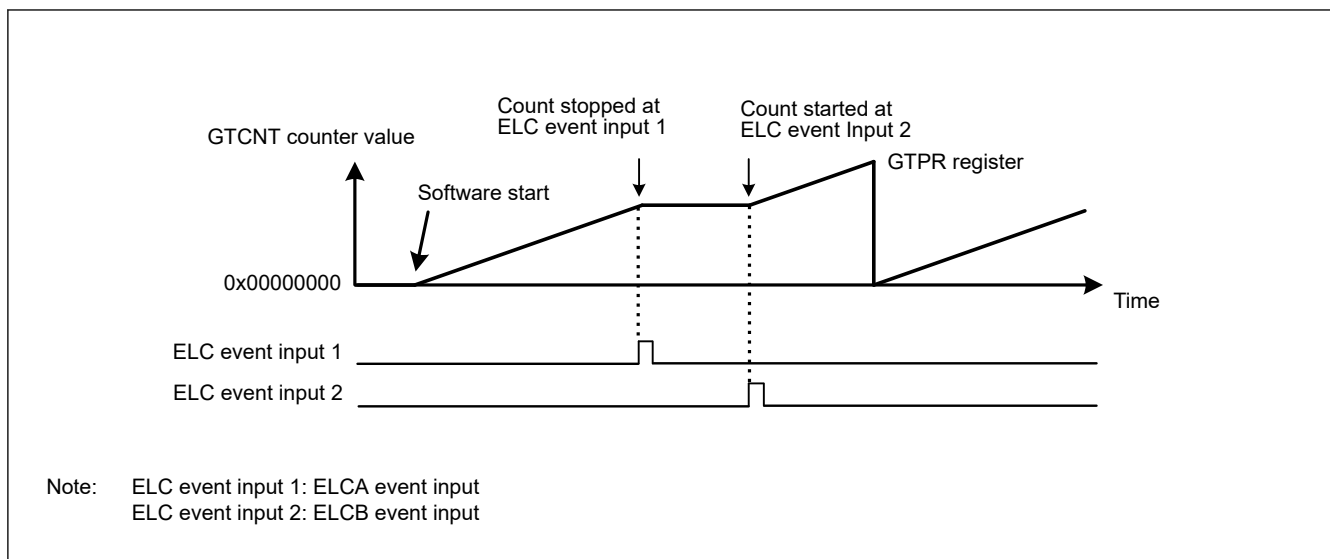


Figure 24.31 Example of count stop operation by hardware source started by software, stopped at ELCA input, and restarted at ELCB input

Table 24.25 Example setting for count stop operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.31 , 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 24.31 , after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 24.31 , 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation in GTSSR register, and wait for count start by the hardware source. In Figure 24.31 , GTSSR.SSELCB = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation in GTPSR register and wait for count stop by the hardware source. In Figure 24.31 , GTPSR.PSELCA = 1.
8	Set hardware source operation	Set operation of the hardware source selected in GTSSR register or GTPSR register, and start or stop counting. In Figure 24.31 , ELCA input operation and ELCB input operation are set.

[Figure 24.32](#) shows an example of a count start/stop operation by a hardware source. [Table 24.26](#) shows the setting example. In this example, the counter operates during the high-level periods of the external trigger input GTETRGA.

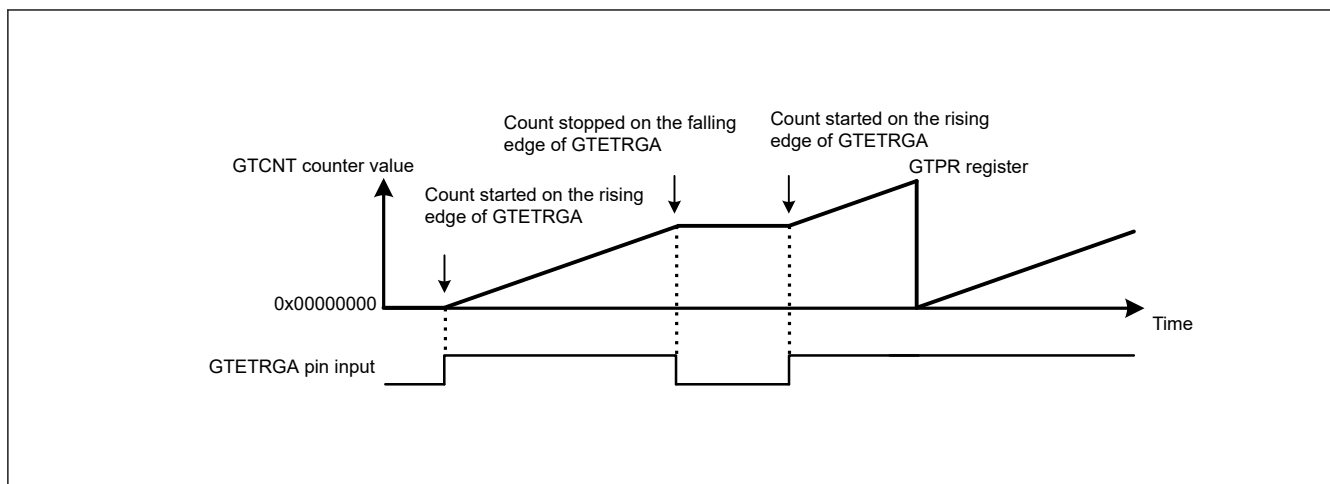


Figure 24.32 Example of count start/stop operation by a hardware source started on the rising edge of GTETRGA pin input, and stopped on the falling edge of GTETRGA pin input

Table 24.26 Example setting for count start/stop operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.32, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 24.32, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 24.32, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation with GTSSR register and wait for count start by the hardware source. In Figure 24.32, GTSSR.SSGTRGAR = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation with GTPSR register and wait for count stop by the hardware source. In Figure 24.32, GTPSR.PSGTRGAF = 1.
8	Set hardware source operation	Set operation of the hardware source selected in GTSSR or GTPSR and start or stop counting. In Figure 24.32, the GTETRGA pin operation is set.

24.3.7.3 Hardware clear operation

The GTCNT counter can be cleared by selecting a hardware source using GTCSR. The GPTn_OVF/GPTn_UDF (n = 0 to 5) interrupt (overflow/underflow interrupt) is not generated when the GTCNT counter is cleared by a hardware source or by software.

Figure 24.33 and Figure 24.34 show examples of the GTCNT counter clearing operation by a hardware source. Table 24.27 shows the setting example. In this example, the GTCNT counter starts at the edge of the ELCA input, and the counter stops and clears at the edge of the ELCB input.

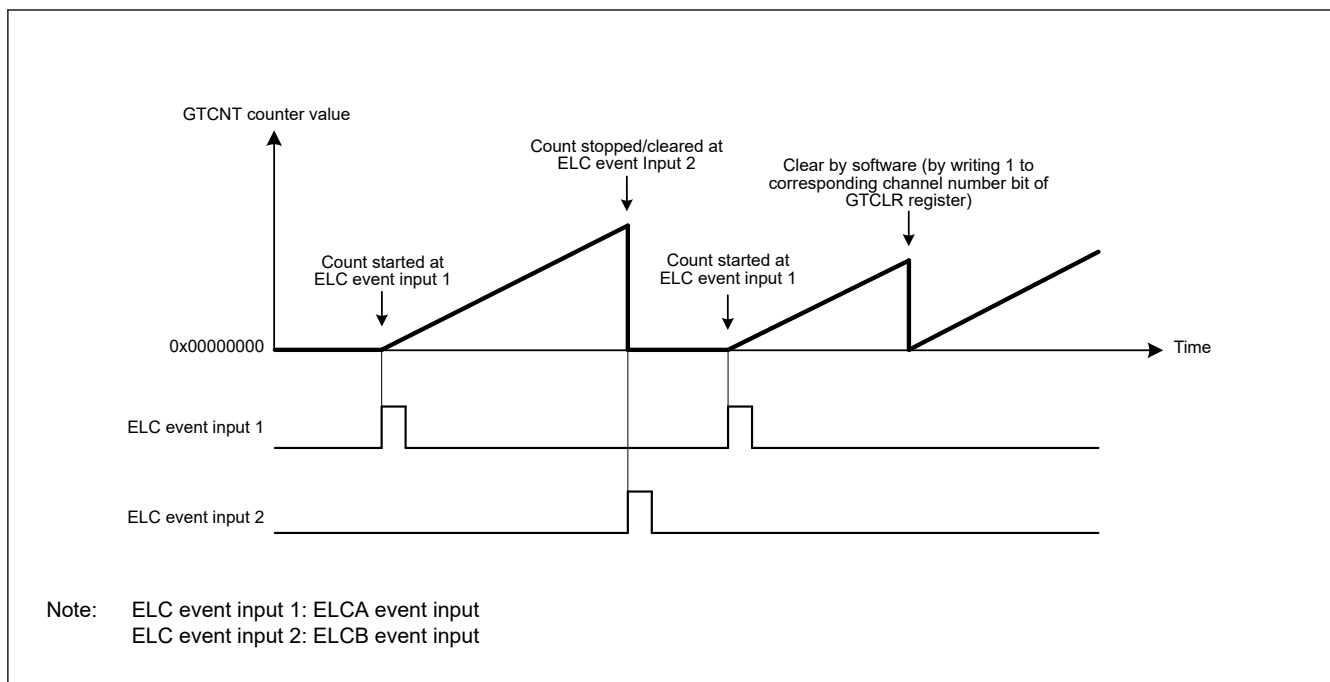


Figure 24.33 Examples of count clearing operation by hardware source in saw wave up-counting, started at ELCA input, and stopped/cleared at ELCB input

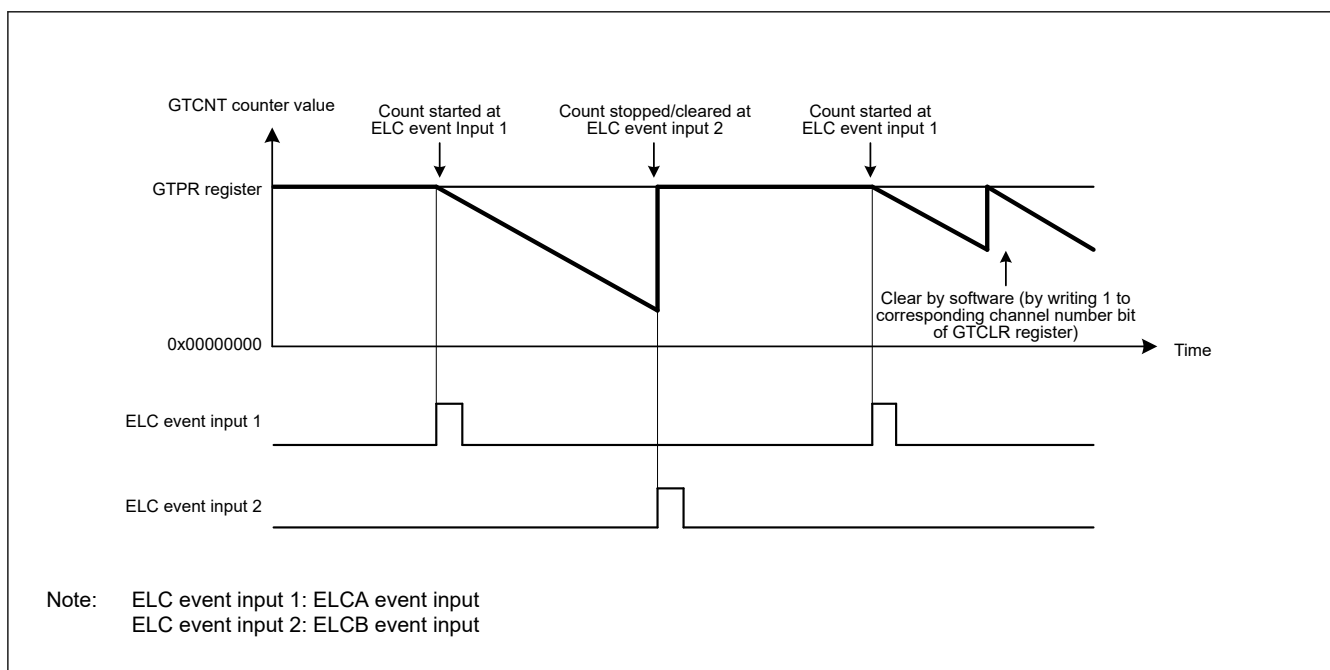


Figure 24.34 Examples of count clearing operation by hardware source in saw wave down-counting, started at ELCA input, and stopped/cleared at ELCB input

Table 24.27 Example setting for count clearing operation by a hardware source (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0]. In Figure 24.33 and Figure 24.34, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 24.33, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting). In Figure 24.34, after 10b is set in GTUDDTYC[1:0], 00b is set in GTUDDTYC[1:0] (down-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].

Table 24.27 Example setting for count clearing operation by a hardware source (2 of 2)

No.	Step Name	Description
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 24.33 , 0x00000000 is set. In Figure 24.34 , the GTPR value is set.
6	Set hardware count start	Select a hardware source for starting count operation in GTSSR register and wait for count start by the hardware source. In Figure 24.33 and Figure 24.34 , GTSSR.SSELCA = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation in GTPSR register and wait for count stop by the hardware source. In Figure 24.33 and Figure 24.34 , GTPSR.PSELCB = 1.
8	Set hardware count clear	Select a hardware source for clearing count operation in GTCSR register and wait for count clear by the hardware source. In Figure 24.33 and Figure 24.34 , GTCSR.CSELCB = 1.
9	Set hardware source operation	Set operation of the hardware source selected in GTSSR register, GTPSR Register or GTCSR register and start, stop or clear counting. In Figure 24.33 and Figure 24.34 , the ELCA input and ELCB input are set.

The GPTn_OVF/GPTn_UDF (n = 0 to 5) interrupt (overflow/underflow interrupt) is not generated when the counter is cleared by a hardware source or by software.

[Figure 24.35](#) shows the relationship between the counter clearing by a hardware source and the GPTn_OVF (n = 0 to 5) interrupt.

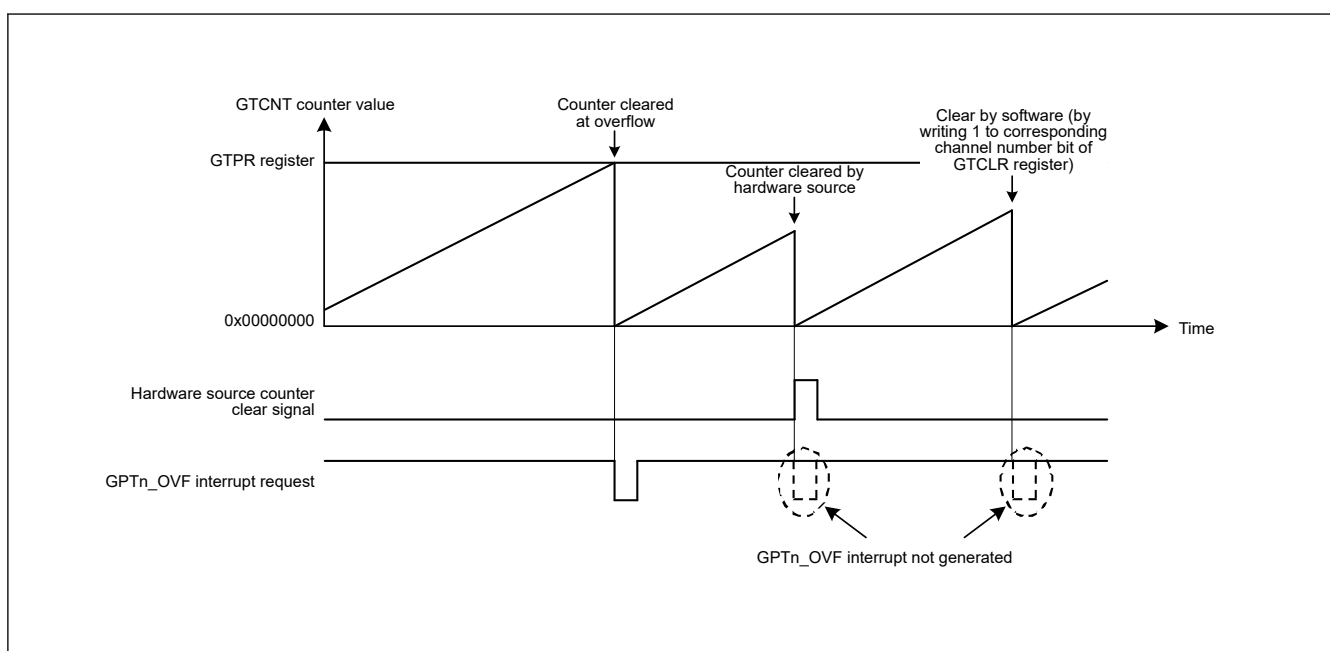


Figure 24.35 Relationship between counter clearing by hardware source and GPTn_OVF (n = 0 to 5) interrupt

24.3.8 Synchronized Operation

Synchronized operation on channels such as a synchronized start, stop, and clear operation can be performed.

24.3.8.1 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 24.36](#) shows an example of a simultaneous start, stop, and clear by software. [Figure 24.37](#) shows an example of phase start operation by software.

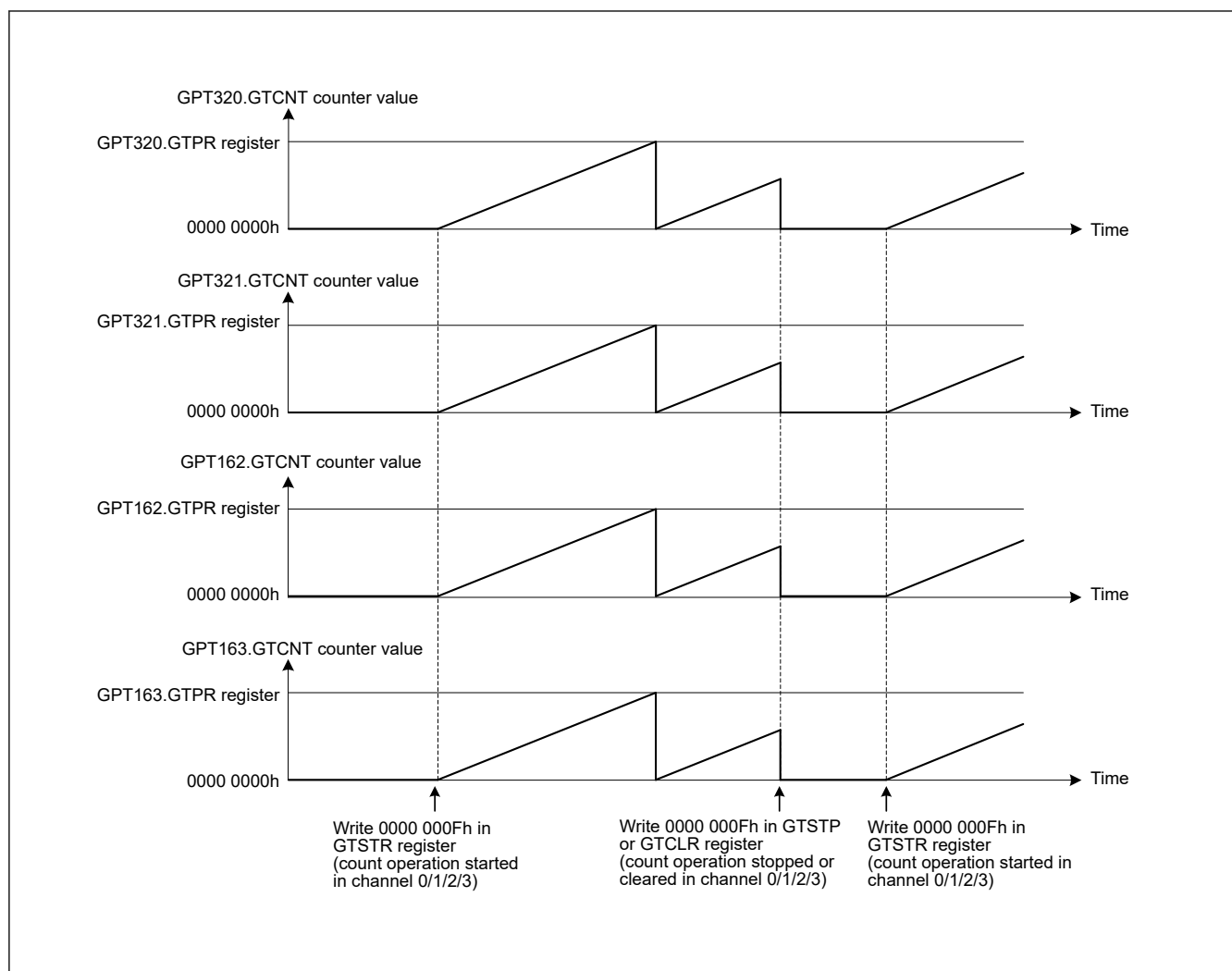


Figure 24.36 Example of a simultaneous start, stop, and clear by software with the same count cycle (GTPR register value)

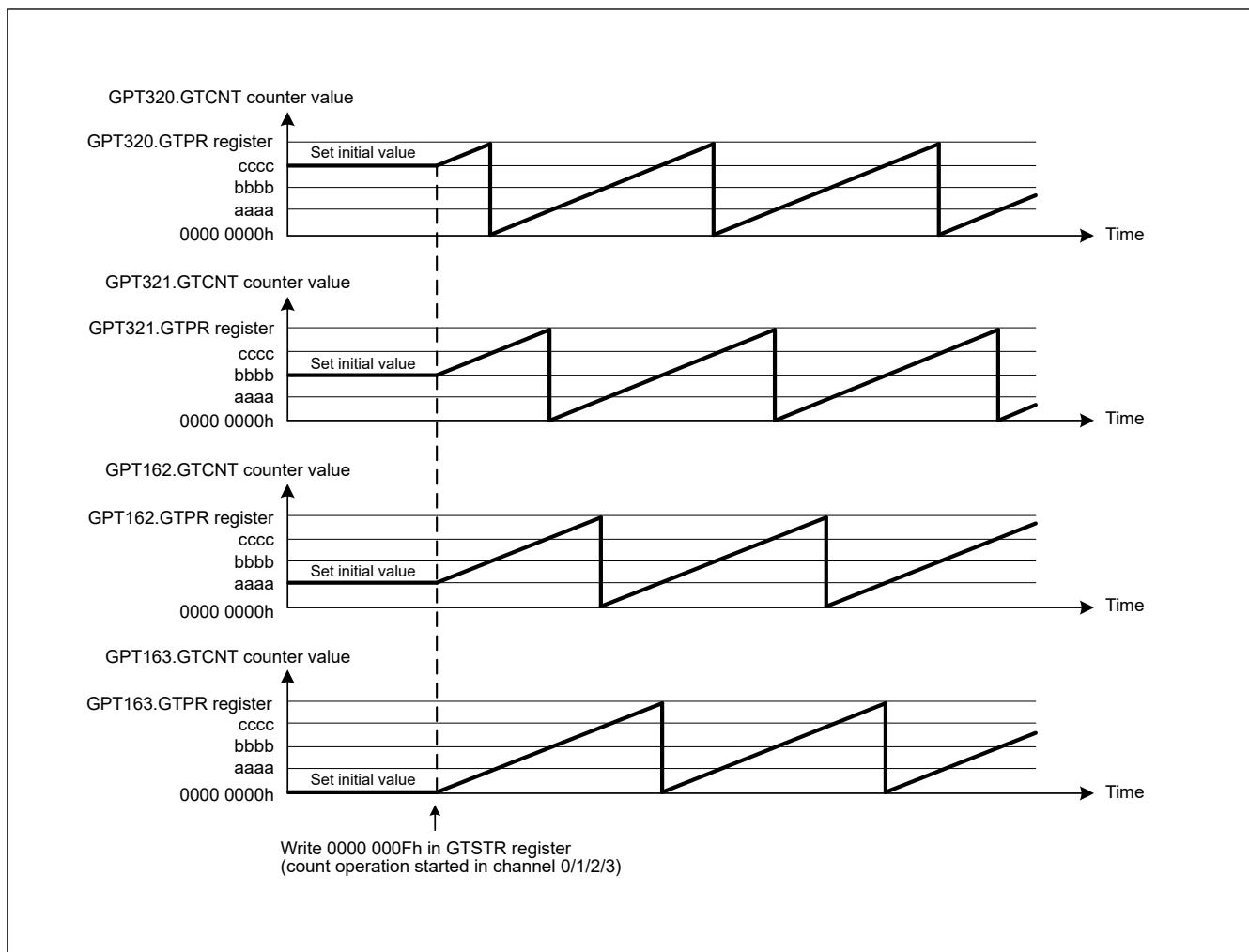


Figure 24.37 Example of software phase start with the same count cycle (GTPR register value)

24.3.8.2 Synchronized operation by hardware

The counters for multiple channels can be started, stopped, and cleared simultaneously by the following hardware sources. Hardware sources that can cause a synchronized operation are external trigger input and ELC event input. Synchronized operations through GTIOCnA and GTIOCnB pin input are possible by setting an ELC event due to input capture as a hardware source ($n = 0$ to 5).

Figure 24.38 shows an example of a simultaneous start, stop, and clear operation by a hardware source. Table 24.28 shows the setting example.

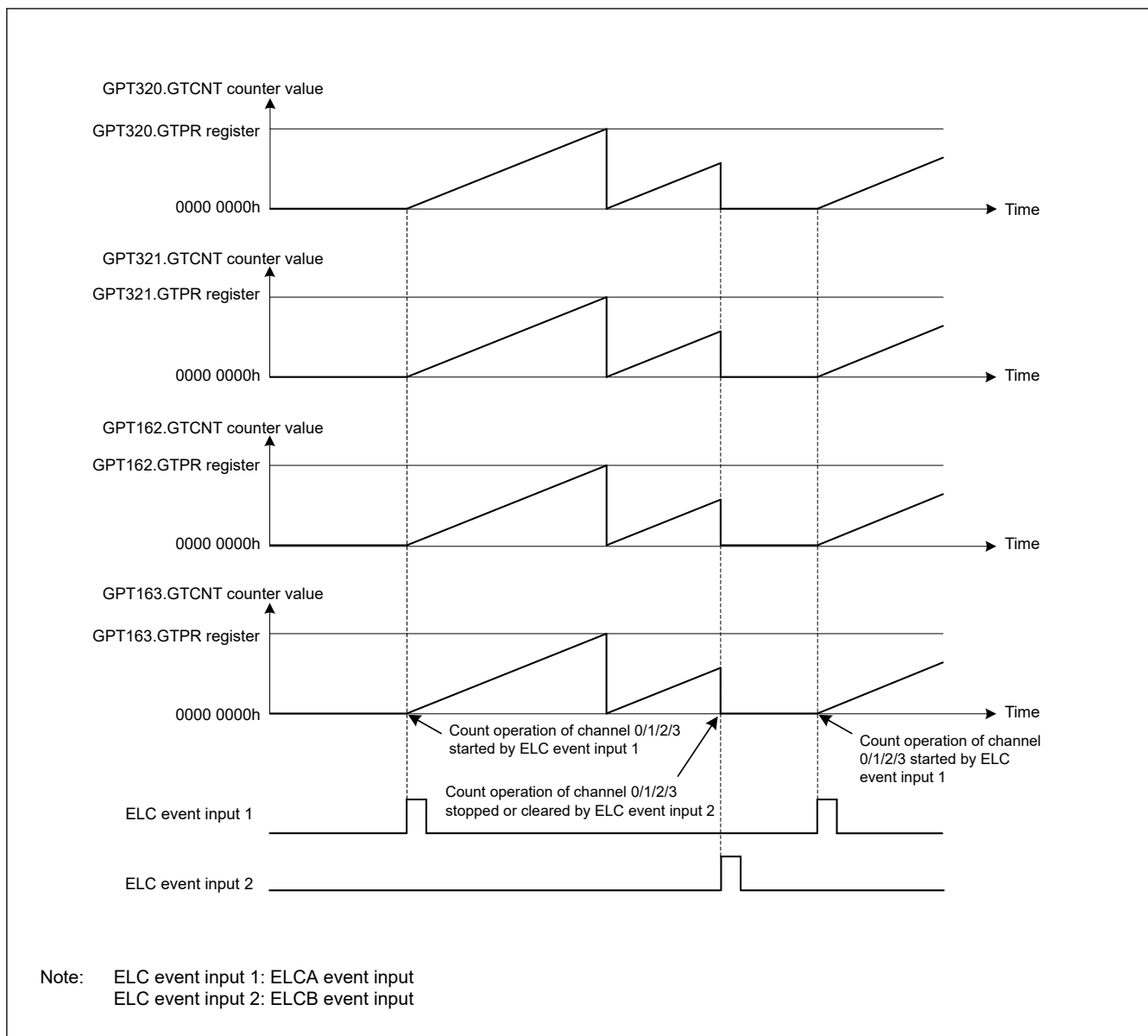


Figure 24.38 Example of a simultaneous start, stop, and clear by a hardware source with the same count cycle (GTPR register value)

Table 24.28 Example setting for simultaneous start by a hardware source (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] In Figure 24.38, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register In Figure 24.38, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[2:0].
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter In Figure 24.38, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation with GTSSR register and wait for count start by the hardware source. In Figure 24.38, GTSSR.SSELCA = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation with GTPSR register and wait for count stop by the hardware source In Figure 24.38, GTPSR.PSELCB = 1.

Table 24.28 Example setting for simultaneous start by a hardware source (2 of 2)

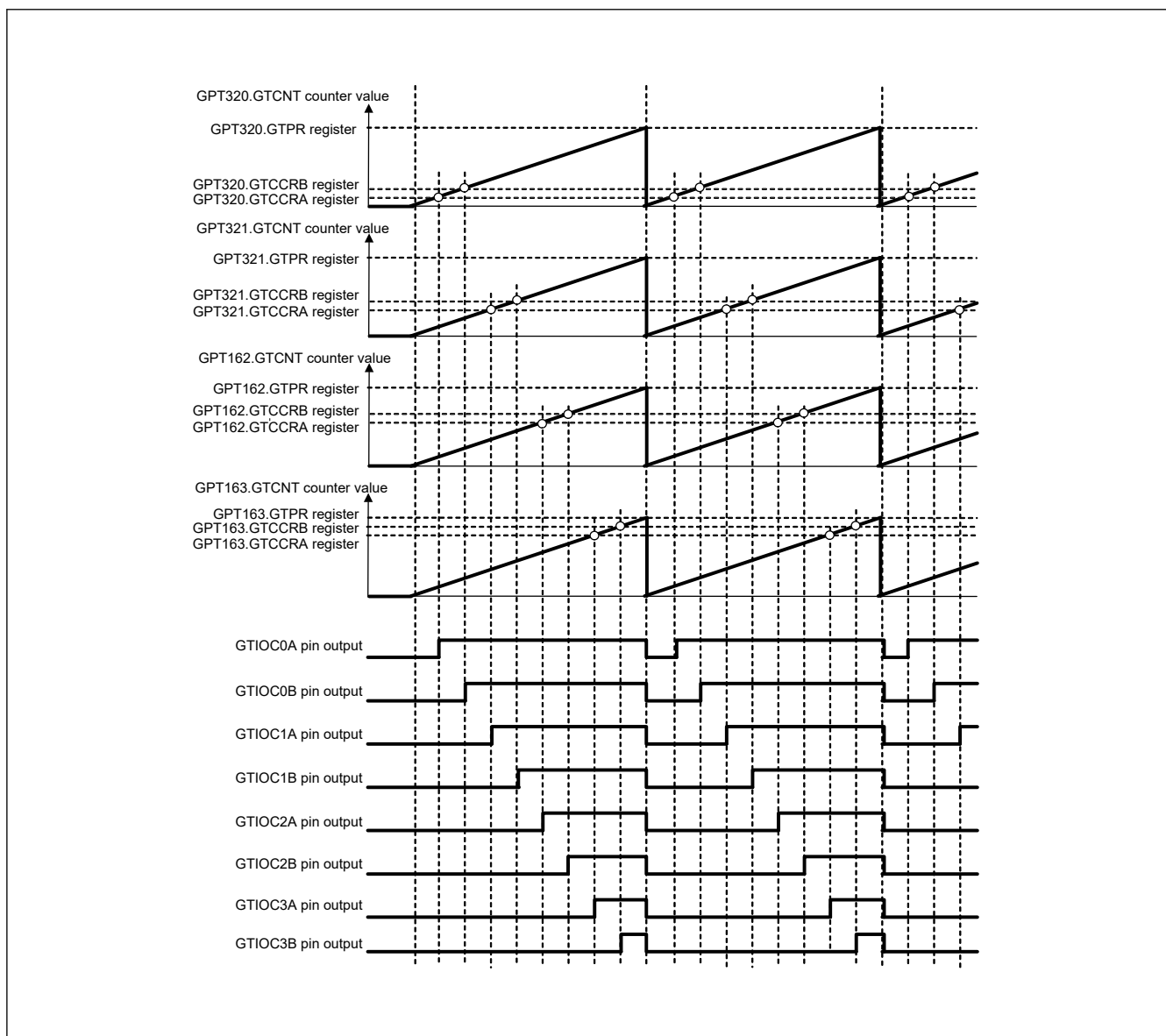
No.	Step Name	Description
8	Set hardware count clear	Select a hardware source for clearing count operation with GTCSR register and wait for count clear by the hardware source In Figure 24.38 , GTCSR.CSELCB = 1.
9	Set hardware source operation	Set operation of the hardware source selected in GTSSR or GTPSR or GTCSR and start or stop or clear counting In Figure 24.38 , ELCA input and ELCB input are set.

24.3.9 PWM Output Operation Examples

(1) Synchronized PWM output

The GPT outputs 6×2 phases of linked PWM waveforms for a maximum of $\text{GPT} \times 6$ channels.

[Figure 24.39](#) shows an example in which four channels perform synchronized operation in saw-wave PWM mode and eight phases of PWM waveforms are output. The GTIOCnA is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCnB is set so that it outputs low as the initial value, high at a GTCCRB compare match, and low at the cycle end.

**Figure 24.39 Example of synchronized PWM output**

(2) 3-phase saw-wave complementary PWM output

Figure 24.40 shows an example in which three channels perform synchronized operation in saw-wave PWM mode and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, low at a GTCCRB compare match, and high at the cycle end.

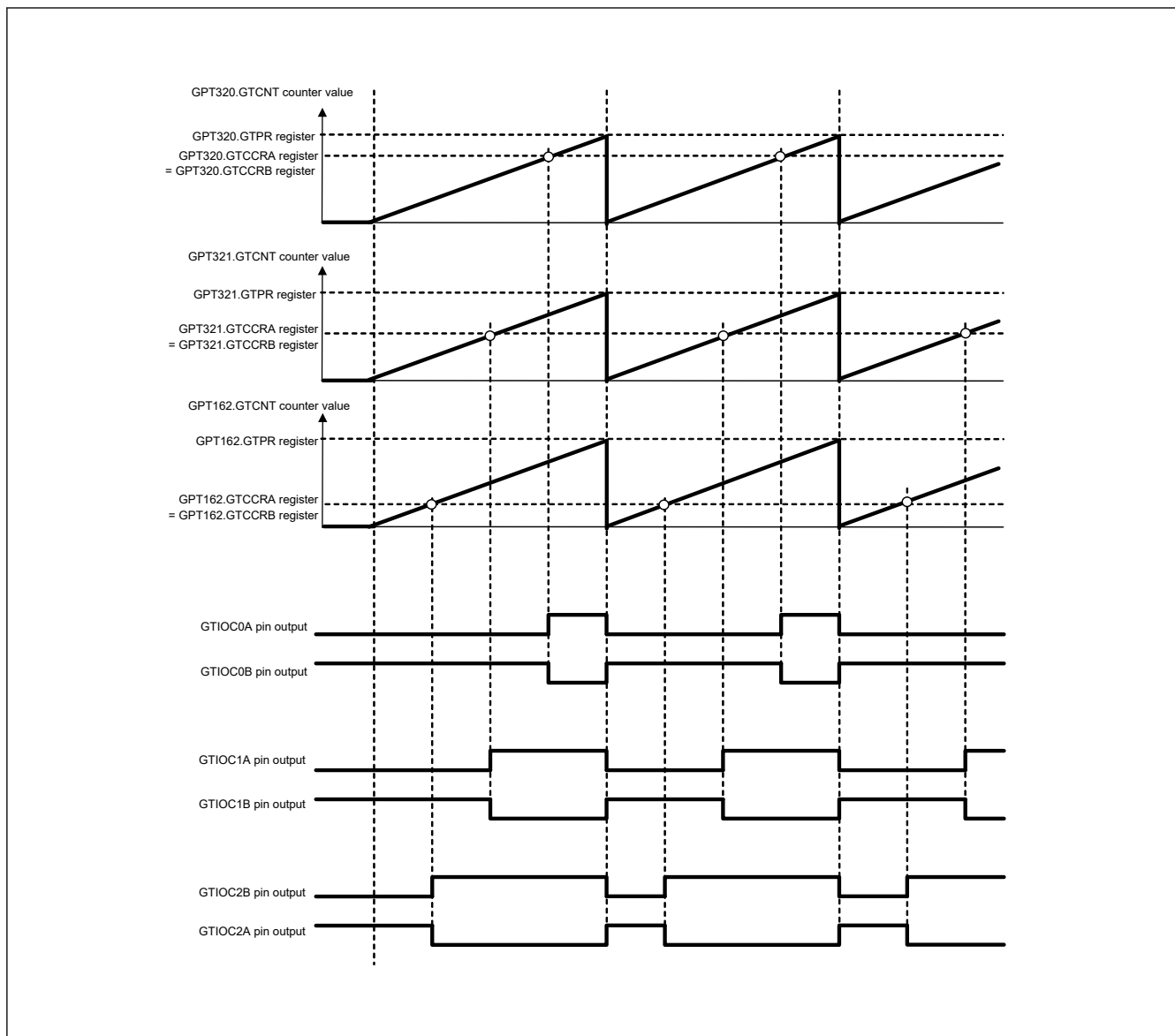


Figure 24.40 Example of 3-phase saw-wave complementary PWM output

(3) 3-phase saw-wave complementary PWM output with automatic dead time setting

Figure 24.41 shows an example in which three channels perform synchronized operation in saw-wave one-shot pulse mode with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTIOCnB compare match, and retains the output at the cycle end.

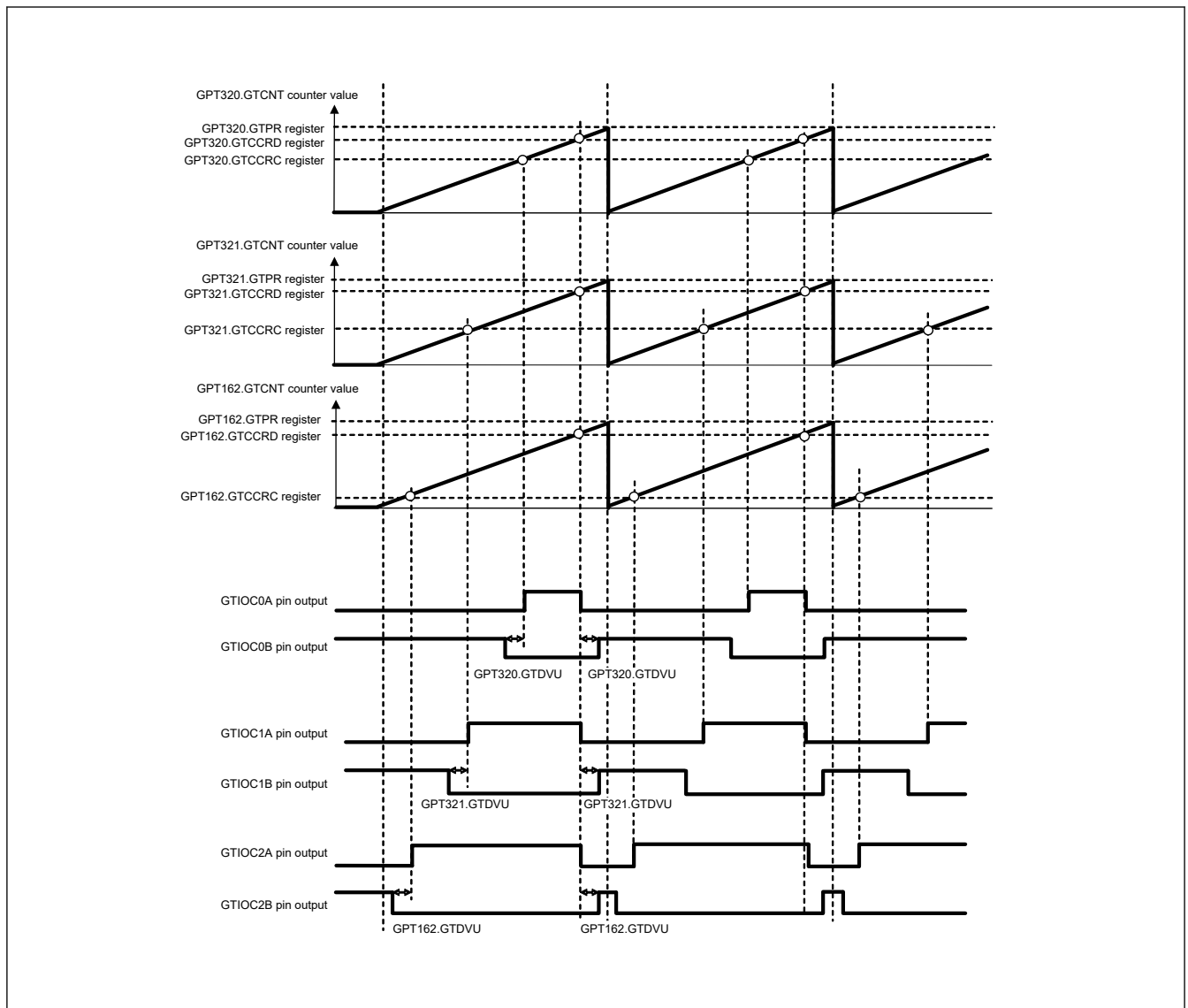


Figure 24.41 Example of 3-phase saw-wave complementary PWM output with automatic dead time setting

(4) 3-phase triangle-wave complementary PWM output

Figure 24.42 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

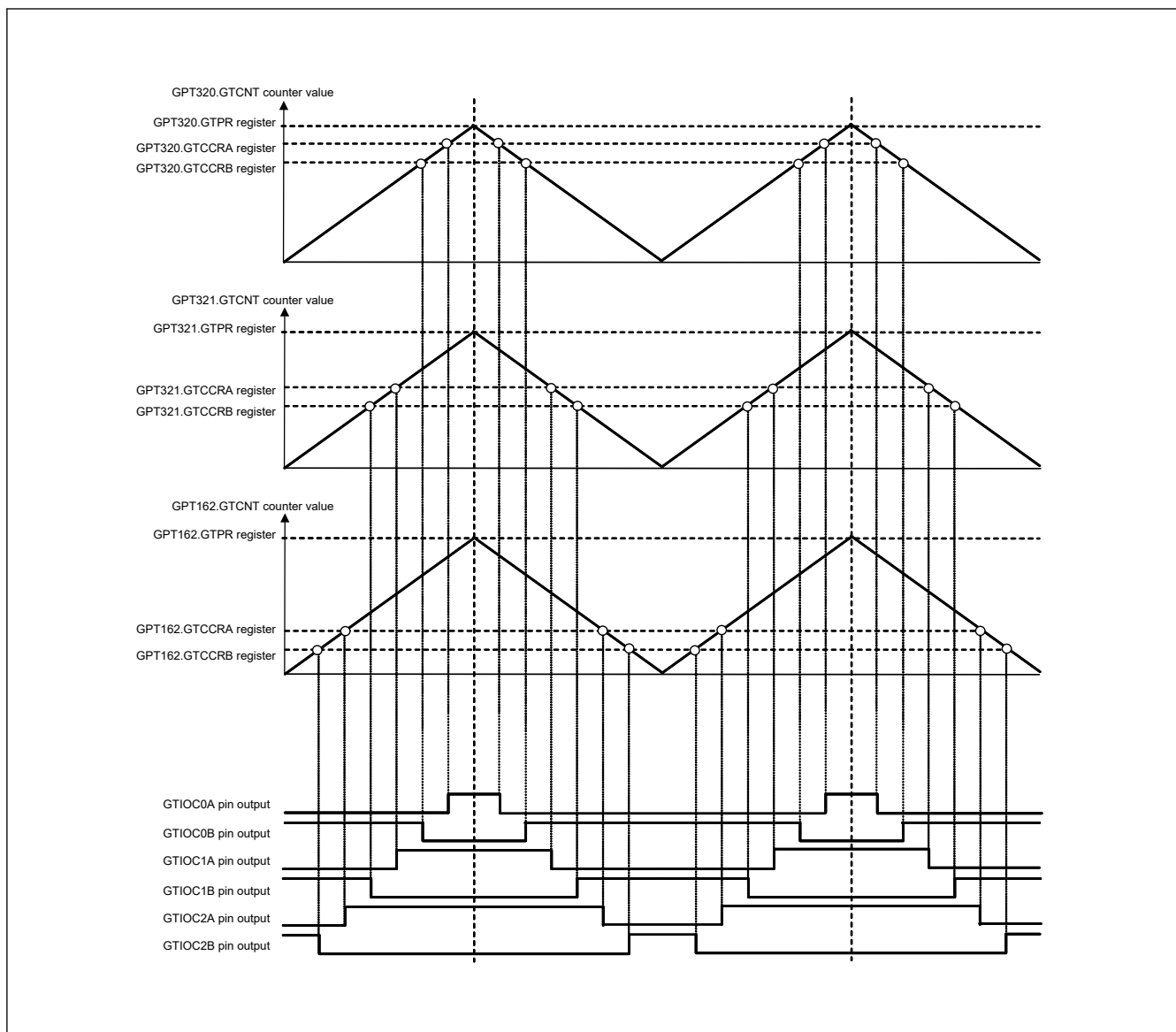


Figure 24.42 Example of 3-phase triangle-wave complementary PWM output

(5) 3-phase triangle-wave complementary PWM output with automatic dead time setting

Figure 24.43 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

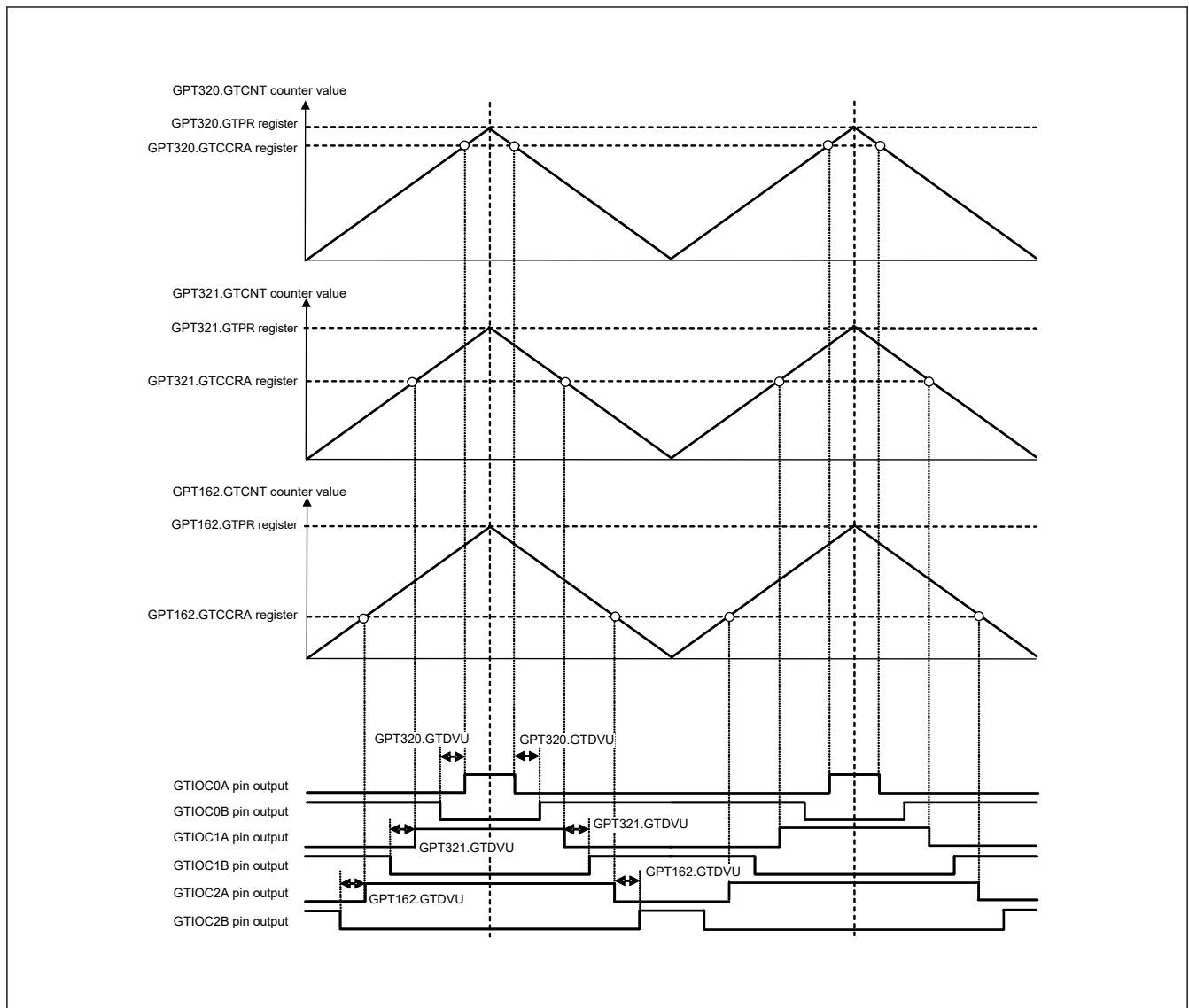


Figure 24.43 Example of 3-phase triangle-wave complementary PWM output with automatic dead time setting

(6) 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

Figure 24.44 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 3 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

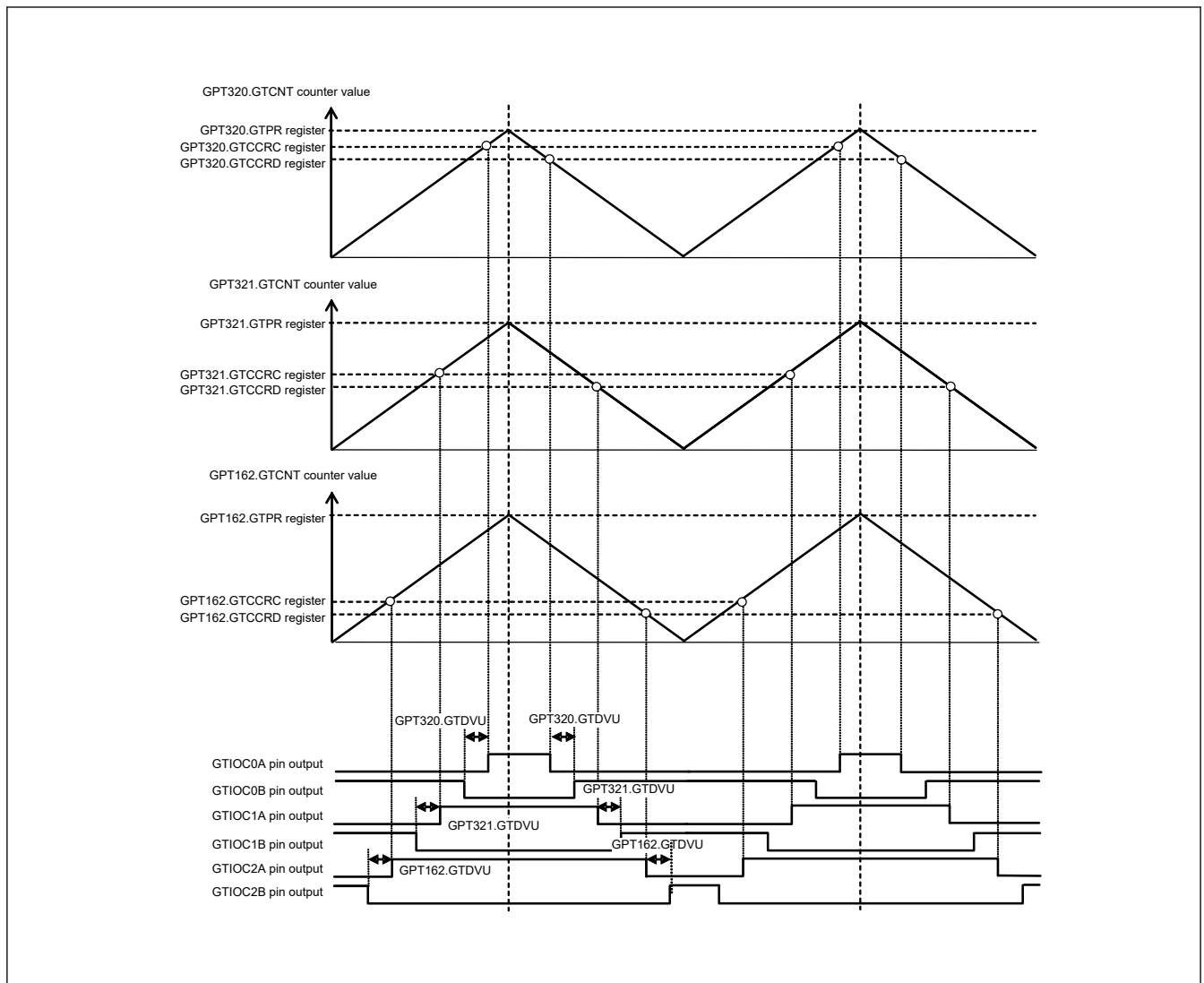


Figure 24.44 Example of 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

24.3.10 Phase Counting Function

The phase difference between the GTIOCnA and GTIOCnB pin ($n = 0$ to 5) inputs is detected and the associated GTCNT counts up or counts down. The detectable phase difference is available in any combination with the relationship between the edge and the level of GTIOCnA and GTIOCnB pin inputs being set in the GTUPSR and GTDNSR registers. For details on count operation, see [section 24.3.1.1. Counter operation](#).

[Figure 24.45](#) to [Figure 24.54](#) show an example of phase counting modes 1 to 5 operation when the GTIOC0A, GTIOC0B pins are used. [Table 24.29](#) to [Table 24.38](#) show conditions of up-counting or down-counting and list settings for the GTUPSR and GTDNSR registers which is corresponding to [Figure 24.45](#) to [Figure 24.54](#).

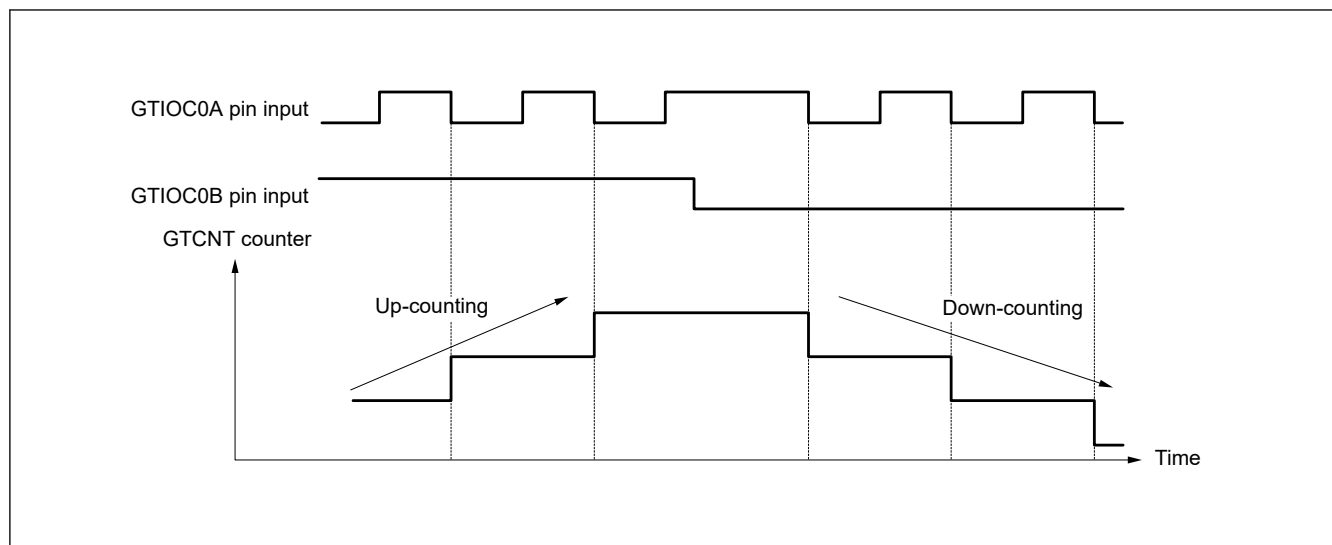


Figure 24.46 Example of phase counting mode 2 (A)

Table 24.30 Conditions of up-counting/down-counting in phase counting mode 2 (A)

 : Rising edge
 : Falling edge

: Falling edge			
GTIOC0A pin input	GTIOC0B pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000800 GTDNSR = 0x00000400
Low			
	Low		
	High	Up-counting	
High		Not counting	
Low			
	High		
	Low	Down-counting	

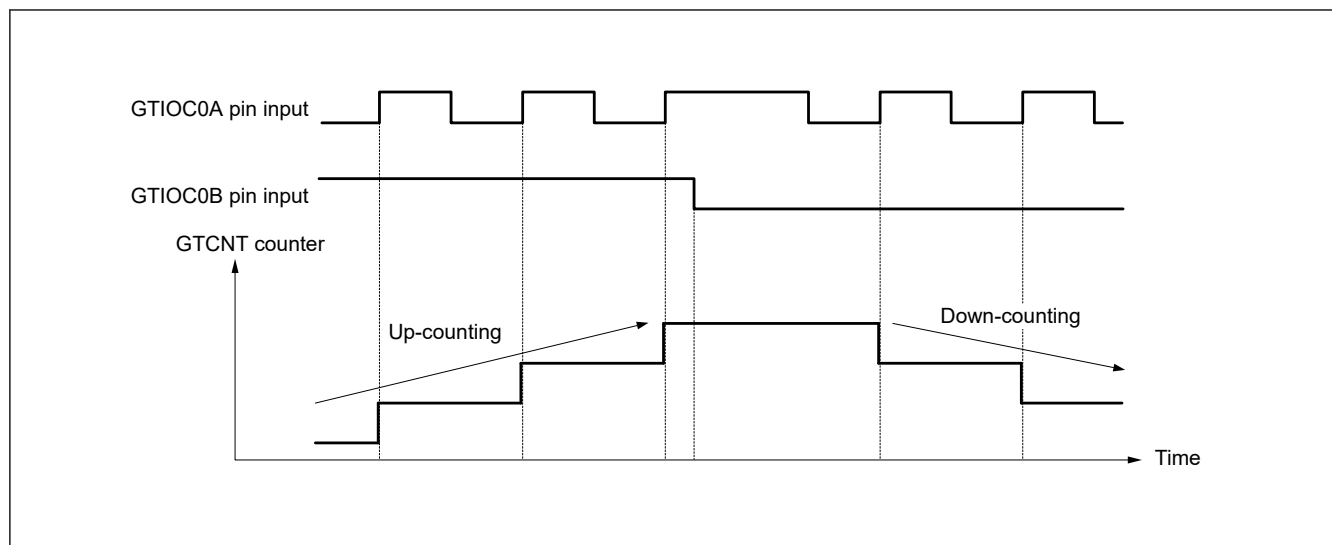


Figure 24.47 Example of phase counting mode 2 (B)

Table 24.31 Conditions of up-counting/down-counting in phase counting mode 2 (B)

 : Rising edge
 : Falling edge

Timing edge			
GTIOC0A pin input	GTIOC0B pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000200 GTDNSR = 0x00000100
Low			
	Low	Down-counting	
	High	Not counting	
High			
Low			
	High	Up-counting	
	Low	Not counting	

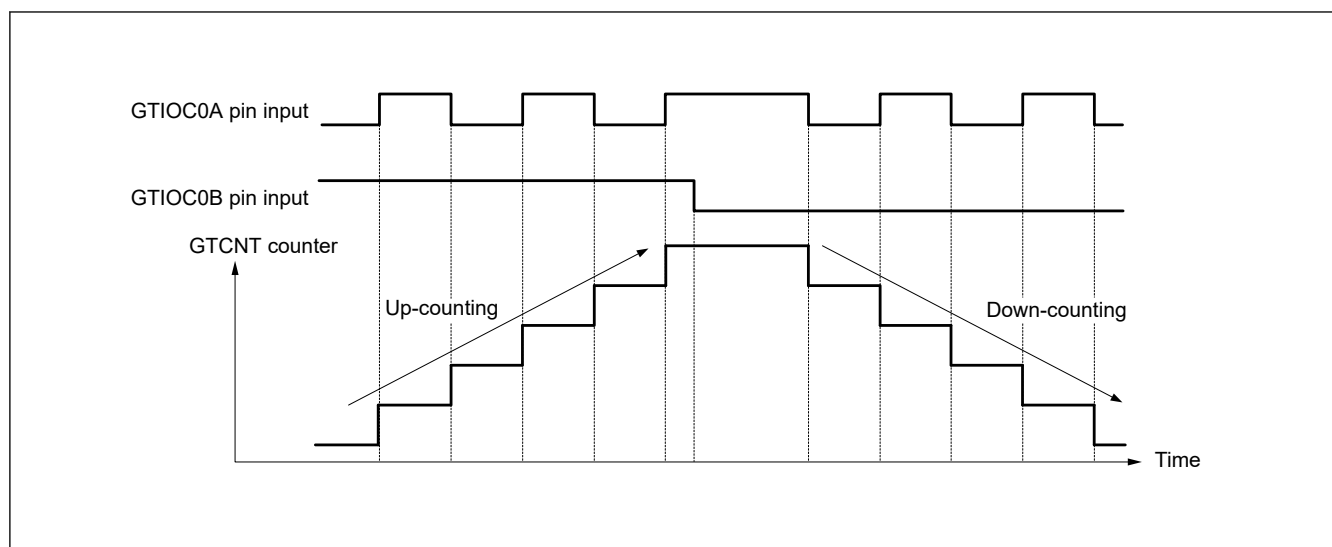


Figure 24.48 Example of phase counting mode 2 (C)

Table 24.32 Conditions of up-counting/down-counting in phase counting mode 2 (C)

 : Rising edge
 : Falling edge

GTIOC0A pin input	GTIOC0B pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000A00 GTDNSR = 0x00000500
Low			
	Low	Down-counting	
	High	Up-counting	
High		Not counting	
Low			
	High	Up-counting	
	Low	Down-counting	

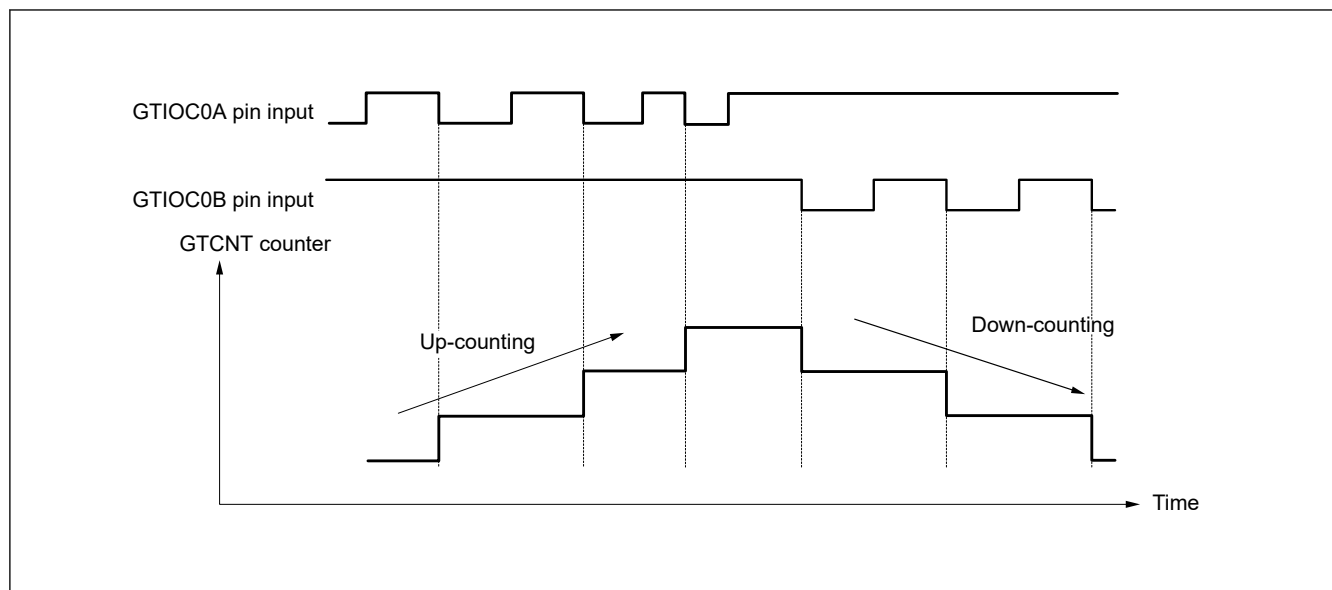


Figure 24.49 Example of phase counting mode 3 (A)

Table 24.33 Conditions of up-counting/down-counting in phase counting mode 3 (A)

 : Rising edge
 : Falling edge

= : Falling edge			
GTIOC0A pin input	GTIOC0B pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000800 GTDNSR = 0x00008000
Low			
	Low		
	High	Up-counting	
High		Down-counting	
Low		Not counting	
	High		
	Low		

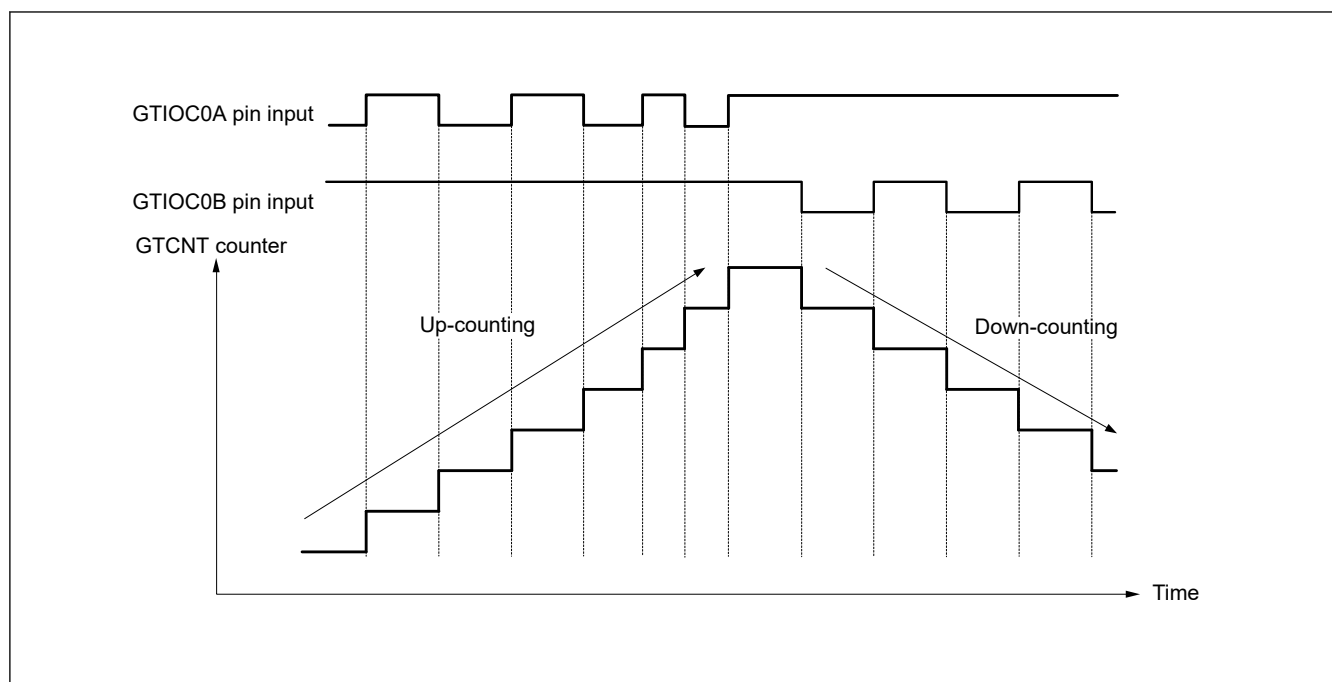


Figure 24.51 Example of phase counting mode 3 (C)

Table 24.35 Conditions of up-counting/down-counting in phase counting mode 3 (C)

 : Rising edge
 : Falling edge

GTIOC0A pin input	GTIOC0B pin input	Operation	Register setting
High		Down-counting	GTUPSR = 0x00000A00 GTDNSR = 0x0000A000
Low		Not counting	
	Low	Not counting	
	High	Up-counting	
High		Down-counting	
Low		Not counting	
	High	Up-counting	
	Low	Not counting	

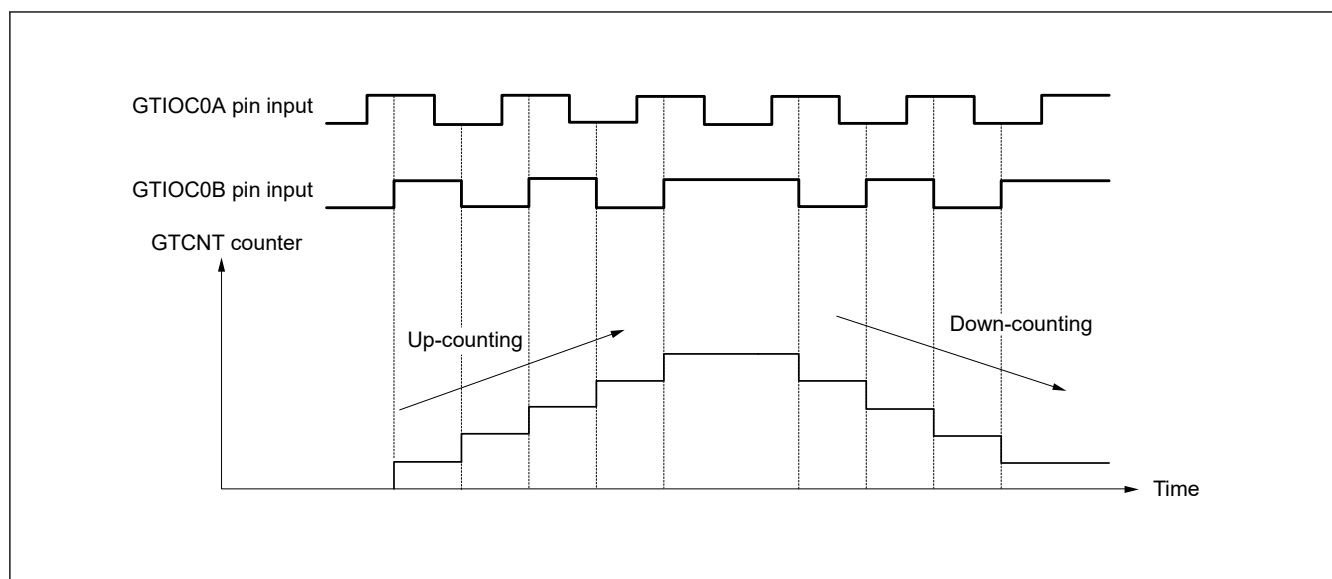


Figure 24.52 Example of phase counting mode 4

Table 24.36 Conditions of up-counting/down-counting in phase counting mode 4

 : Rising edge
 : Falling edge

GTIOC0A pin input	GTIOC0B pin input	Operation	Register setting
High		Up-counting	GTUPSR = 0x00006000 GTDNSR = 0x00009000
Low		Up-counting	
	Low	Not counting	
	High	Not counting	
High		Down-counting	
Low		Down-counting	
	High	Not counting	
	Low	Not counting	

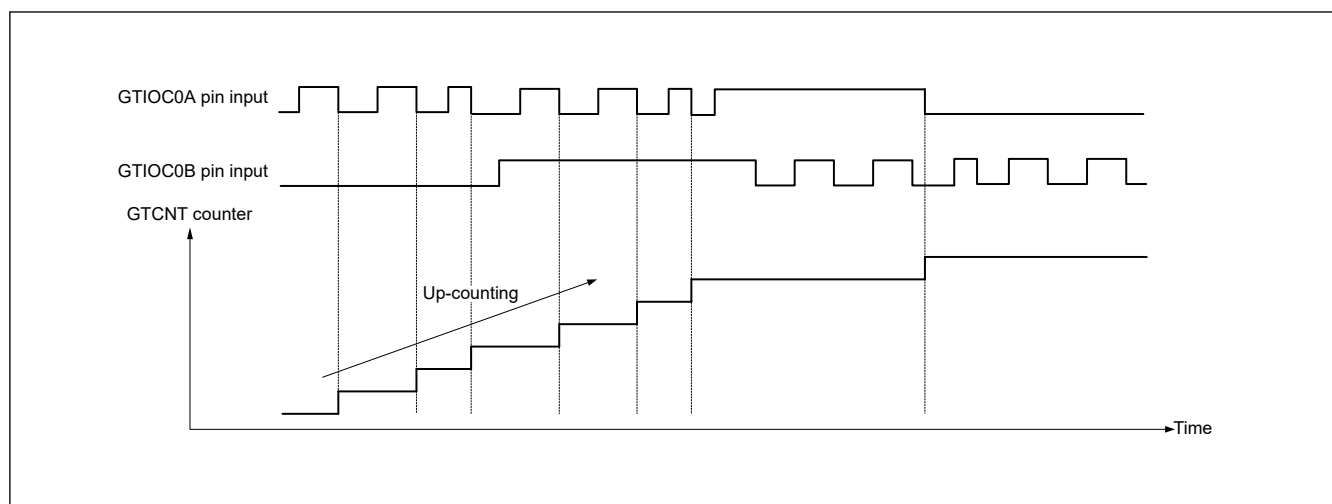


Figure 24.53 Example of phase counting mode 5 (A)

Table 24.37 Conditions of up-counting/down-counting in phase counting mode 5 (A)

 : Rising edge
 : Falling edge

= : Falling edge

GTIOC0A pin input	GTIOC0B pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000C00 GTDNSR = 0x00000000
Low			
	Low		
	High	Up-counting	
High		Not counting	
Low			
	High		
	Low	Up-counting	

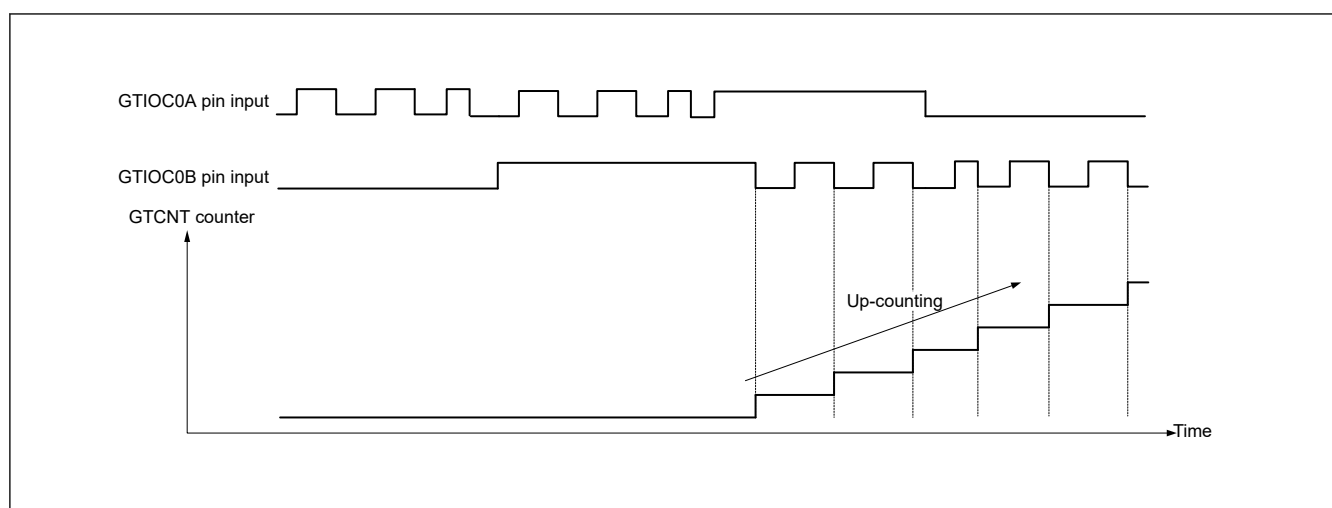


Figure 24.54 Example of phase counting mode 5 (B)

Table 24.38 Conditions of up-counting/down-counting in phase counting mode 5 (B)

 : Rising edge
 : Falling edge

GTIOC0A pin input	GTIOC0B pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x0000C000 GTDNSR = 0x00000000
Low		Up-counting	
	Low	Not counting	
	High		
High		Up-counting	
Low		Not counting	
	High		
	Low		

24.3.11 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 GPT162.GTIOC2A.

Figure 24.55 shows the conceptual diagram of GPT_OPS control flow.

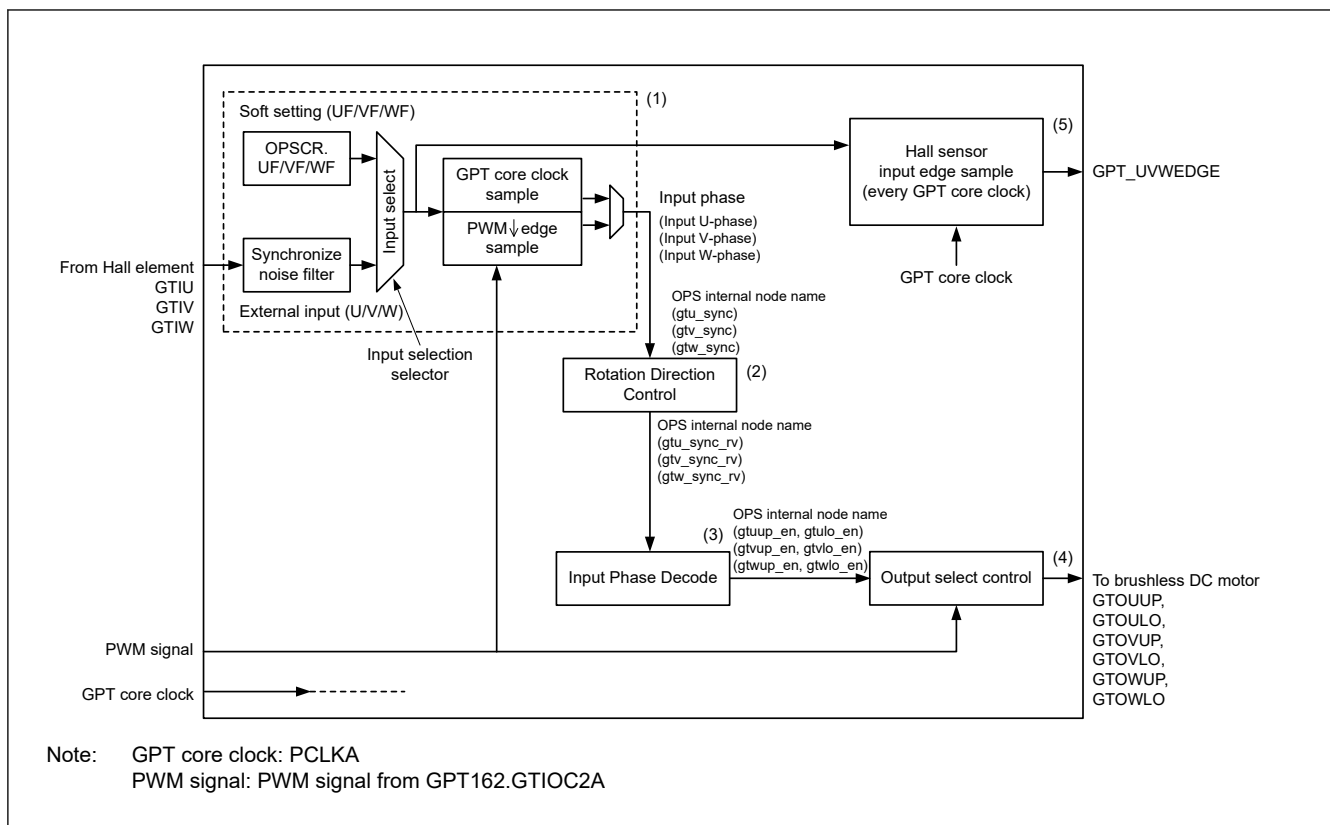


Figure 24.55 Conceptual diagram of GPT_OPS control flow

Figure 24.56 shows a 6-phase level signals output example of a GPT_OPS operation.

The GPT_UVWEDGE signal in Figure 24.56 is the Hall sensor input edge to ELC output.

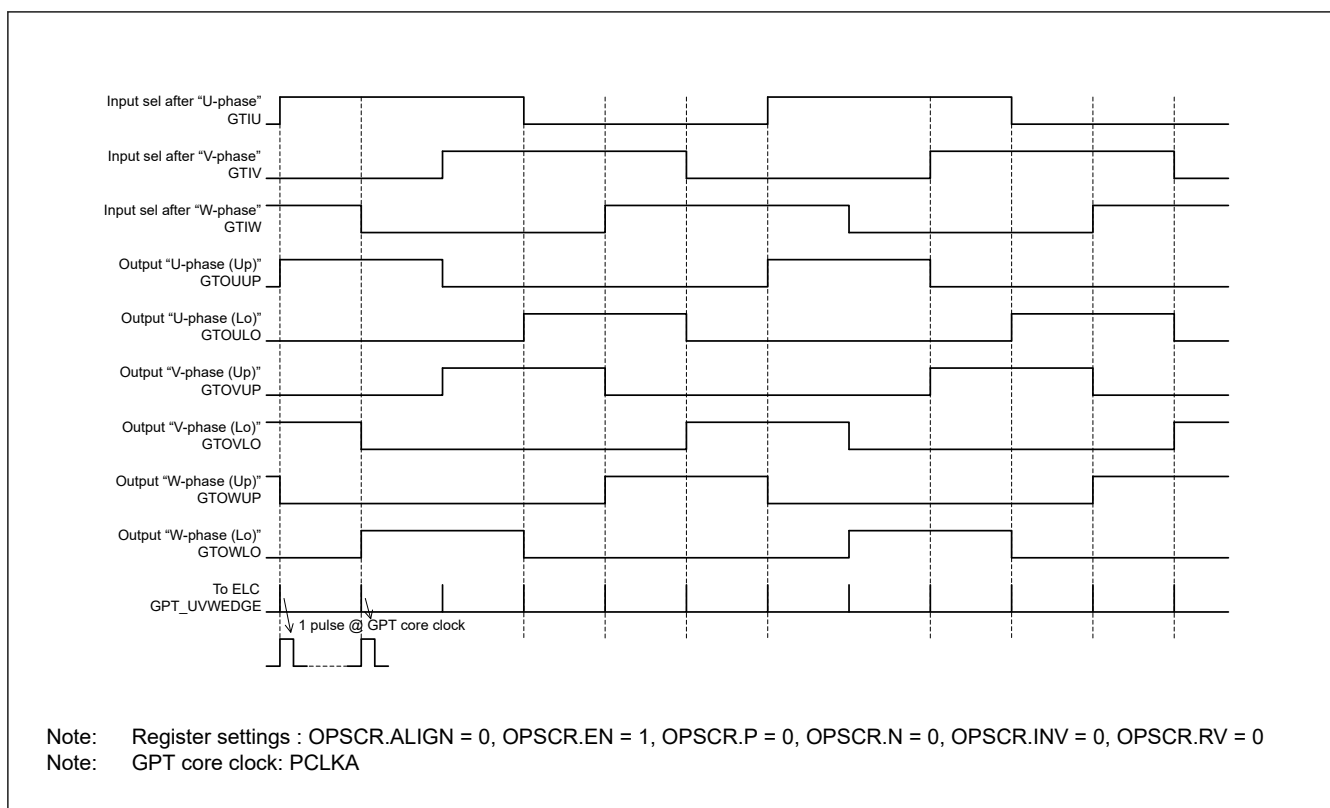
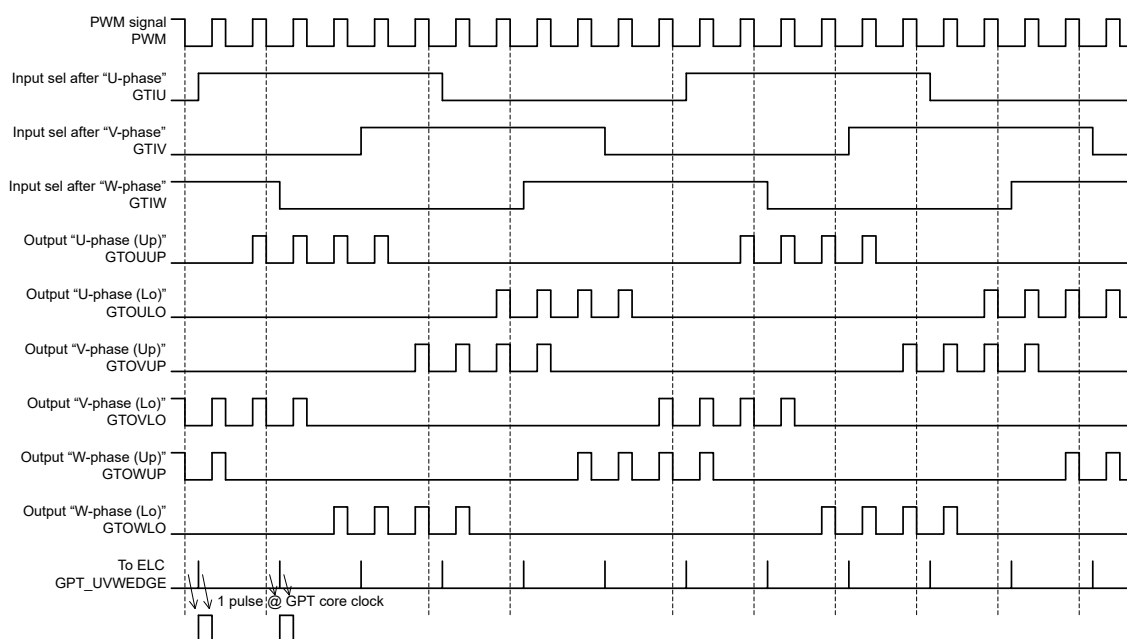


Figure 24.56 Example of 6-phase level output operation

Figure 24.57 shows a 6-phase PWM output example of a GPT_OPS operation with chopper control.



Note: Register settings : OPSCR.ALIGN = 1, OPSCR.EN = 1, OPSCR.P = 1, OPSCR.N = 1, OPSCR.INV = 0, OPSCR.RV = 0
 Note: GPT core clock: PCLKA
 PWM signal: PWM signal from GPT162.GTIOC2A

Figure 24.57 Example of 6-phase PWM output operation with chopper control

Figure 24.58 shows a 6-phase PWM output example of an output disable control operation.

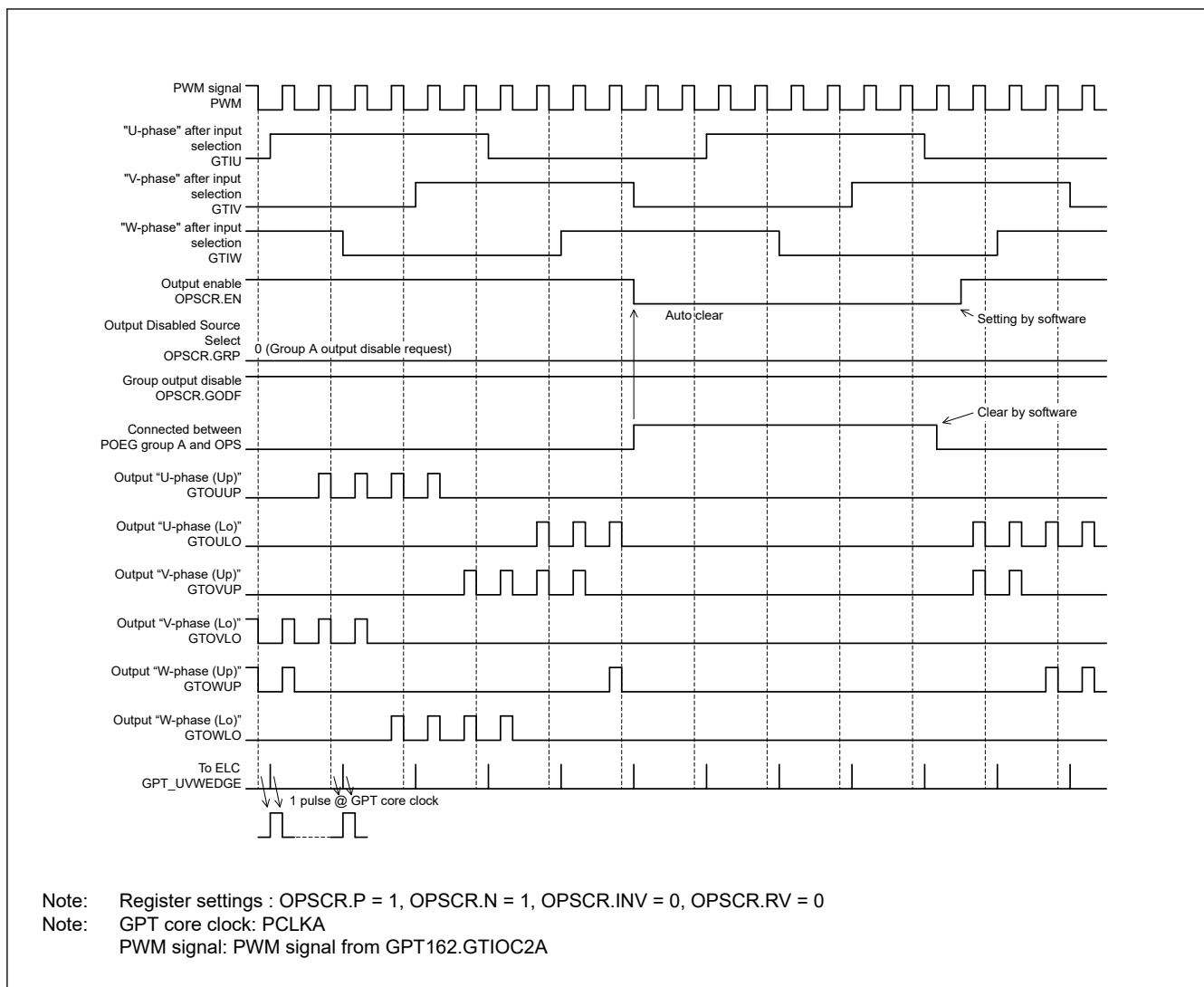


Figure 24.58 Example of group output disable control operation

24.3.11.1 Input selection and synchronization of external input signal

In the GPT_OPS control flow conceptual diagram shown in [Figure 24.55](#), (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 (PCLKA). After carrying out noise filtering (optional), set the external input to the input phase of PWM (PWM of GPT162.GTIOC2A) using falling edge sampling with OPSCR.ALIGN bit set to 1.

When OPSCR.FB bit is 1, select the soft setting (OPSCR.UF, VF, WF) with the value of the input phase of PWM (PWM of GPT162.GTIOC2A) 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 PCLKA synchronization with either OPSCR.FB bit set to 0 or OPSCR.FB bit set to 1. However, there are cases where the PWM pulse width of the output U/V/W phases (PWM output mode) of switch timing (just before/just after) is shortened.

[Table 24.39](#) shows the input selection process and setting of associated OPSCR bits.

Table 24.39 Input selection processing method

Register OPSCR		Selection of input phase sampling method (U/V/W-phase)	Synchronization input/output selection process (GPT_OPS internal node name)
FB bit	ALIGN bit		
0	1	External Input at PWM Falling Edge Sampling (PCLKA 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 PCLKA Synchronization Output (PCLKA 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) (= PCLKA synchronization)	

24.3.11.2 Input sampling

The OPSCR.U, V, W bits indicate the PCLKA sampling results of the input selected in the OPSCR.FB bit.

When OPSCR.FB bit is 0 and after synchronization with the GPT core clock (PCLKA) 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 are the value (OPSCR.UF, VF, WF) of the soft setting.

24.3.11.3 Input phase decode

In the GPT_OPS control flow conceptual diagram shown in [Figure 24.55](#), (3) enables the 6-phase signals by decoding the input phase selected in the OPSCR.FB bit.

[Table 24.40](#) shows the decode table of input phase when OPSCR.RV bit is 0.

Table 24.40 Decode table of input phase (OPSCR.RV = 0)

Input phase (U/V/W) (GPT_OPS internal node name)			6-phase enable {U/V/W (Up/Lo)} by decoding input phase (GPT_OPS internal node name)					
Input U-Phase	Input V-Phase	Input W-Phase	U-phase (Up)	U-phase (Lo)	V-phase (Up)	V-phase (Lo)	W-phase (Up)	W-phase (Lo)
(gtu_sync)	(gtv_sync)	(gtw_sync)	(gtuup_en)	(gtulo_en)	(gtvup_en)	(gtvlo_en)	(gtwup_en)	(gtwlo_en)
1	0	1	1	0	0	1	0	0
1	0	0	1	0	0	0	0	1
1	1	0	0	0	1	0	0	1
0	1	0	0	1	1	0	0	0
0	1	1	0	1	0	0	1	0
0	0	1	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

Table 24.41 Decode table of input phase (OPSCR.RV = 1) (1 of 2)

Input phase (U/V/W) (GPT_OPS internal node name)			6-phase enable {U/V/W (Up/Lo)} by decoding input phase (GPT_OPS internal node name)					
Input U-Phase	Input V-Phase	Input W-Phase	U-phase (Up)	U-phase (Lo)	V-phase (Up)	V-phase (Lo)	W-phase (Up)	W-phase (Lo)
(gtu_sync)	(gtv_sync)	(gtw_sync)	(gtuup_en)	(gtulo_en)	(gtvup_en)	(gtvlo_en)	(gtwup_en)	(gtwlo_en)
1	0	1	0	1	1	0	0	0
1	0	0	0	1	0	0	1	0
1	1	0	0	0	0	1	1	0
0	1	0	1	0	0	1	0	0

Table 24.41 Decode table of input phase (OPSCR.RV = 1) (2 of 2)

Input phase (U/V/W) (GPT_OPS internal node name)			6-phase enable {U/V/W (Up/Lo)} by decoding input phase (GPT_OPS internal node name)					
Input U-Phase	Input V-Phase	Input W-Phase	U-phase (Up)	U-phase (Lo)	V-phase (Up)	V-phase (Lo)	W-phase (Up)	W-phase (Lo)
(gtu_sync)	(gtv_sync)	(gtw_sync)	(gtuup_en)	(gtulo_en)	(gtvup_en)	(gtvlo_en)	(gtwup_en)	(gtwlo_en)
0	1	1	1	0	0	0	0	1
0	0	1	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

24.3.11.4 Rotation Direction Control

In the GPT_OPS control flow conceptual diagram shown in [Figure 24.55](#), (2) controls the direction of rotation of a 3-phase motor using the OPSCR.RV bit.

When the rotation direction is reverse (RV bit = 1), the input phase is inverted.

[Table 24.42](#) shows the assigned output phases based on the OPSCR.RV bit setting (before and after rotation direction control).

Table 24.42 Rotation Direction Control Method

Reversal of Direction of Rotation Using Output Phases as Specified in OPSCR Register	Output of Rotation Direction Control [U/V/W (Positive/Negative)] (GPT_OPS Internal Node Name after Control)					
	(gtuup_en)	(gtulo_en)	(gtvup_en)	(gtvlo_en)	(gtwup_en)	(gtwlo_en)
OPSCR.RV bit	(gtuup_en)	(gtulo_en)	(gtvup_en)	(gtvlo_en)	(gtwup_en)	(gtwlo_en)
0	U-phase (Up) (gtuup_en)	U-phase (Lo) (gtulo_en)	V-phase (Up) (gtvup_en)	V-phase (Lo) (gtvlo_en)	W-phase (Up) (gtwup_en)	W-phase (Lo) (gtwlo_en)
1	U-phase (Up) (gtuup_en)	U-phase (Lo) (gtulo_en)	W-phase (Up) (gtwup_en)	W-phase (Lo) (gtwlo_en)	V-phase (Up) (gtvup_en)	V-phase (Lo) (gtvlo_en)

24.3.11.5 Output selection control

In the GPT_OPS control flow conceptual diagram in [Figure 24.55](#), (4) represents the selection of the output waveform by setting the OPSCR register bit.

For output selection, the following bits are relevant:

- The OPSCR.EN bit controls whether to output the 6-phase output, or to stop
- The OPSCR.P and OPSCR.N bits can select from the level signal or PWM signal (chopper output) for the output phase
- The polarity of the output phase can be set to a positive logic or negative logic by the OPSCR.INV bit.

[Table 24.43](#) and [Table 24.44](#) show the output selection control method using the OPSCR register bit.

Table 24.43 Output selection control method (positive phase) (1 of 2)

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

Table 24.43 Output selection control method (positive phase) (2 of 2)

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
1	0	0	Level signal (gtuup_ren) (gtvup_ren) (gtwup_ren)	Level Output Mode (Positive phase) (Positive logic)
1	0	1	Level signal (~gtuup_ren) (~gtvup_ren) (~gtwup_ren)	Level Output Mode (Positive phase) (Negative logic)
1	1	0	PWM signal (PWM & gtuup_ren) (PWM & gtvup_ren) (PWM & gtwup_ren)	PWM Output Mode (Positive phase) (Positive logic)
1	1	1	PWM signal (~(PWM & gtuup_ren)) (~(PWM & gtvup_ren)) (~(PWM & gtwup_ren))	PWM Output Mode (Positive phase) (Negative logic)

Table 24.44 Output selection control method (negative phase)

Enable-phase output control	Positive-phase output (N) control	Invert-phase output control	Output port name (negative phase = Lo) (output selection internal node allocation)	
OPSCR.EN	OPSCR.N	OPSCR.INV	GTOULO GTOVLO GTOWLO	Mode
0	x	x	0	Output Stop (External pin: Hi-Z) GPT_OPS → 0 output
1	0	0	Level signal (gtulo_ren) (gtvlo_ren) (gtwlo_ren)	Level Output Mode (Negative phase) (Positive logic)
1	0	1	Level signal (~gtulo_ren) (~gtvlo_ren) (~gtwlo_ren)	Level Output Mode (Negative phase) (Negative logic)
1	1	0	PWM signal (PWM & gtulo_ren) (PWM & gtvlo_ren) (PWM & gtwlo_ren)	PWM Output Mode (Negative phase) (Positive logic)
1	1	1	PWM signal (~(PWM & gtulo_ren)) (~(PWM & gtvlo_ren)) (~(PWM & gtwlo_ren))	PWM Output Mode (Negative phase) (Negative logic)

24.3.11.6 Output selection control (group output disable function)

When OPSCR.GODF is 1 and the signal value selected by the OPSCR.GRP bit is high (output disable request), the group output-disable function asynchronously sets the output to Hi-Z. When an output-disable request is generated, the OPSCR.EN bit is cleared to 0. For the return, set the OPSCR.EN bit to 1 after clearing the output disable request by software.

To ensure output-disable control, use the POEG_GROUPn (n = A, B) interrupt to clear the flag in the POE or check that the OPSCR.EN bit is 0 and then clear the flag. For an example of the operation of group output disable control, see [Figure 24.58](#).

24.3.11.7 Event Link Controller (ELC) output

In the GPT_OPS control flow conceptual diagram shown in [Figure 24.55](#), (5) outputs the Hall sensor input signal edge to the ELC.

The Hall sensor input edge signal is the logical OR of the rising and falling edge signals of each U-phase/V-phase/W-phase input sampled at PCLKA. 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 PCLKA.

When OPSCR.FB bit is 1, the Hall sensor input edge signal is the logical OR of the edge of the soft setting (OPSCR.UF, VF, WF) sampled at PCLKA.

See [Figure 24.56](#) to [Figure 24.58](#) for examples of the output signal to the ELC.

24.3.11.8 GPT_OPS start operation setting flow

Table 24.45 Example setting of GPT_OPS start operation

No.	Step Name	Description
1	GPT162 operation mode setting	GPT162.GTIOC2A set the PWM output operation mode of the saw-wave or triangle-wave. For details, see section 24.3.3. PWM Output Operating Mode .
2	Counting of GPT162	Start the count operation of GPT162, and outputs a PWM waveform.
3	GPT_OPS input data set (only software setting is selected)	Set software setting to OPSCR.UF, VF, and WF bits.
4	Noise filter settings of GPT_OPS external input (only external input is selected)	When using a noise filter, set the sampling clock of the noise filter by OPSCR.NFCS[1:0] bits. Then the noise filter is enabled if OPSCR.NFEN = 1.
5	GPT_OPS input phase selection setting/input phase alignment setting	Select the input phase from the external input or software setting by OPSCR.FB bit. Select the alignment of the input phase by OPSCR.ALIGN bit.
6	Setting the GPT_OPS output phase	Set the level output/PWM output of the positive/negative phase output by OPSCR.P/OPSCR.N bit. Set the positive logic/negative logic of the output phase by OPSCR.INV bit. Set the rotation direction by OPSCR.RV bit.
7	GPT_OPS setting the group output disable function	Set the selection of output disable source by OPSCR.GRP bit. Perform the setting of on/off of the group output disable function by OPSCR.GODF bit.
8	GPT_OPS Working	Setting the OPSCR.EN = 1 outputs the 6-phase output to drive the brushless DC motor from the GPT_OPS.

24.4 Interrupt Sources

24.4.1 Interrupt Sources

The GPT provides the following interrupt sources:

- GTCCR input capture/compare match
- GTCNT counter overflow (GTPR compare match)/underflow.

Each interrupt source has its own status flag. When an interrupt source signal is generated, the associated status flag in GTST is set to 1. The associated status flag in GTST can be cleared by writing 0. If flag set and flag clear occur at the same time, flag clear takes priority over flag set. These flags are automatically updated by the internal state. The Interrupt Controller Unit can change the relative channel priorities. However, the priority within a channel is fixed. For details, see [section 16, Interrupt Controller Unit \(ICU\)](#).

[Table 24.46](#) lists the GPT interrupt sources.

Table 24.46 Interrupt sources

Channel	Name	Interrupt source	Interrupt flag	DTC activation
n = 0 to 1	GPTn_CCMPA	GPT32n.GTCCRA input capture/compare match	GTST[0] (TCFA)	Possible
	GPTn_CCMPB	GPT32n.GTCCRB input capture/compare match	GTST[1] (TCFB)	Possible
	GPTn_CMPC	GPT32n.GTCCRC compare match	GTST[2] (TCFC)	Possible
	GPTn_CMPD	GPT32n.GTCCRD compare match	GTST[3] (TCFD)	Possible
	GPTn_OVF	GPT32n.GTCNT overflow (GPT32n.GTPR compare match)	GTST[6] (TCFPO)	Possible
	GPTn_UDF	GPT32n.GTCNT underflow	GTST[7] (TCFPU)	Possible
n = 2 to 5	GPTn_CCMPA	GPT16n.GTCCRA input capture/compare match	GTST[0] (TCFA)	Possible
	GPTn_CCMPB	GPT16n.GTCCRB input capture/compare match	GTST[1] (TCFB)	Possible
	GPTn_CMPC	GPT16n.GTCCRC compare match	GTST[2] (TCFC)	Possible
	GPTn_CMPD	GPT16n.GTCCRD compare match	GTST[3] (TCFD)	Possible
	GPTn_OVF	GPT16n.GTCNT overflow (GPT16n.GTPR compare match)	GTST[6] (TCFPO)	Possible
	GPTn_UDF	GPT16n.GTCNT underflow	GTST[7] (TCFPU)	Possible

Note: This table shows the initial state immediately after a reset. The Interrupt Controller Unit can change the relative channel priorities.

(1) GPTn_CCMPA interrupt (n = 0 to 5)

An interrupt request is generated under the following conditions:

- When the GTCCRA register functions as a compare match register, the GTCNT counter value matches with the GTCCRA register
- When the GTCCRA register functions as an input capture register, the input-capture signal causes transfer of the GTCNT counter value to the GTCCRA register.

(2) GPTn_CCMPB interrupt (n = 0 to 5)

An interrupt request is generated under the following conditions:

- When the GTCCRB register functions as a compare match register, the GTCNT counter value matches with the GTCCRB register
- When the GTCCRB register functions as an input capture register, the input-capture signal causes transfer of the GTCNT counter value to the GTCCRB register.

(3) GPTn_CMPC interrupt (n = 0 to 5)

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 5)

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 5)

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 5)

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.

About Interrupt signals and interrupt status flags, see [section 24.2.16. GTST : General PWM Timer Status Register](#).

24.4.2 DMAC and DTC Activation

The DMAC and DTC can be activated by the interrupt in each channel. For details, see [section 16, Interrupt Controller Unit \(ICU\)](#), [section 19, DMA Controller \(DMAC\)](#), and [section 20, Data Transfer Controller \(DTC\)](#).

24.5 Operations Linked by ELC

24.5.1 Event Signal Output to ELC

The GPT can perform operation linked with another module set in advance when its interrupt request signal is used as an event signal by the Event Link Controller (ELC).

The GPT has the following ELC event signals:

- Generation of compare match and input capture A interrupt (GPTn_CCMPIA)
- Generation of compare match and input capture B interrupt (GPTn_CCMPIB)
- 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 5

24.5.2 Event Signal Inputs from ELC

The GPT can perform the following operations in response to a maximum of 4 events from the ELC:

- Start counting, stop counting, clear counting
- Up-counting, down-counting
- Input capture.

See [section 21, Event Link Controller \(ELC\)](#) for the connection between the ELC and the event signal input.

24.6 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 24.59 shows the timing of noise filtering.

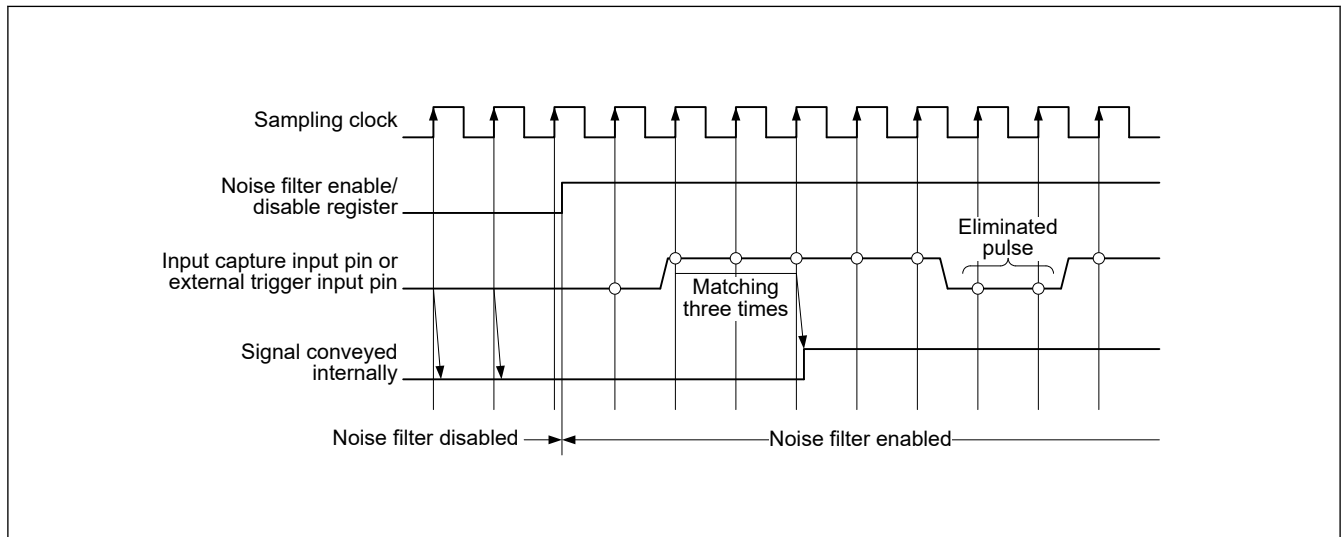


Figure 24.59 Timing of noise filtering

If noise filtering is enabled, the input capture operation or external trigger operation is performed on the edges of the noise filtered signal after a delay of a sampling interval $\times 2 + \text{PCLKA}$. This is due to the noise filtering for the input capture input or external trigger operation.

24.7 Protection Function

24.7.1 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, GTCSSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTDTCR, GTDVU.

Protection using the GTWP register is only for write operations by the CPU. This protection does not cover updates to registers that occur in association with CPU writes.

24.7.2 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 buffer register write.

Figure 24.60 shows an example of operation for disabling buffer operation.

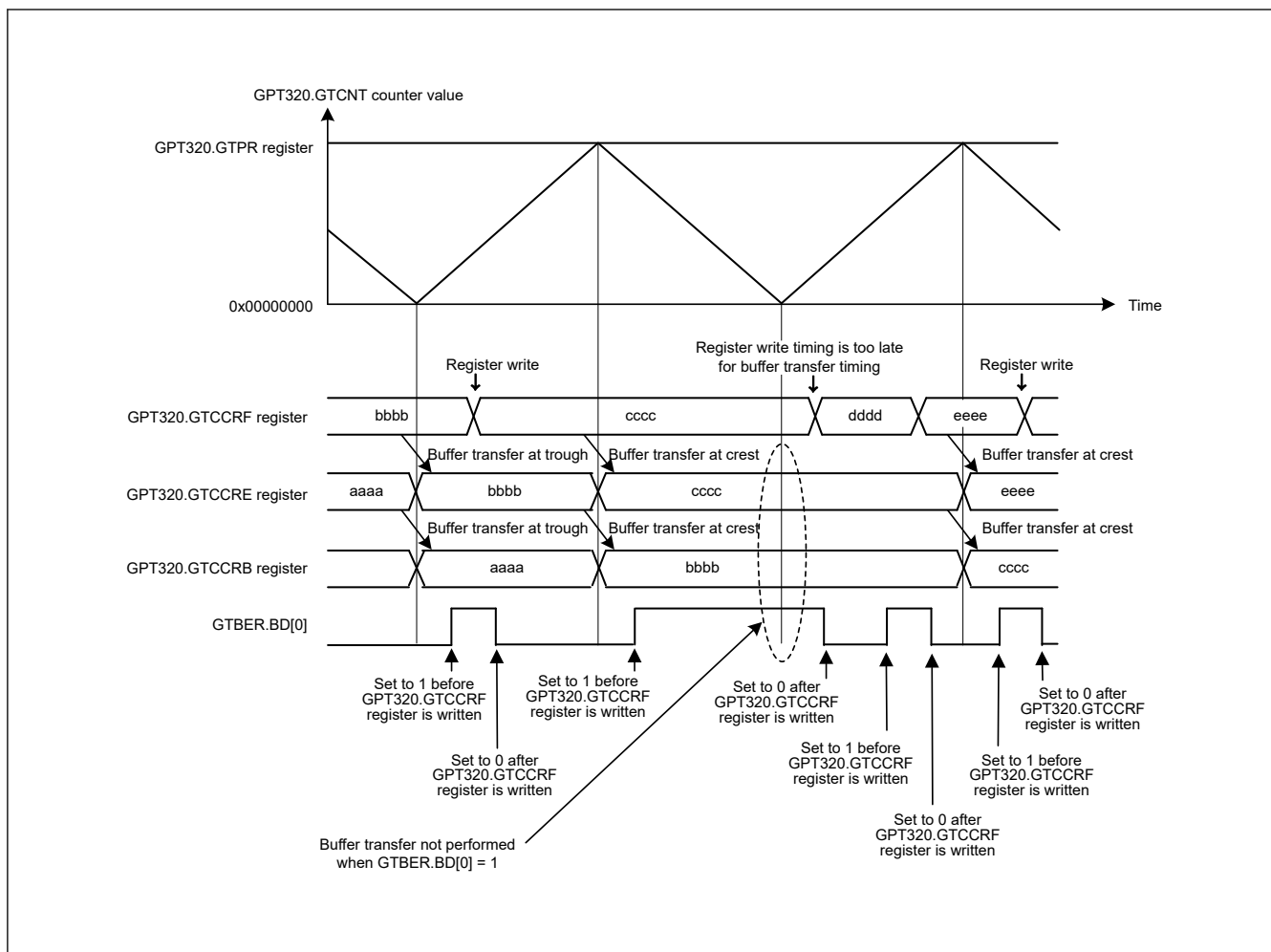


Figure 24.60 Example of operation for disabling buffer operation with triangle waves, double buffer operation, and buffer transfer at both troughs and crests

24.7.3 GTIOCnm Pin Output Negate Control (n = 0 to 5, m = A, B)

For protection from system failure, the output disable control that changes the GTIOCnm pin output value forcibly is provided for GTIOCnm pin output by the request of output disable from POEG. When the GTIOCnA pin output value is the same as the GTIOCnB pin output value, output protection is required. GPT detects this condition and generates output disable requests to POEG according to the setting of the output disable request permission bits, such as GTINTAD.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 GTIOCnA pin and the GTIOCnB pin) out of 2 output disable requests generated by the POEG is selected by setting GTINTAD.GRP[1:0]. The status of the selected disable output request is monitored by reading the GTST.ODF bit. The output level during output disable is set based on the GTIOR.OADF[1:0] bits for the GTIOCnA pin and the GTIOR.OBDF[1:0] setting for the GTIOCnB pin.

The change to the output disable state is performed asynchronously by generating the output disable request from the POEG. The release of the output disable state is performed at end of cycle by terminating the output disable request.

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] should be set to 00b (for GTIOCnA pin) or GTIOR.OBDF[1:0] should be set to 00b (for the GTIOCnB pin).

Figure 24.61 shows an example of the GTIOCnm pin output disable control operation. (n = 0 to 5, m = A, B)

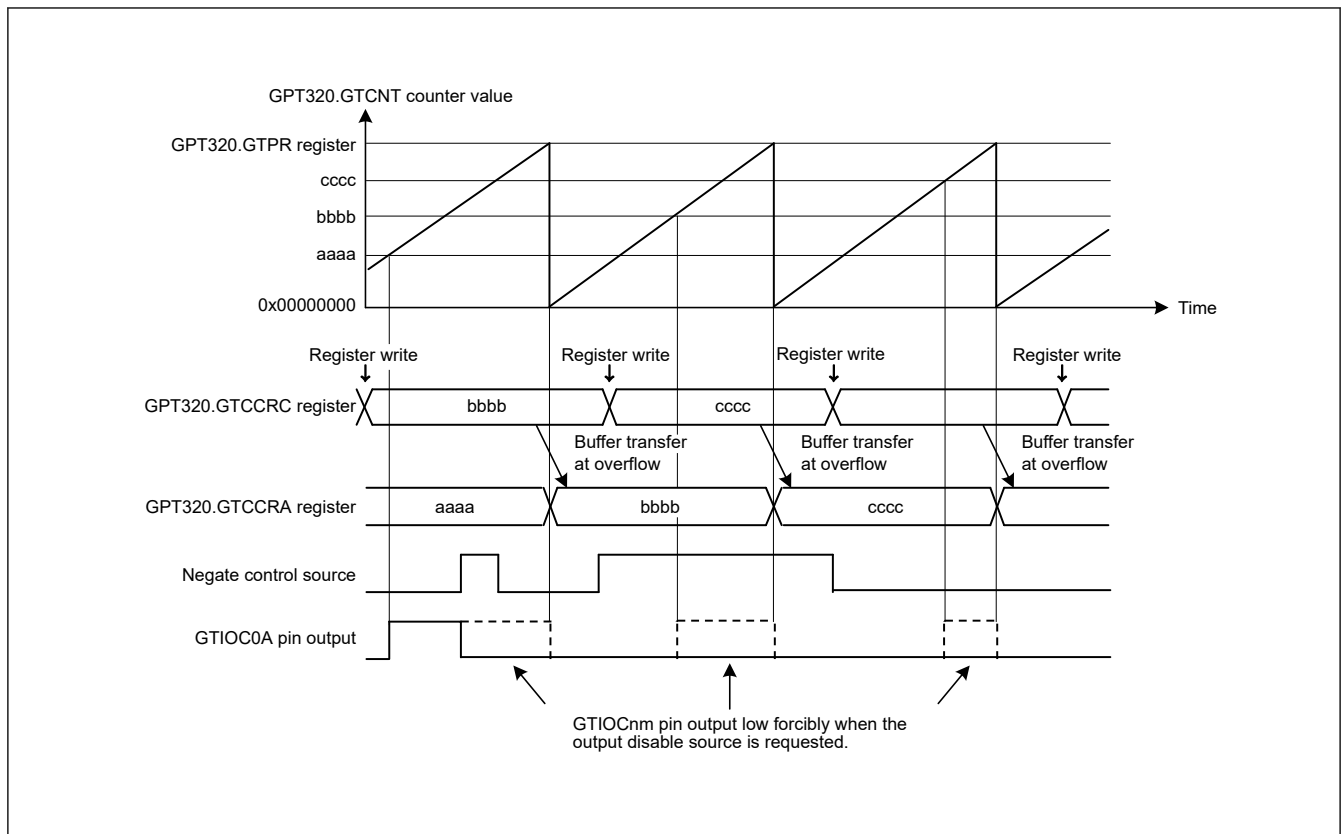


Figure 24.61 Example of GTIOCnm pin output disable control operation in saw-wave up-counting, buffer operation, active level 1, high output at GTCCRA compare match, low output at cycle end, and low output at output disable ($n = 0$ to 5, $m = A, B$)

24.8 Initialization Method of Output Pins

24.8.1 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.

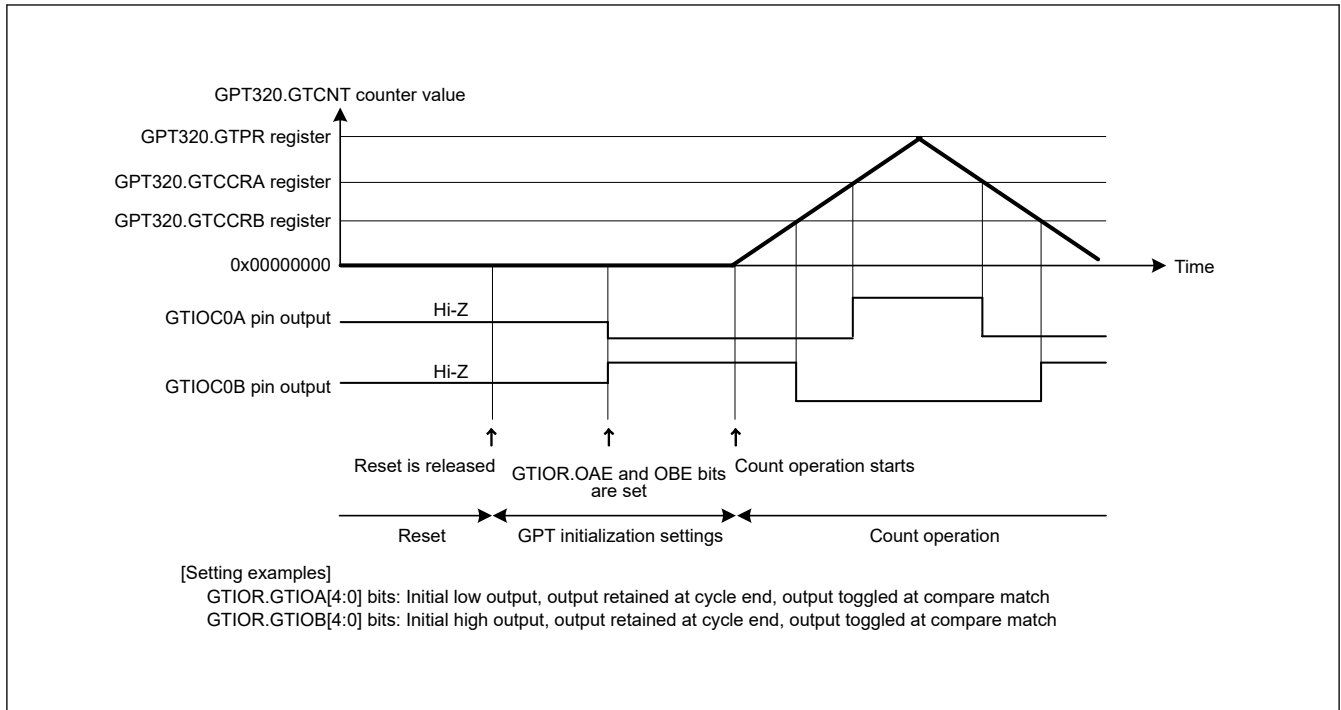


Figure 24.62 Example of pin settings after reset

24.8.2 Pin Initialization Due to Error during Operation

If an error occurs during GPT operation, the following four types of pin processing can be performed before pin initialization:

- Set the OAHLD and OBHLD bits in GTIOR to 1 and retain the outputs at count stop
- Set the OAHLD and OBHLD bits in GTIOR to 0, specify arbitrary output values at OADFLT and OBDFLT in GTIOR, and output the arbitrary values at count stop
- Set the pin to output an arbitrary value as a general output port by setting the PDR, PODR registers and PmnPFS.PMR bit of the I/O port in advance. Set the OAE and OBE bits in GTIOR to 0, and the control bit associated with the pin in the PMR to 0 to allow arbitrary values to be output from the pin set as a general output port when an error occurs.
- Drive the output to a high impedance state using the POEG function.

After 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.

24.9 Usage Notes

24.9.1 Module-Stop Function Setting

The Module Stop Control Register can enable or disable GPT operation. The GPT is initially stopped after a reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

24.9.2 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:

- $GTDVU < GTCCRA$
- $0 < GTCCRA < GTPR$

When the setting of $GTCCRA = 0$ or $GTCCRA \geq GTPR$ is made for the $GTCCRA$ register during count operation, the output protection function is activated. However, if the following condition is not satisfied, the output protection function does not work normally:

- The value of the $GTCCRA$ register at the start of counting is larger than 0 and less than $GTPR$.

For details, see [section 24.7.3. GTIOCnm Pin Output Negate Control \(n = 0 to 5, m = A, B\)](#)

(2) When automatic dead time setting is not made in triangle-wave PWM mode

The $GTCCRA$ register must be set within the range of $0 < GTCCRA < GTPR$. If $GTCCRA = 0$ or $GTCCRA = GTPR$ is set, a compare match occurs within the cycle only when $GTCCRA = 0$ or $GTCCRA = GTPR$ is satisfied. When $GTCCRA > GTPR$, no compare match occurs.

Similarly, $GTCCRB$ must be set within the range of $0 < GTCCRB < GTPR$. If $GTCCRB = 0$ or $GTCCRB = GTPR$ is set, a compare match occurs within the cycle only when $GTCCRB = 0$ or $GTCCRB = GTPR$ is satisfied. When $GTCCRB > GTPR$, no compare match occurs.

(3) When automatic dead time setting is made in saw-wave one-shot pulse mode

The $GTCCRC$ and $GTCCRD$ registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, the correct output waveforms with secured dead time may not be obtained.

- In up-counting: $GTCCRC < GTCCRD$, $GTCCRC > GTDVU$, $GTCCRD < GTPR - GTDVU$
- In down-counting: $GTCCRC > GTCCRD$, $GTCCRC < GTPR - GTDVU$, $GTCCRD > GTDVU$

Similarly, the $GTCCRE$ and $GTCCRF$ registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, correct output waveforms with secured dead time might not be obtained:

- In up-counting: $GTCCRE < GTCCRF$, $GTCCRE > GTDVU$, $GTCCRF < GTPR - GTDVU$
- In down-counting: $GTCCRE > GTCCRF$, $GTCCRE < GTPR - GTDVU$, $GTCCRF > GTDVU$

(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.

24.9.3 Setting Range for GTCNT Counter

The $GTCNT$ counter register must be set with the range of $0 \leq GTCNT \leq GTPR$.

24.9.4 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, resulting in situations in which an event is accepted or an interrupt occurs after GTCR.CST is set to 0.

24.9.5 Priority Order of Each Event

(1) GTCNT register

Table 24.47 shows a priority order of events updating the GTCNT register.

Table 24.47 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 GTCR	↑
Count up or down by hardware sources set in GTUPSR/GTDNSR	↑
Count operation	Low

If up-counting and down-counting by hardware sources occur at the same time, the GTCNT counter value does not change. When there is a conflict between updating the GTCNT register and reading by the CPU, pre-update data is read.

(2) GTCR.CST bit

When there is a conflict between starting/stopping by hardware sources set in the GTSSR/GTPSR registers and writing by the CPU (writing to GTCR/GTSTR/GTSTP registers), the writing by CPU has priority over the starting/stopping by hardware sources.

When there is a conflict between starting by hardware sources set in the GTSSR register and stopping by hardware sources set in GTPSR register, the GTCR.CST bit value does not change. When there is a conflict between updating the GTCR.CST bit and reading by the CPU, pre-update data is read.

(3) GTCCRm registers (m = A to F)

When there is a conflict between input capture/buffer transfer operation and writing to the GTCCRm registers, the writing to GTCCRm registers has priority over input capture/buffer transfer operation. When there is a conflict between input capture and writing to the counter register by the CPU or updating the counter register by hardware sources, the pre-update counter value is captured. When there is a conflict between updating the GTCCRm registers and reading by the CPU, pre-update data is read.

(4) GTPR 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.

25. Low Power Asynchronous General Purpose Timer (AGT, AGTW)

25.1 Overview

The low power Asynchronous General Purpose Timer (AGT, AGTW) is a 16-bit, 32-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events. This timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and can be accessed with the AGT register.

Table 25.1 lists the AGT, AGTW specifications, Figure 25.1 shows a block diagram, and Table 25.2 lists the I/O pins.

Table 25.1 AGT, AGTW specifications

Parameter		Description
Operating modes	Timer mode	The count source is counted
	Pulse output mode	The count source is counted and the output is inverted at each timer underflow
	Event counter mode	An external event is counted
	Pulse width measurement mode	An external pulse width is measured
	Pulse period measurement mode	An external pulse period is measured
Number of Channels		16 bits × 2 channels (AGTn (n = 0, 1)) 32 bits × 2 channels (AGTWn (n = 0, 1))
Count source (operating clock) ^{*2}	Timer mode	PCLKB, PCLKB/2, PCLKB/8, AGTLCLK/d, AGTSCLK/d (d = 1, 2, 4, 8, 16, 32, 64, or 128), or underflow signal of AGT0, AGTW0 selectable. ^{*1}
	Pulse output mode	
	Pulse width measurement mode	
	Pulse period measurement mode	
	Event counting mode	External event input
Interrupt and Event Link function ^{*3}		- Underflow event signal or measurement complete event signal - When the counter underflows - When the measurement of the active width of the external input pin (AGTIO_n, AGTWIO_n) complete in pulse width measurement mode - When the set edge of the external input pin (AGTIO_n, AGTWIO_n) are input in pulse period measurement mode. - Compare match A event signal - When the values of AGT register and AGTCMA register matched (compare match A function enabled). - Compare match B event signal - When the values of AGT register and AGTCMB register matched (compare match B function enabled). - Return from Snooze mode or Software Standby mode can be performed with AGT1_AGTI, AGT1_AGTCMAI, AGT0_AGTCMAI, AGTW1_AGTCMAI, AGTW0_AGTCMAI^{*4}
Selectable functions		Compare match function One or two of the AGT Compare Match A register and AGT Compare Match B register is selectable.

Note 1. AGT0, AGTW0 cannot use underflow signal. AGT1, AGTW1 connects directly with the underflow event signal from the AGT0, AGTW0 timer.

Note 2. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency × 4 of the count source clock.

Note 3. The AGT1 cannot use the Event Link function. AGT0 and AGTWn (n = 0, 1) event signals can be used.

Note 4. In details, see [section 13, Power-Saving Functions](#).

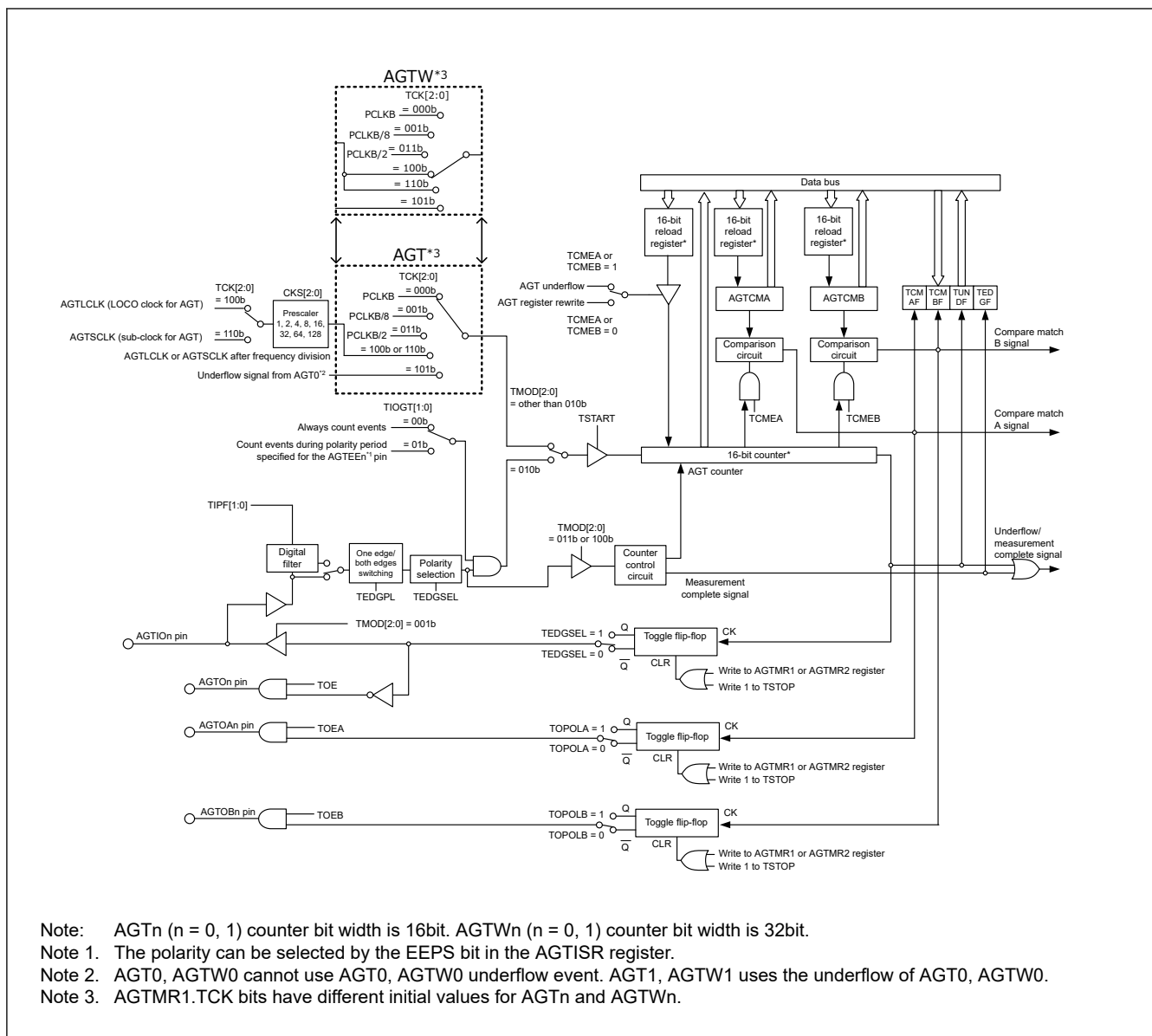


Figure 25.1 AGT, AGTW block diagram

Table 25.2 AGT I/O pins

Pin name	I/O	Function
AGTEEn	Input	External event input enable for AGT
AGTIO n	Input/output	External event input and pulse output for AGT
AGTO n	Output	Pulse output for AGT
AGT	Output	Compare match A output for AGT
AGTOB n	Output	Compare match B output for AGT

Note: Channel number: n = 0, 1

Table 25.3 AGTW I/O pins (1 of 2)

Pin name	I/O	Function
AGTWEE n	Input	External event input enable for AGTW
AGTWIO n	Input/output	External event input and pulse output for AGTW

Table 25.3 AGTW I/O pins (2 of 2)

Pin name	I/O	Function
AGTWOn	Output	Pulse output for AGTW
AGTWOAn	Output	Compare match A output for AGTW
AGTWOBn	Output	Compare match B output for AGTW

Note: Channel number: n = 0, 1

25.1.1 Difference between AGT and AGTW

AGT is a 16-bit timer. AGTW is a 32-bit timer.

AGT and AGTW have almost the same specifications. In the rest of this chapter, it is described in the following policy.

- The following matters will be described in each specification of AGT and AGTW.
 - base address
 - offset
- In the rest of chapter, described as AGT spec. About AGTW spec, replace based on following information.

Table 25.4 Difference between AGT and AGTW

item	AGT	AGTW
module symbol	AGT	AGTW
Recovery trigger from Snooze mode or Software Standby mode	AGT1_AGTI, AGT1_AGTCMAI, AGT0_AGTCMAI	AGTW1_AGTCMAI, AGTW0_AGTCMAI
AGT register	bit width is 16-bit	bit width is 32-bit
AGTCMA register	bit width is 16-bit	bit width is 32-bit
AGTCMB register	bit width is 16-bit	bit width is 32-bit
I/O pins	AGTEEn	AGTWEEEn
	AGTIOOn	AGTWIOOn
	AGTOOn	AGTWOn
	AGTOAn	AGTWOAn
	AGTOBn	AGTWOBn

25.2 Register Descriptions

25.2.1 AGT : AGT Counter Register

Base address: $AGTn = 0x4008\_4000 + 0x0100 \times n$ (n = 0, 1)
 $AGTWn = 0x4008\_4200 + 0x0100 \times n$ (n = 0, 1)

Offset address: 0x00 (AGT)
 0x00 (AGTW)

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:

Value after reset: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
15:0	n/a	16-bit counter and reload register Setting range : 0x0000 to 0xFFFF	R/W

AGTn.AGT is a 16-bit register. AGTWn.AGT is a 32-bit register. The value after reset of AGTWn.AGT is 0xFFFFFFFF. 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 25.3.1. Reload Register and Counter Rewrite Operation](#).

When 1 is written to the TSTOP bit in the AGTCR register, AGT counter is forcibly stopped and set to 0xFFFF(AGT), 0xFFFFFFFF(AGTW).

When the TCK[2:0] bits setting in the AGTMR1 register are a value other than 001b (PCLKB/8) or 011b (PCLKB/2), if the AGT register is set to 0x0000(AGT), 0x00000000(AGTW), a request signal to the ICU, the DTC, the DMAC, and the ELC is generated once immediately after the count starts. The AGTOn, AGTIOn and AGTWOn, AGTWIOn pin output are toggled.

When the AGT register is set to 0x0000(AGT), 0x00000000(AGTW) in event counter mode, regardless of the value of TCK[2:0] bits, a request signal to the ICU, the DTC, the DMAC, and the ELC is generated once immediately after the count starts.

In addition, the AGTOn pin output is toggled even during a period other than the specified count period. When the AGT register is set to 0x0001 or more, AGTW register is set to 0x00000001 or more, a request signal is generated each time AGT underflows.

25.2.2 AGTCMA : AGT Compare Match A Register

Base address: AGTn = 0x4008_4000 + 0x0100 × n (n = 0, 1)
AGTWn = 0x4008_4200 + 0x0100 × n (n = 0, 1)

Offset address: 0x02 (AGT)
0x04 (AGTW)

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

Bit	Symbol	Function	R/W
15:0	n/a	16-bit compare match A data is stored.* ¹ Setting range : 0x0000 to 0xFFFF	R/W

Note 1. Set the AGTCMA register to 0xFFFF(AGT), 0xFFFF FFFF(AGTW) 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. AGTn.AGTCMA is a 16-bit register. AGTWn.AGTCMA is a 32-bit register. The value after reset of AGTWn.AGTCMA is 0xFFFFFFFF. The states of the reload register and compare register A change according to the TSTART bit in the AGTCR register. For details, see [section 25.3.2. Reload Register and AGT Compare Match A/B Register Rewrite Operation](#).

25.2.3 AGTCMB : AGT Compare Match B Register

Base address: AGTn = 0x4008_4000 + 0x0100 × n (n = 0, 1)
AGTWn = 0x4008_4200 + 0x0100 × n (n = 0, 1)

Offset address: 0x04 (AGT)
0x08 (AGTW)

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

Bit	Symbol	Function	R/W
15:0	n/a	16-bit compare match B data is stored.* ¹ Setting range : 0x0000 to 0xFFFF	R/W

Note 1. Set the AGTCMB register to 0xFFFF(AGT), 0xFFFF FFFF(AGTW) 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. AGTn.AGTCMB is a 16-bit register. AGTWn.AGTCMB is a 32-bit register. The value after reset of AGTWn.AGTCMB is 0xFFFFFFFF. The

states of the reload register and compare register B change according to the TSTART bit in the AGTCR register. For details, see [section 25.3.2. Reload Register and AGT Compare Match A/B Register Rewrite Operation](#).

25.2.4 AGTCR : AGT Control Register

Base address: $AGTn = 0x4008\_4000 + 0x0100 \times n$ ($n = 0, 1$)
 $AGTWn = 0x4008\_4200 + 0x0100 \times n$ ($n = 0, 1$)

Offset address: 0x08 (AGT)
 0x0C (AGTW)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TCMB F	TCMA F	TUND F	TEDG F	—	TSTO P	TCST F	TSTA RT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSTART	AGT Count Start*2 0: Count stops 1: Count starts	R/W
1	TCSTF	AGT Count Status Flag*2 0: Count stopped 1: Count in progress	R
2	TSTOP	AGT Count Forced Stop*1 0: Writing is invalid 1: The count is forcibly stopped	W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TEDGF	Active Edge Judgment Flag 0: No active edge received 1: Active edge received	R/(W)*3
5	TUNDF	Underflow Flag 0: No underflow 1: Underflow	R/(W)*3
6	TCMAF	Compare Match A Flag 0: No match 1: Match	R/(W)*3
7	TCMBF	Compare Match B Flag 0: No match 1: Match	R/(W)*3

Note 1. When 1 (count is forcibly stopped) is written to the TSTOP bit, the TSTART bit and TCSTF flag are initialized at the same time. The pulse output level is also initialized. The read value is 0.

Note 2. For information on using the TSTART bit and TCSTF flag, see [section 25.4.1. Count Operation Start and Stop Control](#).

Note 3. Only 0 can be written to clear the flag.

TSTART bit (AGT Count Start)

The count operation is started by writing 1 to the TSTART bit and stopped by writing 0. When the TSTART bit is set to 1 (count starts), the TCSTF flag is set to 1 (count in progress) in synchronization with the count source. Also, after 0 is written to the TSTART bit, the TCSTF flag is set to 0 (count stops) in synchronization with the count source. For details, see [section 25.4.1. Count Operation Start and Stop Control](#).

TCSTF flag (AGT Count Status Flag)

The TCSTF flag indicates the AGT count status.

[Setting condition]

- When 1 is written to the TSTART bit (the TCSTF flag is set to 1 in synchronization with the count source).

[Clearing conditions]

- When 0 is written to the TSTART bit (the TCSTF flag is set to 0 in synchronization with the count source)

- When 1 is written to the TSTOP bit.

TSTOP bit (AGT Count Forced Stop)

When 1 is written to the TSTOP bit, the count is forcibly stopped. The read value is 0.

TEDGF flag (Active Edge Judgment Flag)

The TEDGF flag indicates that an active edge was detected.

[Setting condition]

- When the measurement of the active width of the external input pin (AGTIO_n, AGTWIO_n) is complete in pulse width measurement mode
- When the set edge of the external input pin (AGTIO_n, AGTWIO_n) 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.

25.2.5 AGTMR1 : AGT Mode Register 1

Base address: AGT_n = 0x4008_4000 + 0x0100 × n (n = 0, 1)
AGTW_n = 0x4008_4200 + 0x0100 × n (n = 0, 1)

Offset address: 0x09 (AGT)
0x0D (AGTW)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—		TCK[2:0]		TEDG PL	TMOD[2:0]		

Value after reset: 0 1 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	TMOD[2:0]	Operating Mode* ³ 0 0 0: Timer mode 0 0 1: Pulse output mode 0 1 0: Event counter mode 0 1 1: Pulse width measurement mode 1 0 0: Pulse period measurement mode Others: Setting prohibited	R/W
3	TEDGPL	Edge Polarity* ⁴ 0: Single-edge 1: Both-edge	R/W
6:4	TCK[2:0]	Count Source* ¹ * ² * ⁵ * ⁷ 0 0 0: PCLKB 0 0 1: PCLKB/8 0 1 1: PCLKB/2 1 0 0: Divided clock AGTLCLK specified by CKS[2:0] bits in the AGTMR2 register 1 0 1: Underflow event signal from AGT0, AGTW0* ⁶ 1 1 0: Divided clock AGTSCLK specified by CKS[2:0] bits in the AGTMR2 register Others: Setting prohibited	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: Write access to the AGTMR1 register initializes the output from the AGTOn, AGTIOOn, AGTOAn, AGTOBn, AGTWOn, AGTWIOOn, AGTWOAn, and AGTWOBn pins of the AGT and AGTW (n = 0, 1). For details on the output level at initialization, see [section 25.2.7. AGTIOC : AGT I/O Control Register](#).

Note 1. When event counter mode is selected, the external input pin (AGTIOOn, AGTWIOOn) is selected as the count source regardless of the setting of TCK[2:0] bits.

Note 2. Do not switch count sources during count operation. Only switch count sources when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).

Note 3. The operating mode can only be changed when the count is stopped while both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count is stopped). Do not change the operating mode during count operation.

Note 4. The TEDGPL bit is enabled only in event counter mode.

Note 5. To run AGT in Software Standby and Deep Software Standby mode, select AGTLCLK or AGTSCLK (TCK[2:0] = 100b, 110b).

Note 6. AGT0, AGTW0 cannot use AGT0 underflow (setting prohibited). AGT1, AGTW1 uses the AGT0, AGTW0 underflow.

Note 7. Do not change the TCK[2:0] bits when the CKS[2:0] bits in the AGTMR2 register is not 000b. First, change the CKS[2:0] bits in the AGTMR2 register to 000b. Then change the TCK[2:0] bits and wait for one cycle of the count source.

25.2.6 AGTMR2 : AGT Mode Register 2

Base address: AGTn = 0x4008_4000 + 0x0100 × n (n = 0, 1)
AGTWn = 0x4008_4200 + 0x0100 × n (n = 0, 1)

Offset address: 0x0A (AGT)
0x0E (AGTW)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LPM	—	—	—	—	CKS[2:0]		

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	CKS[2:0]	AGTLCLK or AGTSCLK Count Source Clock Frequency Division Ratio* ¹ * ² * ³ 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	LPM	Low Power Mode 0: Normal mode 1: Low power mode	R/W

Note 1. Do not rewrite the CKS[2:0] bits during count operation. Only rewrite the CKS[2:0] bits when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).

Note 2. When count source is AGTLCLK or AGTSCLK, the switch of CKS[2:0] bits is valid.

Note 3. Do not switch the TCK[2:0] bits in the AGTMR1 register when CKS[2:0] bits are not 000b. Switch the TCK[2:0] bits in the AGTMR1 register after CKS[2:0] bits are set to 000b, and wait for 1 cycle of the count source.

CKS[2:0] bit (AGTLCLK or AGTSCLK Count Source Clock Frequency Division Ratio)

CKS[2:0] bits select the Count Source Clock Frequency Division Ratio for AGTLCLK or AGTSCLK.

LPM bit (Low Power Mode)

The LPM bit sets the low power operation, which impacts access to certain AGT registers. Set this bit to 1 to operate in low power.

When this bit is 1, access to the following registers is prohibited:

- AGT/AGTCMA/AGTCMB/AGTCR.

After this bit is switched from 1 to 0, the first access to the register is constrained as follows:

- When reading from the AGT register, read AGT register twice. Only the second reading of data is valid.
- When writing to the AGT, AGTCMA, AGTCMB, and AGTCR register, allow at least 2 cycles of the count source clock when writing to the register.
- When confirm the value written to the AGT, AGTCMA, AGTCMB, and AGTCR registers.
 - When the count operation is stopped; after writing data, it can be read in the next cycle.
 - When the count operation is operating; after writing data, it can be written 4 cycles after the count source clock.

Figure 25.2 shows the flow of how to write LPM bit

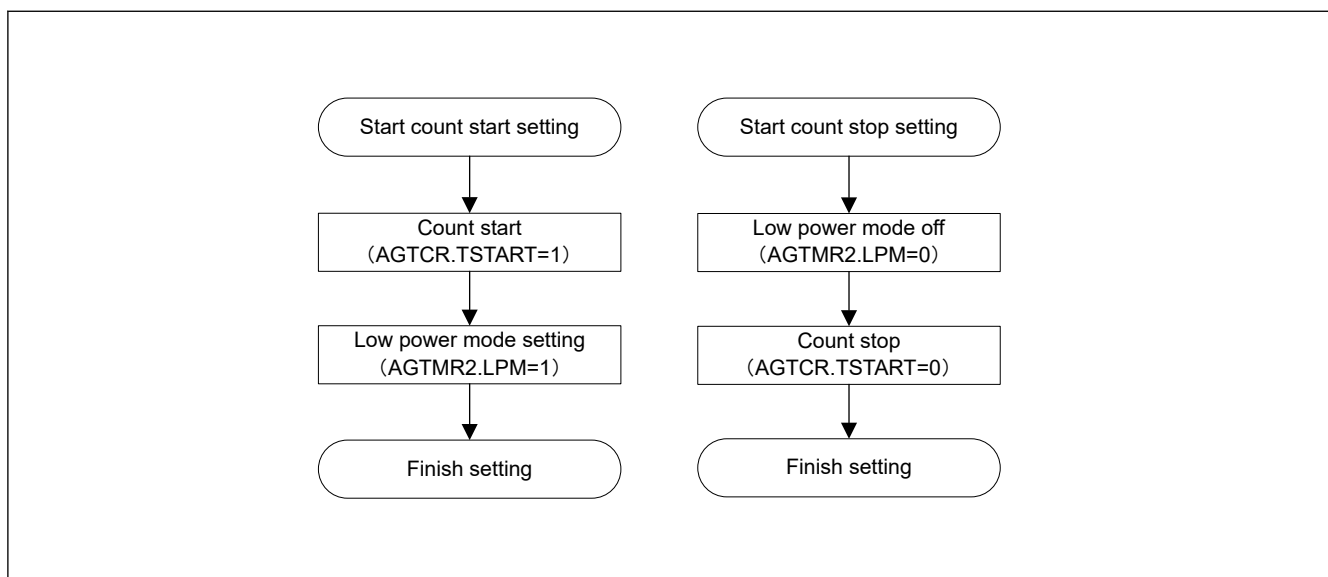


Figure 25.2 LPM how to write flow chart

25.2.7 AGTIOC : AGT I/O Control Register

Base address: AGTn = 0x4008_4000 + 0x0100 × n (n = 0, 1)
AGTWn = 0x4008_4200 + 0x0100 × n (n = 0, 1)

Offset address: 0x0C (AGT)
0x10 (AGTW)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIOGT[1:0]		TIPF[1:0]		—	TOE	—	TEDGSEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TEDGSEL	I/O Polarity Switch Function varies depending on the operating mode (see Table 25.5 and Table 25.6).	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	TOE	AGTOn pin Output Enable 0: AGTOn pin output disabled 1: AGTOn pin output enabled	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TIPF[1:0]	Input Filter*3 These bits specifies the sampling frequency of the filter for the AGTIO pin. If the input to the AGTIO pin is sampled and the value matches three successive times, that value is taken as the input value. 0 0: No filter 0 1: Filter sampled at PCLKB 1 0: Filter sampled at PCLKB/8 1 1: Filter sampled at PCLKB/32	R/W
7:6	TIOGT[1:0]	Count Control*1 *2 0 0: Event is always counted 0 1: Event is counted during polarity period specified for AGTEEn pin Others: Setting prohibited	R/W

Note 1. When AGTEEn pin is used, the polarity to count an event can be selected with the EEPS bit in the AGTISR register.

Note 2. TIOGT[1:0] bits are enabled only in event counter mode.

Note 3. When event counter mode operation is performed during Software Standby and Deep Software Standby mode, the digital filter function cannot be used.

TEDGSEL bit (I/O Polarity Switch)

The TEDGSEL bit switches the AGTOn pin output polarity and the AGTIO pin input/output edge and polarity.

In pulse output mode, it only controls polarity of the AGTOn pin output and AGTIO pin output. AGTOn pin output and AGTIO pin output are initialized when the AGTMR1 register is written or the TSTOP bit in the AGTCR register is written with 1.

TOE bit (AGTOn pin Output Enable)

The TOE bit selects whether the AGTOn pin output is disabled or enabled.

TIPF[1:0] bit (Input Filter)

The TIPF[1:0] bits specify the sampling frequency of the AGTIO pin input filter. When the input to the AGTIO pin is sampled and the values match three times in succession, the value is regarded as the input value.

TIOGT[1:0] bit (Count Control)

The TIOGT[1:0] bits control the event count.

Table 25.5 AGTIO pin I/O edge and polarity switching (1 of 2)

Operating mode	Function
Timer mode	Not used

Table 25.5 AGTIO pin I/O edge and polarity switching (2 of 2)

Operating mode	Function
Pulse output mode	0: Output is started at high (initialization level: high) i.e. inverted output 1: Output is started at low (initialization level: low). i.e. normal output
Event counter mode	0: Count on rising edge 1: Count on falling edge.
Pulse width measurement mode	0: Low-level width is measured 1: High-level width is measured.
Pulse period measurement mode	0: Measure from one rising edge to the next rising edge 1: Measure from one falling edge to the next falling edge.

Note: When the TOE bit is 0, the pin state is Hi-Z.

Table 25.6 AGTOn pin output polarity switching

Operating mode	Function
All modes	0: Output is started at low (initial level: low): Normal output 1: Output is started at high (initial level: high): Inverted output

Note: When the TOE bit is 0, a value according to the set value of the TEDGSEL bit in the pulse output mode is output.

25.2.8 AGTISR : AGT Event Pin Select Register

Base address: $AGTn = 0x4008\_4000 + 0x0100 \times n$ ($n = 0, 1$)
 $AGTWn = 0x4008\_4200 + 0x0100 \times n$ ($n = 0, 1$)

Offset address: 0x0D (AGT)
0x11 (AGTW)

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

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	EEPS	AGTEEn Polarity Selection 0: An event is counted during the low-level period 1: An event is counted during the high-level period	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

EEPS bit (AGTEEn Polarity Selection)

The EEPS bit selects the polarity of events to be counted.

25.2.9 AGTCMSR : AGT Compare Match Function Select Register

Base address: $AGTn = 0x4008\_4000 + 0x0100 \times n$ ($n = 0, 1$)
 $AGTWn = 0x4008\_4200 + 0x0100 \times n$ ($n = 0, 1$)

Offset address: 0x0E (AGT)
0x12 (AGTW)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TOPO LB	TOEB	TCME B	—	TOPO LA	TOEA	TCME A
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCMEA	AGT Compare Match A Register Enable ^{*1 *2} 0: AGT Compare match A register disabled 1: AGT Compare match A register enabled	R/W
1	TOEA	AGTOAn Pin Output Enable ^{*1 *2} 0: AGTOAn pin output disabled 1: AGTOAn pin output enabled	R/W
2	TOPOLA	AGTOAn Pin Polarity Select ^{*1 *2} 0: AGTOAn pin output is started on low. i.e. normal output 1: AGTOAn pin output is started on high. i.e. inverted output	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TCMEB	AGT Compare Match B Register Enable ^{*1 *2} 0: Compare match B register disabled 1: Compare match B register enabled	R/W
5	TOEB	AGTOBn Pin Output Enable ^{*1 *2} 0: AGTOBn pin output disabled 1: AGTOBn pin output enabled	R/W
6	TOPOLB	AGTOBn Pin Polarity Select ^{*1 *2} 0: AGTOBn pin output is started on low. i.e. normal output 1: AGTOBn pin output is started on high. i.e. inverted output	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. Do not rewrite the AGTCMSR register during a count operation. Only rewrite the AGTCMSR register when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).

Note 2. Do not set 1 when in pulse width measurement mode or pulse period measurement mode.

25.2.10 AGTIOSEL : AGT Pin Select Register

Base address: AGTn = 0x4008_4000 + 0x0100 × n (n = 0, 1)
AGTWn = 0x4008_4200 + 0x0100 × n (n = 0, 1)

Offset address: 0x0F (AGT)
0x13 (AGTW)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TIES	—	—	—	—

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	TIES	AGTIOn Pin Input Enable 0: External event input is disabled during Software Standby mode 1: External event input is enabled during Software Standby mode	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The AGTIOSEL register sets the AGTIO_n, AGTWIO_n pin when using the AGTIO_n, AGTWIO_n pin in Deep Software Standby mode and Software Standby mode.

TIES bit (AGTIO_n Pin Input Enable)

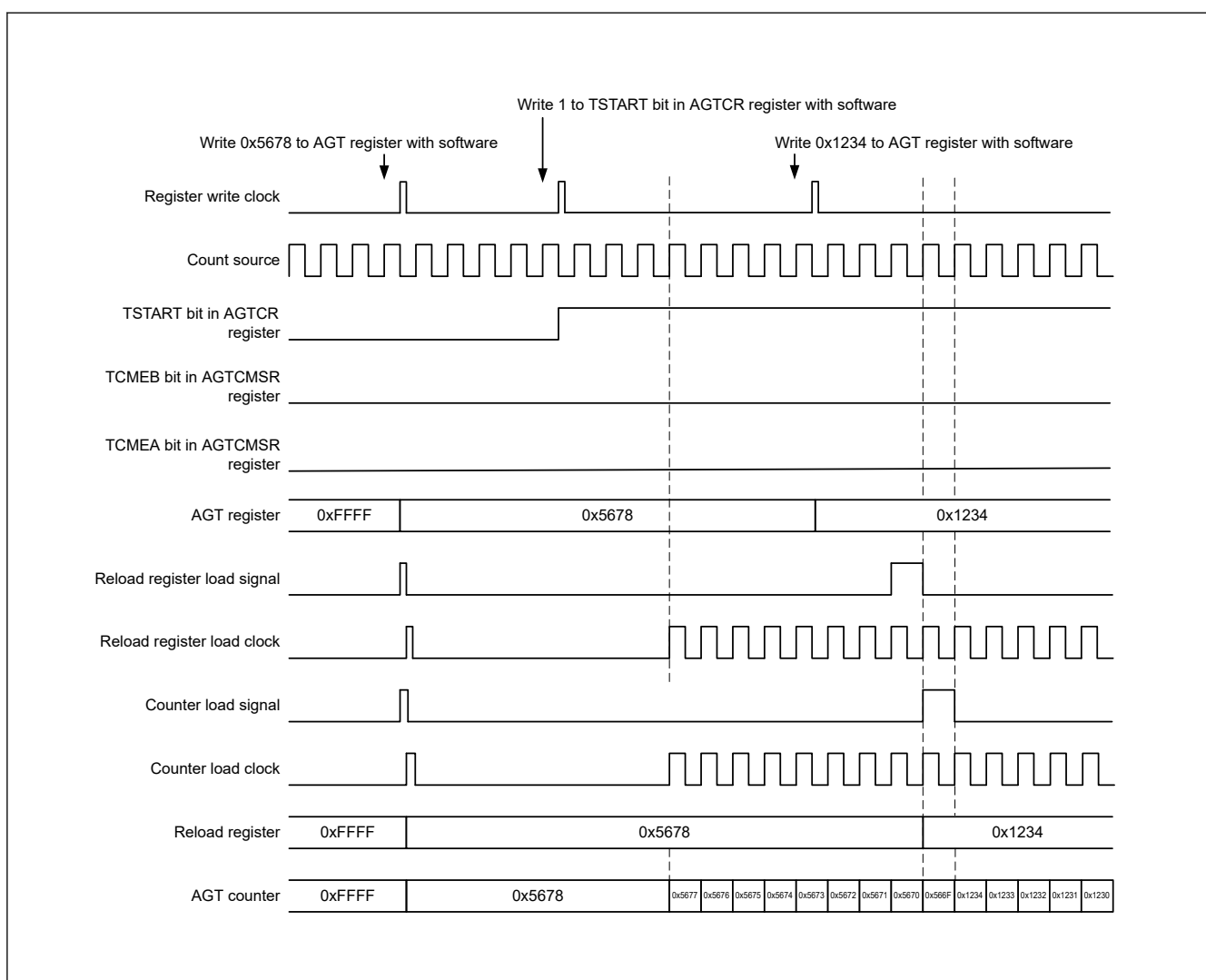
The TIES bit enables or disables an external event input.

25.3 Operation

25.3.1 Reload Register and Counter Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and the counter differs depending on the value of the TSTART bit in the AGTCR register and of the TCMEA or TCMEB bit in the AGTCMSR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and the counter. When the TSTART bit is 1 (count starts) and the TCMEA bit and TCMEB bit are 0 (AGT compare match A/B register are invalid), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the next count source. When the TSTART bit is 1 (count starts) and the TCMEA bit or the TCMEB bit is 1 (AGT compare match A register or compare match B register is valid), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the underflow of the counter.

Figure 25.3 and Figure 25.4 show the timing of rewrite operation with TSTART bit value and TCMEA/TCMEB bit value.



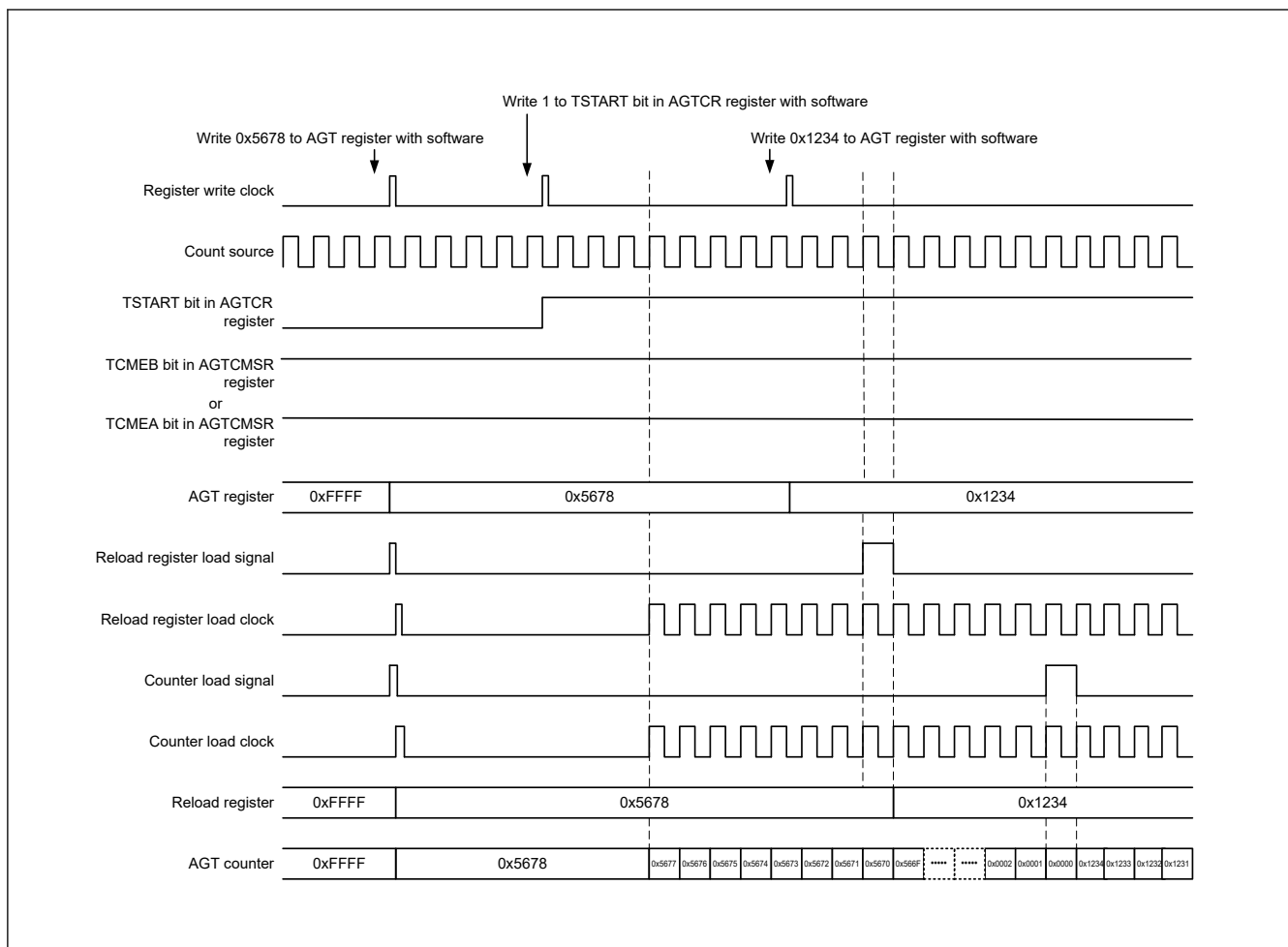


Figure 25.4 Timing of rewrite operation with TSTART bit value and TCMEA or TCMEB bit value when AGT compare match A register or AGT compare match B register is valid

25.3.2 Reload Register and AGT Compare Match A/B Register Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and AGT compare register A/B depends on the value of the TSTART bit in the AGTCR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and AGT compare register A/B. When the TSTART bit is 1 (count starts), the value is written to the reload register in synchronization with the count source, and then to the compare register in synchronization with the underflow of the counter.

Figure 25.5 shows the timing of rewrite operation with TSTART bit value for compare register A. AGT Compare register B is of the same timing as AGT compare register A.

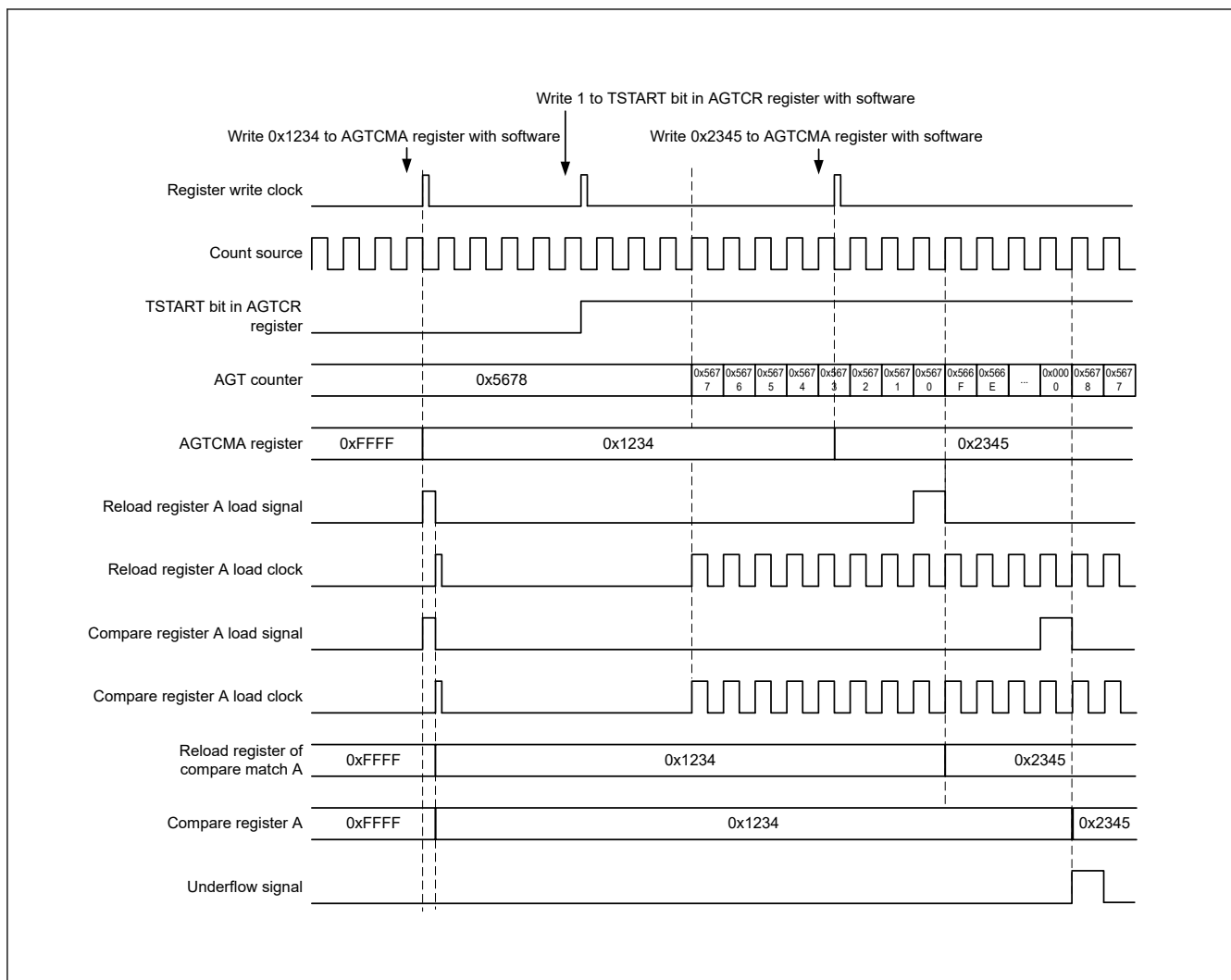


Figure 25.5 Timing of rewrite operation with the TSTART bit value for AGT compare register A

25.3.3 Timer Mode

In this mode, the AGT counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register. In timer mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 0x0000 and the next count source is input, an underflow occurs and an interrupt request is generated.

Figure 25.6 shows the operation example in timer mode.

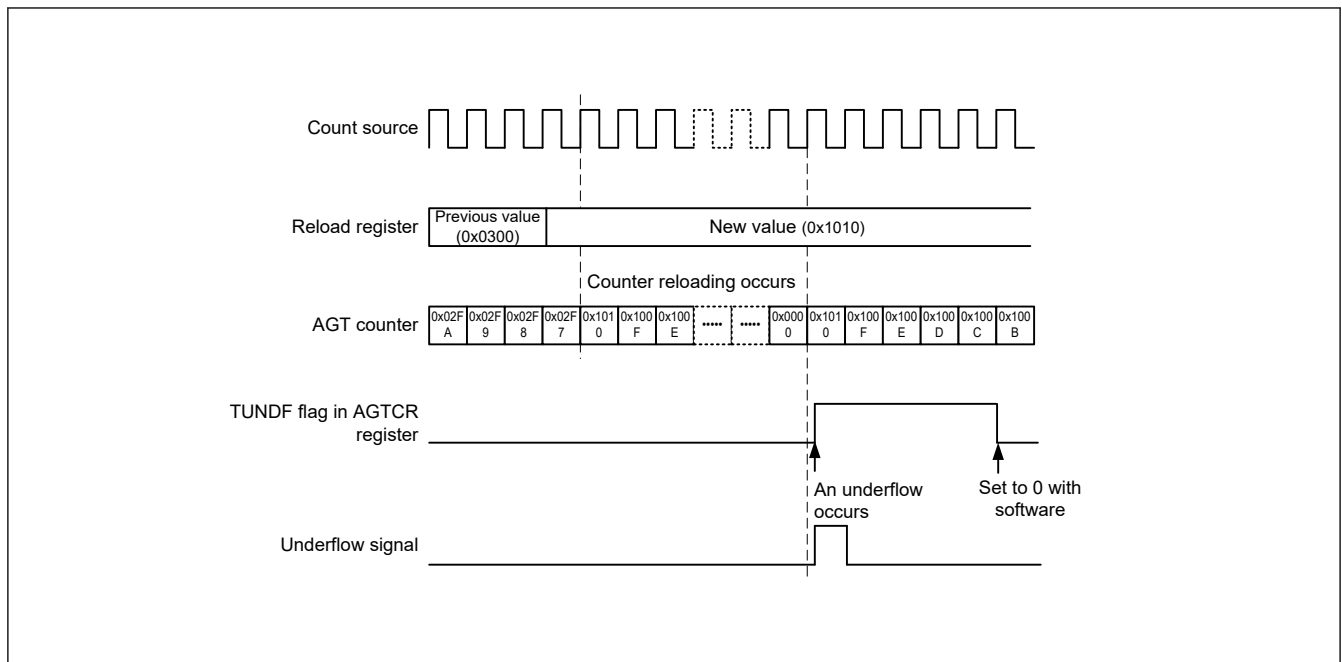


Figure 25.6 Operation example in timer mode

25.3.4 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 AGTIO_n and AGTON pins inverted each time an underflow occurs.

In pulse output mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 0x0000 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 AGTIO_n 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 25.7 shows the operation example in pulse output mode.

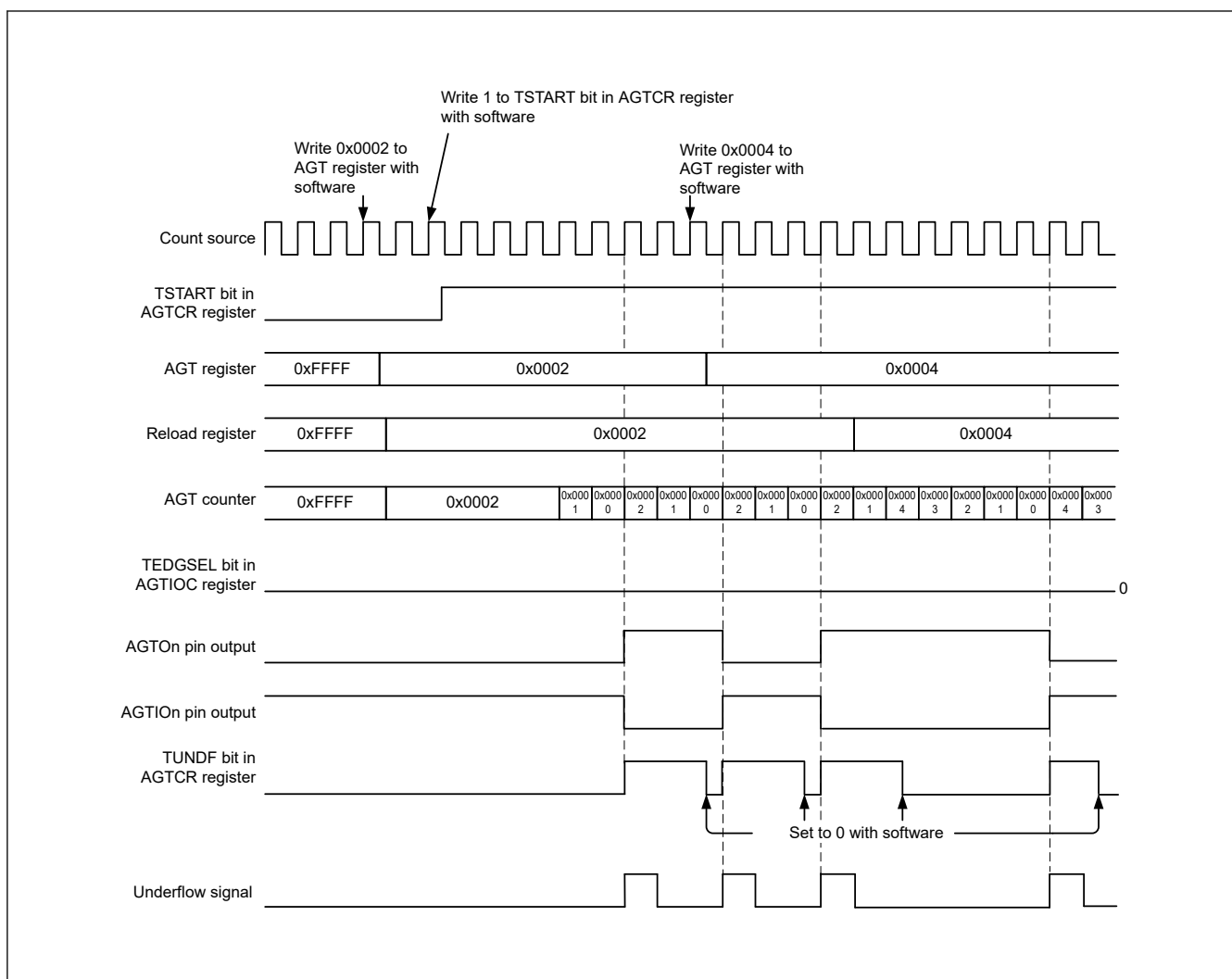


Figure 25.7 Operation example in pulse output mode

25.3.5 Event Counter Mode

In event counter mode, the counter is decremented by an external event signal (count source) input to the AGTOn pin. Various periods for counting events can be set with the TIOGT[1:0] bits in the AGTIOC register and AGTISR registers. In addition, the filter function for the AGTOn pin input can be specified with bits TIPF[1:0] in the AGTIOC register. The output from the AGTOn pin can be toggled even in event counter mode.

Figure 25.8 shows the operation example in event counter mode.

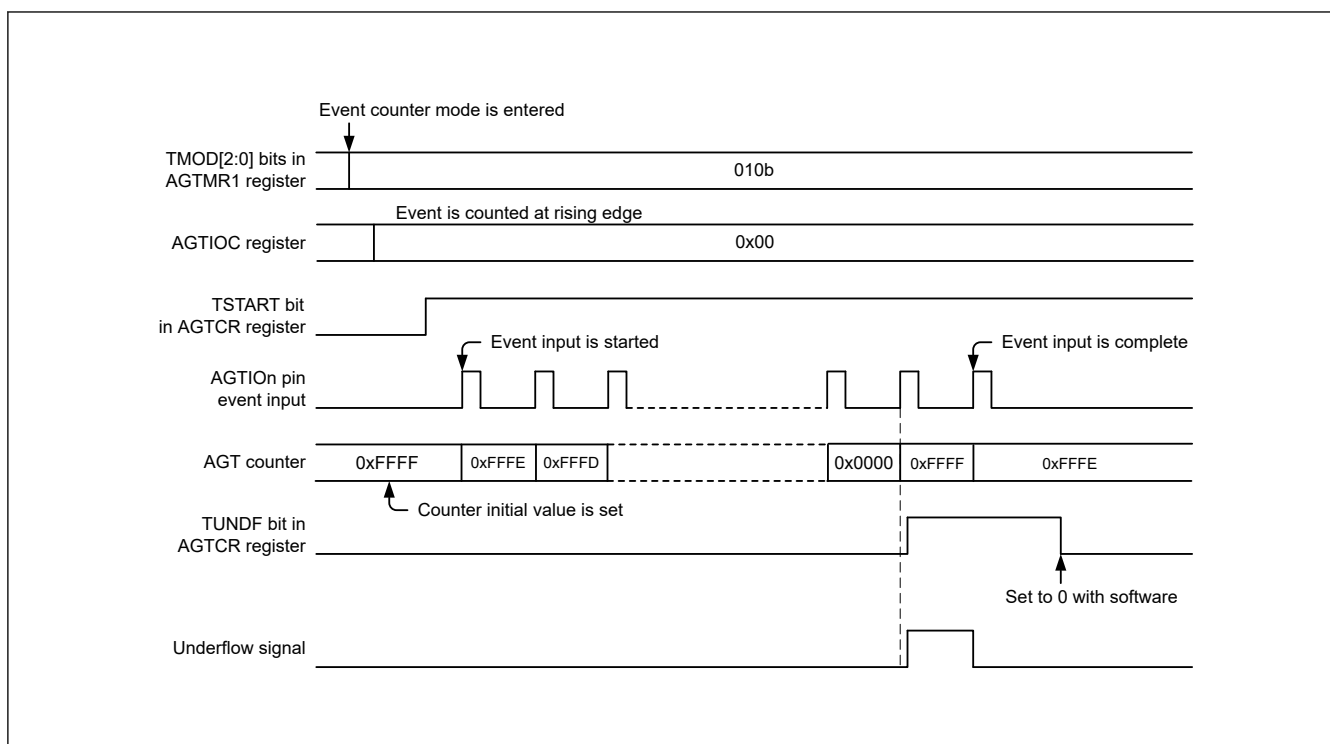


Figure 25.8 Operation example 1 in event counter mode

Figure 25.9 shows an operation example for counting during the specified period in event counter mode (TIOGT[1:0] bits in the AGTIOC register are set to 01b).

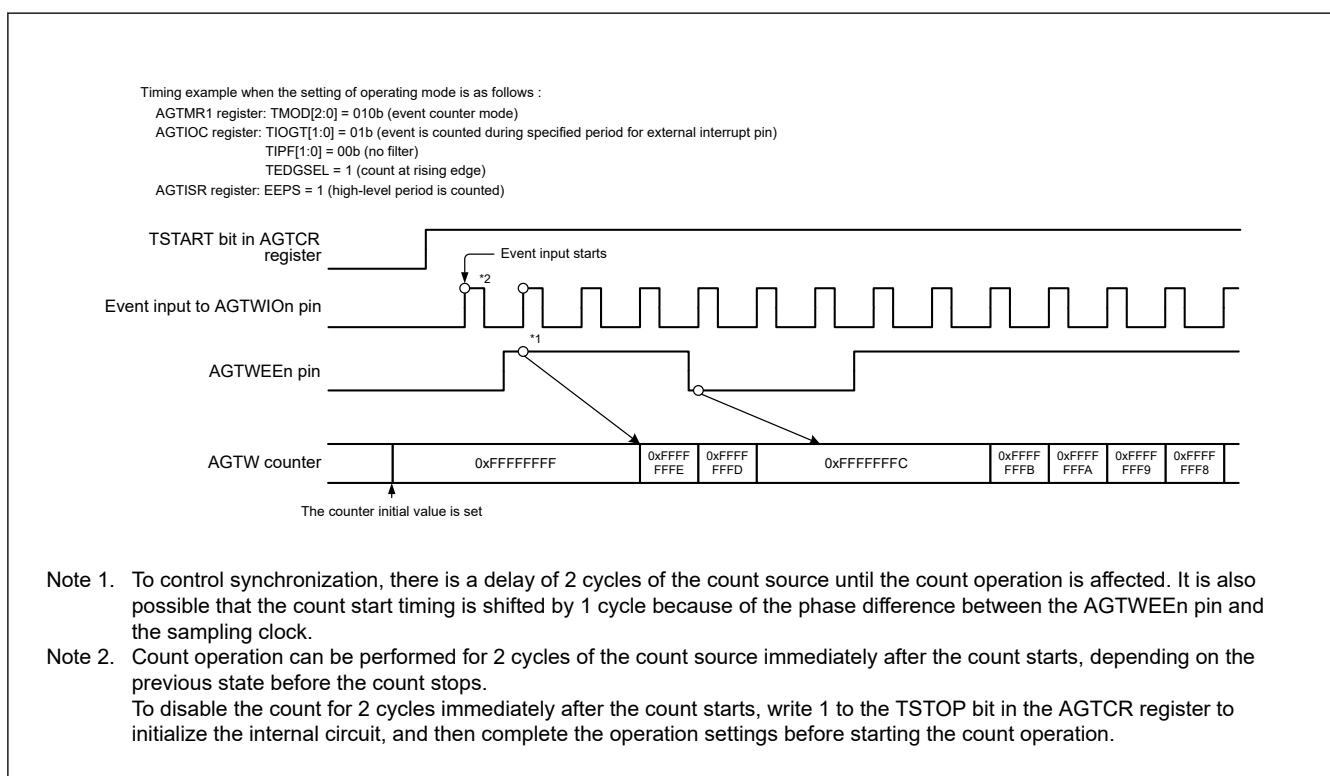


Figure 25.9 Operation example 2 in event counter mode

25.3.6 Pulse Width Measurement Mode

In pulse width measurement mode, the pulse width of an external signal input to the AGTIO pin is measured. When the level specified by the TEDGSEL bit in the AGTIOC register is input to the AGTIO 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 AGTIO 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 25.10 shows the operation example in pulse width measurement mode.

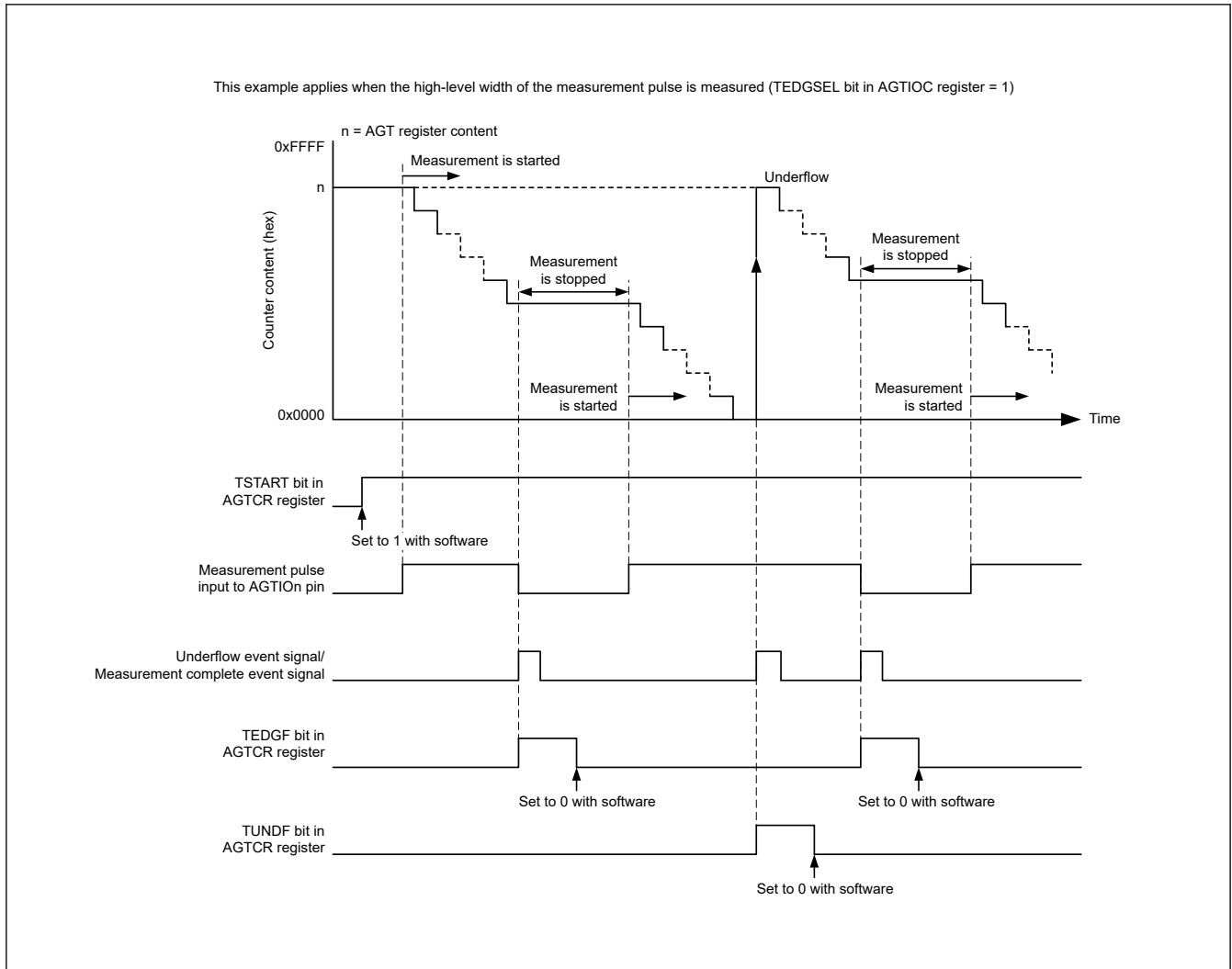


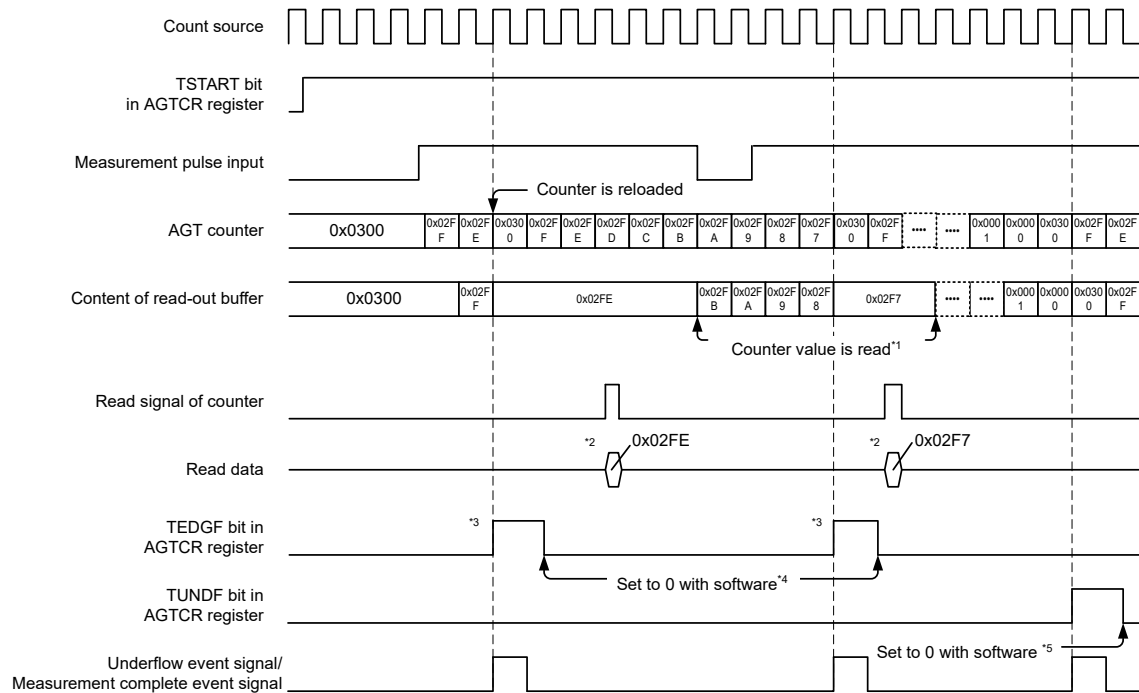
Figure 25.10 Operation example in pulse width measurement mode

25.3.7 Pulse Period Measurement Mode

In pulse period measurement mode, the pulse period of an external signal input to the AGTIO pin is measured. The counter is decremented by the count source selected with TCK[2:0] bits in the AGTMR1 register. When a pulse with the period specified by the TEDGSEL bit in the AGTIOC register is input to the AGTIO pin, the count value is transferred to the read-out buffer on the rising edge of the count source. The value in the reload register is loaded to the counter at the next rising edge. Simultaneously, the TEDGF flag in the AGTCR register is set to 1 (active edge received) and an interrupt request is generated. The read-out buffer (AGT register) is read at this time and the difference from the reload value (see [section 25.4.6. How to Calculate Event Number, Pulse Width, and Pulse Period](#)) is the period data of the input pulse. The period data is retained until the read-out buffer is read. When the counter underflows, the TUNDF flag in the AGTCR register is set to 1 (underflow) and an interrupt request is generated.

Figure 25.11 shows the operation example in pulse period measurement mode.

Only input pulses with a period longer than twice the period of the count source are measured. Also, the low-level and high-level widths must both be longer than the period of the count source. If a pulse period shorter than these conditions is input, the input might be ignored.



This example applies when the initial value of the AGT register is set to 0x0300, the TEDGSEL bit in the AGTIOC register is set to 0, and the period from one rising edge to the next edge of the measurement pulse is measured.

- Note 1. Reading from the AGT register must be performed during the period from when the TEDGF flag is set to 1 (active edge received) until the next active edge is input. The content of the read-out buffer is retained until the AGT register is read. If it is not read before the active edge is input, the measurement result of the previous period is retained.
- Note 2. When the AGT register is read in pulse period measurement mode, the content of the read-out buffer is read.
- Note 3. When the active edge of the measurement pulse is input and then the set edge of an external pulse is input, the TEDGF flag in the AGTCR register is set to 1 (active edge received).
- Note 4. To set to 0 with software, write 0 to the TEDGF flag in the AGTCR register with an 8-bit memory manipulation instruction.
- Note 5. To set to 0 with software, write 0 to the TUNDF flag in the AGTCR register with an 8-bit memory manipulation instruction.

Figure 25.11 Operation example in pulse period measurement mode

25.3.8 Compare Match function

The compare match function detects matches (compare match) between the content of the AGTCMA or AGTCMB register and the content of the AGT register. This function is enabled when the TCMEA or TCMEB bit in the AGTCMSR register is 1 (compare match A register or compare match B register is valid). The counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register, and when the values of AGT and AGTCMA or AGTCMB match, the TCMAF/TCMBF flag in the AGTCR register is set to 1 (match), and an interrupt request is generated.

When the compare match function is enabled, the timing of the rewrite operation to the reload register and the counter differs. See [section 25.3.1. Reload Register and Counter Rewrite Operation](#) for details. In addition, the output level of the AGTOAn, 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 25.12 shows the operation example in compare match mode.

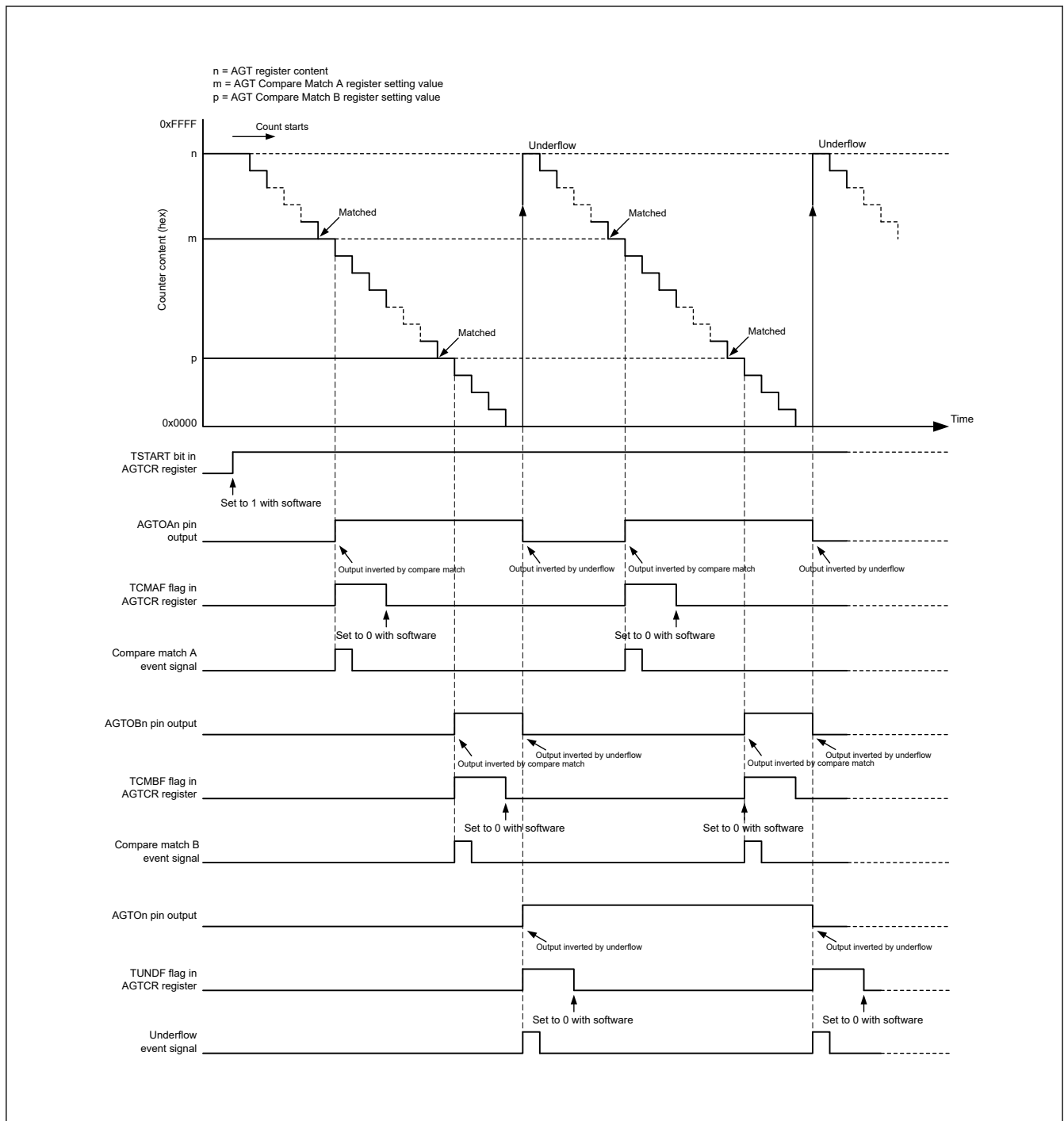


Figure 25.12 Operation example in compare match mode (TOPOLA = 0, TOPOLB = 0)

25.3.9 Output Settings for Each Mode

Table 25.7 to Table 25.10 list the states of pins AGTON, AGTIO, AGTOA, and AGTOB pins in each mode.

Table 25.7 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 25.8 AGTIO pin setting

Operating mode	AGTIOC register	
	TEDGSEL bit	AGTIO pin I/O
Timer mode	0 or 1	Input (not used)
Pulse output mode	1	Normal output
	0	Inverted output
Event counter mode	0 or 1	Input
Pulse width measurement mode		
Pulse period measurement mode		

Table 25.9 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 25.10 AGTOBn pin setting

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)
Pulse width measurement mode	0	0	Prohibited
Pulse period measurement mode			

25.3.10 Standby Mode

The AGT can operate in Software Standby and Deep Software Standby mode. Set it to Software Standby or Deep Software Standby mode with count operation start (TSTART = 1, TCSTF = 1).

Table 25.11 and Table 25.12 show the setting that can be used in Software Standby and Deep Software Standby mode.

Table 25.11 Usable settings in Software Standby and Deep 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	—	AGTIO ⁿ *1	—
Pulse width measurement mode	100b or 110b	AGTLCLK or AGTSCLK	—
Pulse period measurement mode	100b or 110b	AGTLCLK or AGTSCLK	—

Note: —: invalid

Note 1. When using the AGTIOⁿ pin for external event input in Software Standby mode, set AGTIOSEL.TIES = 1.

Table 25.12 Usable settings in Software Standby and Deep 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	—	AGTIO ⁿ *2	- 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: —: invalid

Note: Release of Software Standby or Deep Software Standby mode is only AGT1.

Note: Compare match A/B is resurgence factor of CPU from Software Standby mode.

Note 1. Only when AGT0 operates in [Table 25.11](#)

Note 2. When using the AGTIOⁿ pin for external event input in Software Standby mode, set AGTIOSEL.TIES = 1.

25.3.11 Interrupt Sources

The AGT_n and AGTW_n has three interrupt sources for channels n (n = 0, 1) as listed in [Table 25.13](#).

AGT0 and AGTW0 interrupt factors and AGT1 and AGTW1 interrupt factors are used exclusively. AGT0 and AGTW1 interrupt factors, and AGT1 and AGTW0 interrupt factors can be used simultaneously.

Table 25.13 AGT, AGTW interrupt sources (1 of 2)

Name	Interrupt source	Interrupt to CPU	Start DTC/DMAC	Cancel standby
AGT0_AGTI	- When the counter underflows - When measurement of the active width of the external input pin (AGTIO0) is complete in pulse width measurement mode - When the set edge of the external input pin (AGTIO0) is input in pulse period measurement mode.	Possible	Possible	Impossible
AGT0_AGTCMAI	When the values of AGT0.AGT register and AGT0.AGTCMA register match	Possible	Possible	Possible
AGT0_AGTCMBI	When the values of AGT0.AGT register and AGT0.AGTCMB register match	Possible	Possible	Impossible
AGT1_AGTI	- When the counter underflows - When measurement of the active width of the external input pin (AGTIO1) is complete in pulse width measurement mode - When the set edge of the external input pin (AGTIO1) is input in pulse period measurement mode.	Possible	Possible	Possible
AGT1_AGTCMAI	When the values of AGT1.AGT register and AGT1.AGTCMA register match	Possible	Possible	Possible

Table 25.13 AGT, AGTW interrupt sources (2 of 2)

Name	Interrupt source	Interrupt to CPU	Start DTC/DMAC	Cancel standby
AGT1_AGTCMBI	When the values of AGT1.AGT register and AGT1.AGTCMB register match	Impossible	Impossible	Impossible
AGTW0_AGTI	- When the counter underflows - When measurement of the active width of the external input pin (AGTWIO0) is complete in pulse width measurement mode - When the set edge of the external input pin (AGTWIO0) is input in pulse period measurement mode.	Possible	Possible	Impossible
AGTW0_AGTCMAI	When the values of AGTW0.AGT register and AGTW0.AGTCMA register match	Possible	Possible	Possible
AGTW0_AGTCMBI	When the values of AGTW0.AGT register and AGTW0.AGTCMB register match	Possible	Possible	Impossible
AGTW1_AGTI	- When the counter underflows - When measurement of the active width of the external input pin (AGTWIO1) is complete in pulse width measurement mode - When the set edge of the external input pin (AGTWIO1) is input in pulse period measurement mode.	Possible	Possible	Impossible
AGTW1_AGTCMAI	When the values of AGTW1.AGT register and AGTW1.AGTCMA register match	Possible	Possible	Possible
AGTW1_AGTCMBI	When the values of AGTW1.AGT register and AGTW1.AGTCMB register match	Impossible	Impossible	Impossible

Note: Channel number (n = 0, 1)

25.3.12 Event Signal Output to ELC

The AGT0 and AGTWn (n = 0, 1) uses the Event Link Controller (ELC) to perform a link operation to a specified module using the interrupt request signal as the event signal. The AGT0 and AGTWn (n = 0, 1) outputs compare match A, compare match B, and underflow/measurement complete signals as event signals. For details, see [section 21, Event Link Controller \(ELC\)](#).

25.4 Usage Notes

25.4.1 Count Operation Start and Stop Control

- When the operating mode (see [Table 25.1](#)) is set to other than the event counter mode, or the count source is set to other than AGTn/AGTWn underflow event signal (TCK[2:0] = 101b):
 - After 1 (count starts) is written to the TSTART bit in the AGTCR register while the count is stopped, the TCSTF flag in the AGTCR register remains 0 (count stops) for 3 cycles of the count source. Do not access the registers associated with AGT, AGTW other than the TCSTF flag 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 flag remains 1 for 3 cycles of the count source. When the TCSTF flag is set to 0, the count is stopped. Do not access the registers associated with AGT, AGTW. Other than the TCSTF flag until this bit is set to 0.
- When the operating mode (see [Table 25.1](#)) is set to event counter mode, or the count source is set to AGT1/AGTW1 underflow event signal (TCK[2:0] = 101b):
 - After 1 (count starts) is written to the TSTART bit in the AGTCR register while the count is stopped, the TCSTF flag in the AGTCR register remains 0 (count stops) for 2 PCLKB cycles. Do not access the registers associated with AGT/AGTW other than the TCSTF flag 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 flag remains 1 for 2 PCLKB cycles. When the TCSTF flag is set to 0, the count is stopped. Do not access the registers associated with AGT/AGTW other than the TCSTF flag until this bit is set to 0.

25.4.2 Access to Counter Register

When the TSTART bit and TCSTF flag in the AGTCR register are both 1 (count starts), allow at least 3 cycles of the count source clock between writes when writing to the AGT register successively.

25.4.3 When Changing Mode

The registers associated with AGT operating mode (AGTMR1, AGTMR2, AGTIOC, AGTISR, AGTCMSR and AGTIOC) can be changed only when the count is stopped with both the TSTART bit and TCSTF flag set to 0 (count stops). Do not change these registers during count operation.

When the registers associated with AGT operating mode are changed, the values of TEDGF, TUNDF, TCMAF, and TCMBF flags are undefined. Before starting the count, write 0 to the following flags:

- TEDGF (no active edge received)
- TUNDF (no underflow)
- TCMAF (no match)
- TCMBF (no match).

25.4.4 Output pin setting

When using the AGTOn, AGTIOOn, AGTOAn, or AGTOBn as an output pin, set up the Operation and determine the initial output values. Then set an output mode in the port register.

When using the AGTIOOn as an input pin in pulse width measurement mode or pulse period measurement mode, set up the Operation and start count operation. Then start to enter external events from the AGTIOOn pin. Invalidate the first measurement and validate the second and later completed measurements.

25.4.5 Digital Filter

When using the digital filter, do not start the timer operation for 5 cycles of the digital filter clock after setting TIFP[1:0] bits and when the TEDGSEL bit in the AGTIOC register changes.

25.4.6 How to Calculate Event Number, Pulse Width, and Pulse Period

- In event counter mode, event number is expressed mathematically as follows:
Event number = initial value of counter [AGT register] - counter value of active event end
- In pulse width measurement mode, pulse width is expressed mathematically as follows:
Pulse width = counter value of stopping measurement - counter value of next stopping measurement
- In pulse period measurement mode, input pulse period is expressed mathematically as follows:
Period of input pulse = (initial value of counter [AGT register] - reading value of the read-out buffer) + 1.

25.4.7 When Count is Forcibly Stopped by TSTOP Bit

After the counter is forcibly stopped by the TSTOP bit in the AGTCR register, do not access the following I/O registers for 1 cycle of the count source:

- AGT
- AGTCMA
- AGTCMB
- AGTCR
- AGTMR1
- AGTMR2.

25.4.8 When Selecting AGT0 Underflow as the Count Source

Operate according to the following procedures described in this section when selecting the underflow event signal as the count source.

(1) Procedure for starting operation

1. Set AGT.
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 AGTMR1.TCK[2:0] bits).

25.4.9 Module-stop function

AGT operation can be disabled or enabled using Module Stop Control Register D (MSTPCRD). The AGT module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#)

26. 8-Bit Timers (TMR)

26.1 Overview

8-bit timer (TMR) can count external events and provide multiple functions such as clearing counters, and outputting interrupt requests and pulses of required duty cycles, using the compare match signals with two registers.

In this section, PCLK refers to PCLKB.

Table 26.1 lists the TMR specifications, Table 26.2 lists the TMR functions, and Figure 26.1 shows a block diagram.

Table 26.1 TMR specifications

Parameter	Specifications
Count clock	- Frequency dividing clocks: PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/8192 - External clock: External count clock
Number of channels	(8 bits × 2 channels) × 1 unit
Compare match	Compare match A, compare match B
Counter clear	Selected from compare match A, compare match B, or external counter clear signal
Timer output	Output of pulse with desired duty cycles or of PWM signals
Cascading of two channels	16-bit count mode 16-bit timer using TMR0 for the upper 8 bits and TMR1 for the lower 8 bits - Compare match count mode The TMR1 counts the TMR0 compare matches
Interrupt sources	Compare match A, compare match B, overflow (TMR0, TMR1)
Event Link function	Compare match A, compare match B, overflow (TMR0)
DMAC/DTC activation	DMAC/DTC can be activated by a compare match A interrupt or a compare match B interrupt (TMR0, TMR1)
Trigger for starting conversion by A/D converter	TMR0 compare match A
Function for reducing power consumption	The module-stop state can be set for each unit

Table 26.2 TMR functions (1 of 2)

Parameter		Function		
Counter mode		8 bits		16 bits
Channel		TMR0	TMR1	TMR0 + TMR1
Count clock		PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMC10	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMC11	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMC11
Counter clear		TMR0.TCORA TMR0.TCORB TMR10	TMR1.TCORA TMR1.TCORB TMR11	TMR0.TCORA + TMR1.TCORA TMR0.TCORB + TMR1.TCORB TMR10
Compare match	Compare match A	✓	✓	✓
	Compare match B	✓	✓	✓
Timer output	Low-level output	✓	✓	✓
	High-level output	✓	✓	✓
	Toggled output	✓	✓	✓

Table 26.2 TMR functions (2 of 2)

Parameter		Function		
Counter mode		8 bits		16 bits
Channel		TMR0	TMR1	TMR0 + TMR1
DMAC/DTC activation	Compare match A	✓	✓	✓
	Compare match B	✓	✓	✓
	TCNT overflow	—	—	—
Interrupt	Compare match A	TMR_CMIA0	TMR_CMIA1	TMR_CMIA0
	Compare match B	TMR_CMIB0	TMR_CMIB1	TMR_CMIB0
	TCNT overflow	TMR_OVF0	TMR_OVF1	TMR_OVF0
Cascaded connection		TMR1 overflow	TMR0 compare match A	—
A/D converter conversion start trigger* ¹		✓	—	✓
ELC output event	Compare match A	✓	—	✓* ³
	Compare match B	✓	—	✓* ³
	TCNT overflow	✓	—	✓* ³
Module-stop setting* ²		MSTPCRD.MSTPD1 bit		

✓: Possible

—: Impossible

Note 1. For details, see [section 44, 14-Bit A/D Converter \(ADC14\)](#).Note 2. For details, see [section 13, Power-Saving Functions](#).Note 3. Only TMR0 events can be used. For details, see [section 21, Event Link Controller \(ELC\)](#).

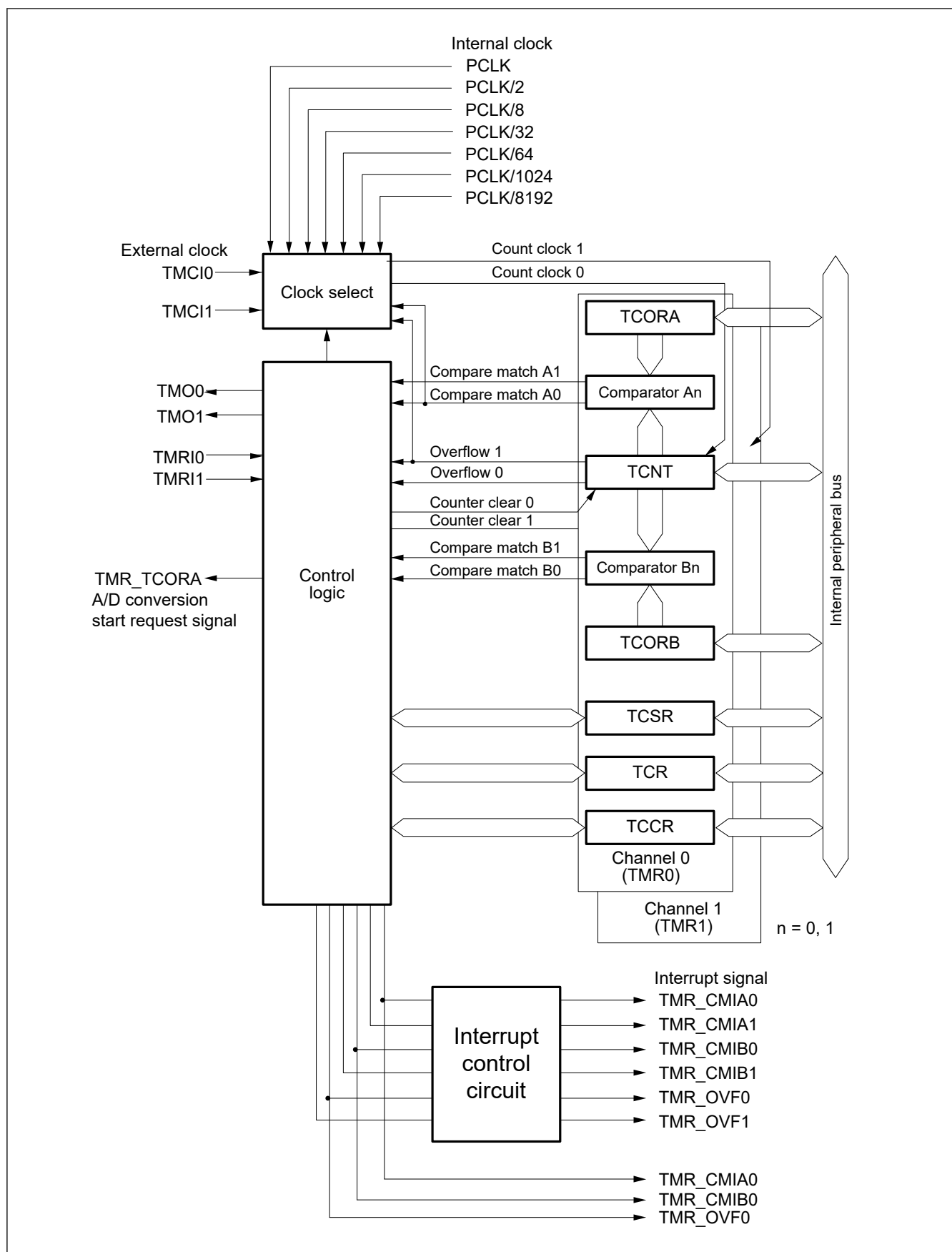


Figure 26.1 TMR block diagram

Table 26.3 shows TMR I/O pins.

Table 26.3 TMR I/O pins

Channel	Pin name	I/O	Function
TMR0	TMO0	Output	Compare match output
	TMCi0	Input	External count clock input
	TMRI0	Input	External counter reset input
TMR1	TMO1	Output	Compare match output
	TMCi1	Input	External count clock input
	TMRI1	Input	External counter reset input

26.2 Register Descriptions

26.2.1 TCNT : Timer Counter

Base address: $\text{TMRn} = 0x4005\_2000 + 0x0001 \times n$ ($n = 0, 1$)

Offset address: 0x08

Bit position: 7 6 5 4 3 2 1 0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	n/a	The TCNT counter is an 8-bit read/write up counter. The TMR0.TCNT counter and the TMR1.TCNT counter can be combined to function as a 16-bit counter. The TCCR.CSS[1:0] and CKS[2:0] bits are used to select the count clock. The TCNT counter can be cleared by an external counter reset signal, compare match A, or compare match B. Use the TCR.CCLR[1:0] bits to determine which compare match to use. When the TCNT counter overflows (its value changes from 0xFF to 0x00), an overflow interrupt is output if the interrupt request is enabled by the TCR.OVIE bit. See section 16, Interrupt Controller Unit (ICU) for information on the corresponding interrupt vector number.	R/W

26.2.2 TCORA : Time Constant Register A

Base address: $\text{TMRn} = 0x4005\_2000 + 0x0001 \times n$ ($n = 0, 1$)

Offset address: 0x04

Bit position: 7 6 5 4 3 2 1 0

Bit field:

Value after reset: 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
7:0	n/a	The TCORA register is an 8-bit read/write register. The TMR0.TCORA register and the TMR1.TCORA register can be combined to function as a 16-bit register. The value in the TCORA register is compared with the value in the TCNT counter. If the values match, compare match A is generated, and a compare match A interrupt is output if the interrupt request is enabled by the TCR.CMIEA bit. However, comparison is not performed during writes to the TCORA register. The timer output from the TMO_n pin can be controlled by the compare match A and the settings of the TCSR.OSA[1:0] bits.	R/W

26.2.3 TCORB : Time Constant Register B

Base address: $TMRn = 0x4005\_2000 + 0x0001 \times n$ ($n = 0, 1$)

Offset address: 0x06

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

Bit	Symbol	Function	R/W
7:0	n/a	The TCORB register is an 8-bit read/write register. The TMR0.TCORB register and the TMR1.TCORB register can be combined to function as a 16-bit register. The value in the TCORB register is compared with the value in the TCNT counter. If the values match, compare match B is generated, and a compare match B interrupt is output if the interrupt request is enabled by the TCR.CMIEB bit. However, comparison is not performed during writes to the TCORB register. The timer output from the TMON pin can be controlled by the compare match B and the settings of the TCSR.OSB[1:0] bits.	R/W

26.2.4 TCR : Timer Control Register

Base address: $TMRn = 0x4005\_2000 + 0x0001 \times n$ ($n = 0, 1$)

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CMIEB	CMIEA	OVIE	CCLR[1:0]	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
4:3	CCLR[1:0]	Counter Clear*1 0 0: Disable clearing 0 1: Clear by compare match A 1 0: Clear by compare match B 1 1: Clear by an external counter reset signal Select an edge or level with the TCCR.TMRIS bit	R/W
5	OVIE	Timer Overflow Interrupt Enable 0: Disable overflow interrupt requests (TMR_OVF _n) 1: Enable overflow interrupt requests (TMR_OVF _n)	R/W
6	CMIEA	Compare Match Interrupt Enable A 0: Disable compare match A interrupt requests (TMR_CMIAn) 1: Enable compare match A interrupt requests (TMR_CMIAn)	R/W
7	CMIEB	Compare Match Interrupt Enable B 0: Disable compare match B interrupt requests (TMR_CMIBn) 1: Enable compare match B interrupt requests (TMR_CMIBn)	R/W

Note 1. To use an external counter reset signal, the corresponding pin must be set. For details, [section 22, I/O Ports](#).

The TCR register is used to set clearing conditions for the TCNT counter, and enable or disable interrupt requests.

CCLR[1:0] bits (Counter Clear)

The CCLR[1:0] bits select the clearing conditions for the TCNT counter.

OVIE bit (Timer Overflow Interrupt Enable)

The OVIE bit enables or disables an interrupt request generated by an overflow of the TCNT counter (TMR_OVF_n).

CMIEA bit (Compare Match Interrupt Enable A)

The CMIEA bit enables or disables an interrupt request generated by compare match A that is output when the values of the TCORA register and the TCNT counter match (TMR_CMIA_n).

CMIEB bit (Compare Match Interrupt Enable B)

The CMIEB bit enables or disables an interrupt request generated by compare match B that is output when the values of the TCORB register and the TCNT counter match (TMR_CMIB_n).

26.2.5 TCCR : Timer Counter Control Register

Base address: TMR_n = 0x4005_2000 + 0x0001 × n (n = 0, 1)

Offset address: 0x0A

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

Bit	Symbol	Function	R/W
2:0	CKS[2:0]	Clock Select*1 See Table 26.4 .	R/W
4:3	CSS[1:0]	Clock Source Select See Table 26.4 .	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	TMRIS	Timer Reset Detection Condition Select 0: Clear on the falling edge of an external counter reset signal 1: Clear when an external counter reset signal is low level	R/W

Note 1. To use an external count clock, you must set the associated pin. For details, [section 22, I/O Ports](#).

The TCCR register is used to select a count clock for the TCNT counter and set conditions for detecting an external counter reset signal.

CKS[2:0] bits (Clock Select), CSS[1:0] bits (Clock Source Select)

These bits select the count clock. For details, see [Table 26.4](#).

TMRIS bit (Timer Reset Detection Condition Select)

The TMRIS bit is enabled when the TCR.CCLR[1:0] bits are set to 11b (clear by an external counter reset signal) and selects the conditions for detecting a counter reset (level or edge).

Table 26.4 Clock input to TCNT counter and count conditions

Channel	TCCR register					Description	
	CSS[1:0]		CKS[2:0]				
	b4	b3	b2	b1	b0		
TMR0	0	0	—	0	0	Disable clock input	
					1	Count on the rising edge of an external count clock ^{*1}	
				1	0	Count on the falling edge of an external count clock ^{*1}	
					1	Count on both rising and falling edges of an external count clock ^{*1}	
		1	0	0	0	Internal clock: count at PCLK	
					1	Internal clock: count at PCLK/2	
					1	0	Internal clock: count at PCLK/8
						1	Internal clock: count at PCLK/32
			1	0	0	Internal clock: count at PCLK/64	
					1	Internal clock: count at PCLK/1024	
				1	0	Internal clock: count at PCLK/8192	
					1	Disable clock input	
	1	0	—	—	—	Settings prohibited	
	1	1	—	—	—	Count at TMR1. TCNT overflow signal ^{*2} .	
TMR1	0	0	—	0	0	Disable clock input	
					1	Count on the rising edge of an external count clock ^{*1}	
				1	0	Count on the falling edge of an external count clock ^{*1}	
					1	Count on both rising and falling edges of an external count clock ^{*1}	
		1	0	0	0	Internal clock: count at PCLK	
					1	Internal clock: count at PCLK/2	
					1	0	Internal clock: count at PCLK/8
						1	Internal clock: count at PCLK/32
			1	0	0	Internal clock: count at PCLK/64	
					1	Internal clock: count at PCLK/1024	
				1	0	Internal clock: count at PCLK/8192	
					1	Disable clock input	
	1	0	—	—	—	Settings prohibited	
	1	1	—	—	—	Count at TMR0. TCNT compare match A ^{*2} .	

Note 1. To use an external count clock, the corresponding pin must be set. For details, see [section 22, I/O Ports](#).

Note 2. If the clock input of the TMR0 is the overflow signal of the TMR1.TCNT counter and the clock input of the TMR1 is the compare match signal of the TMR0.TCNT counter, the TCNT count clock is not generated. Do not use this setting.

26.2.6 TCSR : Timer Control/Status Register

Base address: TMRn = 0x4005_2000 + 0x0001 × n (n = 0, 1)

Offset address: 0x02

Bit position: 7 6 5 4 3 2 1 0

Bit field:	—	—	—	ADTE ^{*3}	OSB[1:0]	OSA[1:0]

Value after reset: x x x 0^{*3} 0 0 0 0

Bit	Symbol	Function	R/W
1:0	OSA[1:0]	Output Select A* ¹ 0 0: No change occurs 0 1: High-level output 1 0: Low-level output 1 1: Inverted output (toggled output)	R/W
3:2	OSB[1:0]	Output Select B* ¹ 0 0: No change occurs 0 1: High-level output 1 0: Low-level output 1 1: Inverted output (toggled output)	R/W
4	ADTE	A/D Trigger Enable* ² * ³ 0: Disable A/D conversion start request by compare match A 1: Enable A/D conversion start request by compare match A	R/W
7:5	—	The read values are undefined. The write value should be 0.	R/W

Note 1. When the OSA[1:0] and OSB[1:0] bits are set to 0, the output enable for the TMO0 pin or TMO1 pin is negated and the I/O port output high impedance. When the OSA[1:0] or OSB[1:0] bits is 1, the timer output pin is held low until the first compare match occurs after a reset.

Note 2. The ADTE bit is incorporated only in the TMR0.TCSR register. For details on the associated A/D channels, see [section 44, 14-Bit A/D Converter \(ADC14\)](#).

Note 3. For the TMR1.TCSR register, bit 4 is a reserved bit. This bit is read as 1. The write value should be 1.

The TCSR register controls output based on the compare matches.

OSA[1:0] bits (Output Select A)

The OSA[1:0] bits select the output type of the TMO_n pin when compare match A occurs from the match between the TCORA register and the TCNT counter.

OSB[1:0] bits (Output Select B)

The OSB[1:0] bits select the output type of the TMO_n pin when compare match B occurs from the match between the TCORB register and the TCNT counter.

ADTE bit (A/D Trigger Enable)

The ADTE bit enables or disables the A/D conversion start request triggered by compare match A.

26.3 Operation

26.3.1 Pulse Output

[Figure 26.2](#) shows an example of outputting pulses with a certain duty cycle.

1. Set the TCR.CCLR[1:0] bits to 01b (clear by compare match A) so that the TCNT counter is cleared at a compare match with the TCORA register.
2. Set the TCSR.OSA[1:0] bits to 10b (high-level output) so that the TMO_n pin is driven high when a compare match with the TCORA register occurs. Set the TCSR.OSB[1:0] bits to 01b (low-level output) so that the TMO_n pin is driven low when a compare match with the TCORB register occurs.

With these settings, the TMR provides a waveform output at a cycle specified in the TCORA register and a pulse width specified in the TCORB register without software intervention.

The timer output pin is held low after the TCSR.OSA[1:0] bits or the TCSR.OSB[1:0] bits are set until the first compare match occurs after a reset.

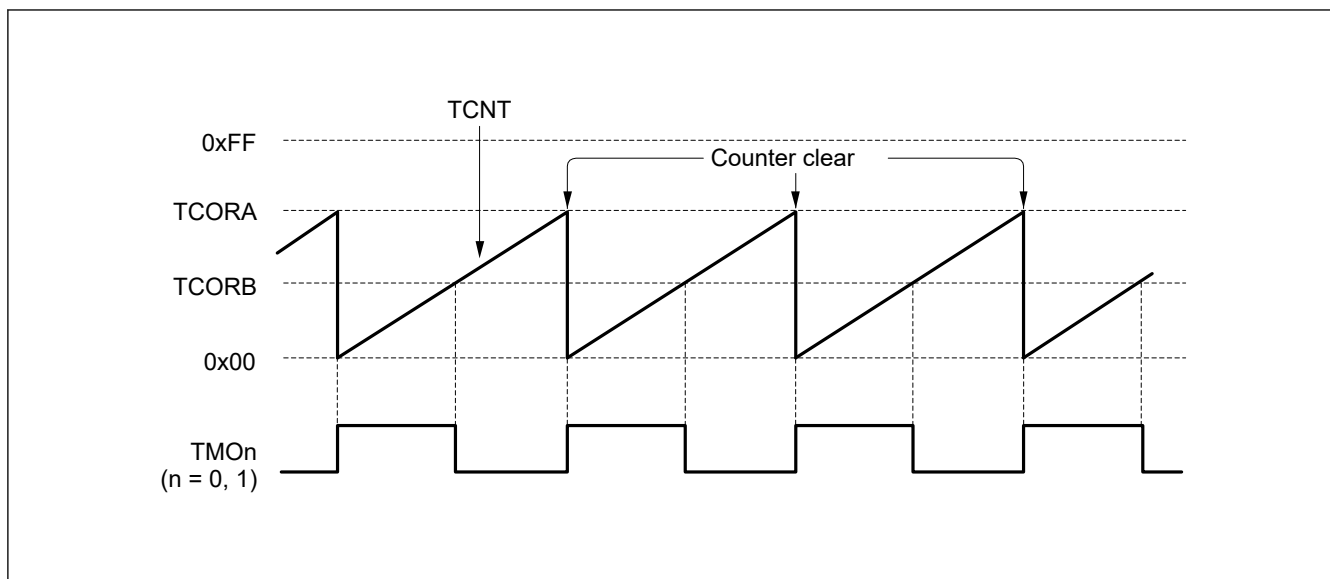


Figure 26.2 Pulse output example

26.3.2 Delayed Pulse Output Caused by External Counter Reset Input

Figure 26.3 shows an example of outputting a pulse at a desired delay time by inputting a reset signal to the TMRIn input pin.

1. Set the TCR.CCLR[1:0] bits to 11b (clear by an external counter reset signal) and the TCCR.TMRIS bit to 1 (clear when an external counter reset signal is low) so that the TCNT counter is cleared when the TMRIn input pin is driven high.
2. Set the TCSR.OSA[1:0] bits to 01b (high-level output) so that the TMO_n pin is driven high when a compare match with the TCORA register occurs. Set the TCSR.OSB[1:0] bits to 10b (low-level output) so that the TMO_n pin is driven low when a compare match with the TCORB register occurs.

With these settings, the TMR can provide a waveform output at a delay time due to the value set in the TCORA register from an input to the TMRIn pin. The pulse width can be set by the TCORA and TCORB register. The difference between the TCORA and TCORB register value determines the pulse width.

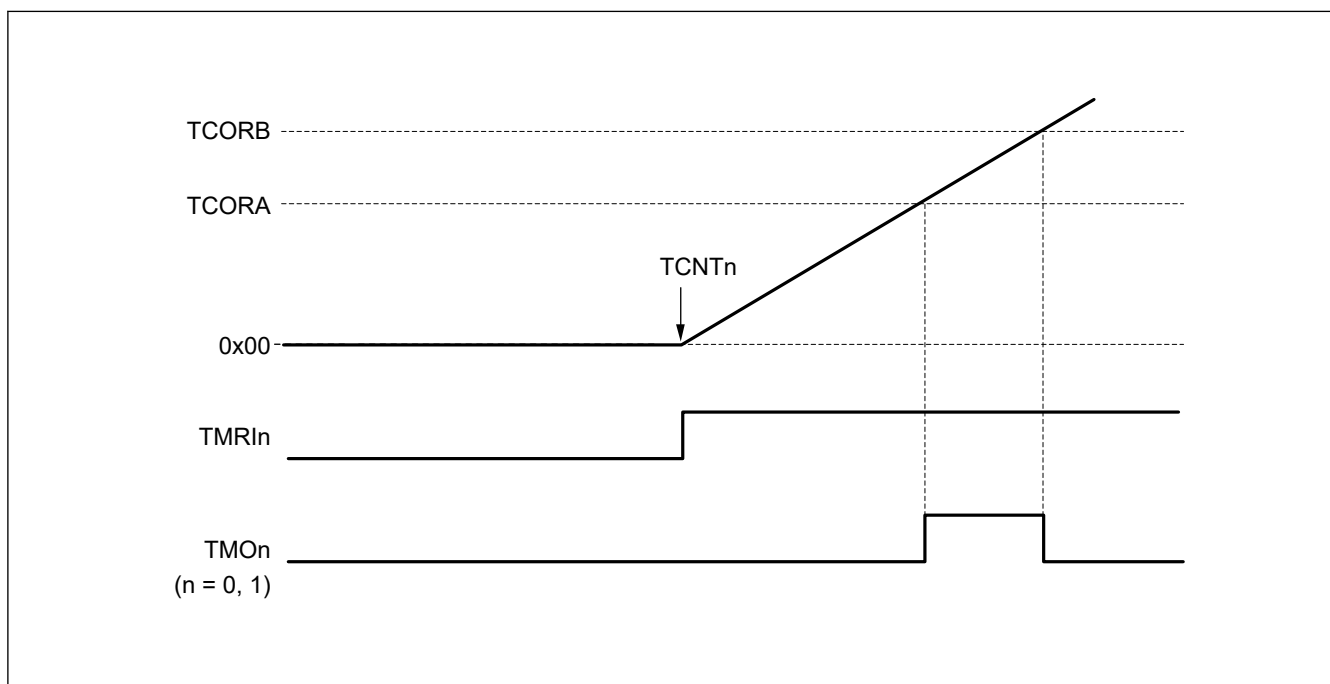


Figure 26.3 Pulse output example caused by external counter reset input

26.4 Operation Timing

26.4.1 TCNT Count Timing

Figure 26.4 shows the count timing of the TCNT counter when the internal clock is operating. Figure 26.5 shows the count timing of the TCNT counter when an external count clock is operating.

The pulse width of the external count clock must be at least 1.5 PCLK for single-edge detection, and at least 2.5 PCLK for both-edge detection. The counter does not operate properly at narrower pulse widths.

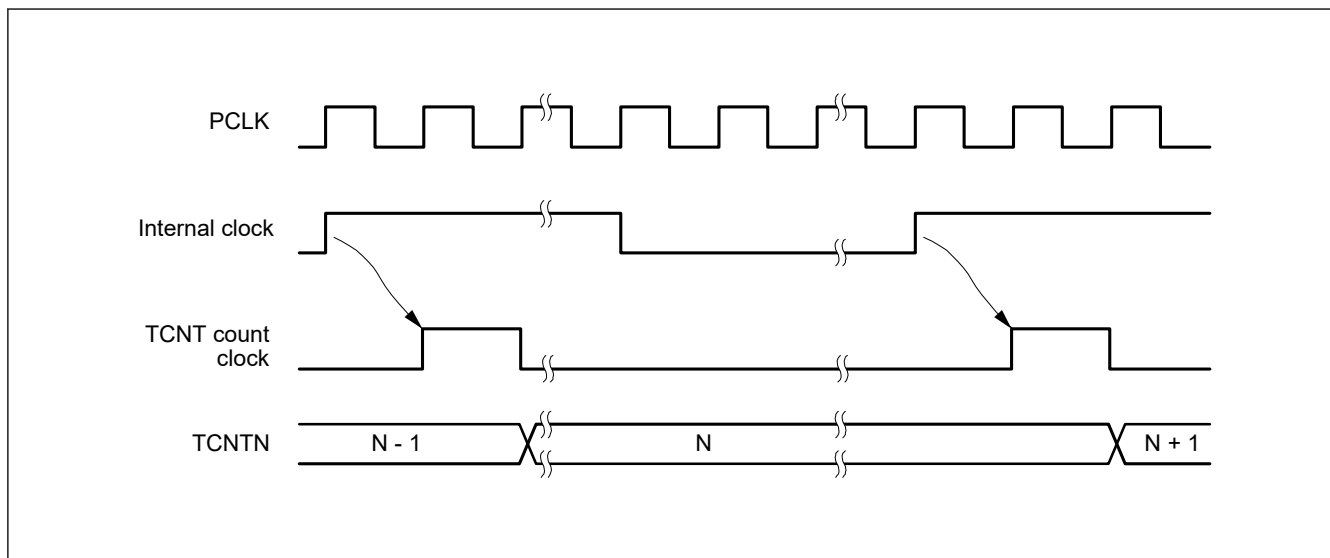


Figure 26.4 Count timing when internal clock is operating

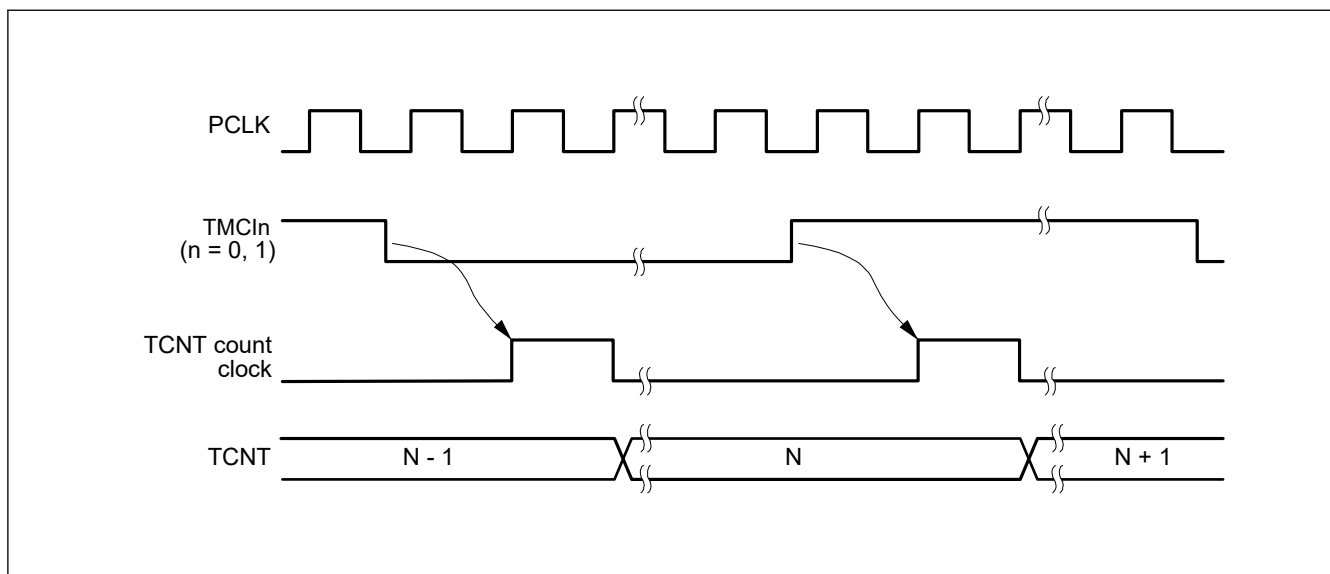


Figure 26.5 Count timing when external clock is operating (for both-edge detection)

26.4.2 Timing of Interrupt on Compare Match

When the value of the TCORA or TCORB register matches the value of the TCNT counter, a compare match occurs. When the interrupt request is enabled, a compare match interrupt signal is output. A compare match is generated in the final state in which TCNT and TCORA or TCORB match (the point where the count value matched is when TCNT is updated). After a match between TCNT and TCORA or TCORB occurred, no compare match is generated until the TCNT count clock is generated.

Figure 26.6 shows the timing for output of the interrupt signal.

See [section 16, Interrupt Controller Unit \(ICU\)](#) for information on the corresponding interrupt vector number.

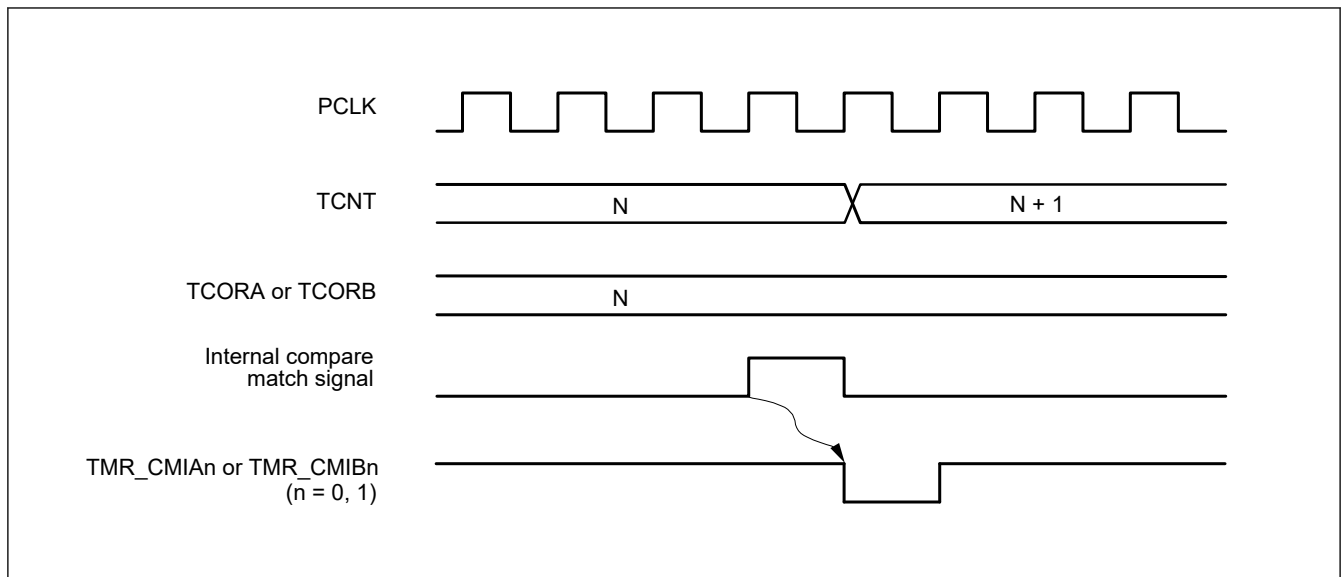


Figure 26.6 Timing of interrupt on compare match

26.4.3 Timing of Signal Output on Compare Match

When a compare match signal occurs, the output value specified by the TCSR.OSA[1:0] and TCSR.OSB[1:0] bits are output to the timer output pin (TMO_n). [Figure 26.7](#) shows the timing for signal output triggered by a compare match A signal.

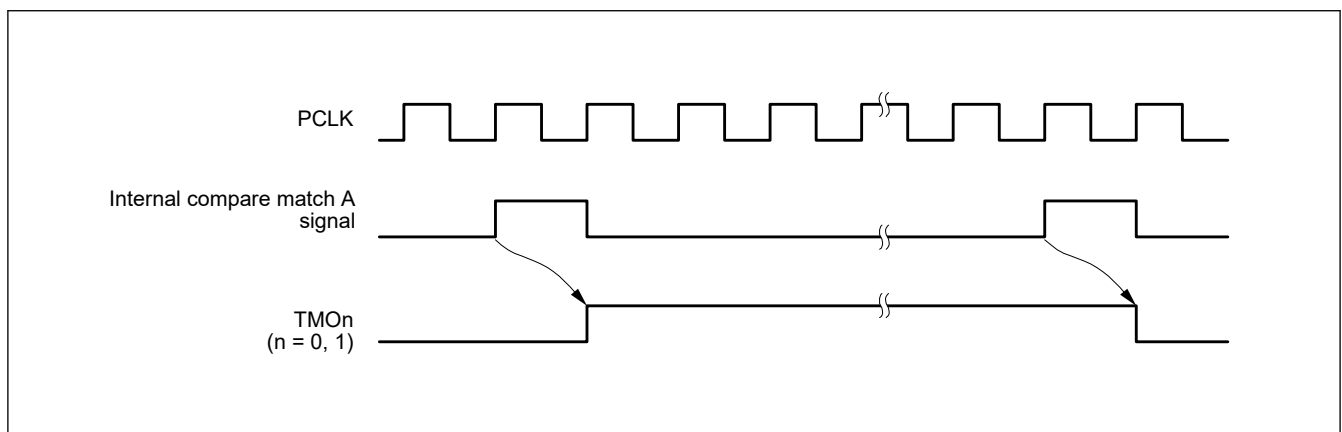


Figure 26.7 Timing of signal output triggered by compare match A signal

26.4.4 Timing of Counter Clearing by Compare Match

The TCNT counter can be cleared by compare match A or compare match B depending on the setting of the TCR.CCLR[1:0] bits.

[Figure 26.8](#) shows the timing of counter clearing caused by a compare match.

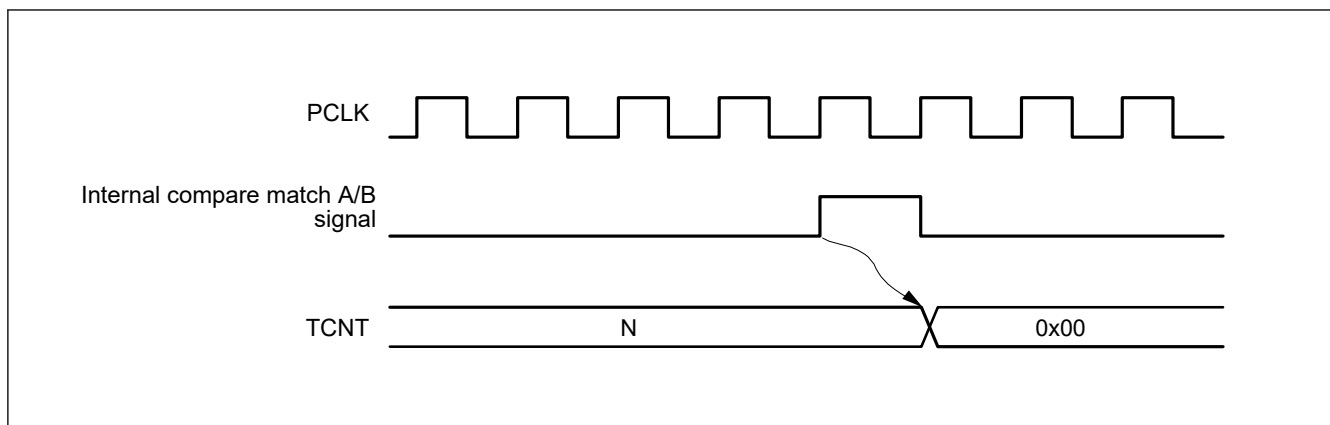


Figure 26.8 Timing of counter clearing by a compare match

26.4.5 Timing of TCNT Counter Clearing by External Counter Reset

The TCNT counter can be cleared on the falling edge of an external counter reset signal or when the signal is low depending on the settings of the TCR.CCLR[1:0] bits. At least 2 PCLK cycles are required from the reset signal input to clear the TCNT counter. Figure 26.9 and Figure 26.10 show the timing of clearing triggered by an external counter reset signal.

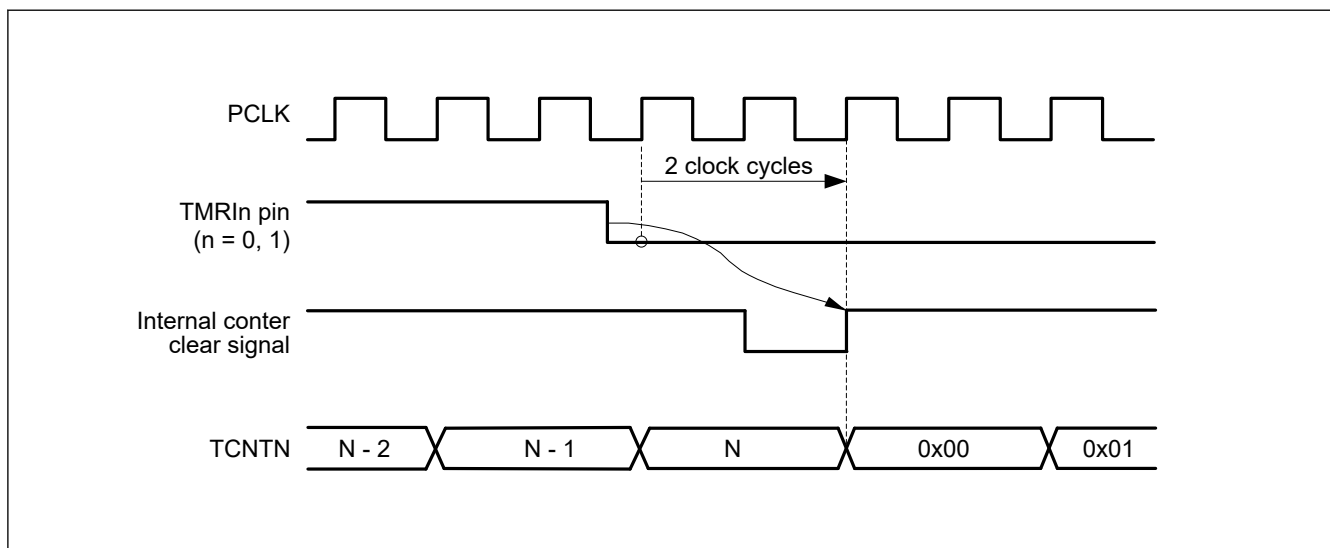


Figure 26.9 Timing of clearance by external counter reset signal (on falling edge)

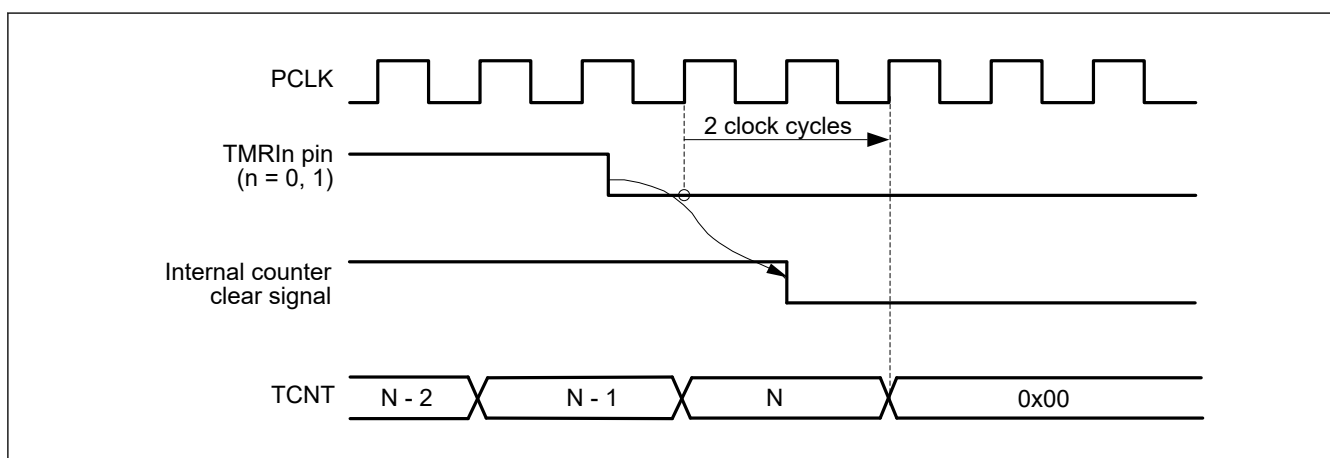


Figure 26.10 Timing of clearance by external counter reset signal (low level)

26.4.6 Timing of Interrupt on Overflow

When the TCNT counter overflows (the value changes from 0xFF to 0x00), an overflow interrupt signal is output if the interrupt request is enabled. Figure 26.11 shows the timing for output of the interrupt signal. See section 16, [Interrupt Controller Unit \(ICU\)](#) for information on the corresponding interrupt vector number.

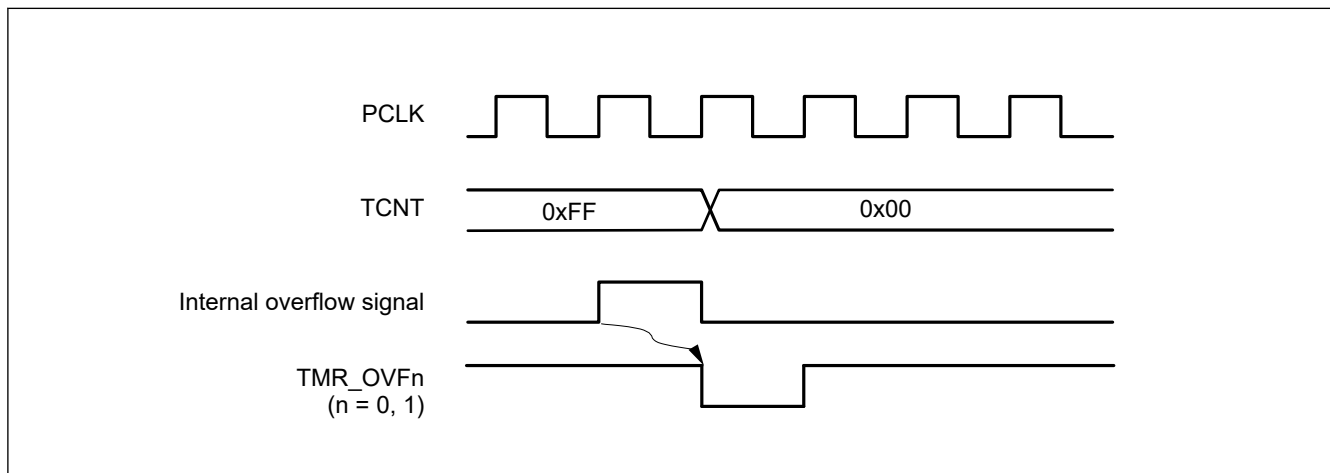


Figure 26.11 Timing of interrupt on overflow

26.5 Operation with Cascaded Connection

If the CSS[1:0] bits in either the TMR0.TCCR or TMR1.TCCR register are set to 11b, the two TMR channels are cascaded. In this configuration, the two timers can be used as a single 16-bit timer (16-bit count mode) or can be operated in compare match count mode where TMR1 counts the compare match value of TMR0.

26.5.1 16-Bit Count Mode

When the TMR0.TCCR.CSS[1:0] bits are set to 11b, the TMR0 and the TMR1 function as a 1-channel 16-bit timer with TMR0 occupying the upper 8 bits and TMR1 occupying the lower 8 bits.

(1) Counter clear specification

- The settings of the TMR0.TCR.CCLR[1:0] bits are valid for 16-bit counter. If the TMR0.TCR.CCLR[1:0] bits are set for counter clear of a compare match, the 16-bit counter of both TMR0.TCNT and TMR1.TCNT is cleared when a 16-bit compare match event occurs. The 16-bit counter of both TMR0.TCNT and TMR1.TCNT is also cleared when counter clear using the TMRI0 pin is set.
- The settings of the TMR1.TCR.CCLR[1:0] bits are invalid.

(2) Pin output

- Control of output from the TMO0 pin by the TMR0.TCSR.OSA[1:0] and OSB[1:0] bits is in accordance with the 16-bit compare match conditions
- Control of output from the TMO1 pin by the TMR1.TCSR.OSA[1:0] and OSB[1:0] bits is in accordance with the lower 8-bit compare match conditions.

26.5.2 Compare Match Count Mode

When the TMR1.TCCR.CSS[1:0] bits are set to 11b, the TMR1.TCNT counter counts the number of occurrences of compare match A for TMR0. The TMR0 and TMR1 are controlled separately. Conditions such as the generation of interrupts, output from the TMO_n (n = 0, 1) pin, and counter clear are in accordance with the settings for each channel.

26.6 Interrupt Sources

26.6.1 Interrupt Sources and DMAC/DTC Activation

Three interrupt sources are available for the TMR_n:

- TMR_CMIA_n
- TMR_CMIB_n
- TMR_OVF_n

Table 26.5 shows the interrupt sources and priorities. TMR_CMIA_n and TMR_CMIB_n interrupts can be used to start the DMAC or DTC.

Table 26.5 TMR interrupt sources

Name	Interrupt source	DMAC/DTC activation
TMR_CMIA0	Compare match with TMR0.TCORA	Possible
TMR_CMIB0	Compare match with TMR0.TCORB	Possible
TMR_OVF0	TMR0.TCNT overflow	Not possible
TMR_CMIA1	Compare match with TMR1.TCORA	Possible
TMR_CMIB1	Compare match with TMR1.TCORB	Possible
TMR_OVF1	TMR1.TCNT overflow	Not possible

26.6.2 A/D Converter Activation

The 14-bit A/D converter^{*1} can be activated by compare match A for the TMR0. If the TMR_n.TCSR.ADTE bit is set to 1 (enable A/D conversion start request by compare match A), the compare match A requests the 14-bit A/D converter to start A/D conversion. If the 8-bit timer conversion start trigger is selected on the 14-bit A/D converter, A/D conversion starts. Table 26.6 shows A/D conversion start request source.

Note 1. For the associated A/D converter units, see [section 44, 14-Bit A/D Converter \(ADC14\)](#).

Table 26.6 A/D conversion start request source

Module symbol	A/D conversion start request source	A/D conversion start request
ADC14	Compare match between TMR0.TCORA and TMR0.TCNT	TMR_TCORA

26.7 Link Operation by the ELC

26.7.1 Event Signal Output to the ELC

The TMR 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 TMR outputs compare match A, compare match B, and overflow signals as event signals. The applicable channel is TMR0.

An event signal can be output regardless of the corresponding interrupt request enable bit settings (TMR0.TCR.OVIE, TMR0.TCR.CMIEA, TMR0.TCR.CMIEB). For details, [section 21, Event Link Controller \(ELC\)](#).

The event output function supports operation with cascaded connection. In 16-bit count mode, an interrupt from TMR0 can be used as an event signal. However, the event output function does not support compare match count mode and TMR1 interrupt signals in 16-bit count mode.

26.8 Usage Notes

26.8.1 Settings for the Module-Stop Function

The Module Stop Control Register D (MSTPCRD) can enable or disable TMR operation. The TMR module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

26.8.2 Notes on Cycle Setting

When counter clearing by a compare match is set, the TCNT counter is cleared when its value matches the value in the TCORA or TCORB register in the final PCLK cycle (the point where the count value matched is when TCNT is updated). The counter frequency is given by the following formula:

$$f = \text{PCLK} / (N + 1)$$

f: counter frequency

PCLK: operating frequency

N: TCORA or TCORB register setting value

26.8.3 Conflict between Writing to and Clearing of the TCNT Counter

As shown in Figure 26.12, if the counter clear signal is generated at the same time when the CPU writes to the TCNT counter, TCNT clearing takes precedence, and the TCNT write operation is not performed.

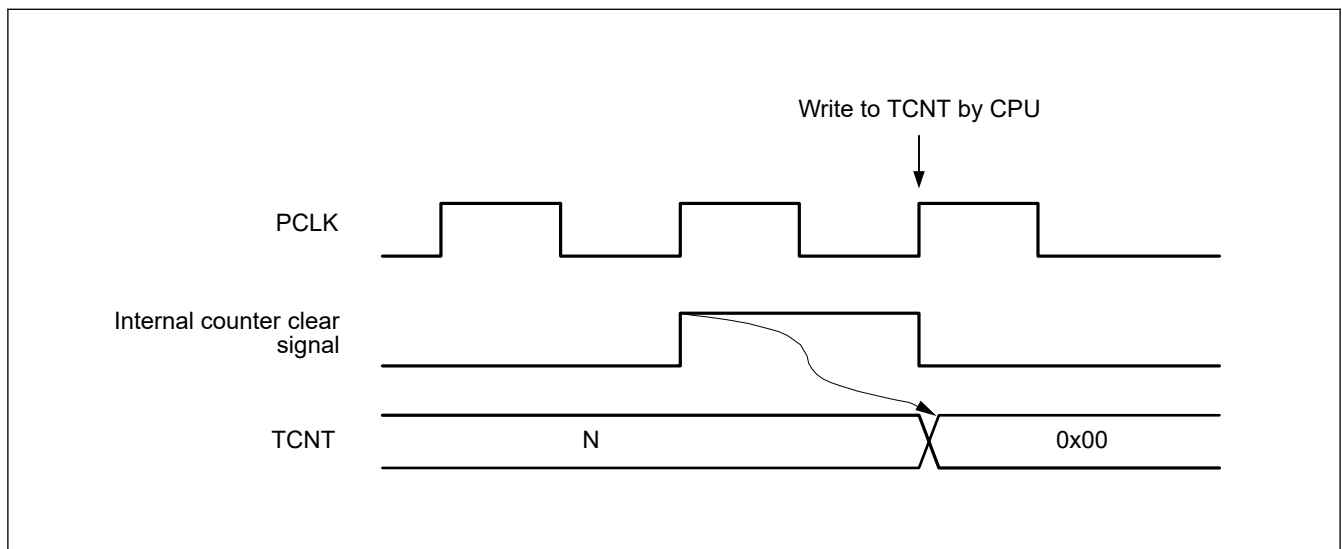


Figure 26.12 Conflict between writing to and clearing of the TCNT counter

26.8.4 Conflict between Writing to and Incrementing of the TCNT Counter

As shown in Figure 26.13, if incrementing occurs at the same time when the CPU writes to the TCNT counter, the TCNT write operation takes precedence and TCNT is not incremented.

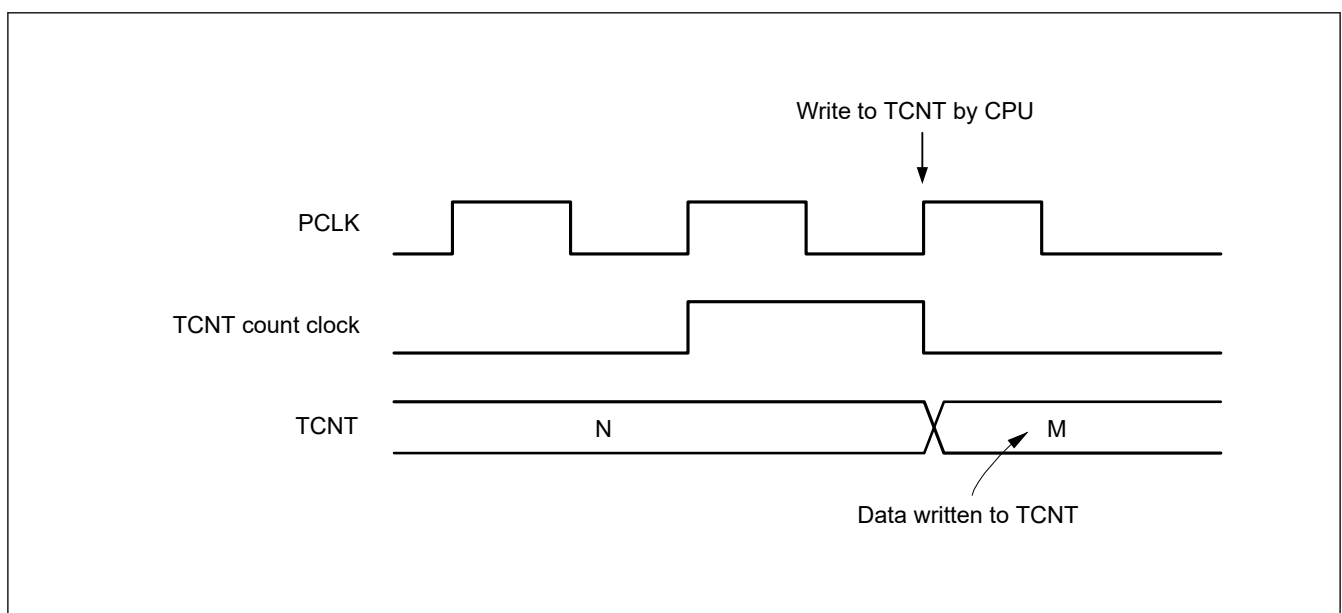


Figure 26.13 Conflict between writing to and incrementing of the TCNT counter

26.8.5 Conflict between Writing to the TCORA or TCORB Register and Compare Match

As shown in [Figure 26.14](#), if a compare match is generated at the same time when the CPU writes to the TCORA or TCORB register, the write takes precedence and the compare match signal does not go high.

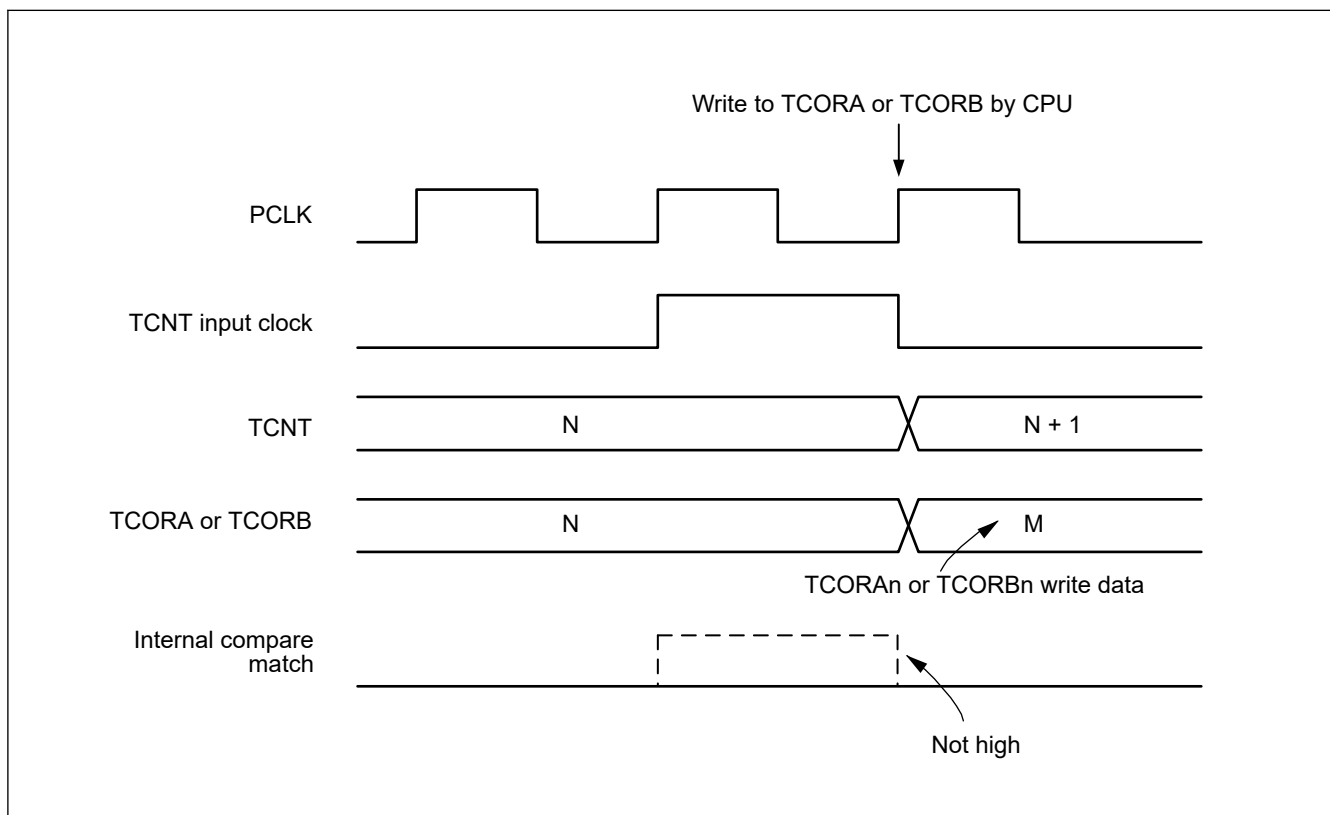


Figure 26.14 Conflict between writing to the TCORA or TCORB register and compare match

26.8.6 Conflict between Compare Matches A and B

If compare match events A and B occur at the same time, between the output types set for compare match A and compare match B, the output with higher priority is performed, as listed in [Table 26.7](#).

Table 26.7 Timer output priorities

Timer output setting	Priority
Toggled output	High
High-level output	↑
Low-level output	↑
No change	Low

26.8.7 Switching of Internal Clocks and TCNT Operation

The TCNT counter might be incremented erroneously depending on when the internal clocks are switched. [Table 26.8](#) shows the relationship between the timing at which the internal clocks are switched (by rewriting the TCCR.CKS[2:0] bits) and the operation of the TCNT counter.

When the TCNT count clock is generated from an internal clock, the rising edges of the internal clock pulses are detected. If the clocks are changed with the signal level switching from low to high as shown in No. 2 in [Table 26.8](#), the timing of switching is at an edge. If this is the case, a TCNT count clock is generated and the TCNT counter is incremented.

The erroneous increment of the TCNT counter can also occur when switching between an internal clock and an external count clock.

Table 26.8 Switching of internal clocks and TCNT operation (1 of 2)

No.	Timing to change TCCR.CKS[2:0] bits settings	TCNT counter operation
1	Switching from low to low ^{*1}	TCNT TCCR.CKS[2:0] bits changed
2	Switching from low to high ^{*2}	TCNT TCCR.CKS[2:0] bits changed
3	Switching from high to low ^{*4}	TCNT TCCR.CKS[2:0] bits changed

Table 26.8 Switching of internal clocks and TCNT operation (2 of 2)

No.	Timing to change TCCR.CKS[2:0] bits settings	TCNT counter operation
4	Switching from high to high	TCNT count clock TCNT TCCR.CKS[2:0] bits changed

Note 1. Includes switching from low to stop, and from stop to low.

Note 2. Includes switching from stop to high.

Note 3. Generated because the timing of switching of the signal level is at an edge, and TCNT is incremented.

Note 4. Includes switching from high to stop.

26.8.8 Clock Source Setting with Cascaded Connection

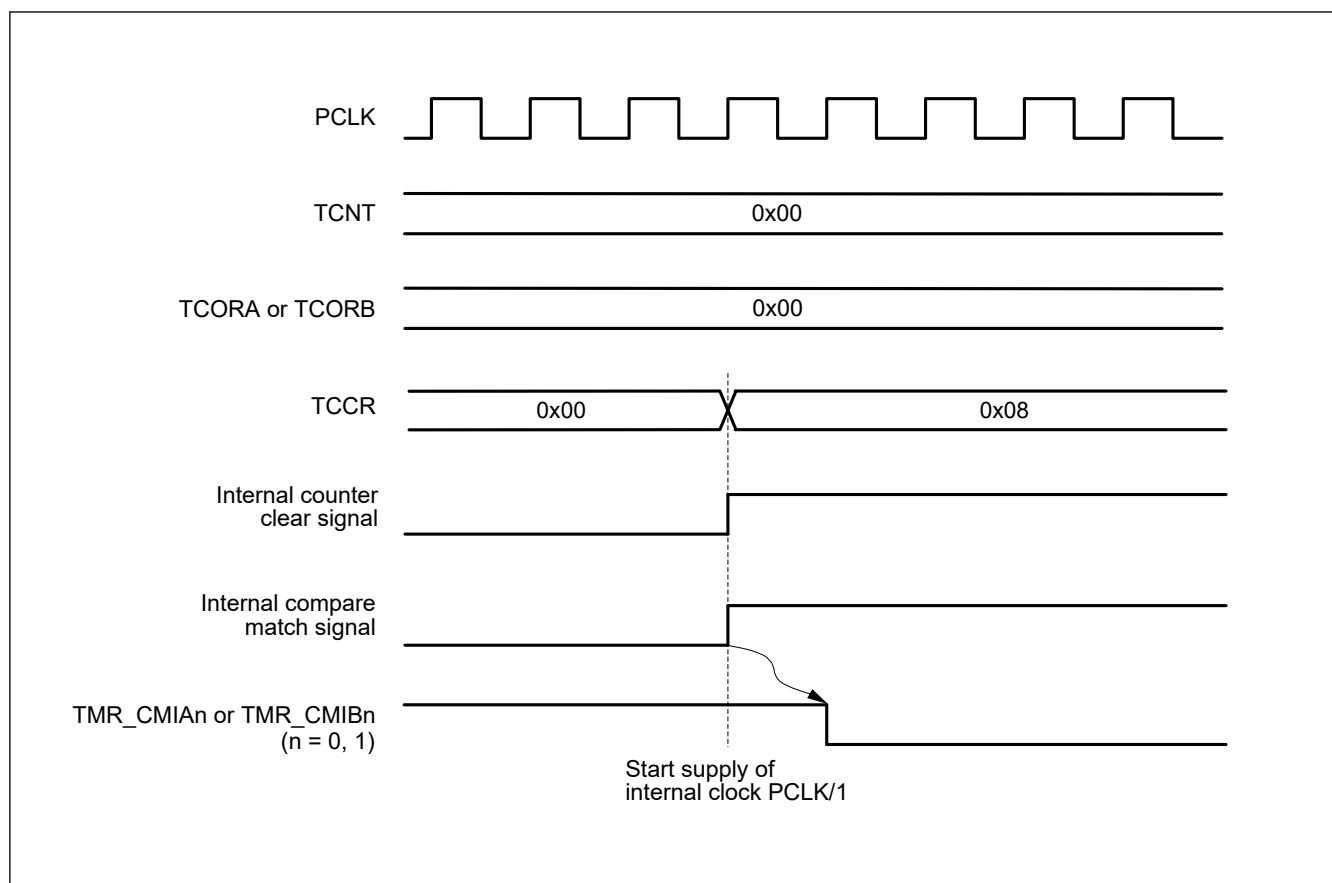
If 16-bit count mode and compare match count mode are specified at the same time, the count clocks for TMR0.TCNT and TMR1.TCNT counters are not generated, and the counters stop. Do not specify 16-bit count mode and compare match count mode simultaneously.

For details on the settings of each mode in the cascaded connection, see [section 26.5.1. 16-Bit Count Mode](#), and [section 26.5.2. Compare Match Count Mode](#).

26.8.9 Continuous Output of Compare Match Interrupt Signal

When the TCORA or TCORB register is set to 0x00, PCLK/1 is selected as the internal clock, and compare match is set as the counter clear source, the TCNT counter remains 0x00 and is not updated, and a compare match A/B interrupt remains asserted. At this time, the interrupt controller cannot detect the second and later interrupts.

[Figure 26.15](#) shows the timing for continuous output of the interrupt signal in response to a compare match.

**Figure 26.15** Continuous output of compare match interrupt signal

27. Wake Up Timer (WUPT)

27.1 Overview

This MCU has wake up timer (WUPT) based on 32-bit counters. The wake up timer provides multiple functions such as resetting count, and outputting interrupt requests and pulses to external pins when an overflow occurs.

Table 27.1 lists the WUPT specifications and Figure 27.1 shows a block diagram of the WUPT.

Table 27.1 WUPT specifications

Item	Description
Count clock	SOSC clock (32.768 kHz) Count operation is possible during the Deep Software Standby mode.
Number of channels	32 bits × 1 channel
Overflow	Overflow can be occurred at an arbitrary setting value.
Count clear	When an overflow occurs, or count reset
Pulse output	Synchronizing an overflow interrupt and pulse output to the external pin (TMWO) Pulse output function is also available in the Deep Software Standby mode.
Interrupt sources	When an overflow occurs (WUPT_OVI) Interrupt function is also issued in the Deep Software Standby mode and the recovery from the Deep Software Standby mode can be executed.
Event link function	When an overflow occurs (WUPT_OVI_ELC)
Function for reducing power consumption	- The module-stop state can be set. - In the Deep Software Standby mode, the operation is possible.

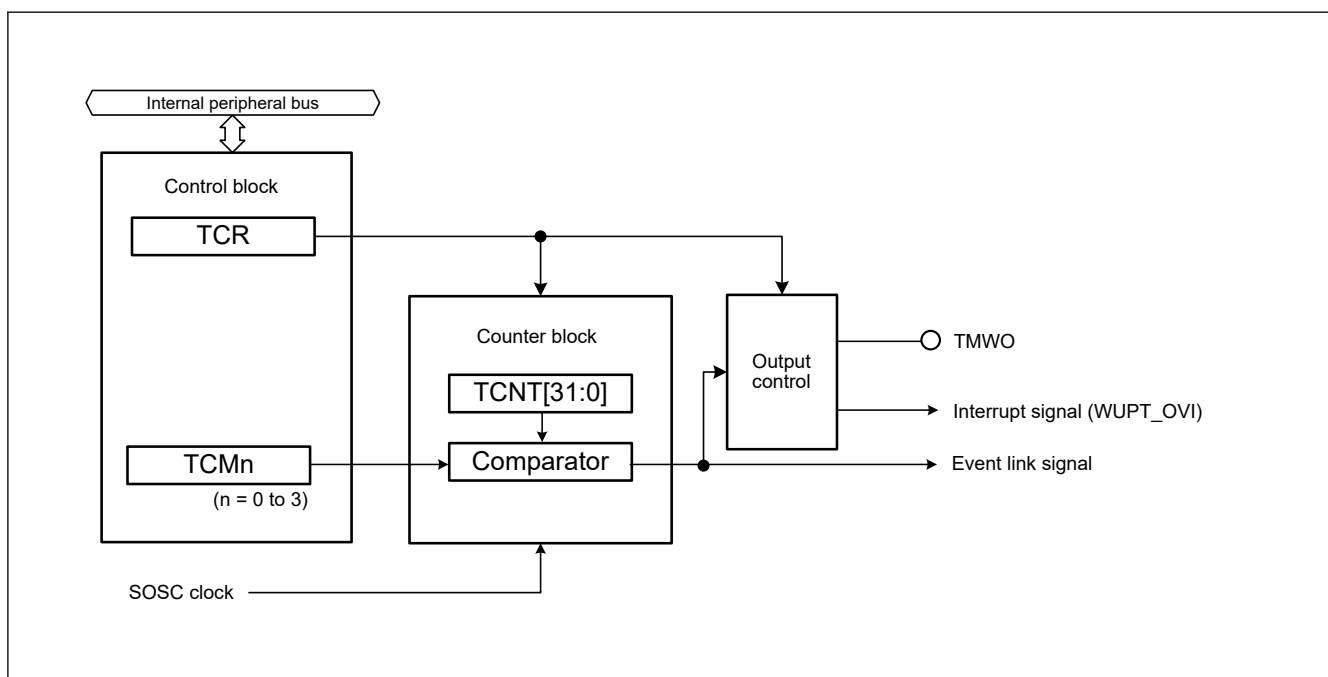


Figure 27.1 WUPT block diagram

Table 27.2 lists the input and output pins used in the WUPT.

Table 27.2 WUPT I/O pin

Pin Name	I/O	Function
TMWO	Output	Pulse output pin

27.2 Register Descriptions

27.2.1 TCMn : Timer Compare Match Register (n = 0 to 3)

Base address: WUPT = 0x4008_4480

Offset address: 0x00 + n

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

Bit	Symbol	Function	R/W
7:0	n/a	When the setting value in the TCMn register matches the value in the TCNT counter, an overflow occurs. The TCMn register and TCNT counter compare the values according to the following relationships. TCM0 = TCNT[7:0] TCM1 = TCNT[15:8] TCM2 = TCNT[23:16] TCM3 = TCNT[31:24]	R/W

Before the TCMn registers are written, stop the counter by the TCR.TCST bit and reset it by the TCR.TCRS bit.

The value of TCMn register is always compared with the value in the TCNT counter, and if the values match, an overflow occurs. When the interrupt request is enabled by the TCR.OVIE bit, an overflow interrupt is output.

When writing to the TCMn register, the value of these bits is not compared with the value in the TCNT counter. The settings of the TCMn register and TCR.OS bits can control the timer output from the TMWO pin.

27.2.2 TCR : Timer Control Register

Base address: WUPT = 0x4008_4480

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TCST	TCRS	—	TCCE	OVIE	TMOE	OS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OS[1:0]	Output Select 0 0: Low-level output 0 1: Low-level output 1 0: High-level output 1 1: Inverted output (toggled output)	R/W
2	TMOE	Timer Output Enable 0: No pulse output from the TMWO pin 1: Pulse output from the TMWO pin	R/W
3	OVIE	Timer Overflow Interrupt Enable 0: Disable overflow interrupt requests (WUPT_OVI) 1: Enable overflow interrupt requests (WUPT_OVI)	R/W
4	TCCE	Timer Count Clock Enable 0: Stop providing the TCNT count clock 1: Provide the TCNT count clock	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6	TCRS	Timer Count Reset 0: The TCNT counter is in normal operation 1: Reset the value in the TCNT counter	R/W
7	TCST	Timer Count Start 0: Stop count operation in the TCNT counter 1: Start count operation in the TCNT counter	R/W

OS[1:0] bits (Output Select)

The OS[1:0] bits select the output type of the TMWO pin when an overflow occurs. Before writing to these bits, stop the counter by the TCST bits.

TMOE bit (Timer Output Enable)

The TMOE bit sets the pulse output from the TMWO pin. Before writing to this bit, stop the counter by the TCST bit.

OVIE bit (Timer Overflow Interrupt Enable)

The OVIE bit enables or disables an interrupt request generated by an overflow of the TCNT counter (WUPT_OVI). Before writing to this bit, stop the counter by the TCST bit.

TCCE bit (Timer Count Clock Enable)

TCCE bit sets clock providing to the TCNT counter. When the clock providing is stopped, stop the counter by the TCST bit before the counter reset by the TCRS bit.

TCRS bit (Timer Count Reset)

The TCRS bit resets the value in the TCNT counter. When writing to this bit, stop the counter by the TCST bit before writing. When the TCNT counter is reset, set this bit to 1 during one cycle or more of the SOSC clock.

TCST bit (Timer Count Start)

The TCST bit sets count operation start in the TCNT counter. Before starting the count operation, provide the clock to the counter by the TCCE bit.

27.3 Operation

27.3.1 Pulse Output

Figure 27.2 shows an example of outputting pulses from the TMWO pin synchronizing an overflow interrupt.

1. Set the TCR.OS[1:0] bits to 11b (toggled output on overflow).
2. Every time an overflow occurs in the TCNT counter, toggled pulse is output from the TMWO pin.

The TMWO pin output is low until the first overflow occurs after a reset.

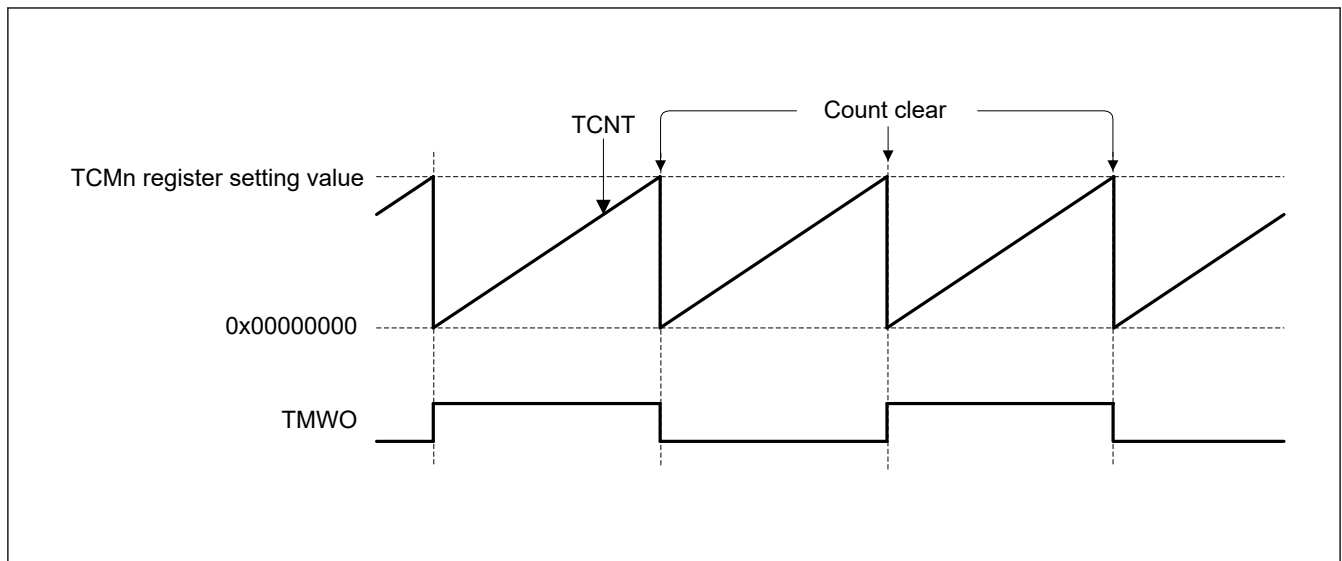


Figure 27.2 Example of TMWO Pulse Output (1)

Figure 27.3 shows an example of outputting pulses when output selection is switched in the middle of count.

1. Set the TCR.OS[1:0] bits to 10b (high-level output on overflow).
2. After timer count starts (TCR.TCST = 1), high-level output on the first overflow.
3. In the middle of TCNT, timer count stops (TCR.TCST = 0).
4. After the TCR.OS[1:0] bits are set to 01b (high-level output on overflow), timer count resets (TCR.TCRS = 1).
5. After timer count starts (TCR.TCST = 1), low-level output on the first overflow.

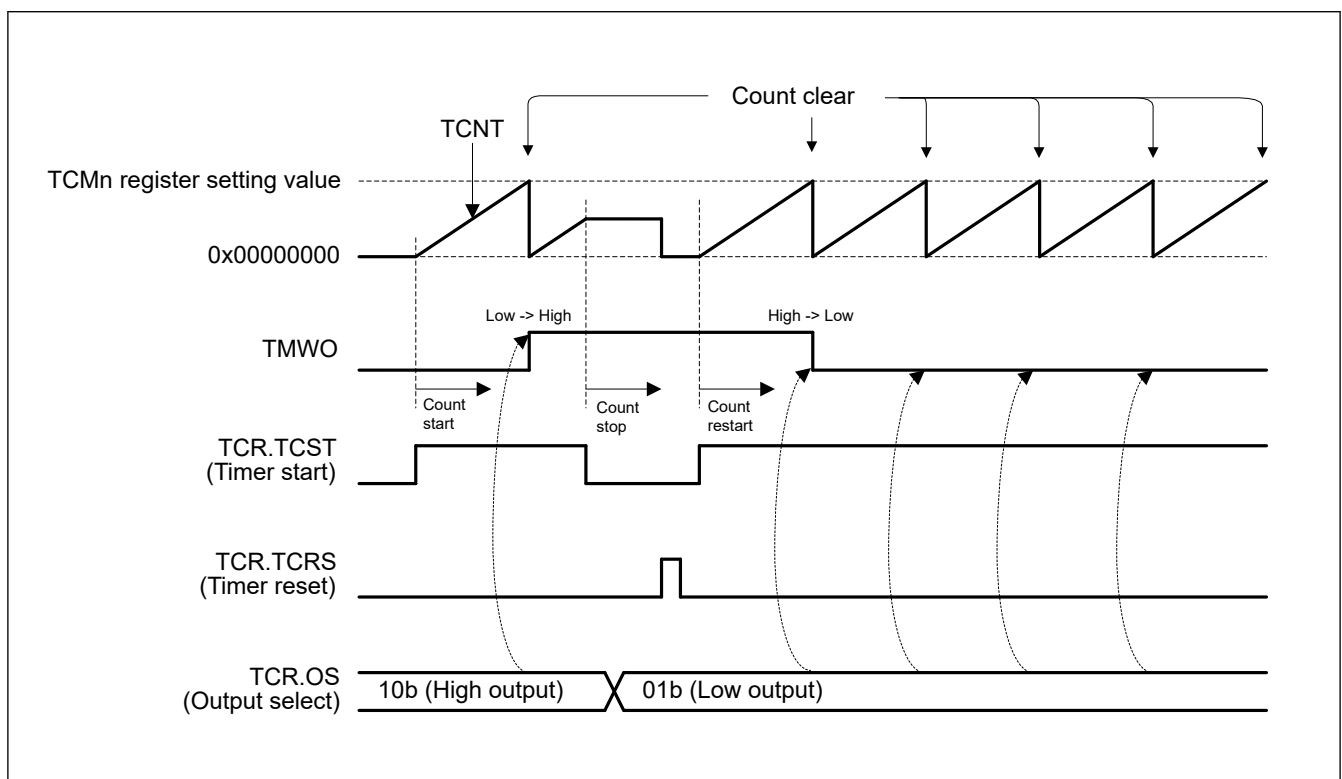


Figure 27.3 Example of TMWO Pulse Output (2)

27.3.2 Operation Timing

27.3.2.1 TCNT Count Timing

Figure 27.4 shows the count timing of the TCNT counter when the SOSC clock is operating.

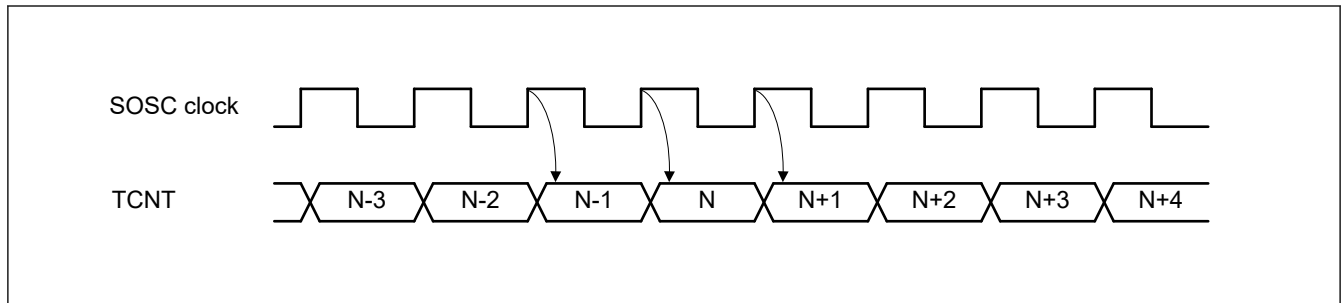


Figure 27.4 TCNT Count Timing

27.3.2.2 TCNT Count Start/Reset Timing

When 2 cycles elapse after the TCR.TCST bit is enabled at high level, TCNT starts count. Figure 27.5 shows TCR.TCST count start timing.

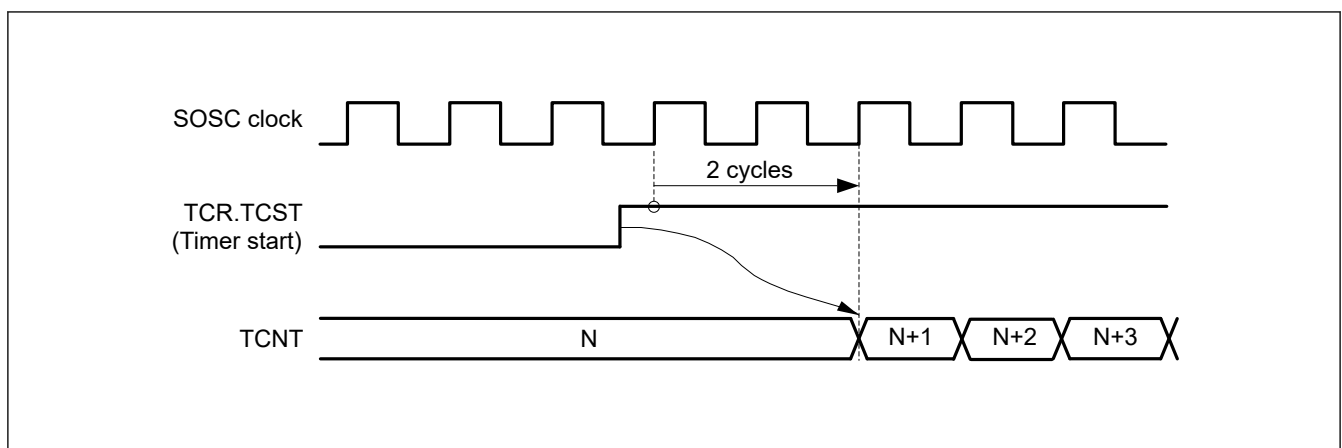


Figure 27.5 TCR.TCST Count Start Timing

When 2 cycles elapse after the TCR.TCST bit is disabled at low level, TCNT stops count.

Figure 27.6 shows TCR.TCST count stop timing.

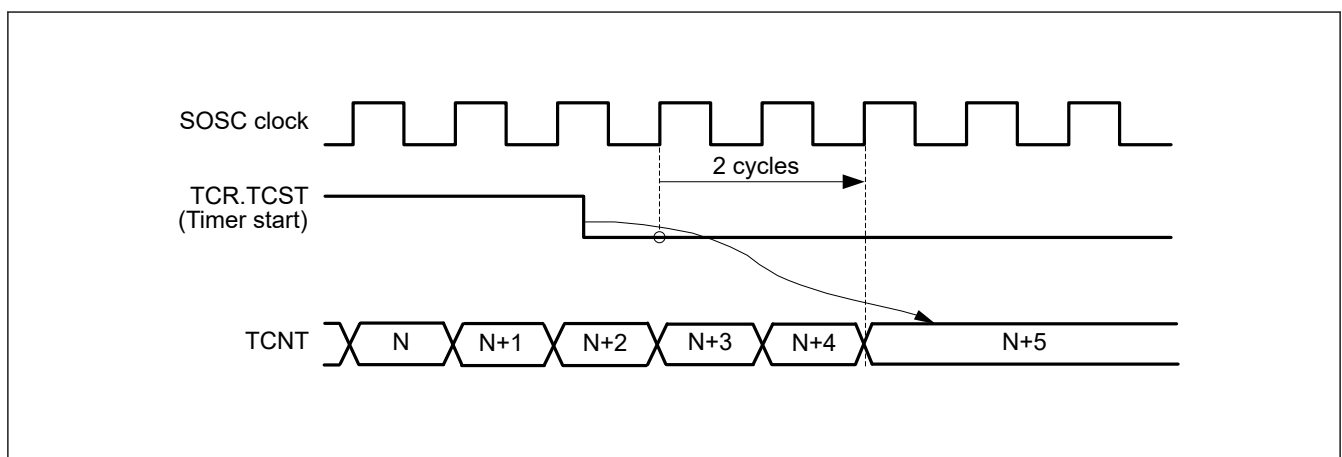


Figure 27.6 TCR.TCST Count Stop Timing

At the next clock after the TCR.TCRS bit is enabled at high level, TCNT resets count.

Figure 27.7 shows TCR.TCRS count reset timing.

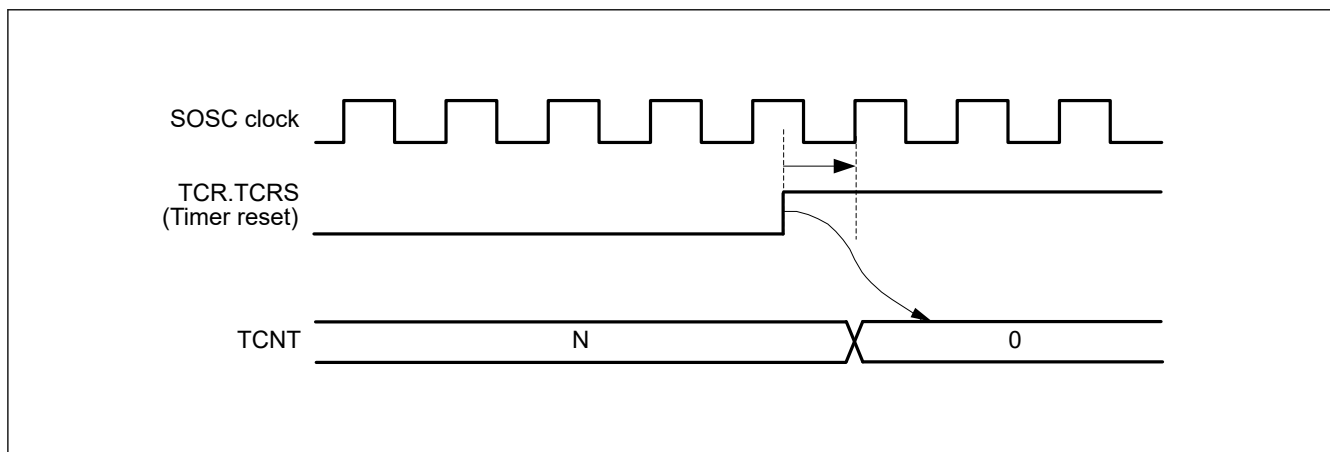


Figure 27.7 TCR.TCRS Count Reset Timing

27.4 Interrupt Sources

27.4.1 Interrupt Sources

The WUPT has one interrupt source, TCNT overflow (WUPT_OVI). Table 27.3 shows the interrupt source.

Table 27.3 WUPT Interrupt Source

Name	Interrupt Source	CPU Activation	Recovery from Software Standby mode	Recovery from Deep Software Standby mode
WUPT_OVI	TCNT overflow	Possible	Possible	Possible

27.5 Link Operation by the ELC

Event link is the system function that operates the multiple modules beforehand set by the CPU via the Event Link Controller (ELC) instead of the CPU.

The WUPT issues the event to the Event Link Controller (ELC).

27.6 Usage Notes

27.6.1 Module-Stop Function Settings

Operation of the WUPT can be disabled or enabled with the module stop control register D (MSTPCRD). For the initial value, the WUPT operation is stopped. Access to the WUPT registers is enabled by releasing the WUPT from the module-stop state. For details, see [section 13, Power-Saving Functions](#).

27.6.2 Note on Cycle Setting

The TCNT counter is cleared when its value matches the value in the TCMn register (the next state at which the count value matched). Consequently, the counter frequency is given by the following formula (f: TCNT counter frequency, CLK: sub-clock frequency, N: TCMn register setting value) :

$$f = \frac{\text{CLK}}{N + 1}$$

28. Realtime Clock (RTC)

28.1 Overview

The realtime clock (RTC) has three counting modes, calendar count mode, binary count mode, and 32-kHz 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 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.

For 32-kHz count mode, the binary counter can be counted with 32.768 kHz.

Table 28.1 lists the RTC specifications, Figure 28.1 shows a block diagram, and Table 28.2 lists the I/O pins.

Table 28.1 RTC specifications

Parameter	Specifications
Count mode	Calendar count mode/binary count mode/32-kHz count mode
Count source*1	Sub-clock (XCIN)
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 32-KHz count mode - Count 32.768 KHz in 32 bits, binary display - Shared by three 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*2 - 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 32-KHz count mode: Upper 30 bits of 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. (32-KHz count mode is only for 64-Hz counter reading)*3 - Return from Software Standby or Deep Software Standby mode mode can be performed by an alarm interrupt or periodic interrupt
Time capture function	Times can be captured when the edge of the time capture event input pin is detected. For every event input, month, date, hour, minute, and second are captured or the 32-bit binary counter value is captured.*2
Event link function	Periodic event output (RTC_PRD)

Note 1. The frequency of the peripheral module clock (PCLKB) $\geq$ the frequency of the count source should be satisfied.

Note 2. Not supported in 32-kHz count mode.

Note 3. There is no carry to the counter / binary counter 0 in 32KHz count mode.

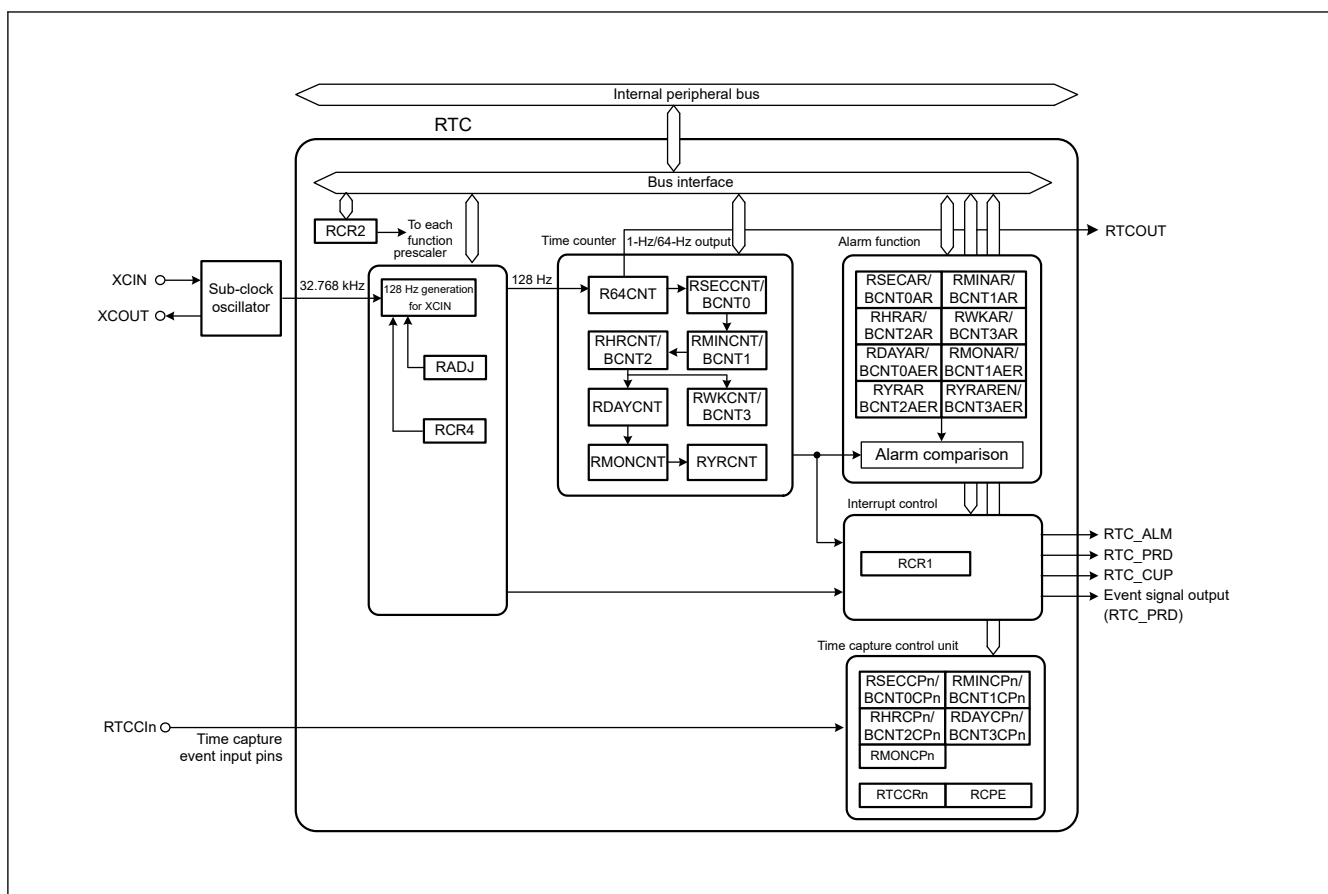


Figure 28.1 RTC block diagram

Table 28.2 RTC I/O pins

Pin name	I/O	Description
XCIN	Input	Connect a 32.768-kHz crystal to these pins
XCOU	Output	
RTCOUT	Output	This pin is used to output a 1-Hz/64-Hz waveform, but not in Deep Software Standby mode
RTCCIn (n = 0 to 2)	Input	Time capture event input pins

28.2 Register Descriptions

Write or read from the RTC registers as described in [section 28.6.5. 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 or Deep Software Standby mode immediately after setting any of these registers. For details, see [section 28.6.4. Transitions to Low Power Modes after Setting Registers](#).

28.2.1 R64CNT : 64-Hz Counter

Base address: RTC = 0x4004_4000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	F1HZ	F2HZ	F4HZ	F8HZ	F16HZ	F32HZ	F64HZ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	F64HZ	64-Hz Flag This bit indicates the 64-Hz state of the sub-second digit.	R
1	F32HZ	32-Hz Flag This bit indicates the 32-Hz state of the sub-second digit.	R
2	F16HZ	16-Hz Flag This bit indicates the 16-Hz state of the sub-second digit.	R
3	F8HZ	8-Hz Flag This bit indicates the 8-Hz state of the sub-second digit.	R
4	F4HZ	4-Hz Flag This bit indicates the 4-Hz state of the sub-second digit.	R
5	F2HZ	2-Hz Flag This bit indicates the 2-Hz state of the sub-second digit.	R
6	F1HZ	1-Hz Flag This bit indicates the 1-Hz state of the sub-second digit.	R
7	—	This bit is read as 0.	R

The R64CNT counter is commonly used in the calendar count mode, binary count mode, and 32-kHz 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 0x00 by an RTC software reset or an execution of a 30-second adjustment. To read this counter, follow the procedure in [section 28.3.5. Reading 64-Hz Counter and Time](#).

28.2.2 RSECCNT : Second Counter (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SEC10[2:0]			SEC1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	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
6:4	SEC10[2:0]	10-Second Count Counts from 0 to 5 for 60-second counting.	R/W
7	—	This bit is read as 0. The write value should be 0.	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, you must stop the count operation using the START bit in RCR2.

To read this counter, follow the procedure in [section 28.3.5. Reading 64-Hz Counter and Time](#).

28.2.3 RMINCNT : Minute Counter (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	MIN10[2:0]			MIN1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	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
6:4	MIN10[2:0]	10-Minute Count Counts from 0 to 5 for 60-minute counting.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

The RMINCNT counter sets and counts the BCD-coded minute value. It counts the carries generated once every minute in the second counter.

A value from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in [section 28.3.5. Reading 64-Hz Counter and Time](#).

28.2.4 RHRCNT : Hour Counter (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	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
5:4	HR10[1:0]	10-Hour Count Counts from 0 to 2 once per carry from the ones place.	R/W
6	PM	AM/PM select for time counter setting. 0: AM 1: PM	R/W
7	—	This bit is read as 0. The write value should be 0.	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 of this range is specified, the RTC does not operate correctly. Before writing to this register, you must 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 28.3.5. Reading 64-Hz Counter and Time](#).

28.2.5 RWKCNT : Day-of-Week Counter (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x08

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

Bit	Symbol	Function	R/W
2:0	DAYW[2:0]	Day-of-Week Counting 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
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The RWKCNT counter sets and counts in the coded day-of-week value. It counts the 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, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in [section 28.3.5. Reading 64-Hz Counter and Time](#).

28.2.6 BCNTn : Binary Counter n (n = 0 to 3) (in Binary Count Mode and 32-kHz Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x02 + 0x02 × n

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

Bit	Symbol	Function	R/W
7:0	BCNT[7:0]	Binary Counter	R/W

BCNTn is a read/write 8-bit register to access BCNT[31:0] that is a 32-bit binary counter. BCNT3 is assigned to the BCNT[31:24] bits, BCNT2 is assigned to the BCNT[23:16] bits, BCNT1 is assigned to the BCNT[15:8] bits, and BCNT0 is assigned to the BCNT[7:0] bits. BCNTn performs count operation by a carry generated for each second of the 64-Hz counter. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in [section 28.3.5. Reading 64-Hz Counter and Time](#).

In the 32-kHz count mode, the binary counter counts in synchronization with the SOSC clock. Before writing to this register, set RCR2.START = 1 to stop counting. In the count operation, the BCNT0 [1:0] bits are fixed to 0, and the value of this register is updated by the rising and falling edges of BCNT0 [2].

28.2.7 RDAYCNT : Day Counter

Base address: RTC = 0x4004_4000

Offset address: 0x0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	DATE10[1:0]	DATE1[3:0]				
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	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
5:4	DATE10[1:0]	10-Day Count Counts from 0 to 3 once per carry from the ones place.	R/W
7:6	—	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 the 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, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in [section 28.3.5. Reading 64-Hz Counter and Time](#).

28.2.8 RMONCNT : Month Counter

Base address: RTC = 0x4004_4000

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MON10	MON1[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	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
4	MON10	10-Month Count Counts from 0 to 1 once per carry from the ones place.	R/W
7:5	—	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 the 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, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in [section 28.3.5. Reading 64-Hz Counter and Time](#).

28.2.9 RYRCNT : Year Counter

Base address: RTC = 0x4004_4000

Offset address: 0x0E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	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
7:4	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
15:8	—	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, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in [section 28.3.5. Reading 64-Hz Counter and Time](#).

28.2.10 RSECAR : Second Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	SEC10[2:0]			SEC1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	SEC1[3:0]	1 Second Value for the ones place of seconds.	R/W
6:4	SEC10[2:0]	10 Seconds Value for the tens place of seconds.	R/W
7	ENB	ENB 0: Do not compare register value with RSECCNT counter value 1: Compare register value with 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.

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 0x00 by an RTC software reset.

28.2.11 RMINAR : Minute Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x12

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	MIN10[2:0]			MIN1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	MIN1[3:0]	1 Minute Value for the ones place of minutes.	R/W
6:4	MIN10[2:0]	10 Minutes Value for the tens place of minutes.	R/W
7	ENB	ENB 0: Do not compare register value with RMINCNT counter value 1: Compare register value with 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.

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 0x00 by an RTC software reset.

28.2.12 RHRAR : Hour Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	HR1[3:0]	1 Hour Value for the ones place of hours.	R/W
5:4	HR10[1:0]	10 Hours Value for the tens place of hours.	R/W
6	PM	AM/PM select for alarm setting. 0: AM 1: PM	R/W
7	ENB	ENB 0: Do not compare register value with RHRCNT counter value 1: Compare register value with 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.

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, you must 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 0x00 by an RTC software reset.

28.2.13 RWKAR : Day-of-Week Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x16

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

Bit	Symbol	Function	R/W
2:0	DAYW[2:0]	Day-of-Week Setting 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
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	ENB	ENB 0: Do not compare register value with RWKCNT counter value 1: Compare register value with 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.

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 0x00 by an RTC software reset.

28.2.14 BCNTnAR : Binary Counter n Alarm Register (n = 0 to 3) (in Binary Count Mode and 32-kHz Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x10 + 0x02 × n

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

Bit	Symbol	Function	R/W
7:0	BCNTAR	Alarm register associated with the 32-bit binary counter	R/W

BCNTnAR is a read/write alarm register associated with the 32-bit binary counter. BCNT3AR is assigned to the BCNTAR[31:24] bits, BCNT2AR is assigned to the BCNTAR[23:16] bits, BCNT1AR is assigned to the BCNTAR[15:8] bits, and BCNT0AR is assigned to the BCNTAR[7:0] (In 32kHz count mode only, assigned to BCNTAR[7:2]). This register is set to 0x00 by an RTC software reset.

28.2.15 RDAYAR : Date Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x18

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	—	DATE10[1:0]	DATE1[3:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	DATE1[3:0]	1 Day Value for the ones place of days.	R/W
5:4	DATE10[1:0]	10 Days Value for the tens place of days.	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	ENB	ENB 0: Do not compare register value with RDAYCNT counter value 1: Compare register value with 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.

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 0x00 by an RTC software reset.

28.2.16 RMONAR : Month Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x1A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	—	—	MON10	MON1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	MON1[3:0]	1 Month Value for the ones place of months.	R/W
4	MON10	10 Months Value for the tens place of months.	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	ENB	ENB 0: Do not compare register value with RMONCNT counter value 1: Compare register value with 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.

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 0x00 by an RTC software reset.

28.2.17 RYRAR : Year Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x1C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	YR1[3:0]	1 Year Value for the ones place of years.	R/W
7:4	YR10[3:0]	10 Years Value for the tens place of years.	R/W
15:8	—	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 0x0000 by an RTC software reset.

28.2.18 RYRAREN : Year Alarm Enable Register (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x1E

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	ENB	ENB 0: Do not compare register value with the RYRCNT counter value 1: Compare register value with the RYRCNT counter value	R/W

When the ENB bit in the RYRAREN register is set to 1, the RYRAR value is compared with the RYRCNT value.

This register is set to 0x00 by an RTC software reset.

28.2.19 BCNTnAER : Binary Counter n Alarm Enable Register (n = 0, 1) (in Binary Count Mode and 32-kHz Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x18 + 0x02 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB[7:0] ([1:0] = Reserved)							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
7:2	ENB[7:2]	Setting the alarm enable associated with the 32-bit binary counter	R/W

BCNTnAER is a read/write register for setting the alarm enable (BCNTAER) associated with the 32-bit binary counter. BCNT3AER is assigned to the BCNTAER.ENB[31:24] bits, BCNT2AER register is assigned to the BCNTAER.ENB[23:16] bits, BCNT1AER is assigned to the BCNTAER.ENB[15:8] bits, and BCNT0AER is assigned to the BCNTAER.ENB[7:0] bits (In 32 kHz count mode only, assigned to BCNTAER.ENB[7:2]). This register is set to 0x00 by an RTC software reset.

28.2.20 BCNT2AER : Binary Counter 2 Alarm Enable Register (in Binary Count Mode and 32-kHz Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x1C

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

Bit	Symbol	Function	R/W
1:0	END[1:0]	These bits are read as 0. The write value should be 0.	R/W
7:2	ENB[7:2]	Setting the alarm enable associated with the 32-bit binary counter	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

BCNT2AER is a read/write register for setting the alarm enable (BCNTAER) associated with the 32-bit binary counter. BCNT3AER is assigned to the BCNTAER.ENB[31:24] bits, BCNT2AER register is assigned to the BCNTAER.ENB[23:16] bits, BCNT1AER is assigned to the BCNTAER.ENB[15:8] bits, and BCNT0AER is assigned to the BCNTAER.ENB[7:0] bits (In 32 kHz count mode only, assigned to BCNTAER.ENB[7:2]). This register is set to 0x00 by an RTC software reset.

28.2.21 BCNT3AER : Binary Counter 3 Alarm Enable Register (in Binary Count Mode and 32-kHz Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x1E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB[7:0] ([1:0] = Reserved)							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
7:2	ENB[7:2]	Setting the alarm enable associated with the 32-bit binary counter	R/W

BCNT3AER is a read/write register for setting the alarm enable (BCNTAER) associated with the 32-bit binary counter. BCNT3AER is assigned to the BCNTAER.ENB[31:24] bits, BCNT2AER register is assigned to the BCNTAER.ENB[23:16] bits, BCNT1AER is assigned to the BCNTAER.ENB[15:8] bits, and BCNT0AER is assigned to the BCNTAER.ENB[7:0] bits (In 32 kHz count mode only, assigned to BCNTAER.ENB[7:2]). This register is set to 0x00 by an RTC software reset.

28.2.22 RCR1 : RTC Control Register 1

Base address: RTC = 0x4004_4000

Offset address: 0x22

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PES[3:0]				RTCO S	PIE	CIE	AIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AIE	Alarm Interrupt Enable 0: Disable alarm interrupt requests 1: Enable alarm interrupt requests	R/W
1	CIE	Carry Interrupt Enable 0: Disable carry interrupt requests 1: Enable carry interrupt requests	R/W
2	PIE	Periodic Interrupt Enable 0: Disable periodic interrupt requests 1: Enable periodic interrupt requests	R/W
3	RTCOS	RTCOUT Output Select 0: Outputs 1 Hz on RTCOUT 1: Outputs 64 Hz RTCOUT	R/W
7:4	PES[3:0]	Periodic Interrupt Select 0x6: Generate periodic interrupt every 1/256 second 0x7: Generate periodic interrupt every 1/128 second 0x8: Generate periodic interrupt every 1/64 second 0x9: Generate periodic interrupt every 1/32 second 0xA: Generate periodic interrupt every 1/16 second 0xB: Generate periodic interrupt every 1/8 second 0xC: Generate periodic interrupt every 1/4 second 0xD: Generate periodic interrupt every 1/2 second 0xE: Generate periodic interrupt every 1 second 0xF: Generate periodic interrupt every 2 seconds Others: Do not generate periodic interrupts	R/W

The RCR1 register is commonly used in the calendar count mode, binary count mode, and 32-kHz 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.

The AIE, PIE, and PES[3:0] can be cleared to 0 by a power-on reset, but cannot be cleared by any other reset.

AIE bit (Alarm Interrupt Enable)

The AIE bit enables or disables alarm interrupt requests.

If the times indicated in the counters and alarm settings match in Deep Software Standby mode, the MCU returns from the regardless of the AIE bit value.

CIE bit (Carry Interrupt Enable)

In the calendar mode and binary count mode, 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.

In the 32-kHz count mode, the CIE bit enables or disables interrupt requests 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 disabled a periodic interrupt.

If the periods indicated in the counters and PES[3:0] settings match in Deep Software Standby mode, the MCU returns from the regardless of the PIE bit value.

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 (RCR2.START = 0) and the RTCOUT output is disabled (RCR2.RTCOE = 0). When RTCOUT is output to an external pin, the RCR2.RTCOE bit must be enabled.

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 in these bits.

28.2.23 RCR2 : RTC Control Register 2 (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x24

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CNTM D	HR24	AADJ P	AADJ E	RTCO E	ADJ30	RESE T	START
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	START	Start 0: Stop prescaler and time counter 1: Operate prescaler and time counter normally	R/W
1	RESET	RTC Software Reset 0: In writing: Invalid (writing 0 has no effect). In reading: Normal time operation in progress, or an RTC software reset has completed. 1: In writing: Initialize the prescaler and target registers for RTC software reset* ¹ . In reading: RTC software reset in progress.	R/W
2	ADJ30	30-Second Adjustment 0: In writing: Invalid (writing 0 has no effect). In reading: Normal time operation in progress, or 30-second adjustment has completed. 1: In writing: Execute 30-second adjustment. In reading: 30-second adjustment in progress.	R/W
3	RTCOE	RTCOUT Output Enable 0: Disable RTCOUT output 1: Enable RTCOUT output	R/W
4	AADJE	Automatic Adjustment Enable* ² 0: Disable automatic adjustment 1: Enable automatic adjustment	R/W
5	AADJP	Automatic Adjustment Period Select* ² 0: The RCR2.ADJ[5:0] setting from the count value of the prescaler every minute. 1: The RCR2.ADJ[5:0] setting value is adjusted from the count value of the prescaler every 10 seconds.	R/W
6	HR24	Hours Mode* ² 0: Operate RTC in 12-hour mode 1: Operate RTC in 24-hour mode	R/W
7	CNTMD	Count Mode Select* ³ 0: Calendar count mode 1: Binary count mode or 32-kHz count mode	R/W

Note 1. R64CNT, RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, RYRAR, RYRAREN, RADJ, RTCCRN, RSECCPN, RMINCPN, RHRCNP, RDAYCPN, RMONCPN, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP.

Note 2. When rewriting this bit, confirm that the value has been rewritten before performing the following processing. See [section 28.6.5. Notes on Writing to and Reading from Registers](#) for notes on register writing/reading.

Note 3. When rewriting this bit, confirm that the value has been rewritten before performing the following processing.

The RCR2 register is related to hours mode, automatic adjustment function, enabling RTCOUT output, 30-second adjustment, RTC software reset, and controlling count operation.

The START, HR24, and CNTMD can be cleared to 0 by a power-on reset, but cannot be cleared by any other reset.

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 this bit, initialization starts in synchronization with the count source. When the initialization is complete, the RESET bit is automatically set to 0. Check that this bit is 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 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)

The AADJE bit controls (enables or disables) automatic adjustment.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit.

The AADJE bit is set to 0 by an RTC software reset.

AADJP bit (Automatic Adjustment Period Select)

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 in synchronization with the count source. However, the count mode switches only after the RTC software reset. (Bit switches before RTC reset, mode switches after RTC reset.)

For details on initial settings, see [section 28.3.1. Outline of Initial Settings of Registers after Power On](#).

28.2.24 RCR2 : RTC Control Register 2 (in Binary Count Mode and 32-kHz Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x24

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CNTMD	—	AADJP	AADJE	RTCOE	—	RESET	START
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	START	Start 0: Stop the 32-bit binary counter, 64-Hz counter, and prescaler 1: Operate the 32-bit binary counter, 64-Hz counter, and prescaler normally	R/W
1	RESET	RTC Software Reset 0: In writing: Invalid (writing 0 has no effect). In reading: Normal time operation in progress, or an RTC software reset has completed. 1: In writing: Initialize the prescaler and target registers for RTC software reset*1. In reading: RTC software reset in progress.	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	RTCOE	RTCOUNT Output Enable 0: Disable RTCOUNT output 1: Enable RTCOUNT output	R/W
4	AADJE	Automatic Adjustment Enable*2 0: Disable automatic adjustment 1: Enable automatic adjustment	R/W
5	AADJP	Automatic Adjustment Period Select*2 0: Add or subtract RADJ.ADJ [5:0] bits from prescaler count value every 32 seconds 1: Add or subtract RADJ.ADJ [5:0] bits from prescaler countvalue every 8 seconds.	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	CNTMD	Count Mode Select*3 0: Calendar count mode 1: Binary count mode or 32-kHz count mode	R/W

Note 1. R64CNT, BCNTnAR, BCNTnAER, RADJ, RTCCRN, BCNTnCPm, RCR2.AADJE, RCR2.AADJP.

Note 2. When rewriting this bit, confirm that the value has been rewritten before performing the following processing. See [section 28.6.5. Notes on Writing to and Reading from Registers](#) for notes on register writing/reading.

Note 3. When rewriting this bit, confirm that the value has been rewritten before performing the following processing.

RCR2 in the binary count mode is a register related to the automatic correction function, RTCOUNT output enable, RTC software reset, and count mode control.

The START and CNTMD can be cleared to 0 by a power-on reset, but cannot be cleared by any other reset.

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 this bit, initialization starts in synchronization with the count source. When the initialization is complete, the RESET bit is automatically set to 0. When 1 is written to the RESET bit, check that the bit is 0 before proceeding.

RTCOE bit (RTCOUNT Output Enable)

The RTCOE bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUNT pin.

Use the START bit to stop counting before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time. When an RTCOUT signal is to be output from an external pin, enable the port control in addition to setting this bit.

AADJE bit (Automatic Adjustment Enable)

The AADJE bit controls (enables or disables) automatic adjustment.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit. The AADJE bit is 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 the calendar count mode, binary count mode or 32-kHz count mode.

When setting the count mode, execute an RTC software reset and start again from the initial settings. This bit is updated in synchronization with the count source. However, the count mode switches only after the RTC software reset. (Bit switches before RTC reset, mode switches after RTC reset.)

For details on initial settings, see [section 28.3.1. Outline of Initial Settings of Registers after Power On](#).

28.2.25 RCR4 : RTC Control Register 4

Base address: RTC = 0x4004_4000

Offset address: 0x28

Bit position:	7	6	5	4	3	2	1	0
Bit field:	R32K MD	—	—	—	—	—	RSKS TP	—

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	RSKSTP	SOSC clock supply setting 0: Enable SOSC clock supply to RTC 1: Enable SOSC clock supply to RTC	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	R32KMD	32-kHz Count Mode Enable 0: Disable 32-kHz count mode 1: Enable 32-kHz count mode	R/W

The RCR4 register is commonly used in the calendar count mode, binary count mode, and 32-kHz count mode. The RSKSTP and R32KMD can be cleared to 0 by a power-on reset, but cannot be cleared by any other reset.

RSKSTP bit (SOSC clock supply setting)

Setting the RSKSTP bit is set to 1 supplies the OSOCS clock to the RTC.

R32KMD bit (32-kHz Count Mode Enable)

Setting the R32KMD bit to 1, the 32-bit binary counter to count in synchronization with the SOCS clock. Writing to this bit should be set only once before initializing the RTC register after power-on.

28.2.26 RADJ : Time Error Adjustment Register

Base address: RTC = 0x4004_4000

Offset address: 0x2E

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

Bit	Symbol	Function	R/W
5:0	ADJ[5:0]	Adjustment Value These bits specify the adjustment value from the prescaler.	R/W
7:6	PMADJ[1:0]	Plus-Minus 0 0: Do not perform adjustment. 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

The RADJ register is used in both calendar count mode and binary count mode. Adjustment is performed by the addition to or subtraction from the prescaler or 64-Hz counter. If the Automatic Adjustment Enable (RCR2.AADJE) bit is 0, adjustment is performed when writing to the RADJ. If the RCR2.AADJE bit is 1, adjustment is performed in the interval specified in the Automatic Adjustment Period Select (RCR2.AADJP) bit.

The current adjustment by software (disabling automatic adjustment) may 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 more processing. This register is set to 0x00 by an RTC software reset.

ADJ[5:0] bits (Adjustment Value)

The ADJ[5:0] bits specify the adjustment value (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.

28.2.27 RCPE : RTC Time Capture Enable Register

Base address: RTC = 0x4004_4000

Offset address: 0x26

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

Bit	Symbol	Function	R/W
0	RTCEN	Time Capture Event Input Enable (n = 0 to 2) 0: Disable the RTCICn pin 1: Enable the RTCICn pin	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

The RCPE register is used for enabling time capture events. This register is used in the calendar count mode and binary count mode. This register cannot be used in 32-KHz count mode.

When this register is modified, check that all the bits are updated before proceeding to the next processing.

RTCEN is initialized only by a power-on reset and is not initialized by other resets.

RTCEN bit (Time Capture Event Input Enable)

The RTCEN bit enables or disables input of RTC time capture events. For more details, see [section 28.2.28. RTCCRN : Time Capture Control Register n \(n = 0 to 2\)](#).

This bit can be cleared to 0 by a power-on reset, but cannot be cleared by any other reset. Some settings may lead to the acceptance of input through an RTCICn pin that is not as intended. To avoid this, set this bit to 0 when the RTC is not to be used or the RTCICn pins are not to be used.

28.2.28 RTCCRN : Time Capture Control Register n (n = 0 to 2)

Base address: RTC = 0x4004_4000

Offset address: 0x40 + 0x02 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TCEN	TCPSEL	TCNF[1:0]	—	TCST	TCCT[1:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TCCT[1:0]	Time Capture Control 0 0: Do not detect events 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges	R/W
2	TCST	Time Capture Status 0: No event detected 1: Event detected ^{*1}	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TCNF[1:0]	Time Capture Noise Filter Control 0 0: Turn noise filter off 0 1: Setting prohibited 1 0: Turn noise filter on (count source) 1 1: Turn noise filter on (count source by divided by 32)	R/W
6	TCPSEL	RTCICn input pin select 0: RTCICn_A 1: RTCICn_B	R/W
7	TCEN	Time Capture Event Input Pin Enable 0: Disable the RTCICn pin as the time capture event input pin 1: Enable the RTCICn pin as the time capture event input pin	R/W

Note 1. Indicates that an event is detected. Writing 1 to this bit has no effect. Writing 0 sets this bit to 0.

The RTCCRN register is used both in calendar count mode and in binary count mode. RTCCR0, RTCCR1, and RTCCR2 control the RTCIC0, RTCIC1, and RTCIC2 pins, respectively.

This register cannot be used in 32-KHz count mode.

RTCCRN is updated in synchronization with the count source. When RTCCRN is modified, check that all the bits except the TCST bit are updated before continuing with additional processing. This register is cleared to 0x00 by an RTC software reset.

TCCT[1:0] bits (Time Capture Control)

The TCCT[1:0] bits control the edge detection of the time capture event input pins, RTCIC0, RTCIC1, and RTCIC2. The detection edge is selectable. The TCCT[1:0] bits must be set while the VBTICTLR.VCHnIEN bit is 1.

TCST bit (Time Capture Status)

The TCST bit indicates that an event on the time capture event input pins, RTCIC0, RTCIC1, and RTCIC2, was detected. When the TCST bit is 0, no event is detected. When the TCST bit is 1, this bit indicates that an event was detected on the

associated pin and the capture register is valid. When multiple events are detected, the capture time for the first event is retained.

The event is detected only during count operation (RCR2.START bit = 1). Before reading the capture register, make sure that this bit is set to 1.

Set the TCST bit while the TCCT[1:0] bits are 00b (no event is detected). The TCST bit is set to 0 in synchronization with the count source. When the TCST bit is set to 0, check that the bit is updated before continuing with additional processing.

TCNF[1:0] bits (Time Capture Noise Filter Control)

The TCNF[1:0] bits control the noise filter of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2).

When the noise filter is on, the count source divided by 1 or divided by 32 is selectable. In this case, when the input level on the time capture event input pin matches three consecutive times at the set sampling period, the input level is determined.

Set the TCNF[1:0] bits while the TCCT[1:0] bits are 00b (no event is detected). When the noise filter is used, set the TCNF[1:0] bits, wait for 3 cycles of the specified sampling period, then set the TCCT[1:0] bits.

TCPSEL bit (RTCICn input pin select)

The TCPSEL bit is selected RTCICn_A or RTCICn_B as a time capture event input pin. This bit should be set before the RTC operation.

TCEN bit (Time Capture Event Input Pin Enable)

The TCEN bit enables or disables the time capture event input pins RTCIC0, RTCIC1, and RTCIC2.

Table 28.3 lists the conditions for enabling the RTCICn pin. Before setting this bit to 1, be sure to set RTC time capture event enable bit (RCPE.RTCEN), port control setting bits (PmnPFS.PSEL[4:0], and PmnPFS.PMR). For details on the port control setting bits (PmnPFS.PSEL[4:0] and PmnPFS.PMR), see [section 22, I/O Ports](#).

Table 28.3 Conditions for Enabling the RTCICn Pin

PmnPFS.PSEL[4:0] PmnPFS.PMR	RCPE.RTCEN	TCEN	Time Capture Event Input Pin Enabled/Disabled
RTCICn pin not selected	Don't care	Don't care	Disabled
RTCICn pin selected	0 (disabling input)	Don't care	
	1 (enabling input)	0 (disabling input)	
		1 (enabling input)	Enabled

If the TCEN bit is set to 0, also set the TCCT[1:0] bits to 00b.

The TCEN bit is not initialized by a reset. Some settings may lead to the acceptance of input through an RTCICn pin that is not as intended. To avoid this, use the following procedure to disable input when the RTC is not to be used or the RTCICn pins are not to be used.

1. Start the RTC count source.
2. Set the TCEN bit to 0.

28.2.29 RSECCPn : Second Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x52 + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—		SEC10[2:0]			SEC1[3:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	SEC1[3:0]	1-Second Capture Capture value for the ones place of seconds.	R

Bit	Symbol	Function	R/W
6:4	SEC10[2:0]	10-Second Capture Capture value for the tens place of seconds.	R
7	—	This bit is read as 0.	R

RSECCPn is a read-only register that captures the RSECCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RSECCP0, RSECCP1, and RSECCP2 registers, respectively. This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

28.2.30 RMINCPn : Minute Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x54 + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	MIN10[2:0]			MIN1[3:0]			

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
3:0	MIN1[3:0]	1-Minute Capture Capture value for the ones place of minutes.	R
6:4	MIN10[2:0]	10-Minute Capture Capture value for the tens place of minutes.	R
7	—	This bit is read as 0.	R

RMINCPn is a read-only register that captures the RMINCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMINCP0, RMINCP1, and RMINCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

28.2.31 RHRCpN : Hour Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x56 + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PM	HR10[1:0]		HR1[3:0]			

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
3:0	HR1[3:0]	1-Hour Capture Capture value for the ones place of hours	R
5:4	HR10[1:0]	10-Hour Capture Capture value for the tens place of hours	R
6	PM	PM 0: AM 1: PM	R
7	—	This bit is read as 0.	R

RHRCpN is a read-only register that captures the RHRCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RHRCP0, RHRCP1, and RHRCP2 registers, respectively.

The PM bit is only enabled when the RCR2.HR24 bit is 0 (in 12-hour mode).

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, you must stop the time capture event detection using the RTCCRn.TCCT[1:0] bits.

28.2.32 RDAYCPn : Date Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x5A + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	DATE10[1:0]	DATE1[3:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	DATE1[3:0]	1-Day Capture Capture value for the ones place of days.	R
5:4	DATE10[1:0]	10-Day Capture Capture value for the tens place of days.	R
7:6	—	This bit is read as 0.	R

RDAYCPn is a read-only register that captures the RDAYCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RDAYCP0, RDAYCP1, and RDAYCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

28.2.33 RMONCPn : Month Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x5C + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MON10	MON1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	MON1[3:0]	1-Month Capture Capture value for the ones place of months.	R
4	MON10	10-Month Capture Capture value for the tens place of months.	R
7:5	—	These bits are read as 0.	R

RMONCPn is a read-only register that captures the RMONCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMONCP0, RMONCP1, and RMONCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

28.2.34 BCNTnCPm : BCNTn Capture Register m (n= 0 to 3, m = 0 to 2) (in Binary Count Mode)

Base address: RTC = 0x4004_4000

Offset address: 0x52 + 0x10 × m (BCNT0CPm)
 0x54 + 0x10 × m (BCNT1CPm)
 0x56 + 0x10 × m (BCNT2CPm)
 0x5A + 0x10 × m (BCNT3CPm)

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

BCNTnCPm is a read-only register that captures the BCNTn value when a time capture event is detected. BCNT3CPm is assigned to the BCNTCPm[31:24] bits, BCNT2CPm is assigned to the BCNTCPm[23:16] bits, BCNT1CPm is assigned to the BCNTCPm[15:8] bits and BCNT0CPm is assigned to the BCNTCPm[7:0] bits. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNTnCP0, BCNTnCP1, and BCNTnCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, you must stop the time capture event detection using the RTCCRn.TCCT[1:0] bits.

28.3 Operation

28.3.1 Outline of Initial Settings of Registers after Power On

After the power is turned on, perform the initial settings for the clock, count mode, time error adjustment, time, alarm, interrupts, and time capture.

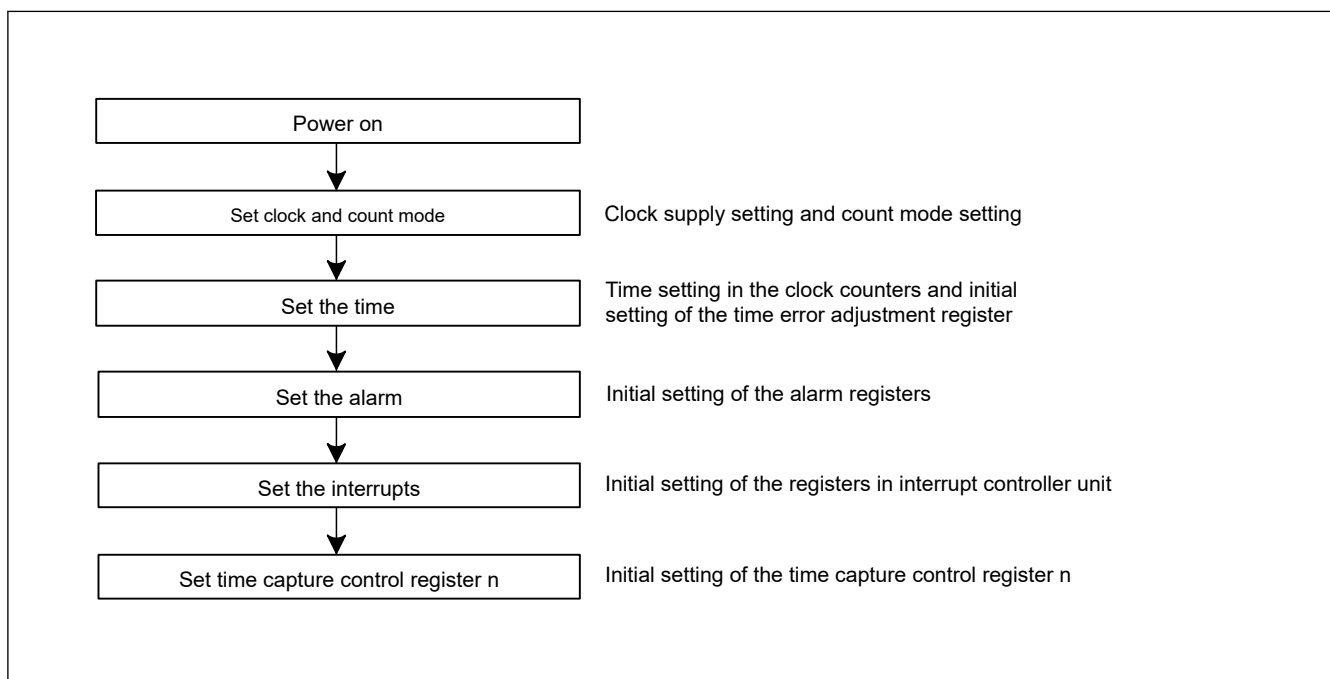
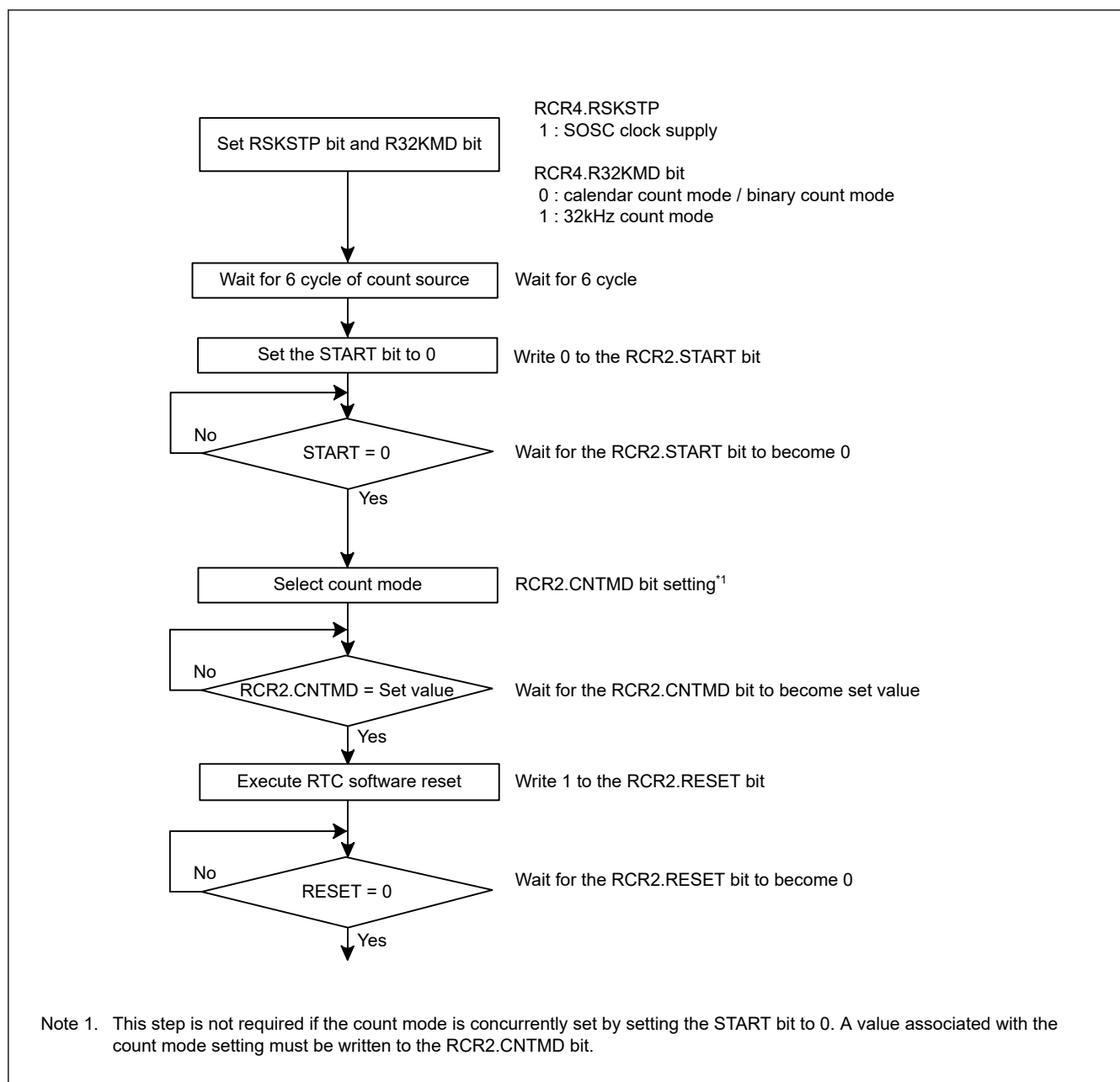


Figure 28.2 Outline of initial settings after a power on

28.3.2 Clock and Count Mode Setting Procedure

Figure 28.3 shows how to set the clock and the count mode.

**Figure 28.3 Clock and count mode setting procedure**

28.3.3 Setting the Time

Figure 28.4 shows how to set the time.

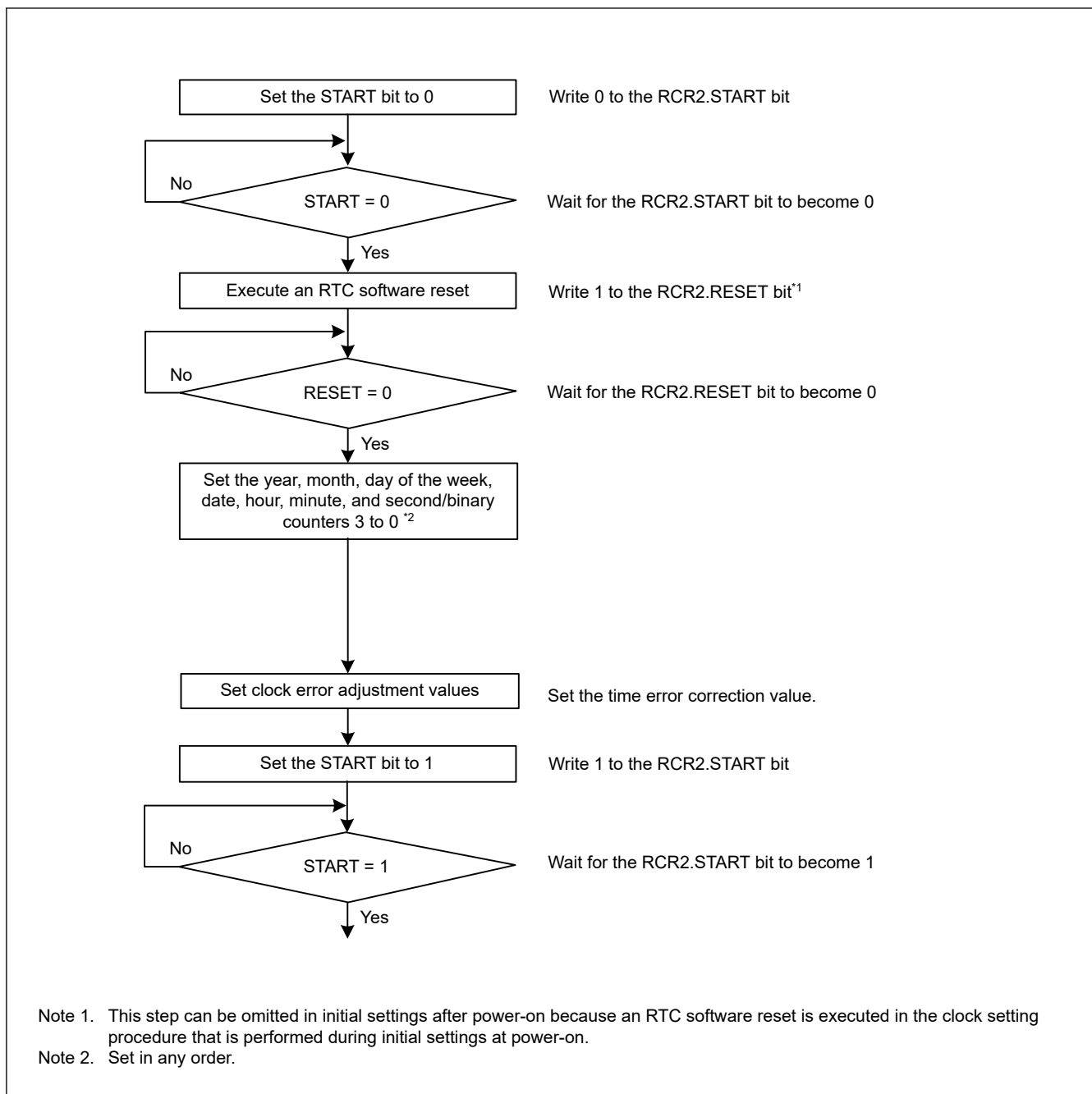


Figure 28.4 Setting the time

28.3.4 30-Second Adjustment

Figure 28.5 shows how to execute a 30-second adjustment.

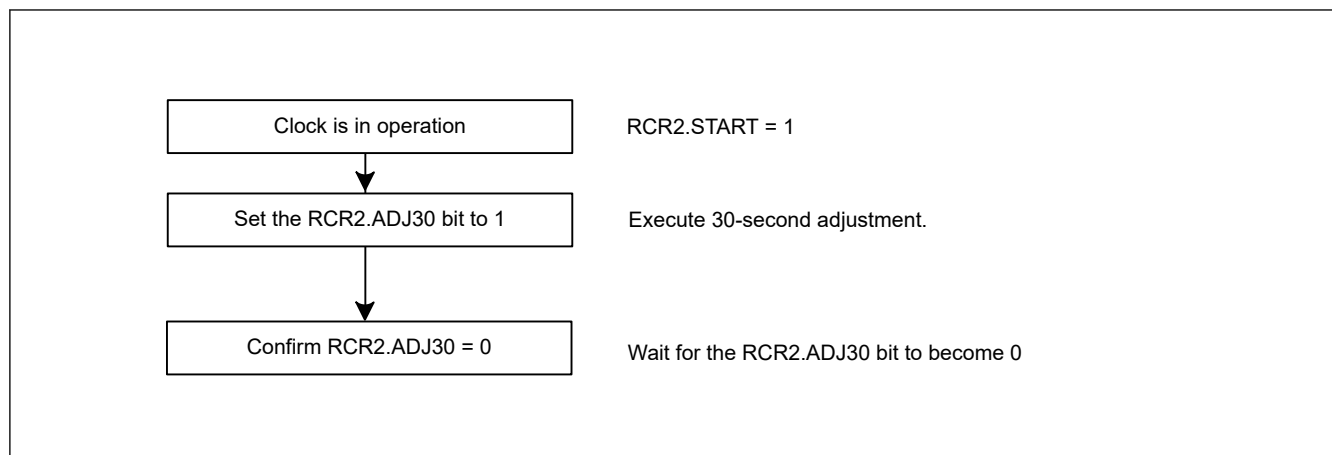
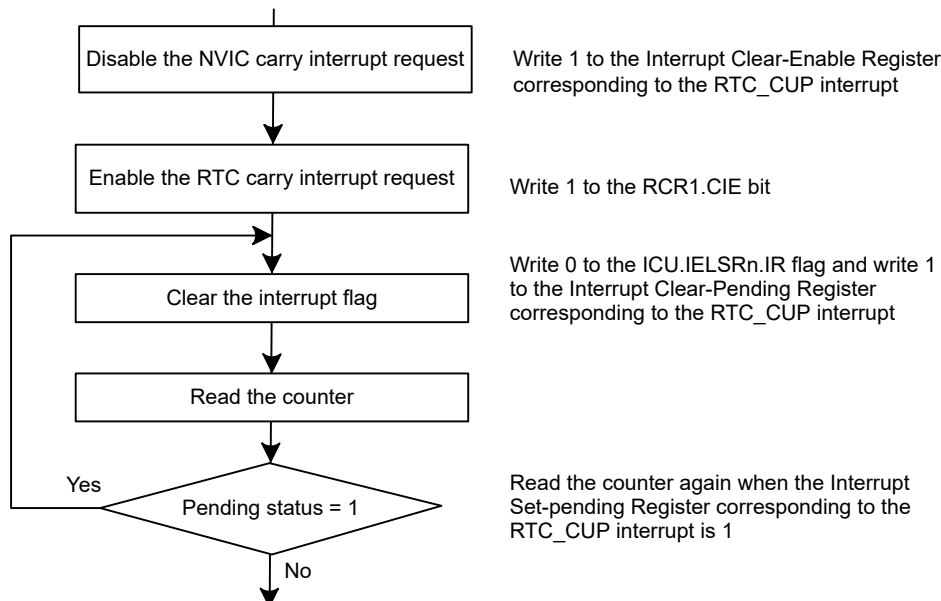


Figure 28.5 30-second adjustment

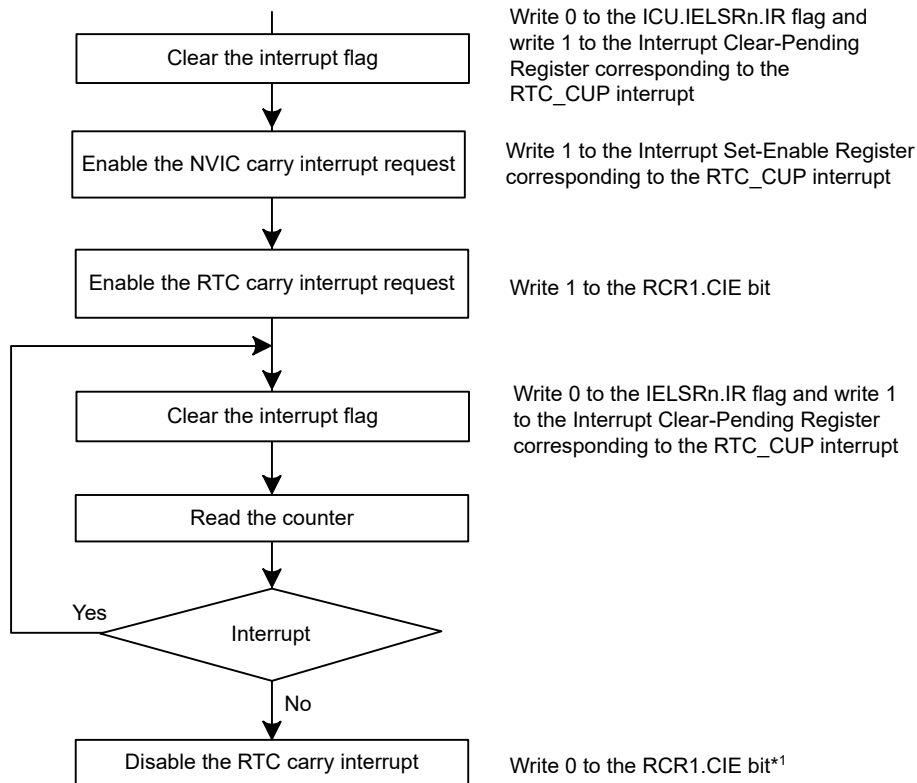
28.3.5 Reading 64-Hz Counter and Time

[Figure 28.6](#) shows how to read a 64-Hz counter and time.

(a) To read the time without using interrupt



(b) To read the time using interrupts



Note 1. Disable interrupts if required.

Figure 28.6 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 28.6, and the procedure using carry interrupts is shown in (b). To keep the program simple, method (a) should be used in most cases.

28.3.6 Alarm Function

Figure 28.7 shows how to use the alarm function.

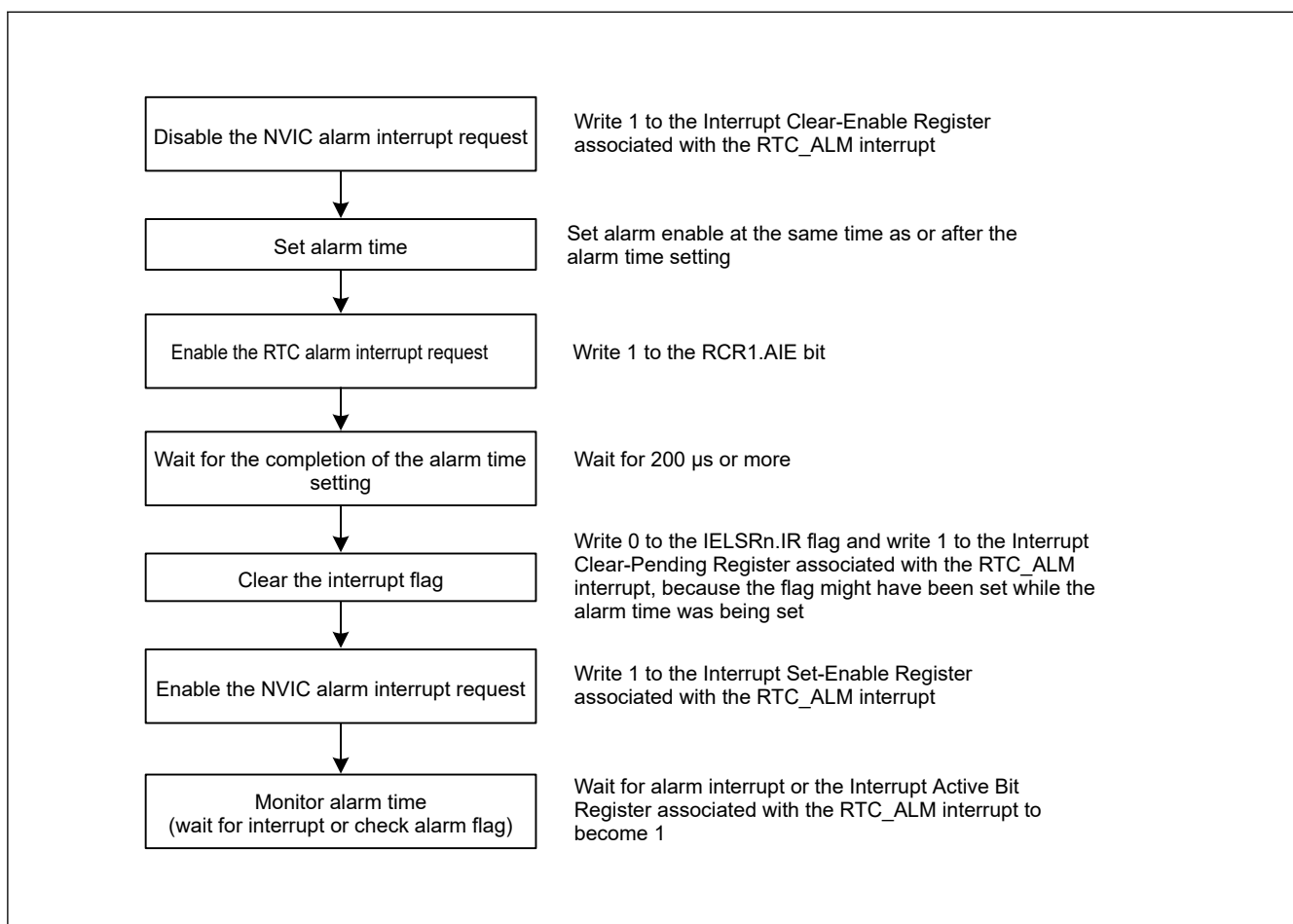


Figure 28.7 Using the alarm function

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 involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

In binary count mode, an alarm can be generated in any bit combination of 32 bits. In 32-kHz count mode, an alarm can be generated using any combination of the upper 30 bits. Write 1 to the ENB bit of the Alarm Enable register associated with the target bit of the alarm, and set the alarm time in the alarm register. For bits that are not the target of the alarm, write 0 to the ENB bit of the Alarm Enable register.^{*1}

For any of the ENB[31:0] bits that are set to 1, the bits in the corresponding positions in the binary counter (BCNT[31:0]) are compared with the values of the corresponding bits in the binary alarm registers^{*1}. When all such bits match, the IR flag associated with the RTC_ALM interrupt is set to 1 and the corresponding bits in the Interrupt Set-Pending/Clear-Pending Registers are set to 1. Alarm detection can be confirmed by reading the Interrupt Set-Pending Register associated with the RTC_ALM interrupt, but an interrupt should be used in most cases. If 1 is set in the Interrupt Set-Enable Register associated with the RTC_ALM interrupt, an alarm interrupt is generated in the event of the alarm, enabling the alarm to be detected.

Writing 0 sets the IELSRn.IR flag associated with the RTC_ALM interrupt to 0. If interrupt is enabled, the Interrupt Set-Pending/Clear-Pending 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. In 32-kHz count mode, the 32-bit binary counter operates by switching the count source from 1 Hz to 32.768 kHz. The alarm register (BCNT3AR, BCNT2AR, BCNT1AR, BCNT0AR) and Alarm Enable register (BCNT3AER, BCNT2AER, BCNT1AER, BCNT0AER) of the binary counter has a setting value of 32.768 kHz accuracy. The binary counter compare match and compare match alarm interrupts operate on the SOSC clock.

Note 1. For any bits in the ENB bits that are set to 1, the values in the corresponding positions in the alarm registers from the following registers are compared with the corresponding bits of the counted values.

Counter registers: RSECCNT, RMINCNT, RHRCNT, RWKCNT, RDAYCNT, RMONCNT, RYRCNT

Alarm registers: RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, RYRAREN

28.3.7 Procedure for Disabling Alarm Interrupt

Figure 28.8 shows the procedure for disabling the enabled alarm interrupt request.

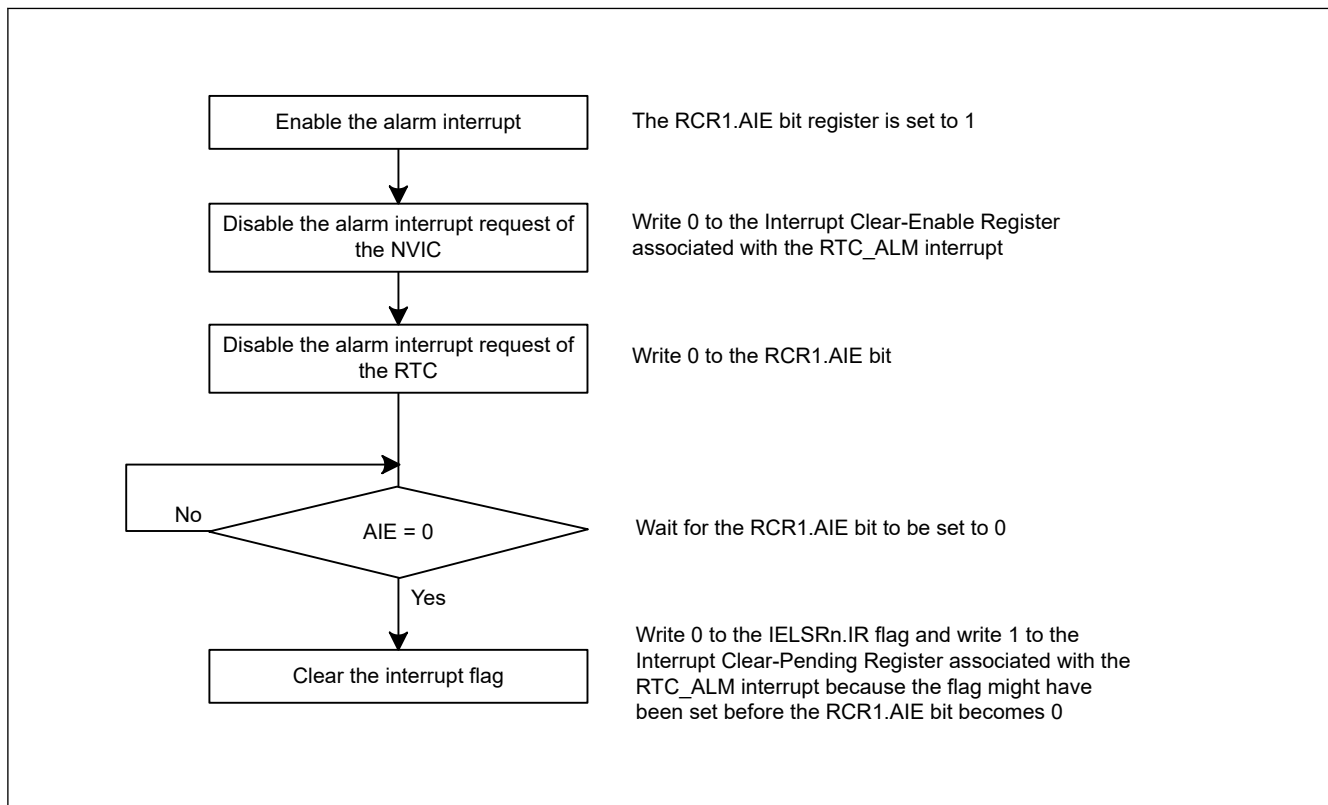


Figure 28.8 Procedure for disabling alarm interrupt request

28.3.8 Time Error Adjustment Function

The time error adjustment function is used to correct errors, running fast or slow, in the time caused 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.

28.3.8.1 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**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 (0x3C)

(2) Example 2: Sub-clock oscillator running at 32.766 kHz**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 (0x14)

(3) Example 3: Sub-clock oscillator running at 32.764 kHz**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 32 clock cycles every 8 seconds.

Register settings when RCR2.CNTMD = 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 (0x20)

28.3.8.2 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 on execution of a write instruction to the RADJ register.

(1) Example 1: Sub-clock oscillator running at 32.769 kHz**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.

Register settings

- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler)
- RADJ.ADJ[5:0] = 1 (0x10)

This is written to the RADJ register once per 1-second interrupt.

28.3.8.3 Procedure to change 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 wanted time. After that, the time is adjusted every time a value is written to the RADJ register.

28.3.8.4 Procedure to stop adjustment

Stop the adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

28.3.9 Capturing the time

The RTC is capable of storing the month, date, hour, minute and second/binary counters 3 to 0 by detecting an edge of a signal on a time capture event input pin in calendar count mode or binary count mode, but it is not available in 32-kHz count mode.

A noise filter can also be used on a time capture event input pin. If the noise filter is enabled, the RTCCRn.TCST bit is set to 1 when the input level on the pin matches three times.

The noise filter can be switched on or off for each of the time capture event input pins. Operation when the noise filter is off is shown in [Figure 28.9](#) and operation when the noise filter is on is shown in [Figure 28.10](#).

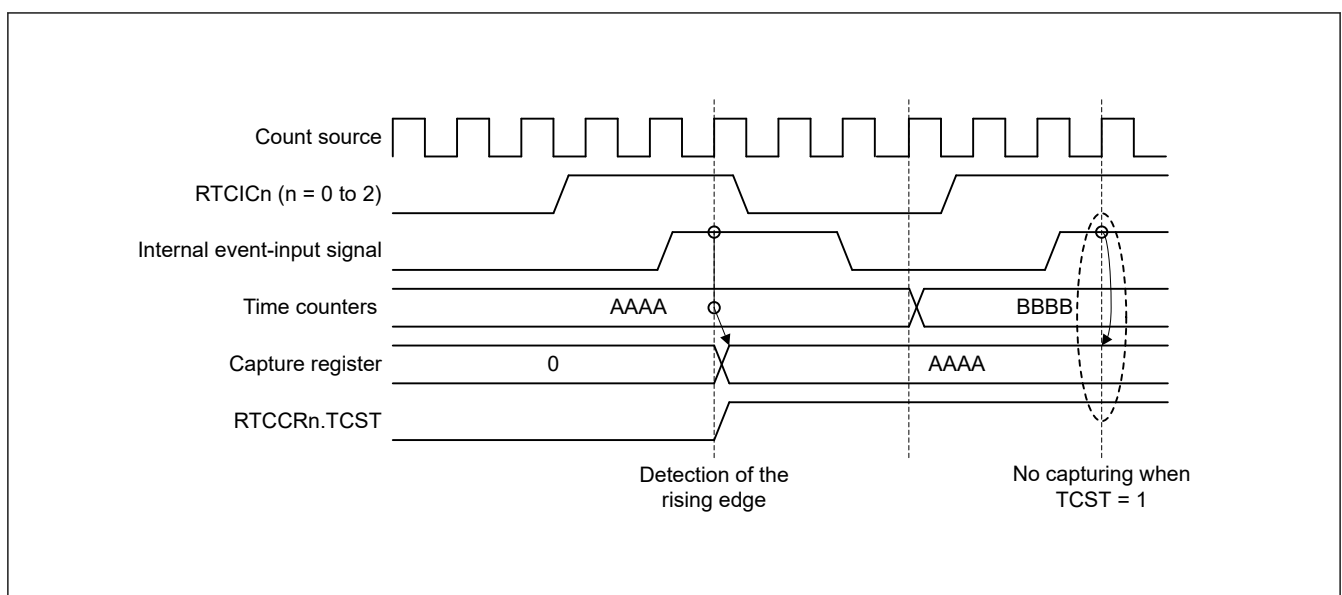


Figure 28.9 Timing of a time capture operation with the noise filter off

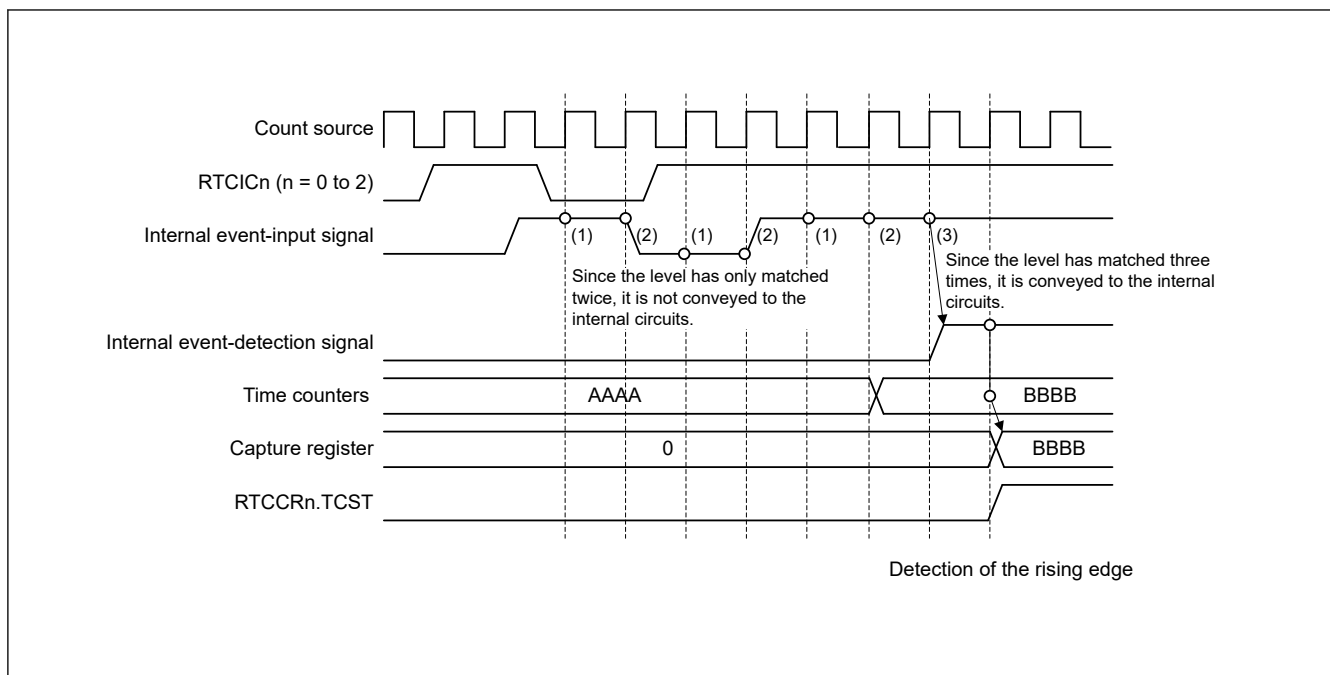


Figure 28.10 Timing of a time capture operation with the noise filter on

28.4 Interrupt Sources

The RTC has three interrupt sources, as listed in [Table 28.4](#).

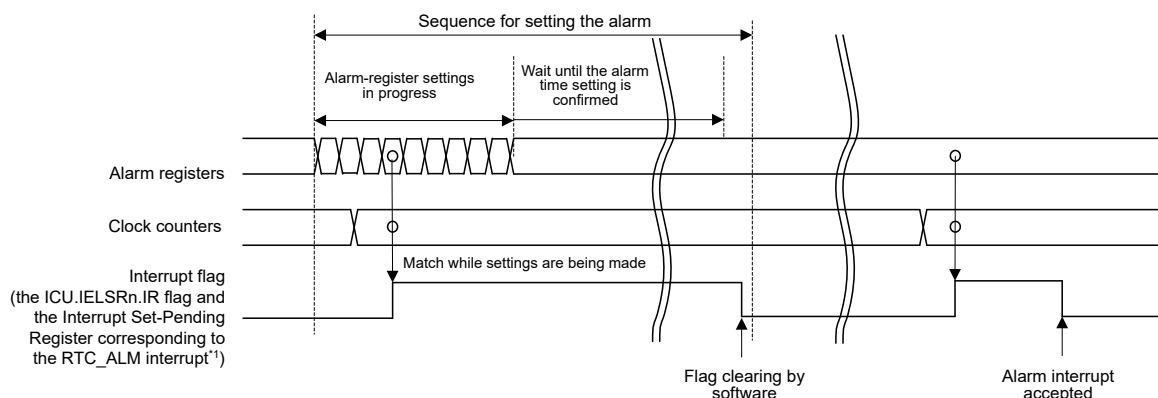
Table 28.4 RTC interrupt sources

Name	Interrupt source
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 28.3.6. 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 flag and the interrupt Set-Pending Register associated with the RTC_ALM interrupt to 0 again after modifying values of the alarm registers. After the interrupt flag for the alarm interrupt is set to 1 and the state is returned to mismatching of the alarm registers and clock counters, the flag is not 1 again until there is another match or the values of the alarm registers are modified again.



Note 1. See [section 16, Interrupt Controller Unit \(ICU\)](#) for details on the associated interrupt vector number.

Figure 28.11 Timing for the alarm interrupt (RTC_ALM)

(2) Periodic interrupt (RTC_PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected 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. In the 32-kHz count mode, select whether to enable or disable the interrupt request when a carry to the reading of the 64-Hz counter occurs.

[Figure 28.12](#) shows the timing of the carry interrupt (RTC_CPU).

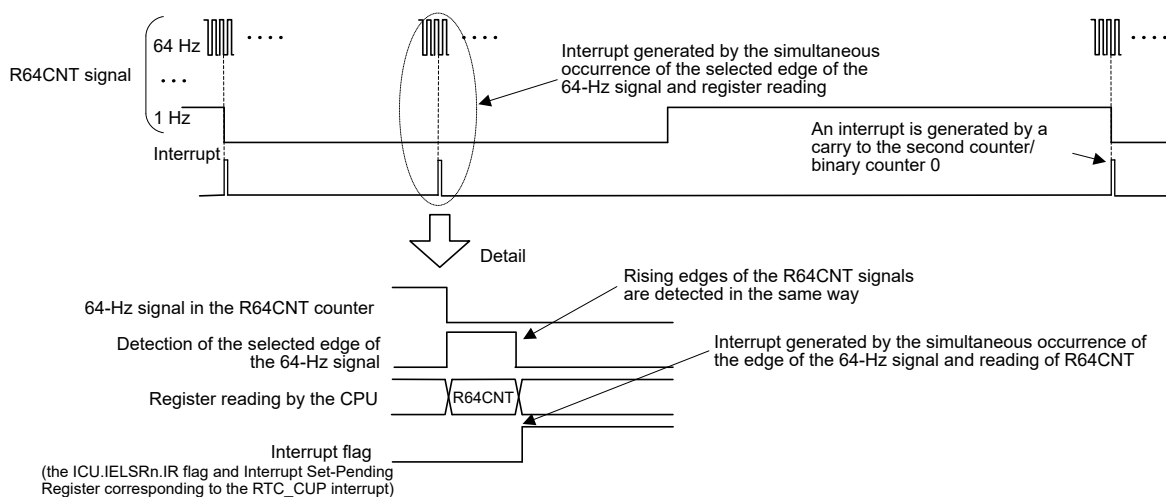


Figure 28.12 Timing for the carry interrupt (RTC_CUP)

28.5 Event Link Output

The RTC generates periodic event output (RTC_PRD) event signal 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.

28.5.1 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 or Deep Software Standby mode, the periodic event signals for the ELC are not output.

28.6 Usage Notes

28.6.1 Register Writing during Counting

The following registers should not be written to during counting, that is, while the RCR2.START bit is 1:

- RSECCNT/BCNT0
- RMINCNT/BCNT1
- RHRCNT/BCNT2
- RDAYCNT
- RWKCNT/BCNT3
- RMONCNT
- RYRCNT
- RCR1.RTCOS
- RCR2.RTCOE
- RCR2.HR24

The counter should be stopped before writing to any of these registers.

28.6.2 Use of Periodic Interrupts

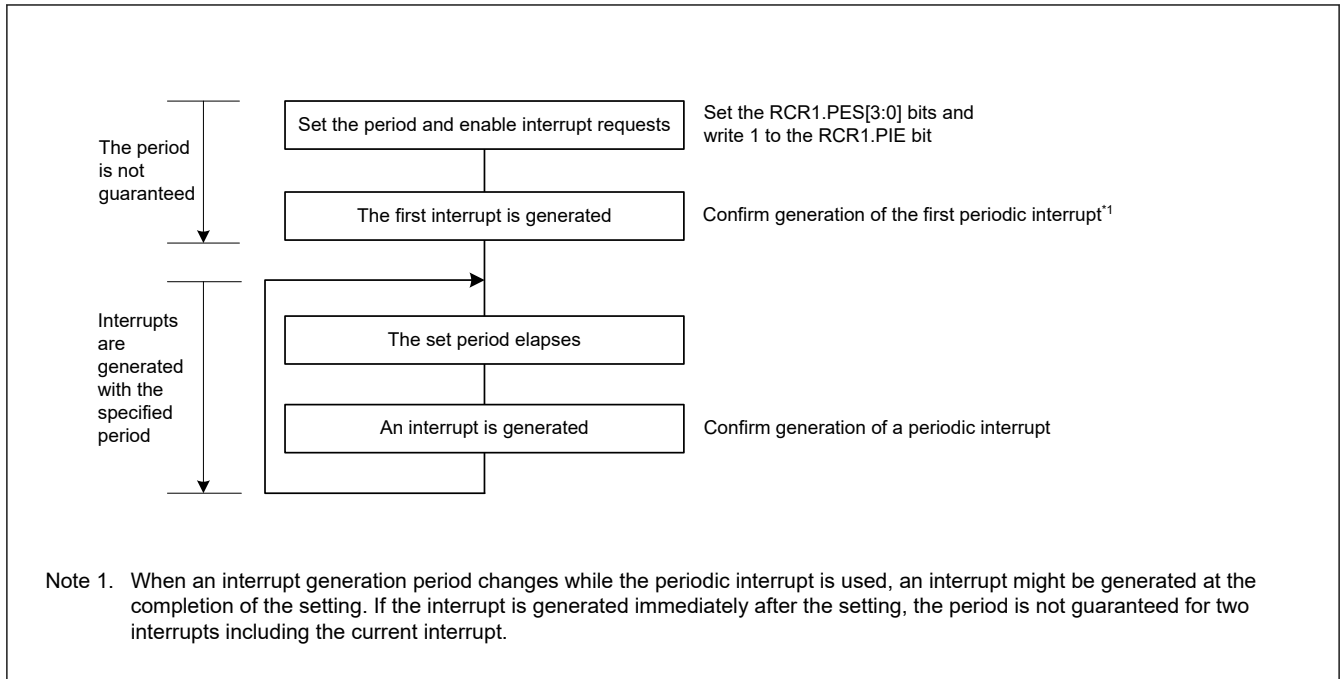
Figure 28.13 shows the procedure for using periodic interrupts.

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 operation can affect the interrupt period:

- Stopping/restarting or resetting counter operation
- Reset by RTC software
- 30-second adjustment by changing the RCR2 value

When the time error adjustment function is used, the interrupt generation period after adjustment is added or subtracted based on the adjustment value.

**Figure 28.13 Using the periodic interrupt function**

28.6.3 RTCOUT (1-Hz/64-Hz) Clock Output

Stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the period of RTCOUT (1-Hz/64-Hz) output. When the time error adjustment function is used, the period of RTCOUT (1-Hz/64-Hz) output after adjustment is added or subtracted based on the adjustment value.

28.6.4 Transitions to Low Power Modes after Setting Registers

A transition to a low power state (Software Standby mode) 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.

28.6.5 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 28.3.5. 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 is reflected when fourth 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, wait for 1/128 second while the clock is operating (RCR2.START bit = 1).
- After a reset is generated, write to the RTC register after 6 cycles of the count source clock have elapsed.

28.6.6 Changing the Count Mode

When changing the count mode (calendar count mode/binary count mode/32-KHz count mode), set the RCR2.START bit to 0, stop the counting operation, then start it again from the initial setting. For details on the initial setting, see [section 28.3.1. Outline of Initial Settings of Registers after Power On](#).

28.6.7 Time Capture Operation While Count is Running or Stopping

To capture the counter value when a time capture event is input, set the time capture control bit (RTCCRn.TCCT) while RTC is running (RCR2.START bit = 1). The time capture function cannot be used during LVD resetting.

28.6.8 Initialization

Registers in RTC are initialized by power-on reset and RTC software reset. However, some bits of the RCR1, RCR2, RCR4, and RCPE registers are initialized only by a power-on reset.

29. Low-Speed Clock Timer (LST)

29.1 Overview

The low-speed clock timer (LST) is a stopwatch count module that converts the 2.048-kHz source clock to 1 kHz and counts from 0.000 seconds to 1.999 seconds.

Table 29.1 lists the LST specifications and Figure 29.1 shows a block diagram.

Table 29.1 LST specifications

Parameter	Specifications
Clock source	2.048 kHz
Count range	0.000 seconds to 1.999 seconds
Resolution	0.001 seconds
Format of count value	BCD format
Function for reducing power consumption	The module-stop state can be set

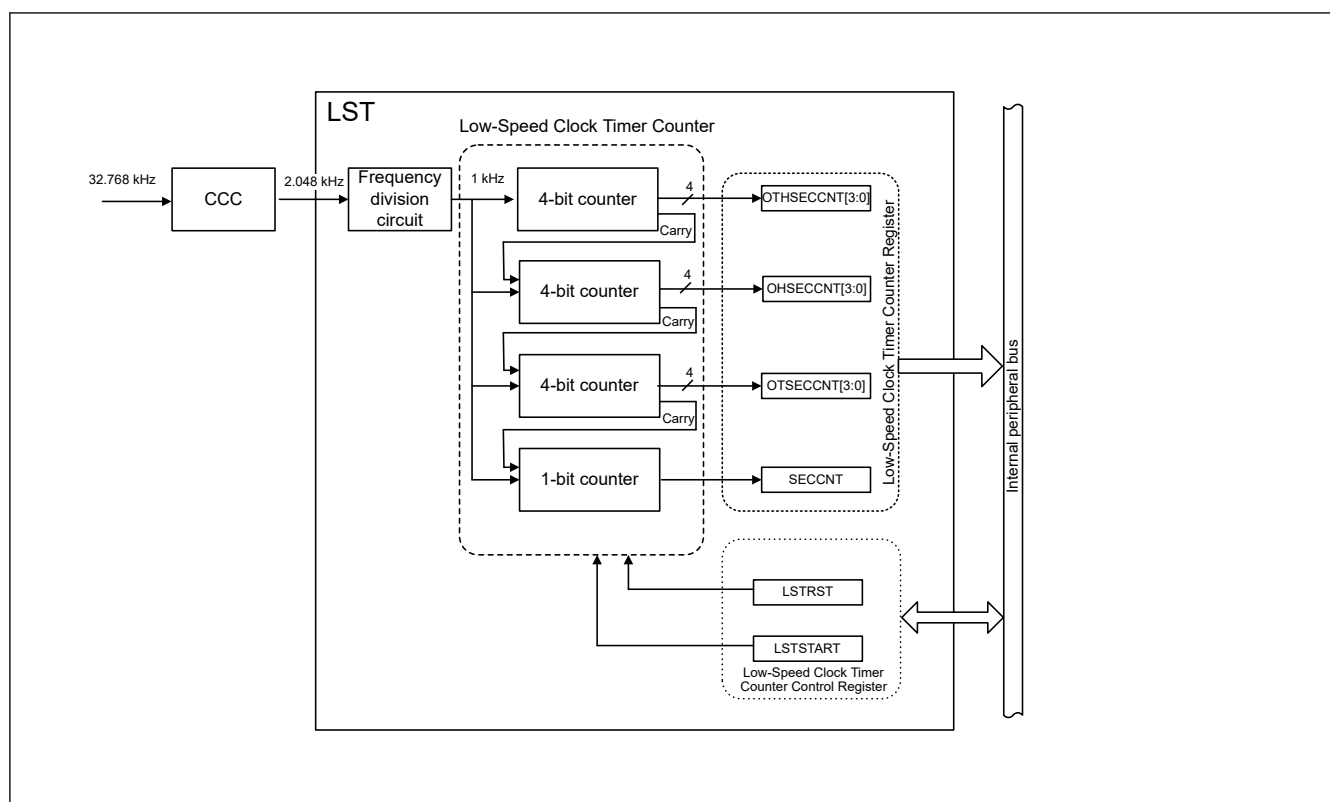


Figure 29.1 LST block diagram

29.2 Register Descriptions

29.2.1 LSTCNT : Low-Speed Clock Timer Count Register

Base address: LST = 0x4008_4400

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SECCNT	OTSECCNT[3:0]				OHSECCNT[3:0]				OTHSECCNT[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	OTHSECCNT[3:0]	0.001-Second Count Counts from 0 to 9 every 0.001 seconds	R
7:4	OHSECCNT[3:0]	0.01-Second Count Counts from 0 to 9 every 0.01 seconds	R
11:8	OTSECCNT[3:0]	0.1-Second Count Counts from 0 to 9 every 0.1 seconds	R
12	SECCNT	1-Second Count Toggles every second	R
31:13	—	These bits are read as 0.	R

29.2.2 LSTCTRL : Low-Speed Clock Timer Counter Control Register

Base address: LST = 0x4008_4400

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LSTRST	LSTSTART
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LSTSTART	Count Start 0: The count stops 1: The count starts	R/W
1	LSTRST	Counter Clear 0: No effect 1: The low-speed clock timer counter and the LSTCNT register are set to 0	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

LSTSTART bit (Count Start)

The LSTSTART bit enables or disables the LST counter.

LSTRST bit (Counter Clear)

When 1 is written to the LSTRST bit, the count source is synchronized, and the low-speed clock timer counter and the LSTCNT register are set to 0. After the operation, the LSTRST bit is automatically set to 0.

29.3 Operation**29.3.1 Basic Operation**

For the LST, set the LSTSTART bit in the Low-Speed Clock Timer Control Register (LSTCTRL) to 1 so that the LSTCNT starts counting. The LSTCNT register value is in the BCD format in the range from 0.000 to 1.999 seconds, in 0.001-second units.

29.4 Usage Notes**29.4.1 Settings for the Module-Stop Function**

The Module Stop Control Register D (MSTPCRD) can enable or disable LST operation. The LST module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

29.4.2 Limits in the Bus Clock Specifications

When the LST register is accessed, the frequency of the PCLKB must be set to at least 8 kHz.

29.4.3 Settings for the Clock Source

The clock source (2.048 kHz) for the LST is provided by the Clock Correction Circuit (CCC). After the Clock Correction Circuit Enable bit (CADJUSCEN) in the 128-Hz Counter Control Register (CCC.R128CTRL) is set to 1 (CCC is enabled), start the LST count. For details, see [section 10, Clock Correction Circuit \(CCC\)](#).

30. Watchdog Timer (WDT)

30.1 Overview

The Watchdog Timer (WDT) is a 14-bit down counter that can be used to reset the MCU when the counter underflows because the system has run out of control and is unable to refresh the WDT. In addition, the WDT can be used to generate a non-maskable interrupt or an underflow interrupt.

[Table 30.1](#) lists the WDT specifications and [Figure 30.1](#) shows a block diagram.

Table 30.1 WDT specifications

Parameter	Specifications
Count source* ¹	Peripheral clock (PCLKB) or CCC_2K clock* ² One of these is selected to be the WDT count clock (WDTCLK).
Clock division ratio	Division by 4, 64, 128, 512, 2048, or 8192
Counter operation	Counting down using a 14-bit down-counter
Condition for starting the counter	- Auto start mode: Counting automatically starts after a reset or after an underflow or refresh error occurs - Register start mode: Counting is started with a refresh by writing to the WDTRR register
Conditions for stopping the counter	- Reset (the down-counter and other registers return to their initial values) - A counter underflows or a refresh error is generated
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Watchdog timer reset sources	- Down-counter underflows - Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt/interrupt sources	- Down-counter underflows - Refreshing outside the refresh-permitted period (refresh error)
Reading of the counter value	The down-counter value can be read by the WDTSR register
Event link function (output)	- Down-counter underflow event output - Refresh error event output
Output signal (internal signal)	- Reset output - Interrupt request output

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) $\geq 4 \times$ (the frequency of the count clock source after division).

Note 2. The WDT count source is selected by WDTCLKSEL in the Option Function Select Register 0 (OFS0). For details, see [section 7.2.1](#).

[OFS0 : Option Function Select Register 0](#).

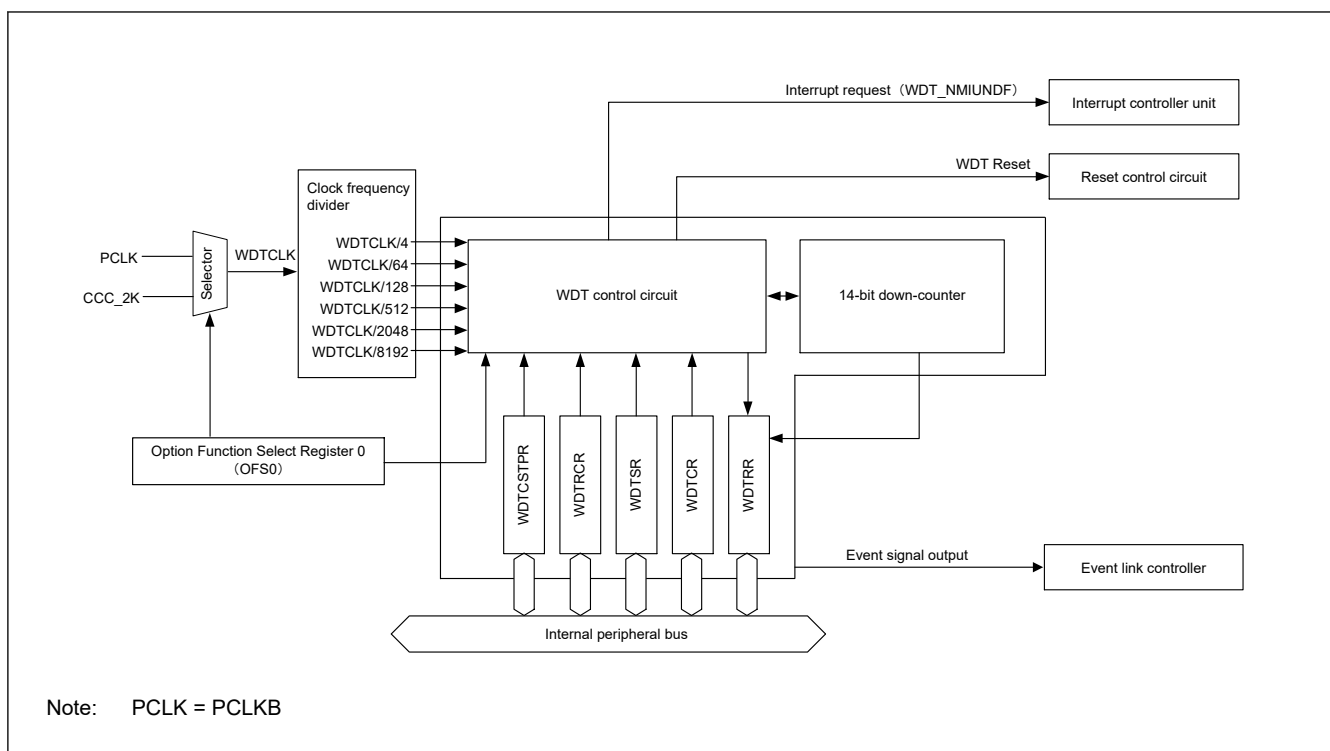


Figure 30.1 WDT block diagram

30.2 Register Descriptions

30.2.1 WDTRR : WDT Refresh Register

Base address: WDT = 0x4004_4200

Offset address: 0x00



Value after reset: 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
7:0	n/a	The down-counter is refreshed by writing 0x00 and then writing 0xFF to this register.	R/W

The WDTRR register refreshes the down-counter of the WDT.

The down-counter of the WDT is refreshed by writing 0x00 and then writing 0xFF to WDTRR register (refresh operation) within the refresh-permitted period.

After the down-counter is refreshed, it starts counting down from the value selected by setting the WDT Timeout Period Select bits (OFS0.WDTPS[1:0]) in the Option Function Select Register 0 in auto start mode. In register start mode, counting down starts from the value selected by setting the Timeout Period Select bits (WDTCSR.TOPS[1:0]) in the WDT Control Register.

When 0x00 is written, the read value is 0x00. When a value other than 0x00 is written, the read value is 0xFF. For details of the refresh operation, see [section 30.3.3. Refresh Operation](#).

30.2.2 WDTCR : WDT Control Register

Base address: WDT = 0x4004_4200

Offset address: 0x02

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

Bit	Symbol	Function	R/W
1:0	TOPS[1:0]	Timeout Period Select 0 0: 1024 cycles (0x03FF) 0 1: 4096 cycles (0x0FFF) 1 0: 8192 cycles (0x1FFF) 1 1: 16384 cycles (0x3FFF)	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	CKS[3:0]	Clock Division Ratio Select 0x1: WDTCLK/4 0x4: WDTCLK/64 0xF: WDTCLK/128 0x6: WDTCLK/512 0x7: WDTCLK/2048 0x8: WDTCLK/8192 Others: Setting prohibited	R/W
9:8	RPES[1:0]	Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (do not specify window end position).	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	RPSS[1:0]	Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (do not specify window start position).	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

The WDTCR register is used to set the clock division ratio, and window start and end positions for refresh, and the timeout period until the down-counter underflows in register start mode.

Some constraints apply to writes to the WDTCR register. For details, see [section 30.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCTPR 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 30.3.8. Association between Option Function Select Register 0 \(OFS0\) and WDT Registers](#).

TOPS[1:0] bits (Timeout Period Select)

The TOPS[1:0] bits select the timeout period, the period until the down-counter underflows, from 1024, 4096, 8192, and 16384 cycles, taking the divided clock specified in the CKS[3:0] bits as 1 cycle. After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the number of WDTCLK cycles until the counter underflows.

[Table 30.2](#) lists the relationship between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of WDTCLK cycles.

Table 30.2 Timeout period settings

CKS[3:0] bits	TOPS[1:0] bits	Clock division ratio	Timeout period (number of cycles)	WDTCLK clock cycles
0x1	00b	WDTCLK/4	1024	4096
	01b		4096	16384
	10b		8192	32768
	11b		16384	65536
0x4	00b	WDTCLK/64	1024	65536
	01b		4096	262144
	10b		8192	524288
	11b		16384	1048576
0xF	00b	WDTCLK/128	1024	131072
	01b		4096	524288
	10b		8192	1048576
	11b		16384	2097152
0x6	00b	WDTCLK/512	1024	524288
	01b		4096	2097152
	10b		8192	4194304
	11b		16384	8388608
0x7	00b	WDTCLK/2048	1024	2097152
	01b		4096	8388608
	10b		8192	16777216
	11b		16384	33554432
0x8	00b	WDTCLK/8192	1024	8388608
	01b		4096	33554432
	10b		8192	67108864
	11b		16384	134217728

CKS[3:0] bits (Clock Division Ratio Select)

The CKS[3:0] bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from the WDTCLK divided by 4, 64, 128, 512, 2048, and 8192. Combined with the TOPS[1:0] bit setting, this allows the WDT to be configured to a count period between 4096 and 134217728 WDTCLK clock cycles.

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

The RPES[1:0] bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. Set the window end position to a value less than the value for the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.

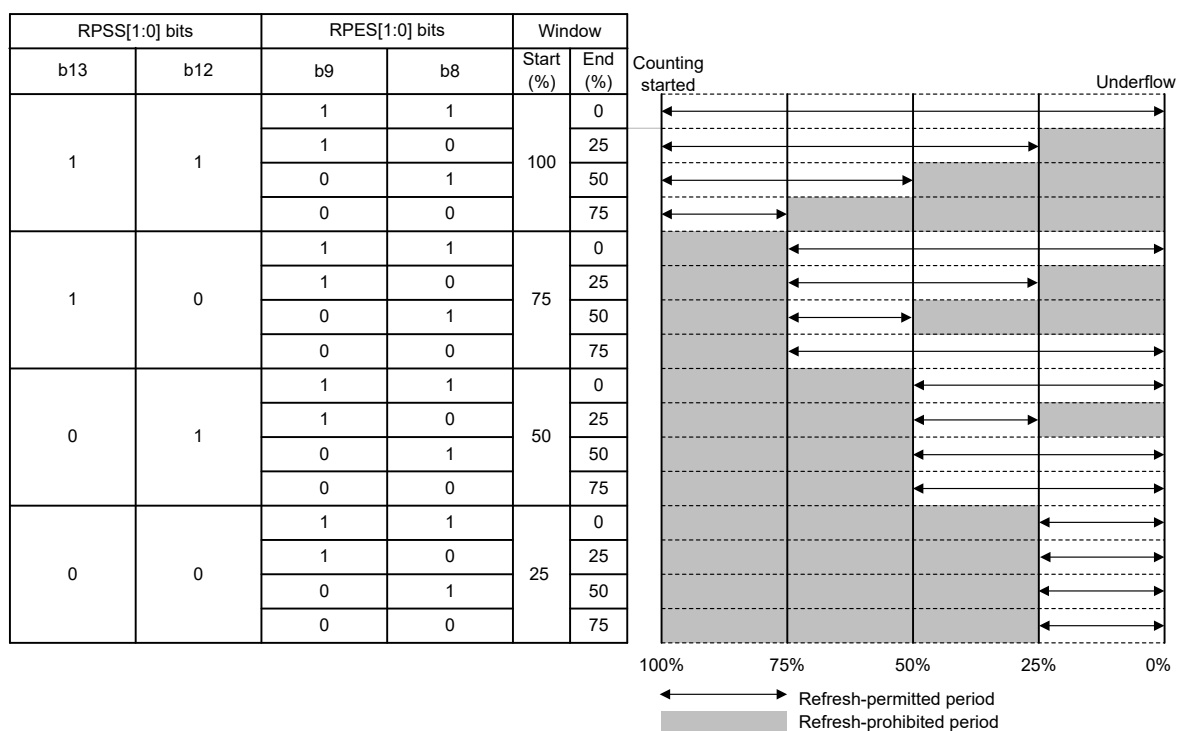
RPSS[1:0] bits (Window Start Position Select)

The RPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. Set the window start position to a value greater than the value for the window end position. If the window start position is set to a value less than or equal to the window end position, the window end position is set to 0%.

Table 30.3 lists the counter values for the window start and end positions, and Figure 30.2 shows the refresh-permitted period set in the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 30.3 Relationship between the timeout period and window start and end counter values

TOPS[1:0]	Timeout period		Window start and end counter value			
	Cycles	Counter value	100%	75%	50%	25%
00b	1024	0x03FF	0x03FF	0x02FF	0x01FF	0x00FF
01b	4096	0x0FFF	0x0FFF	0x0BFF	0x07FF	0x03FF
10b	8192	0x1FFF	0x1FFF	0x17FF	0x0FFF	0x07FF
11b	16384	0x3FFF	0x3FFF	0x2FFF	0x1FFF	0x0FFF



Note: If window end setting $\geq$ window start setting, the window end setting is set to 0%.

Figure 30.2 RPSS[1:0] and RPES[1:0] bits setting and refresh-permitted period

30.2.3 WDTSR : WDT Status Register

Base address: WDT = 0x4004_4200

Offset address: 0x04

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field: REFE F UNDF F CNTVAL[13:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
13:0	CNTVAL[13:0]	Down-Counter Value Value counted by the down-counter	R
14	UNDF	Underflow Flag 0: No underflow occurred 1: Underflow occurred	R/W ¹

Bit	Symbol	Function	R/W
15	REFEF	Refresh Error Flag 0: No refresh error occurred 1: Refresh error occurred	R/W ^{*1}

Note 1. Only 0 can be written to clear the flag.

The WDTSR register indicates the counter value of the down-counter and the status of whether an underflow or refresh error occurred in the down-counter.

CNTVAL[13:0] bits (Down-Counter Value)

Read the CNTVAL[13:0] bits to confirm the value of the down-counter. The read value might differ from the actual count by 1.

UNDFE flag (Underflow Flag)

Read the UNDFE flag to confirm whether an underflow occurred in the counter. A value of 1 indicates that the down counter underflowed. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the UNDFE flag takes (N+2) WDTCLK cycles and two PCLKB cycles. In addition, clearing of the flag is ignored for (N+2) WDTCLK cycles after an underflow. N is specified in the WDTCCR.CKS[3:0] bits as follows:

- When WDTCCR.CKS[3:0] = 0x1, N = 4
- When WDTCCR.CKS[3:0] = 0x4, N = 64
- When WDTCCR.CKS[3:0] = 0xF, N = 128
- When WDTCCR.CKS[3:0] = 0x6, N = 512
- When WDTCCR.CKS[3:0] = 0x7, N = 2048
- When WDTCCR.CKS[3:0] = 0x8, N = 8192

REFEF flag (Refresh Error Flag)

Read the REFEF flag to confirm whether a refresh error occurred, indicating that a refresh operation was performed during a prohibited period. A value of 1 indicates that a refresh error occurred. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the REFEF flag takes (N+2) WDTCLK cycles and two PCLKB cycles. In addition, clearing of the flag is ignored for (N+2) WDTCLK cycles after a refresh error. N is specified in the WDTCCR.CKS[3:0] bits as follows:

- When WDTCCR.CKS[3:0] = 0x1, N = 4
- When WDTCCR.CKS[3:0] = 0x4, N = 64
- When WDTCCR.CKS[3:0] = 0xF, N = 128
- When WDTCCR.CKS[3:0] = 0x6, N = 512
- When WDTCCR.CKS[3:0] = 0x7, N = 2048
- When WDTCCR.CKS[3:0] = 0x8, N = 8192

30.2.4 WDTCCR : WDT Reset Control Register

Base address: WDT = 0x4004_4200

Offset address: 0x06

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	RSTIRQS	Reset Interrupt Request Select 0: Enable non-maskable interrupt request or interrupt request output 1: Enable reset output	R/W

The WDTRCR register controls reset output by a WDT down-counter underflow or interrupt request output.

Some constraints apply to writes to the WDTRCR register. For details, see [section 30.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSNPR 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 30.3.8. Association between Option Function Select Register 0 \(OFS0\) and WDT Registers](#).

30.2.5 WDTCSNPR : WDT Count Stop Control Register

Base address: WDT = 0x4004_4200

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SLCS TP	—	—	—	—	—	—	—

Value after reset: 1 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	SLCSTP	WDT Count Stop Control Register 0: Disable count stop 1: Stop count on transition to Sleep mode	R/W

The WDTCSNPR register controls whether to stop the WDT counter in a low power mode. Some constraints apply to writes to the WDTCSNPR register. For details, see [section 30.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSNPR Registers](#).

In auto start mode, the WDTCSNPR register settings are disabled, and the settings in the Option Function Select register 0 (OFS0) are enabled. The settings for the WDTCSNPR register can also be made for the OFS0 register. For details, see [section 30.3.8. Association between Option Function Select Register 0 \(OFS0\) and WDT Registers](#).

SLCSTP bit (WDT Count Stop Control Register)

The SLCSTP bit selects whether to stop counting on transition to Sleep mode, Snooze, or Software Standby mode.

Count stop control mode varies depending on the clock source specified as the WDTCLK. [Table 30.4](#) lists count stop control by the SLCSTP bit.

Table 30.4 Count Stop Control by SLCSTP Bit

SLCSTP	Low power mode	Count control	
		WDTCLK = PCLKB	WDTCLK = CCC_2K
0	sleep mode	Continues counting	Continues counting
	Snooze / Software Standby mode	Stops counting	
1	Sleep mode	Stops counting	Stops counting
	Snooze / Software Standby mode		

30.2.6 Option Function Select Register 0 (OFS0)

For information on the OFS0 register, see [section 30.3.8. Association between Option Function Select Register 0 \(OFS0\) and WDT Registers](#).

30.3 Operation

30.3.1 Count Operation in each Start Mode

The WDT has two start modes:

- Auto start mode, in which counting automatically starts after a release from the reset state
- Register start mode, in which counting starts with a refresh by writing to the register.

In auto start mode, counting automatically starts after a release from the reset state according to the settings in the Option Function Select register 0 (OFS0) in the flash.

In register start mode, counting starts with a refresh by writing to the WDTRR register after the respective registers are set after a release from the reset state.

Select auto start mode or register start mode by setting the WDT Start Mode Select bit (OFS0.WDTSTRT) in the OFS0 register.

When the auto start mode is selected, the settings in the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDCSTPR) are disabled while the settings in the OFS0 register are enabled.

When the register start mode is selected, the setting for the OFS0 register is disabled while the settings for the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDCSTPR) are enabled.

30.3.1.1 Register start mode

When the WDT Start Mode Select bit (OFS0.WDTSTRT) is 1, register start mode is selected and the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDCSTPR) are enabled.

After the reset state is released, set the following to Sleep mode in the WDCSTPR register:

- Clock division ratio
- Window start and end positions
- Timeout period in the WDTCR register
- Reset output or interrupt request output in the WDTRCR register
- Counter stop control during transitions to Sleep mode in the WDCSTPR register

Refresh the down-counter to start counting down from the value set in the Timeout Period Selection bits (WDTCR.TOPS[1:0]).

Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The WDT does not output the reset signal as long as counting continues. However, if the down-counter underflows because the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the WDT outputs the reset signal or a non-maskable interrupt request/interrupt request (WDT_NMIUNDF). Reset output or interrupt request output can be selected in the WDT Reset Interrupt Request Select bit (WDTRCR.RSTIRQS). Non-maskable interrupt requests or interrupt requests can be selected in the WDT Underflow/Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 30.3 shows an example of operation under the following conditions:

- Register start mode (OFS0.WDTSTRT = 1)
- Reset output is enabled (WDTRCR.RSTIRQS = 1)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b)

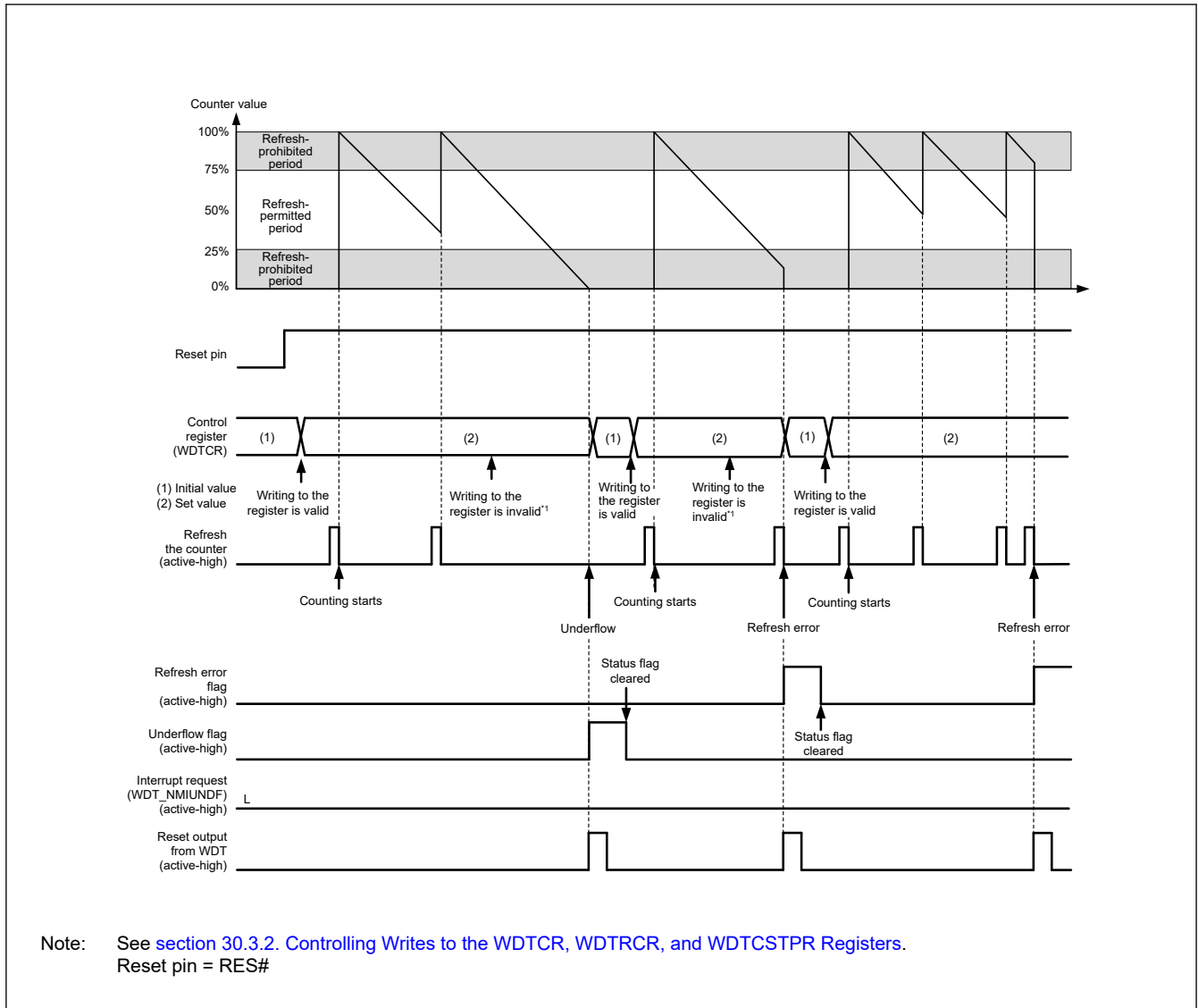


Figure 30.3 Operation example in register start mode

30.3.1.2 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 (WDTCSR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSPTPR) are disabled, and the settings in the OFS0 register are enabled.

Within the reset state, the setting values for the following in the Option Function Select Register 0 (OFS0) are set in the WDT registers:

- Clock division ratio
- Window start and end positions
- Timeout period
- Reset output or interrupt request
- Counter stop control during transition to Sleep mode

When the reset state is released, the down-counter automatically starts counting down from the value set in the WDT Timeout Period Select bits (OFS0.WDTPOPS[1:0]).

Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The WDT does not output the reset signal as long as the counting continues. However, if the down-counter underflows because refreshing of the down-counter is not possible due to a

runaway program or if a refresh error occurs due to refreshing outside the refresh-permitted period, the WDT outputs the reset signal or non-maskable interrupt request/interrupt request (WDT_NMIUNDF).

After the reset signal or non-maskable interrupt request/interrupt request is generated, the counter reloads the timeout period after counting for 1 cycle. The value of the timeout period is set in the down-counter and counting restarts.

Reset output or interrupt request output can be selected by setting the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS). Non-maskable interrupt request or interrupt request can be selected in the WDT Underflow/Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 30.4 shows an example of operation (non-maskable interrupt) under the following conditions:

- Auto start mode (OFS0.WDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.WDTRSTIRQS = 0)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b)

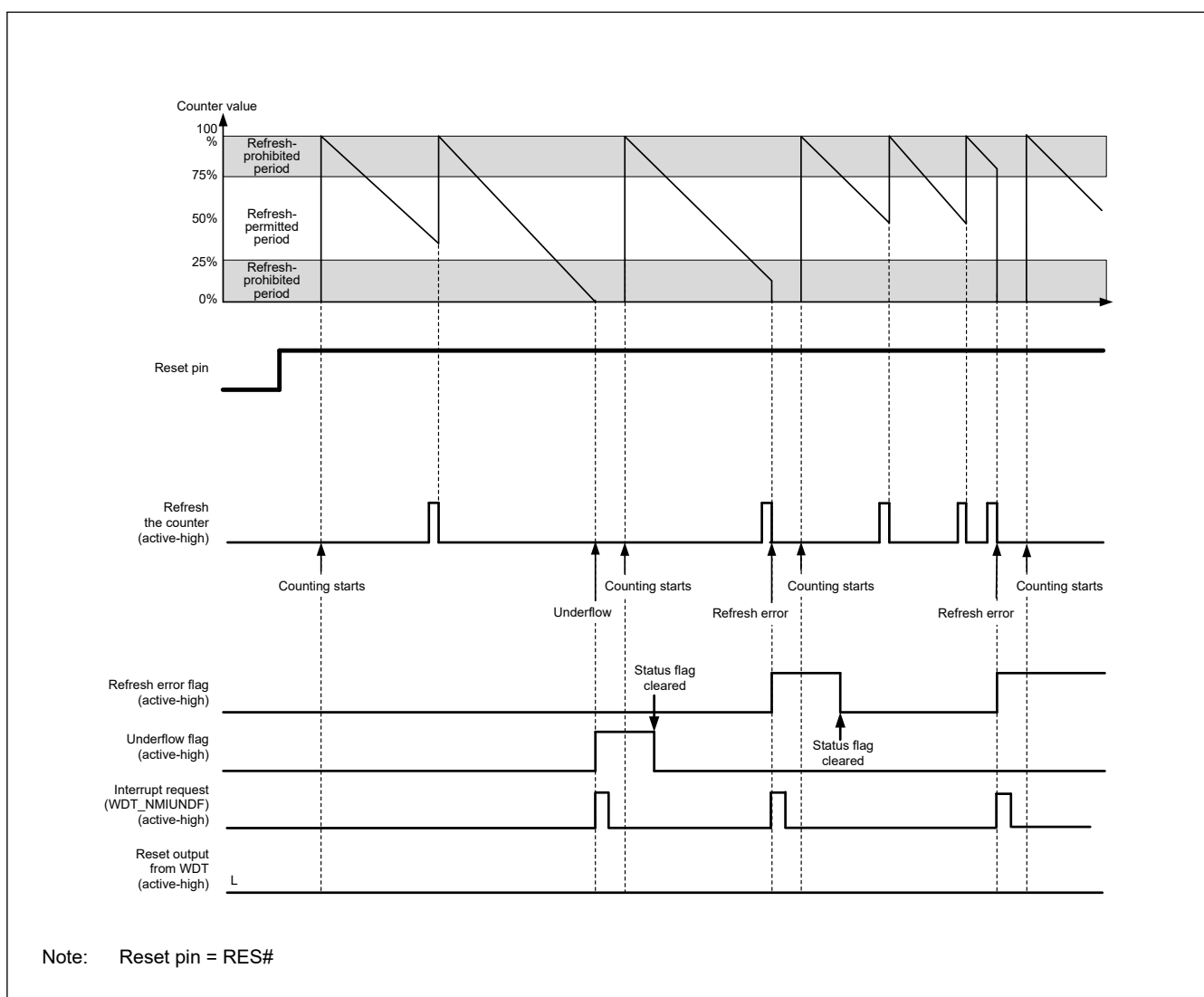


Figure 30.4 Operation example in auto start mode

30.3.2 Controlling Writes to the WDTCR, WDTRCR, and WDTCSSTPR Registers

Writing to the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), or WDT Count Stop Control Register (WDTCSSTPR) is possible once between the release from the reset state and the first refresh operation.

After a refresh (counting starts) or a write to WDTCR, WDTRCR or WDTCSNPR register, the protection signal in the WDT becomes 1 to protect WDTCR, WDTRCR and WDTCSNPR register against subsequent write attempts. This protection is released by the reset source of the WDT. With other reset sources, the protection is not released.

Figure 30.5 shows control waveforms produced in response to writing to the WDTCR.

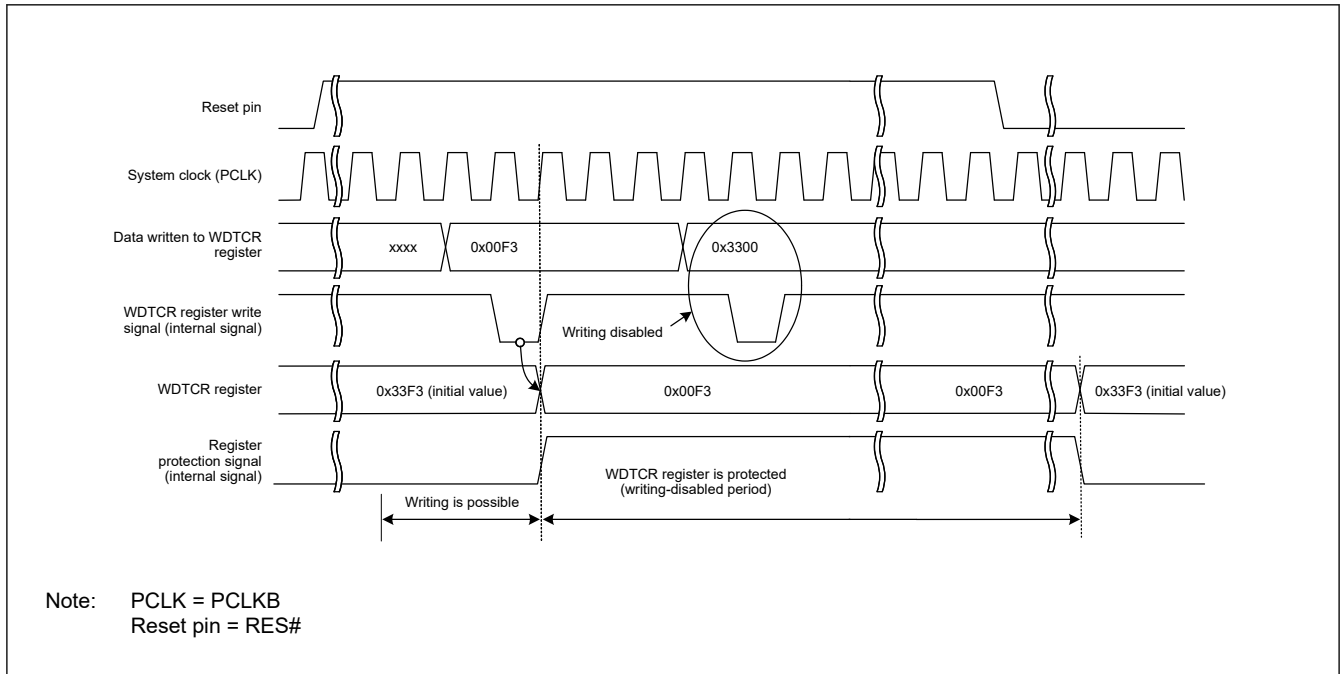


Figure 30.5 Control waveforms produced in response to writes to the WDTCR register

30.3.3 Refresh Operation

The down-counter is refreshed and starts counting operation on a write of the values 0x00 and 0xFF to the WDT Refresh Register (WDTRR). If a value other than 0xFF is written after 0x00, the down-counter is not refreshed. If an invalid value is written, correct refreshing resumes on a write of 0x00 and 0xFF to the WDTRR register.

Correct refreshing is also performed when a register other than WDTRR is accessed or WDTRR is read between writing 0x00 and writing 0xFF to WDTRR. Writes to refresh the counter must be made within the refresh-permitted period, and this is determined by the 0xFF write. For this reason, correct refreshing is performed even when 0x00 is written outside the refresh-permitted period.

[Example write sequences that are valid for refreshing the counter]

- 0x00 → 0xFF
- 0x00 ((n-1)th time) → 0x00 (nth time) → 0xFF
- 0x00 → access to another register or read from WDTRR → 0xFF

[Example write sequences that are invalid for refreshing the counter]

- 0x23 (a value other than 0x00) → 0xFF
- 0x00 → 0x54 (a value other than 0xFF)
- 0x00 → 0xAA (0x00 and a value other than 0xFF) → 0xFF

After 0xFF is written to the WDT Refresh Register (WDTRR), refreshing the down-counter requires up to 4 cycles of the signal for counting. To meet this requirement, complete writing 0xFF to WDTRR 4 count cycles before the down-counter underflows.

Figure 30.6 shows the WDT refresh-operation waveforms when the clock division ratio is WDTCLK/64.

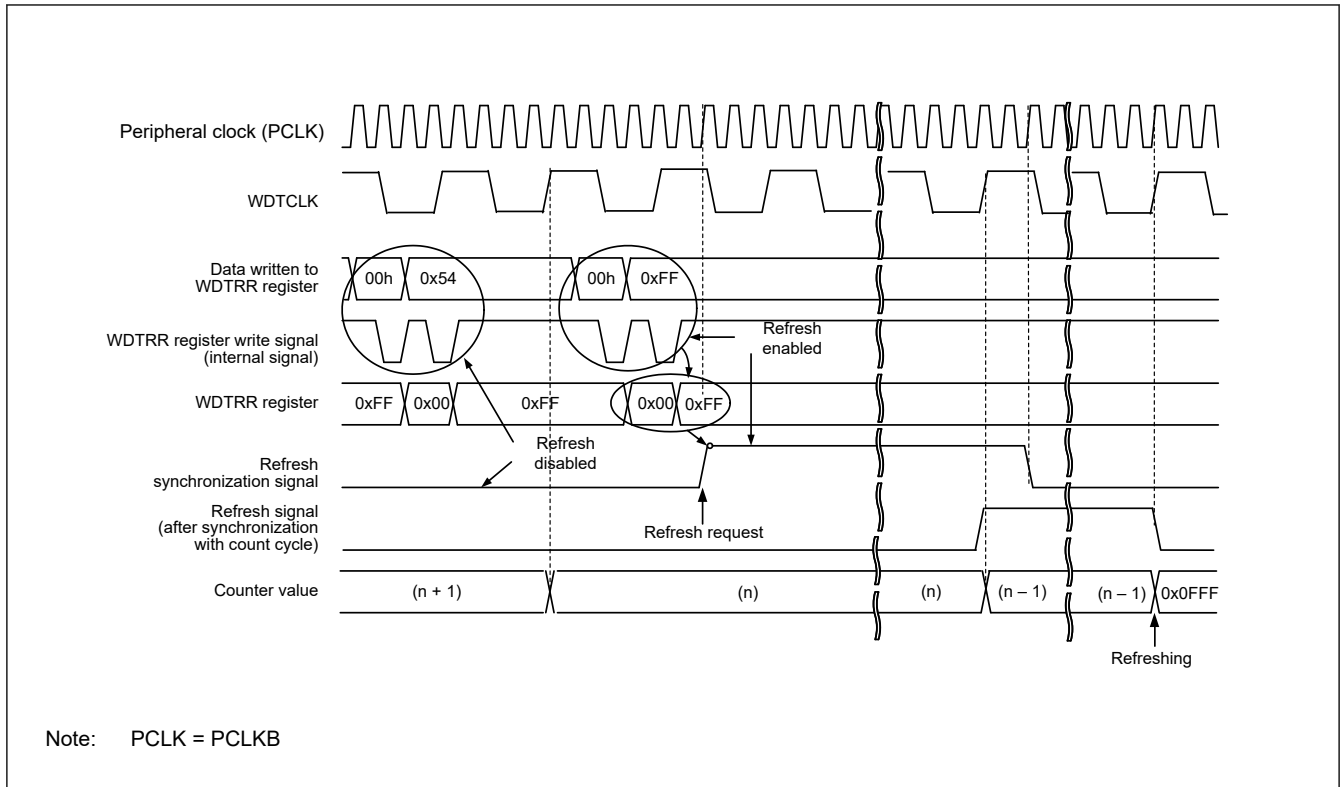


Figure 30.6 WDT refresh operation waveforms when WDTCSR.CKS[3:0] = 0x4 and WDTCSR.TOPS[1:0] = 01b

Note: When setting the refresh time, consider the oscillation accuracy of the clock sources of the PCLKB and WDTCLK. Set values which ensure that refreshing is possible even when the frequency varies in the range of error of the oscillation accuracy.

30.3.4 Status Flags

The refresh error (WDTSR.REFEF) and underflow (WDTSR.UNDF) flags retain the source of the reset signal output from the WDT or the source of the interrupt request from the WDT. After a release from the reset state or interrupt request generation, read the WDTSR.REFEF and WDTSR.UNDF flags to check for the reset or interrupt source. For each flag, writing 0 clears the bit. Writing 1 has no effect. Leaving the status flags unchanged does not affect operation. If the flags are not cleared at the next reset or interrupt request from the WDT, the earlier reset or interrupt source is cleared and the new reset or interrupt source is written. For the time period between when 0 is written in each flag and when its value is reflected, see [section 30.2.3. WDTSR : WDT Status Register](#).

30.3.5 Reset Output

When the Reset Interrupt Select bit (WDTRCR.RSTIRQS) is set to 1 in register start mode, or when the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 1 in auto start mode, a reset signal is output for 1 cycle count when an underflow in the down-counter or a refresh error occurs.

In register start mode, the down-counter is initialized (all bits set to 0) and stopped in that state after output of a reset signal. After the reset state is released and the program is restarted, the counter is set up again and counting down starts again with a refresh. In auto start mode, counting down starts automatically after the reset state is released.

30.3.6 Interrupt Sources

When the Reset Interrupt Select bit (WDTRCR.RSTIRQS) is set to 0 in register start mode or when the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 0 in auto start mode, an interrupt (WDT_NMIUNDF) signal is generated when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or an interrupt. For details, see [section 16, Interrupt Controller Unit \(ICU\)](#).

Table 30.5 WDT interrupt source

Name	Interrupt source	Interrupt to CPU	Start DMAC or DTC
WDT_NMIUNDF	- Down-counter underflow - Refresh error	Possible	Not possible

30.3.7 Reading the Down-Counter Value

The WDT stores the counter value in the down-counter value bits (WDTSR.CNTVAL[13:0]) of the WDT Status Register. Check these bits to obtain the counter value. The read value of the down-counter might differ from the actual count by one.

Figure 30.7 shows the processing for reading the WDT down-counter value when the clock division ratio is WDTCLK/64.

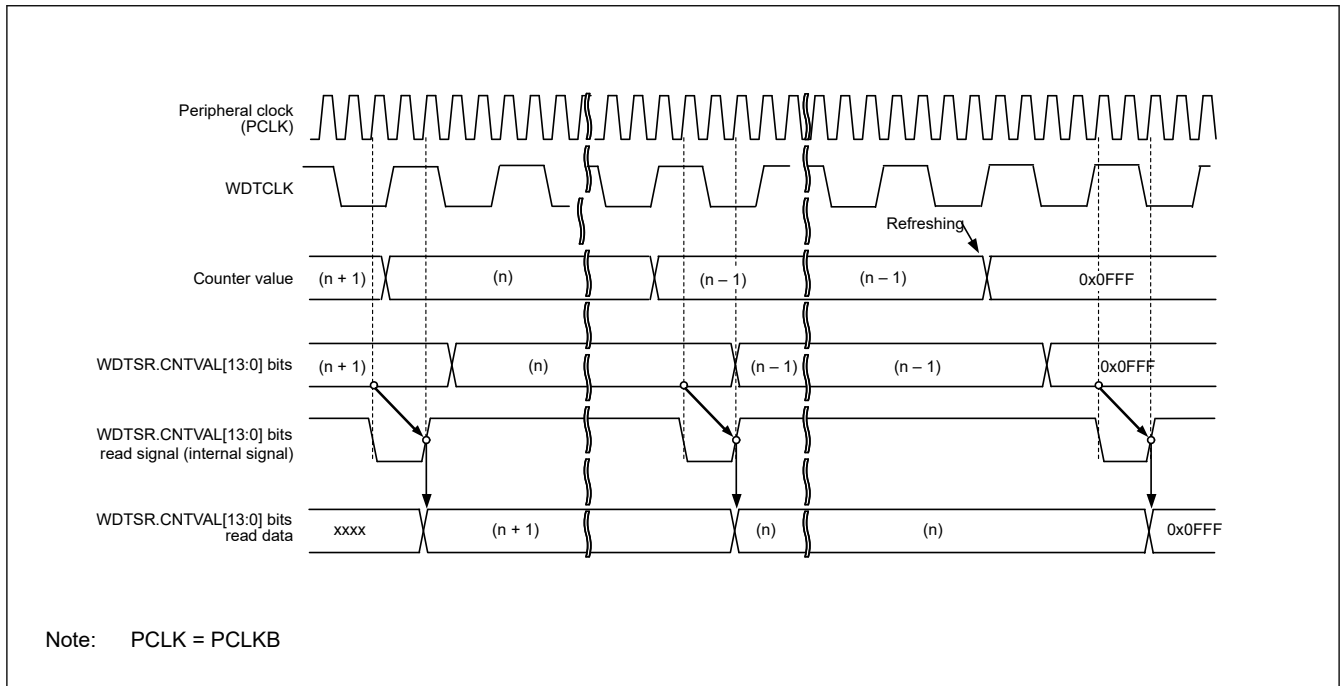


Figure 30.7 Processing for reading WDT down-counter value when WDTCR.CKS[3:0] = 0x4 and WDTCR.TOPS[1:0] = 01b

30.3.8 Association between Option Function Select Register 0 (OFS0) and WDT Registers

Table 30.6 lists the association between the Option Function Select Register 0 (OFS0) used in auto start mode, and the registers used in register start mode. For details on the Option Function Select Register 0 (OFS0), see [section 7.2.1. OFS0 : Option Function Select Register 0](#).

Table 30.6 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.WDTPOPS[1:0]	WDTCR.TOPS[1:0]
	Clock division ratio selection	OFS0.WDTCKS[3:0]	WDTCR.CKS[3:0]
	Window start position selection	OFS0.WDTRPSS[1:0]	WDTCR.RPSS[1:0]
	Window end position selection	OFS0.WDTRPES[1:0]	WDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.WDTRSTIRQS	WDTCCR.RSTIRQS
Count stop	Sleep or Snooze mode count stop control	OFS0.WDTSTPCTL	WDTCSR.SLCSTP

30.4 Output to the Event Link Controller (ELC)

The WDT is capable of a link operation for the previously specified module when interrupt request signal is used as an event signal by the ELC. The event signal is output by the counter underflow and refresh error. An event signal is output regardless of the setting of the Reset Interrupt Request Select bit (WDTRCR.RSTIRQS) in register start mode or auto start mode. An event signal can also be output when the next interrupt source is generated while the Refresh Error flag (WDTSR.REFEF) or Underflow flag (WDTSR.UNDF) is 1. For details, see [section 21, Event Link Controller \(ELC\)](#).

30.5 Usage Notes

30.5.1 ICU Event Link Setting Register n (IELSRn) Setting

Setting 0x07 to ICU Event Link Setting Register n (ICU.IELSRn) is prohibited when enabling the WDT reset assertion (OFS0.WDTRSTIRQS = 0 or WDTRCR.RSTIRQS = 0) or when enabling event link operation (ELSRn.ELS[7:0] = 0x18).

30.5.2 Transition to MINPWON

Counting by the WDT stops when a transition to MINPWON is made. In auto-start mode, however, a transition from MINPWON to EXFPWON or to ALLPWON triggers the start of counting from the initial state.

31. Independent Watchdog Timer (IWDT)

31.1 Overview

The Independent Watchdog Timer (IWDT) consists of a 14-bit down counter that must be serviced periodically to prevent counter underflow. The IWDT provides functionality to reset the MCU or to generate a non-maskable interrupt or an underflow interrupt. Because the timer operates with an independent, dedicated clock source, it is particularly useful in returning the MCU to a known state as a fail-safe mechanism when the system runs out of control. The IWDT can be triggered automatically by a reset, underflow, refresh error, or a refresh of the count value in the registers.

The IWDT functions differ from those of the WDT in the following respects:

- The divided IWDT-dedicated clock (IWDTCLK) is used as the count source (not affected by PCLKB)
- IWDT does not support register start mode

Table 31.1 lists the IWDT specifications and Figure 31.1 shows a block diagram.

Table 31.1 IWDT specifications

Parameter	Description
Count source ^{*1}	IWDT-dedicated clock (IWDTCLK)
Clock division ratio	Division by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Condition for starting the counter	• Counting automatically starts after a reset
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error is generated (counting restarts automatically).
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Reset output sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error).
Non-maskable interrupt/interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error).
Reading the counter value	The down-counter value can be read by the IWDTSR register
Event link function	• Down-counter underflow event output • Refresh error event output.
Output signal (internal signal)	• Reset output • Interrupt request output • Sleep-mode count stop control output.
Auto start mode	Configurable to the following triggers: • Clock frequency division ratio after a reset (OFS0.IWDTCKS[3:0] bits) • Timeout period of the Independent Watchdog Timer (OFS0.IWDTTOPS[1:0] bits) • Window start position in the Independent Watchdog Timer (OFS0.IWDTTPSS[1:0] bits) • Window end position in the Independent Watchdog Timer (OFS0.IWDTRPES[1:0] bits) • Reset output or interrupt request output (OFS0.IWDTSTIRQS 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) $\geq 4 \times$ (the frequency of the count clock source after division).

The bus interface and registers operate with PCLKB, and the 14-bit counter and control circuits operate with IWDTCLK.

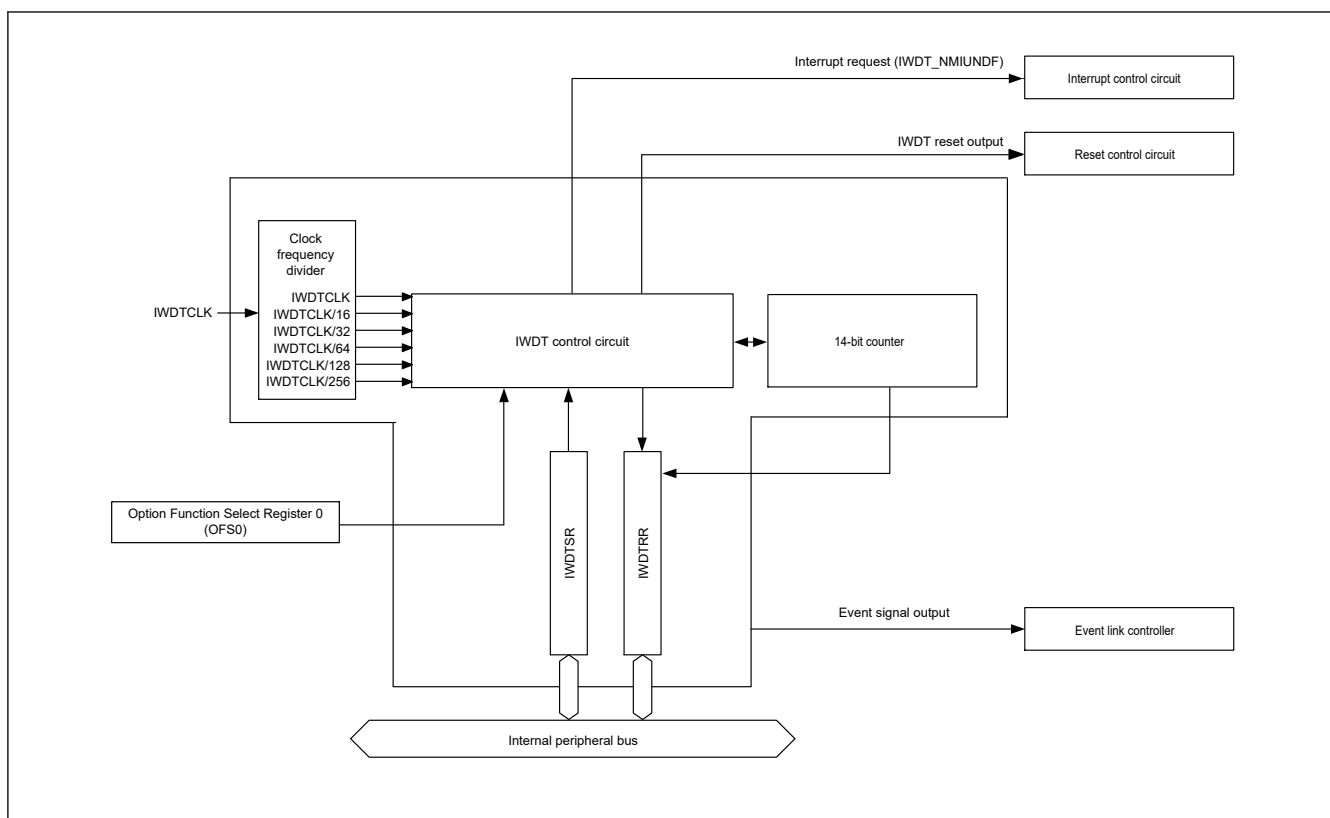


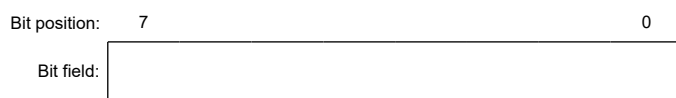
Figure 31.1 IWDT block diagram

31.2 Register Descriptions

31.2.1 IWDTRR : IWDT Refresh Register

Base address: IWDT = 0x4004_4400

Offset address: 0x00



Value after reset: 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
7:0	n/a	The down-counter is refreshed by writing 0x00 and then writing 0xFF to this register	R/W

The IWDTRR register refreshes the down-counter of the IWDT. The down-counter of the IWDT is refreshed by writing 0x00 and then writing 0xFF to IWDTRR (refresh operation) within the refresh-permitted period. After the down-counter is refreshed, it starts counting down from the value selected in the IWDT Timeout Period Select bits (OFS0.IWDTTOPS[1:0]) in the Option Function Select Register 0 (OFS0).

When 0x00 is written, the read value is 0x00. When a value other than 0x00 is written, the read value is 0xFF. For details of the refresh operation, see [section 31.3.2. Refresh Operation](#).

31.2.2 IWDTSR : IWDT Status Register

Base address: IWDTSR = 0x4004_4400

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	REFE F	UNDF F	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	CNTVAL[13:0]	Down-counter Value Value counted by the down-counter	R
14	UNDF	Underflow Flag 0: No underflow occurred 1: Underflow occurred	R/W ¹
15	REFE	Refresh Error Flag 0: No refresh error occurred 1: Refresh error occurred	R/W ¹

Note 1. Only 0 can be written to clear the flag.

The IWDTSR register indicates the counter value of the down-counter and whether an underflow or refresh error occurred in the down-counter.

CNTVAL[13:0] bits (Down-counter Value)

Read the CNTVAL[13:0] bits to confirm the value of the down-counter. The read value might differ from the actual count by 1.

UNDF flag (Underflow Flag)

Read the UNDF flag to confirm whether an underflow occurred in the down-counter. The value 1 indicates that the down-counter underflowed. Write 0 to the UNDF flag to set the value to 0. Writing 1 has no effect.

Clearing of the UNDF flag takes (N + 2) IWDTCCLK cycles and 2 PCLKB cycles. In addition, clearing of this flag is ignored for (N + 2) IWDTCCLK cycles after an underflow. N is specified in the IWDTCCKS[3:0] bits as follows:

- When OFS0.IWDTCCKS[3:0] = 0x0, N = 1
- When OFS0.IWDTCCKS[3:0] = 0x2, N = 16
- When OFS0.IWDTCCKS[3:0] = 0x3, N = 32
- When OFS0.IWDTCCKS[3:0] = 0x4, N = 64
- When OFS0.IWDTCCKS[3:0] = 0xF, N = 128
- When OFS0.IWDTCCKS[3:0] = 0x5, N = 256.

REFE flag (Refresh Error Flag)

Read the REFE flag to confirm whether a refresh error occurred. This indicates that a refresh operation was performed during a prohibited period. The value 1 indicates that a refresh error occurred. Write 0 to the REFE flag to set the value to 0. Writing 1 has no effect.

Clearing of the REFE flag takes (N + 2) IWDTCCLK cycles and 2 PCLKB cycles. In addition, clearing of this flag is ignored for (N + 2) IWDTCCLK cycles following a refresh error. N is specified in the IWDTCCKS[3:0] bits as follows:

- When OFS0.IWDTCCKS[3:0] = 0x0, N = 1
- When OFS0.IWDTCCKS[3:0] = 0x2, N = 16
- When OFS0.IWDTCCKS[3:0] = 0x3, N = 32
- When OFS0.IWDTCCKS[3:0] = 0x4, N = 64
- When OFS0.IWDTCCKS[3:0] = 0xF, N = 128

- When $\text{OFS0.IWDTCKS}[3:0] = 0x5$, $N = 256$.

31.3 Operation

31.3.1 Auto Start Mode

When the IWDT Start Mode Select bit (OFS0.IWDTSTRT) in the Option Function Select Register 0 is 0, auto start mode is selected, otherwise the IWDT is disabled.

Within the reset state, the setting values for the following in the Option Function Select Register 0 (OFS0) are set in the IWDT registers:

- Clock division ratio ($\text{OFS0.IWDTCKS}[3:0]$)
- Window start and end positions ($\text{OFS0.IWDTRPSS}[1:0]$, $\text{OFS0.IWDRPES}[1:0]$)
- Timeout period ($\text{OFS0.IWDTTOPS}[1:0]$)
- Reset output or interrupt request (OFS0.IWDRSTIRQS)

When the reset state is released, the counter automatically starts counting down from the value selected in the IWDT Timeout Period Select bits ($\text{OFS0.IWDTTOPS}[1:0]$).

After that, as long as the program continues normal operation and the counter is refreshed within the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and down-counting continues. The IWDT does not output the reset signal as long as this procedure continues. However, if the counter underflows because the program crashed or because a refresh error occurred when an attempt is made to refresh outside the refresh-permitted period, the IWDT asserts the reset signal or non-maskable interrupt request/interrupt request (IWDT_NMIUNDF).

After the reset signal or non-maskable interrupt request/interrupt request is generated, the counter reloads the timeout period after counting for 1 cycle, the value of the timeout period is set in the down-counter and counting starts. The reset output or interrupt request output can be selected with the IWDT Reset Interrupt Request Select bit (OFS0.IWDRSTIRQS). Non-maskable interrupt request or interrupt request can be selected with the IWDT Underflow/Refresh Error Interrupt Enable bit (NMIER.IWDTEN).

Figure 31.2 shows an example of operation under the following conditions:

- Auto start mode ($\text{OFS0.IWDTSTRT} = 0$)
- Non-maskable interrupt request output is enabled ($\text{OFS0.IWDRSTIRQS} = 0$)
- The window start position is 75% ($\text{OFS0.IWDTRPSS}[1:0] = 10b$)
- The window end position is 25% ($\text{OFS0.IWDRPES}[1:0] = 10b$).

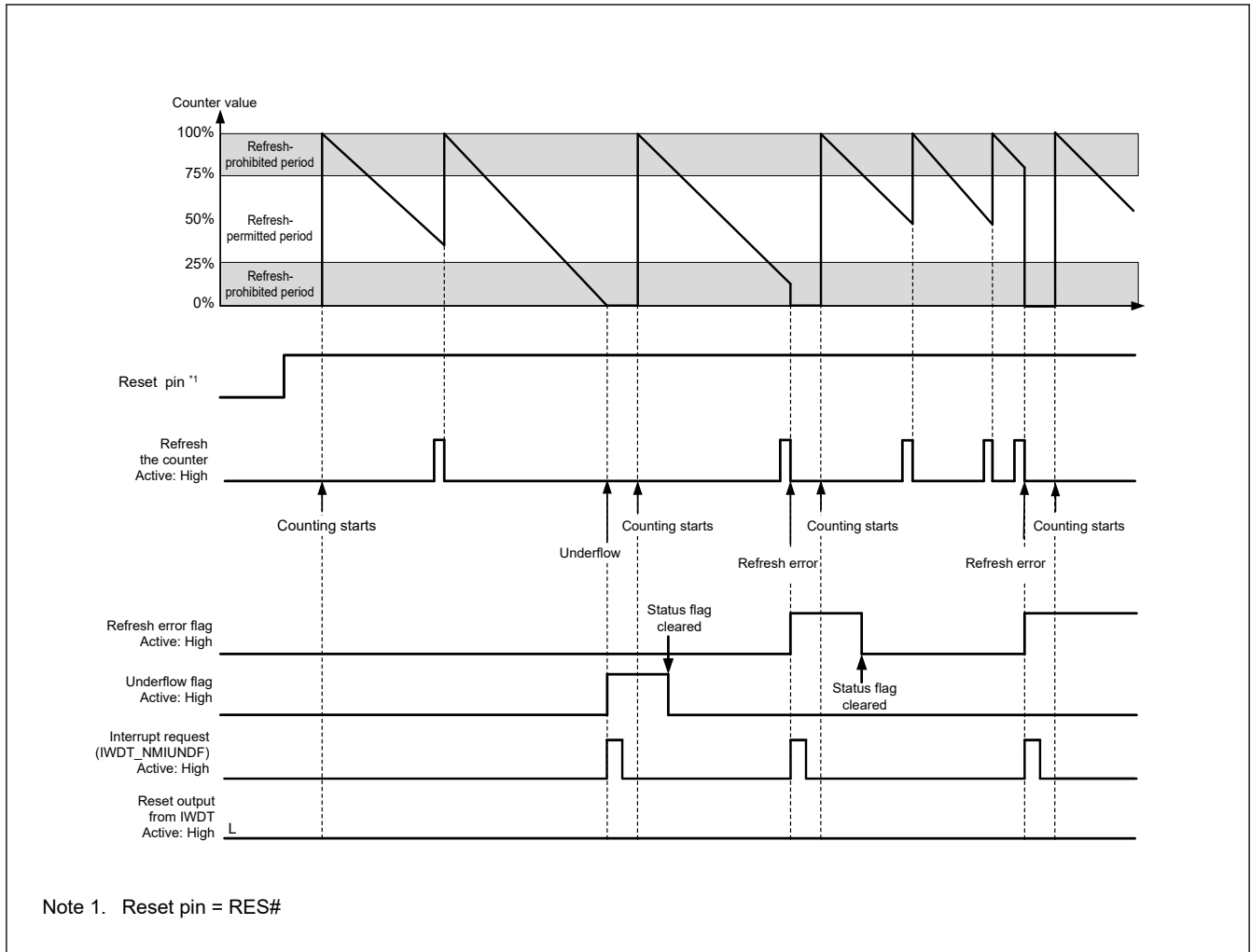


Figure 31.2 Operation example in auto start mode

31.3.2 Refresh Operation

The down-counter is refreshed and operation starts (counting is started by refreshing) by writing the values 0x00 and 0xFF to the IWDT Refresh Register (IWDTRR). If a value other than 0xFF is written after 0x00, the down-counter is not refreshed. If an invalid value is written, correct refreshing resumes on a write of 0x00 and 0xFF to the IWDTRR.

When writes are made in the order of 0x00 (first time) → 0x00 (second time), and if 0xFF is written after that, the writing order 0x00 → 0xFF is satisfied. Writing 0x00 ((n - 1)th time) → 0x00 (nth time) → 0xFF is valid, and the refresh is performed correctly. Even when the first value written before 0x00 is not 0x00, correct refreshing is performed as long as the operation contains the write sequence of 0x00 → 0xFF.

Correct refreshing is also performed regardless of whether a register other than IWDTRR is accessed or IWDTRR is read between writing 0x00 and writing 0xFF to IWDTRR. Writes to refresh the counter must be made within the refresh-permitted period. Whether writing is done within the refresh-permitted period is determined when 0xFF is written. For this reason, correct refreshing is performed even when 0x00 is written outside the refresh-permitted period.

[Example write sequences that are valid to refresh the counter]

- 0x00 → 0xFF
- 0x00 ((n - 1)th time) → 0x00 (nth time) → 0xFF
- 0x00 → access to another register or read from IWDTRR → 0xFF.

[Example write sequences that are not valid to refresh the counter]

- 0x23 (a value other than 0x00) → 0xFF
- 0x00 → 0x54 (a value other than 0xFF)

- 0x00 → 0xAA (0x00 and a value other than 0xFF) → 0xFF.

After 0xFF is written to the IWDTRR register, refreshing the counter requires up to 4 cycles of the signal for counting (the IWDT-Dedicated Clock Frequency Division Ratio Select bits (OFS0.IWDTCKS[3:0]) to determine how many cycles of the IWDT-dedicated clock (IWDTCCLK) make up 1 cycle for counting. To meet this requirement, writing 0xFF to the IWDTRR must be completed 4 count cycles before the end of the refresh-permitted period or a down-counter underflow. The value of the counter can be checked with the counter bits (IWDTSR.CNTVAL[13:0]).

[Example refreshing timings]

- When the window start position is set to 0x1FFF, even if 0x00 is written to IWDTRR before 0x1FFF is reached at (0x2002, for example), refreshing occurs if 0xFF is written to IWDTRR after the value of the IWDTSR.CNTVAL[13:0] bits reaches 0x1FFF
- When the window end position is set to 0x1FFF, refreshing occurs if 0x2003 (4 count cycles before 0x1FFF) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 0x00 → 0xFF to IWDTRR
- When the refresh-permitted period continues until count 0x0000, refreshing can be performed immediately before an underflow. In this case, if 0x0003 (4 count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 0x00 → 0xFF to IWDTRR, no underflow occurs and refreshing is performed.

Figure 31.3 shows the IWDT refresh-operation waveforms when $PCLKB > IWDTCCLK$ and the clock division ratio is IWDTCCLK.

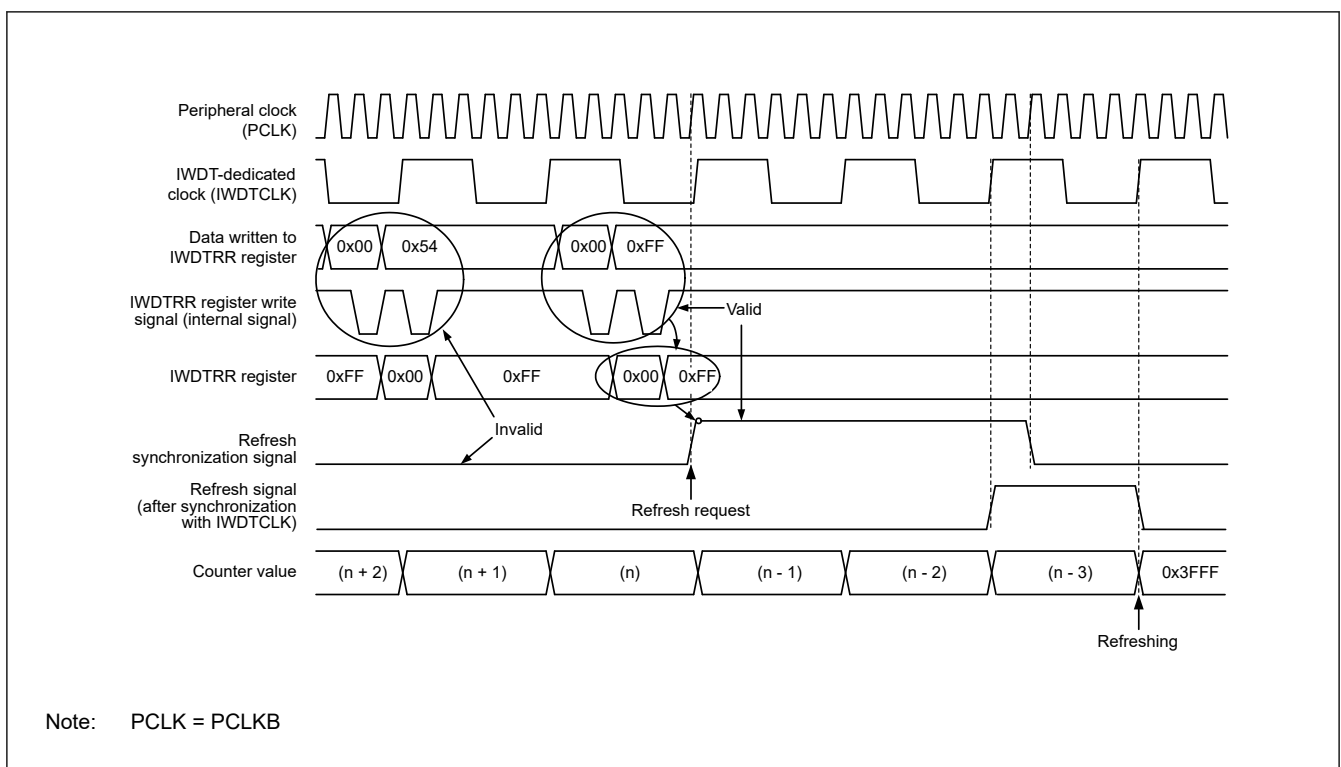


Figure 31.3 IWDT refresh operation waveforms when OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOS[1:0] = 11b

31.3.3 Status Flags

The refresh error (IWDTSR.REFEF) and underflow (IWDTSR.UNDF) flags retain the source of the reset signal output or the source of the interrupt request from the IWDT. Therefore, after a release from the reset state or interrupt request generation, read the IWDTSR.REFEF and UNDF flags to check for the reset or interrupt source. For each flag, writing 0 clears the bit and writing 1 has no effect.

Leaving the status flags unchanged does not affect operation. If the flags are not cleared at the time of the next reset or interrupt request from the IWDT, the earlier reset or interrupt source is cleared and the new reset or interrupt source is

written. For the time period between when 0 is written in each flag and when its value is reflected, see [section 31.2.2. IWDTSR : IWDT Status Register](#).

31.3.4 Reset Output

When the IWDT Reset Interrupt Request Select bit (OFS0.IWDRSTIRQS) 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.

31.3.5 Interrupt Sources

When the IWDT Reset Interrupt Request Select bit (OFS0.IWDRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 0, an interrupt (IWDT_NMIUNDF) signal occurs when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or an interrupt. For details, see [section 16, Interrupt Controller Unit \(ICU\)](#).

Table 31.2 IWDT interrupt source

Name	Interrupt source	Interrupt to CPU	Start DMAC or DTC
IWDT_NMIUNDF	- Down-counter underflow - Refresh error	Possible	Not possible

31.3.6 Reading the Down-Counter Value

As the counter is a IWDT-dedicated clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral clock (PCLKB) and stores it in the down-counter value bits (IWDTSR.CNTVAL[13:0]) of the IWDT Status Register. Check these bits to obtain the counter value indirectly.

Reading the counter value requires multiple PCLKB clock cycles (up to 4 clock cycles), and the read counter value might differ from the actual counter value by a value of one count.

[Figure 31.4](#) shows the processing for reading the IWDT counter value when $PCLKB > IWDTCLK$ and the clock division ratio is IWDTCLK.

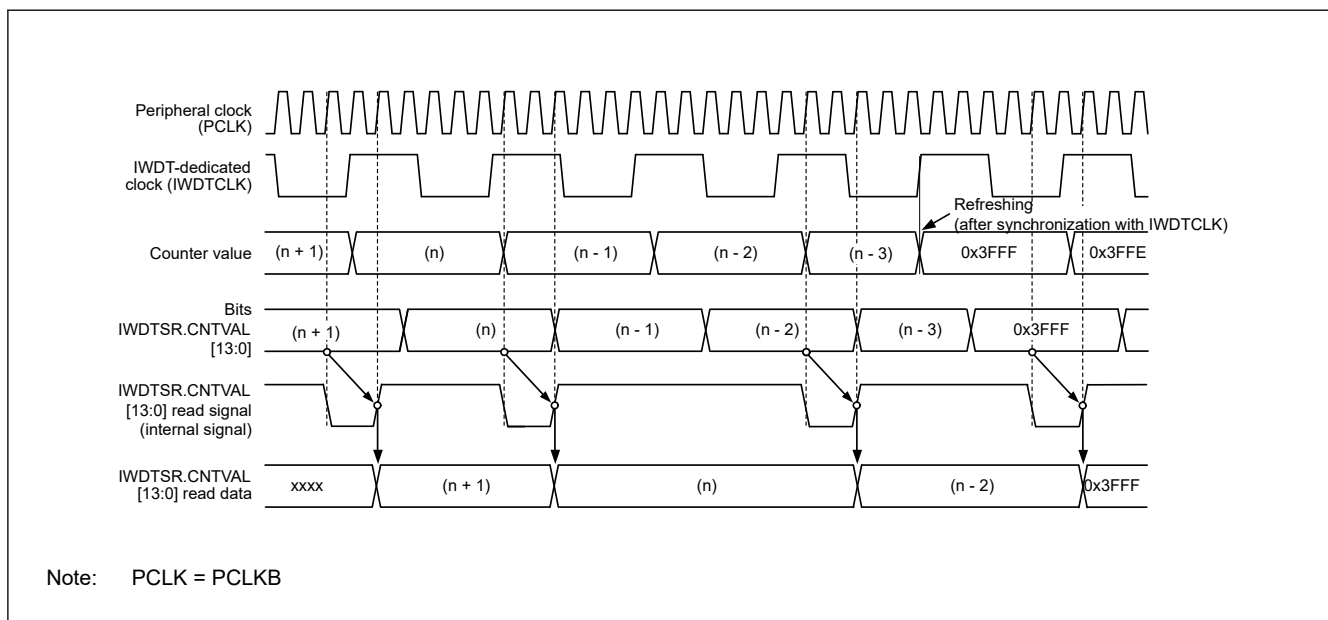


Figure 31.4 Processing for reading IWDT counter value when OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOPS[1:0] = 11b

31.4 Output to the Event Link Controller (ELC)

The IWDT is capable of link operation for a specified module when the interrupt request signal is used as an event signal by the event link controller (ELC). The event signal is output by the counter underflow or refresh error.

An event signal is output regardless of the setting of the OFS0.IWDTRSTIRQS bit. An event signal can also be output at generation of the next interrupt source while the Refresh Error flag (IWDTSR.REFEF) or Underflow flag (IWDTSR.UNDFF) is 1. For details, see [section 21, Event Link Controller \(ELC\)](#).

31.5 Usage Notes

31.5.1 Refresh Operations

While configuring the refresh time, consider variations in the range of errors given the accuracy of PCLKB and IWDTCCLK. Set values that ensure refreshing is possible.

31.5.2 Clock Division Ratio Setting

Satisfy the frequency of the peripheral module clock ($PCLKB \geq 4 \times$ (the frequency of the count clock source after division)).

31.5.3 Constraints on the ICU Event Link Setting Register n (IELSRn) Setting

Setting 0x04 to ICU Event Link Setting Register n (IELSRn.IELS[4:0]) is prohibited when enabling the IWDT reset assertion (OFS0.IWDTRSTIRQS = 0) or when enabling event link operation (ELSRn.ELS[7:0] = 0x17).

32. Serial Communications Interface (SCI)

32.1 Overview

The Serial Communications Interface (SCI) × 7 channels have asynchronous and synchronous serial interfaces:

- Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA))
- 8-bit clock synchronous interface
- Simple IIC (master-only)
- Simple SPI
- Smart card interface

The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. SCIn (n = 0, 1) has FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator.

In this section, PCLK refers to PCLKA for SCIO and SCI1, or PCLKB for SCI2 to SCI5, and SCI9.

[Table 32.1](#) lists the SCI specifications, [Figure 32.1](#) shows a block diagram of SCI, and [Table 32.2](#) lists the I/O pins.

Table 32.1 SCI specifications (1 of 2)

Parameter		Specifications
Number of modules		7 (SCIn (n = 0 to 5, 9))
Serial communication modes		• Asynchronous • Clock synchronous • Simple IIC • Simple SPI • Smart card interface
Transfer speed		Bit rate specifiable with the on-chip baud rate generator
Full-duplex communications		• Transmitter: Continuous transmission possible using double-buffering • Receiver: Continuous reception possible using double-buffering
Data transfer		Selectable as LSB-first or MSB-first transfer
Interrupt sources		Transmit end, transmit data empty, receive data full, receive error, receive data ready, address match. Completion of generation of a start condition, restart condition, or stop condition (for simple IIC mode)
Module-stop function		Module-stop state can be set for each channel
Snooze end request		SCIO address mismatch (SCIO_DCUF)
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Hardware flow control	Transmission and reception controllable with CTSn/RTSn pins
	Transmission and reception	Selectable to 1-stage register or 16-stage FIFO (only SCIn (n = 0, 1) supports FIFO)
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	• Parity error • Overrun error • Framing error
	Hardware flow control	Transmission and reception controllable with CTSn/RTSn pins
	Transmission and reception	Selectable to 1-stage register or 16-stage FIFO (SCIn (n = 0, 1) supports FIFO)

Table 32.1 SCI specifications (2 of 2)

Parameter		Specifications
	Address match	Interrupt request/event output can be issued upon detecting a match between received data and the value in the compare match register
	Address mismatch (SCI0 only) receive data	Snooze end request can be issued when detecting a mismatch between the received data and the value in the compare match register
	Start-bit detection	Selectable to low level or falling edge detection
	Break detection	Breaks from framing errors detectable by read from SPTR register
	Clock source	Selectable to internal or external clock
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Multi-processor communications function	Serial communication enabled among multiple processors
	Noise cancellation	Digital noise filters included on signal paths from the RXDn pin inputs
Smart card interface mode	Error processing	Error signal can be automatically transmitted upon detecting a parity error during reception
		Data can be automatically retransmitted upon receiving an error signal during transmission
	Data type	Both direct and inverse convention supported
Simple IIC mode	Transfer format	I ² C bus format (MSB-first only)
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 400 kbps (PCLK ≥ 20 MHz) Up to 250 kbps (PCLK ≥ 10 MHz) (all channel) Up to 250 kbps (PCLK = 4 MHz) (only SCI3, 4) Up to 125 kbps (PCLK = 2 MHz) (only SCI3, 4)
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters and provide an adjustable interval for noise cancellation
Simple SPI mode	Data length	8 bits
	Error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	SSn input pin function	High impedance state can be invoked on the output pins by driving the SSn pin high.
	Clock settings	Configurable among four clock phase and clock polarity settings
Bit rate modulation function		Error reduction through correction of outputs from the on-chip baud rate generator
Event link function		Error event output for receive error or error signal detection (SCIn_ERI) (n = 2)
		Receive data full event output (SCIn_RXI) (n = 2)
		Transmit data empty event output (SCIn_TXI) (n = 2)
		Transmit end event output (SCIn_TEI) (n = 2)

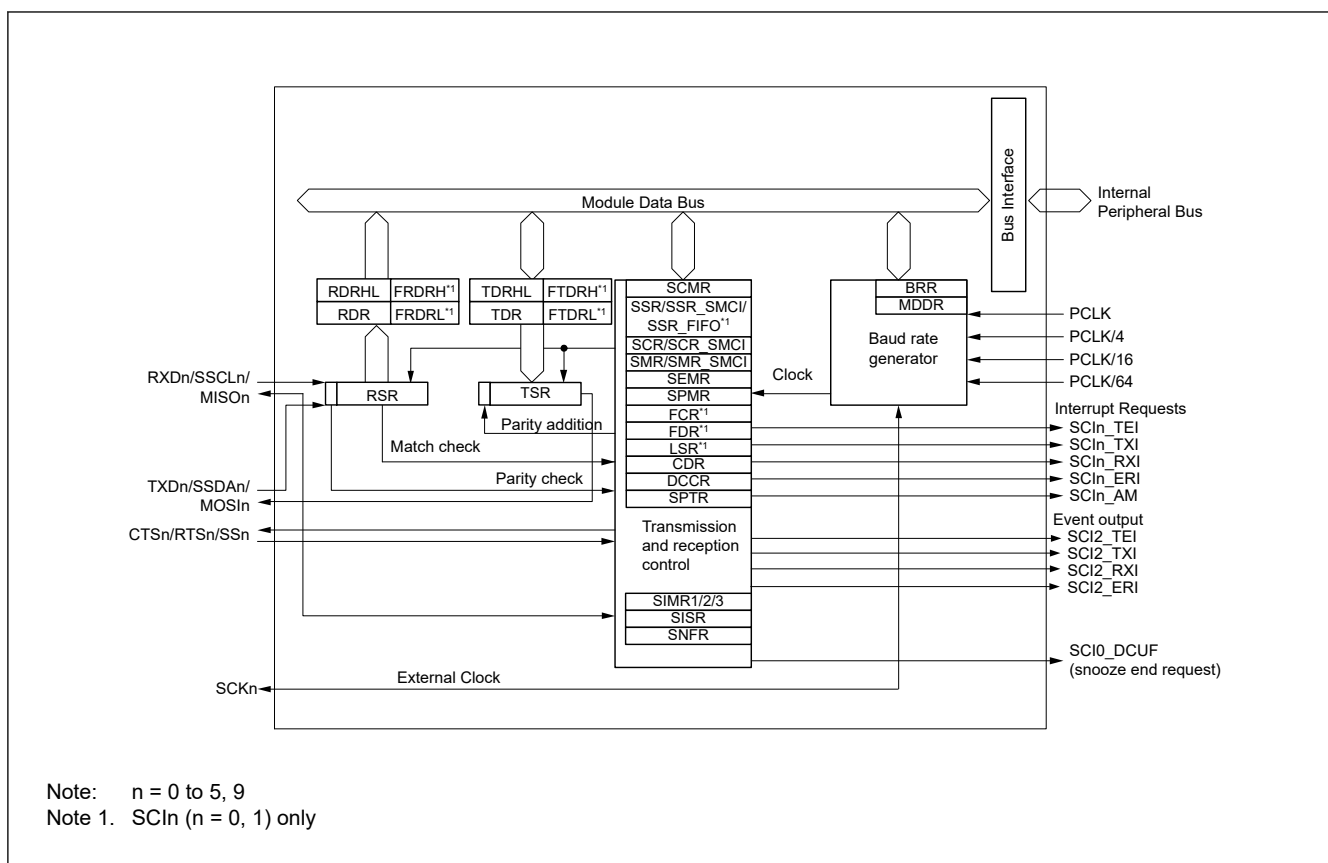


Figure 32.1 SCI block diagram

Table 32.2 SCI I/O pins

Function	Pin name	Input/Output	Description
SCIn (n = 0 to 5, 9)	SCKn	Input/Output	SCIn clock input/output
	RXDn/SSCLn/MISOⁿ	Input/Output	SCIn receive data input SCIn I ² C clock input/output SCIn slave transmit data input/output
	TXDn/SSDAn/MOSⁿ	Input/Output	SCIn transmit data output SCIn I ² C data input/output SCIn master transmit data input/output
	SSn/CTSn/RTSn	Input/Output	SCIn chip select input, active-low SCIn transfer start control input/output, active-low

32.2 Register Descriptions

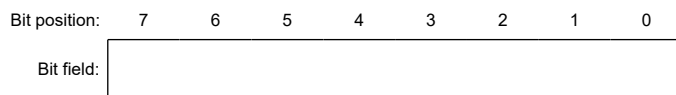
32.2.1 RSR : Receive Shift Register

RSR is a shift register that receives serial data input from the RXDn pin and converts it into parallel data. When one frame of data is received, the data is automatically transferred to the RDR, RDRHL, or the receive FIFO register. The RSR register cannot be directly accessed by the CPU.

32.2.2 RDR : Receive Data Register

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
 SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x05



Value after reset: 0 0 0 0 0 0 0 0

RDR is an 8-bit register that stores received data. When one frame of serial data is received, it is transferred from RSR to RDR, and the RSR register can receive more data. Because RSR and RDR function as a double buffer, continuous received operations can be performed.

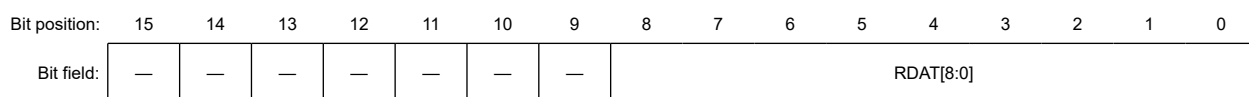
Read the RDR only once after a receive data full interrupt (SCIn_RXI) occurs.

Note: If the next frame of data is received before reading the received data from RDR, an overrun error occurs. The CPU cannot write to the RDR.

32.2.3 RDRHL : Receive Data Register

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
 SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x10



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
8:0	RDAT[8:0]	Serial Receive Data	R
15:9	—	These bits are read as 0.	R

RDRHL is a 16-bit register that stores received data. Use this register when asynchronous mode and 9-bit data length are selected.

The lower 8 bits of RDRHL are the shadow register of RDR, so access to RDRHL affects the RDR register. Access to the RDRHL register is prohibited if 7-bit or 8-bit data length is selected.

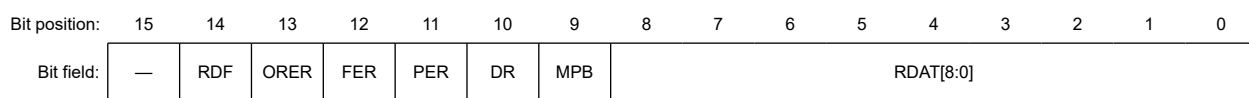
After one frame of data is received, the received data is transferred from the RSR register to the RDR/RDRHL registers, allowing the RSR register to receive more data.

The RSR and RDRHL registers form a double-buffered structure to enable continuous reception. RDRHL should be read only when a receive data full interrupt (SCIn_RXI) request is issued. An overrun error occurs when the next frame of data is received before the received data is read from RDRHL. The CPU cannot write to the RDRHL register.

32.2.4 FRDRHL/FRDRH/FRDRL : Receive FIFO Data Register

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)

Offset address: 0x10 (FRDRHL/FRDRH)
 0x11 (FRDRL)



Value after reset: x 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
8:0	RDAT[8:0]	Serial receive data Stores the serial receive data. Valid only in asynchronous mode, including multi-processor mode, and clock synchronous mode, and with FIFO selected.	R
9	MPB	Multi-Processor Bit Flag Stores the value of the multi-processor bit in the serial receive data, RDAT[8:0]. Valid only in asynchronous mode with SMR.MP = 1, and with FIFO selected. 0: Data transmission cycle 1: ID transmission cycle	R
10	DR	Receive Data Ready Flag This flag is the same as SSR_FIFO.DR. 0: Receiving is in progress, or no received data remains in the FRDRH and FRDRL registers after successfully completed reception 1: Next receive data is not received for a period after successfully completed reception	R ^{*1}
11	PER	Parity Error Flag 0: No parity error occurred in the first data of FRDRH and FRDRL 1: Parity error occurred in the first data of FRDRH and FRDRL	R
12	FER	Framing Error Flag 0: No framing error occurred in the first data of FRDRH and FRDRL 1: Framing error occurred in the first data of FRDRH and FRDRL	R
13	ORER	Overrun Error Flag This flag is the same as SSR_FIFO.ORER. 0: No overrun error occurred 1: Overrun error occurred	R ^{*1}
14	RDF	Receive FIFO Data Full Flag This flag is the same as SSR_FIFO.RDF. 0: The amount of receive data written in FRDRH and FRDRL is less than the specified receive triggering number 1: The amount of receive data written in FRDRH and FRDRL is equal to or greater than the specified receive triggering number	R ^{*1}
15	—	The read value is undefined.	R

Note 1. If this flag is read, it indicates the same value as that read from the SSR_FIFO register. Write 0 to the SSR_FIFO register to clear the flag.

FRDRHL is a 16-bit register that consists of the 8-bit FRDRH and FRDRL registers. FRDRH is assigned to the FRDRHL[15:8] bits, and allocated to the same address as FRDRHL. FRDRL is assigned to the FRDRHL[7:0] bits, and allocated to (the address of FRDRHL + 1) address.

FRDRH and FRDRL constitute a 16-stage FIFO register that stores serial receive data and related status information readable by software. This register is only valid in asynchronous mode, including multi-processor mode, or clock synchronous mode.

The SCI completes reception of one frame of serial data by transferring the received data from the Receive Shift Register (RSR) into FRDRH and FRDRL for storage. Continuous reception is executed until 16 stages are stored. If data is read when there is no received data in FRDRH and FRDRL, the value is undefined. When FRDRH and FRDRL are full, subsequent serial receive data is lost. The CPU can read from the FRDRH and FRDRL registers but cannot write to them.

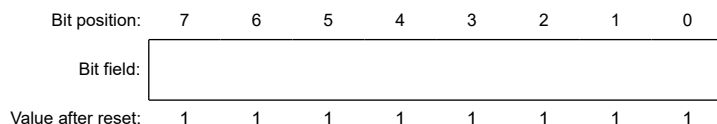
Reading 1 from the RDF, ORER, or DR flags of the FRDRH register is the same as reading from those bits in the SSR_FIFO register. When writing 0 to clear a flag in the SSR_FIFO register after reading the FRDRH register, write 0 only to the flag that is to be cleared and write 1 to the other flags.

When reading both the FRDRH and FRDRL registers, read in order from FRDRH to FRDRL. The FRDRHL register can be accessed in 16-bit units.

32.2.5 TDR : Transmit Data Register

Base address: SCIn = $0x4007\_0000 + 0x0020 \times n$ ($n = 0, 1$)
 SCIm = $0x4004\_1200 + 0x0020 \times m$ ($m = 2$ to $5, 9$)

Offset address: 0x03



Bit	Symbol	Function	R/W
7:0	n/a	Serial Transmit Data	R/W

TDR is an 8-bit register that stores transmit data.

When the SCI detects that the TSR register is empty, it transfers the transmit data written in the TDR register to the TSR register and starts transmission.

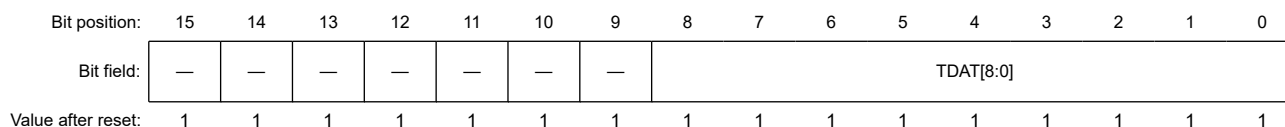
The double-buffered structure of the TDR and TSR registers enables continuous serial transmission. If the next transmit data is already written to TDR when one frame of data is transmitted, the SCI transfers the written data to the TSR register to continue transmission.

The CPU can read from or write to TDR at any time. Only write transmit data to TDR once after each instance of the transmit data empty interrupt (SCIn_TXI).

32.2.6 TDRHL : Transmit Data Register

Base address: SCIn = $0x4007\_0000 + 0x0020 \times n$ ($n = 0, 1$)
 SCIm = $0x4004\_1200 + 0x0020 \times m$ ($m = 2$ to $5, 9$)

Offset address: 0x0E



Bit	Symbol	Function	R/W
8:0	TDAT[8:0]	Serial Transmit Data	R/W
15:9	—	This bit is read as 1. The write value should be 1.	R/W

TDRHL is a 16-bit register that stores transmit data. Use this register when asynchronous mode and 9-bit data length are selected.

The lower 8 bits of TDRHL are the shadow register of TDR, so access to TDRHL affects the TDR register. Access to the TDRHL register is prohibited if 7-bit or 8-bit data length is selected.

When empty space is detected in the TSR register, the transmit data stored in the TDRHL registers is transferred to TSR and transmission starts.

The TSR and TDRHL registers have a double-buffered structure to support continuous transmission. When the next data to be transmitted is stored in TDRHL after one frame of data is transmitted, the transmitting operation continues by transferring the data to the TSR register.

The CPU can read and write to the TDRHL register. Bits [15:9] in TDRHL are fixed to 1. These bits are read as 1. The write value should be 1.

Write transmit data to the TDRHL register only once when a transmit data empty interrupt (SCIn_TXI) request is issued.

32.2.7 FTDRHL/FTDRH/FTDRL : Transmit FIFO Data Register

Base address: $SCIn = 0x4007\_0000 + 0x0020 \times n$ ($n = 0, 1$)

Offset address: 0x0E (FTDRHL/FTDRH)
0x0F (FTDRL)

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

Bit	Symbol	Function	R/W
8:0	TDAT[8:0]	Serial transmit data Specifies the serial transmit data. Valid only in asynchronous mode, including multi-processor mode, and clock synchronous mode, and with FIFO selected.	W
9	MPBT	Multi-Processor Transfer Bit Flag Specifies the multi-processor bit in the transmission frame. Valid only in asynchronous mode and $SMR.MP = 1$, and with FIFO selected. Valid only in asynchronous mode, including multi-processor mode, and clock synchronous mode, and with FIFO selected. 0: Data transmission cycle 1: ID transmission cycle	W
15:10	—	The write value should be 1.	W

FTDRHL is a 16-bit register that consists of the 8-bit FTDRH and FTDRL registers. FTDRH is assigned to the FTDRHL[15:8] bits, and allocated to the same address as FTDRHL. FTDRL is assigned to the FTDRHL[7:0] bits, and allocated to (the address of FTDRHL + 1) address.

FTDRH and FTDRL constitute a 16-stage FIFO register that stores data for serial transmission and a multi-processor transfer bit. This register is only valid in asynchronous mode, including multi-processor mode, or clock synchronous mode.

When the SCI detects that the Transmit Shift Register (TSR) is empty, it transfers data written in the FTDRH and FTDRL registers to the TSR register and starts serial transmission. Continuous serial transmission is executed until no transmit data is left in FTDRH and FTDRL. When FTDRHL is full of transmit data, no more data can be written. If writing new data is attempted, the data is ignored. The CPU can write to the FTDRH and FTDRL registers but cannot read them.

When writing to both the FTDRH and FTDRL registers, write in order from FTDRH to FTDRL.

TDAT[8:0] bits (Serial transmit data)

The TDAT[8:0] bits set the serial transmission data. This is valid only when FIFO is selected in asynchronous mode (including multiprocessor) or clock synchronous mode.

MPBT flag (Multi-Processor Transfer Bit Flag)

The MPBT flag specifies the value of the multi-processor bit of the transmit frame. When $FCR.FM = 1$, $SSR.MPBT$ is invalid.

32.2.8 TSR : Transmit Shift Register

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first automatically transfers transmit data from TDR, TDRHL, or transmit FIFO to TSR, then sends the data to the TXDn pin. The CPU cannot directly access the TSR.

32.2.9 SMR : Serial Mode Register for Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
 SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
1:0	CKS[1:0]	Clock Select 0 0: PCLK clock (n = 0) ^{*1} ⁵ 0 1: PCLK/4 clock (n = 1) ^{*1} 1 0: PCLK/16 clock (n = 2) ^{*1} 1 1: PCLK/64 clock (n = 3) ^{*1}	R/W ⁴
2	MP	Multi-Processor Mode Valid only in asynchronous mode. 0: Disable multi-processor communications function 1: Enable multi-processor communications function	R/W ⁴
3	STOP	Stop Bit Length Valid only in asynchronous mode. 0: 1 stop bit 1: 2 stop bits	R/W ⁴
4	PM	Parity Mode Valid only when the PE bit is 1. 0: Even parity 1: Odd parity	R/W ⁴
5	PE	Parity Enable Valid only in asynchronous mode. 0: When transmitting: Do not add parity bit When receiving: Do not check parity bit 1: When transmitting: Add parity bit When receiving: Check parity bit	R/W ⁴
6	CHR	Character Length Valid only in asynchronous mode. ^{*2} Selects the transmit/receive character length in combination with the SCMR.CHR1 bit. 0: SCMR.CHR1 = 0: Transmit/receive in 9-bit data length SCMR.CHR1 = 1: Transmit/receive in 8-bit data length (initial value) 1: SCMR.CHR1 = 0: Transmit/receive in 9-bit data length SCMR.CHR1 = 1: Transmit/receive in 7-bit data length ^{*3}	R/W ⁴
7	CM	Communication Mode 0: Asynchronous mode or simple IIC mode 1: Clock synchronous mode or simple SPI mode	R/W ⁴

Note 1. n is the decimal notation of the value of n in the BRR register. See [section 32.2.17. BRR : Bit Rate Register](#).

Note 2. In any mode other than asynchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.

Note 3. LSB-first is fixed and the MSB (bit [7]) in the TDR register is not transmitted in transmit mode.

Note 4. Writable only when SCR.TE = 0 and SCR.RE = 0 (both serial transmission and reception are disabled).

Note 5. In simple I2C mode, when 16 clocks are selected as one bit period by the basic clock selection bit of the SEMR register, only n = 0 can be selected.

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 32.2.17. BRR : Bit Rate Register](#).

MP bit (Multi-Processor Mode)

The MP bit disables or enables the multi-processor communications function. The PE and PM bit settings are invalid in multi-processor mode.

STOP bit (Stop Bit Length)

The STOP bit selects the stop bit length in transmission.

In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

PM bit (Parity Mode)

The PM bit selects the parity mode (even or odd) for transmission and reception. The PM bit setting is invalid in multiprocessor mode.

PE bit (Parity Enable)

When the PE bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception. Regardless of the PE bit setting, the parity bit is not added or checked in multi-processor format.

CHR bit (Character Length)

The CHR bit selects the data length for transmission and reception in combination with the SCMR.CHR1 bit. In modes other than asynchronous, a fixed data length of 8 bits is used.

CM bit (Communication Mode)

The CM bit selects the communication mode:

- Asynchronous mode or simple IIC mode
- Clock synchronous mode or simple SPI mode

32.2.10 SMR_SMCI : Serial Mode Register for Smart Card Interface Mode (SCMR.SMIF = 1)

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CKS[1:0]	Clock Select 0 0: PCLK clock (n = 0)*1 0 1: PCLK/4 clock (n = 1)*1 1 0: PCLK/16 clock (n = 2)*1 1 1: PCLK/64 clock (n = 3)*1	R/W*2
3:2	BCP[1:0]	Base Clock Pulse Selects the number of base clock cycles in combination with the SCMR.BCP2 bit. Table 32.3 lists the combinations of the SCMR.BCP2 and SMR.BCP[1:0] bits.	R/W*2
4	PM	Parity Mode Valid only when the PE bit is 1. 0: Even parity 1: Odd parity	R/W*2
5	PE	Parity Enable When this bit is set to 1, a parity bit is added to transmit data, and the parity of received data is checked. Set this bit to 1 in smart card interface mode.	R/W*2

Bit	Symbol	Function	R/W
6	BLK	Block Transfer Mode 0: Normal mode operation 1: Block transfer mode operation	R/W ²
7	GM	GSM Mode 0: Normal mode operation 1: GSM mode operation	R/W ²

Note 1. n is the decimal notation of the value of n in the BRR register. See [section 32.2.17. BRR : Bit Rate Register](#).

Note 2. Writable only when SCR_SMCI.TE = 0 and SCR_SMCI.RE = 0 (both serial transmission and reception are disabled).

The SMR_SMCI register sets the communication format and clock source for the on-chip baud rate generator.

CKS[1:0] bits (Clock Select)

The CKS[1:0] bits select the clock source for the on-chip baud rate generator. For the relationship between the settings of these bits and the baud rate, see [section 32.2.17. BRR : Bit Rate Register](#).

BCP[1:0] bits (Base Clock Pulse)

The BCP[1:0] bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set these bits in combination with the SCMR.BCP2 bit.

For details, see [section 32.6.4. Receive Data Sampling Timing and Reception Margin](#).

Table 32.3 Combinations of SCMR.BCP2 and SMR_SMCI.BCP[1:0] bits

SCMR.BCP2 bit	SMR_SMCI.BCP[1:0] bits	Number of base clock cycles for 1-bit transfer period ^{*1}
0	00b	93 clock cycles (S = 93)
0	01b	128 clock cycles (S = 128)
0	10b	186 clock cycles (S = 186)
0	11b	512 clock cycles (S = 512)
1	00b	32 clock cycles (S = 32) (initial value)
1	01b	64 clock cycles (S = 64)
1	10b	372 clock cycles (S = 372)
1	11b	256 clock cycles (S = 256)

Note 1. S is the value of S in BRR (see [section 32.2.17. BRR : Bit Rate Register](#)).

PM bit (Parity Mode)

The PM bit selects the parity mode for transmission and reception (even or odd). For details on the usage of this bit in smart card interface mode, see [section 32.6.2. Data Format \(Except in Block Transfer Mode\)](#).

PE bit (Parity Enable)

Set the PE bit to 1. The parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

BLK bit (Block Transfer Mode)

Setting the BLK bit to 1 enables block transfer mode operation. For details, see [section 32.6.3. Block Transfer Mode](#).

GM bit (GSM Mode)

Setting the GM bit to 1 enables GSM mode operation. In GSM mode, the SSR_SMCI.TEND flag set timing is moved forward to 11.0 ETUs (elementary time unit = 1-bit transfer time) from the start bit, and clock output control is added. For details, see [section 32.6.6. Serial Data Transmission \(Except in Block Transfer Mode\)](#) and [section 32.6.8. Clock Output Control](#).

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

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
 SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
1:0	CKE[1:0]	Clock Enable 0 0: In asynchronous mode, the SCKn pin is available for use as an I/O port based on the I/O port settings. In clock synchronous mode, the SCKn pin functions as the clock output pin. 0 1: In asynchronous mode, a clock with the same frequency as the bit rate is output from the SCKn pin. In clock synchronous mode, the SCKn pin functions as the clock output pin. Others: In asynchronous mode, input a clock with a frequency 16 times the bit rate from the SCKn pin when the SEMR.ABCS bit is 0. Input a clock signal with a frequency eight times the bit rate when the SEMR.ABCS bit is 1. In clock synchronous mode, the SCKn pin functions as the clock input pin.	R/W ^{*1}
2	TEIE	Transmit End Interrupt Enable 0: Disable SCIn_TEI interrupt requests 1: Enable SCIn_TEI interrupt requests	R/W
3	MPIE	Multi-Processor Interrupt Enable Valid in asynchronous mode when SMR.MP = 1. 0: Normal reception 1: When data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags RDRF, ORER, and FER in SSR to 1 and the status flags SYER, PFER, and SBER in MESR are disabled. When data with the multi-processor bit set to 1 is received, the MPIE bit is automatically set to 0, and normal reception is resumed.	R/W ^{*3}
4	RE	Receive Enable 0: Disable serial reception 1: Enable serial reception	R/W ^{*2}
5	TE	Transmit Enable 0: Disable serial transmission 1: Enable serial transmission	R/W ^{*2}
6	RIE	Receive Interrupt Enable 0: Disable SCIn_RXI and SCIn_ERI interrupt requests 1: Enable SCIn_RXI and SCIn_ERI interrupt requests	R/W
7	TIE	Transmit Interrupt Enable 0: Disable SCIn_TXI interrupt requests 1: Enable SCIn_TXI interrupt requests	R/W

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0, when the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written to TE and RE. When the SMR.CM bit is 0 and the SIMR1.IICM bit is 0, writing is enabled under any condition.

Note 3. When writing a new value to a bit other than the MPIE bit of this register in multi-processor mode (SMR.MP bit = 1), write 0 to the MPIE bit using the store instruction to avoid accidentally setting the MPIE bit to 1 by a read-modify-write operation when using a bit manipulation instruction.

The SCR register controls operation and clock source selection for transmission and reception.

CKE[1:0] bits (Clock Enable)

The CKE[1:0] bits select the clock source and the SCKn pin function.

TEIE bit (Transmit End Interrupt Enable)

The TEIE bit enables or disables SCIn_TEI interrupt requests. Set TEIE to 0 to disable an SCIn_TEI interrupt request.

In simple IIC mode, SCIn_TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STIn). In this case, the TEIE bit can be used to enable or disable the STI.

MPIE bit (Multi-Processor Interrupt Enable)

When the MPIE bit is set to 1 and data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags RDRF, ORER, FER, RDF, and DR in SSR/SSR_FIFO to 1 is disabled. When data with the multi-processor bit set to 1 is received, the MPIE bit is automatically set to 0, and normal reception resumes. For details, see [section 32.4. Multi-Processor Communication Function](#).

When the MPB bit in the SSR register is 0, the receive data is not transferred from the RSR register to the RDR register, a receive error is not detected, and setting the flags ORER and FER to 1 is disabled.

When the MPB bit is set to 1, the MPIE bit is automatically set to 0, SCIn_RXI and SCIn_ERI interrupt requests are enabled (if the RIE bit in SCR is set to 1), and setting of the ORER and FER flags to 1 is enabled.

Set MPIE to 0 if the multi-processor communications function is not used.

RE bit (Receive Enable)

The RE bit enables or disables serial reception. When the RE bit is set to 1, serial reception starts by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Set the reception format in the SMR register before setting the RE bit to 1.

In non-FIFO operation, when reception is halted by setting the RE bit to 0, the RDRF, ORER, FER, and PER flags in the SSR register are not affected, and the previous values are retained.

When FIFO operation is selected and reception is halted by setting the RE bit to 0, the RDF, ORER, FER, PER, and DR flags in SSR_FIFO are not affected and the previous values are retained.

TE bit (Transmit Enable)

The TE bit enables or disables serial transmission.

When the TE bit is set to 1, serial transmission is started by writing transmit data to the TDR register. Set the transmission format in the SMR register before setting the TE bit to 1.

RIE bit (Receive Interrupt Enable)

The RIE bit enables or disables SCIn_RXI and SCIn_ERI interrupt requests.

SCIn_RXI and SCIn_ERI interrupt requests are disabled by setting the RIE bit to 0.

An SCIn_ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in SSR/SSR_FIFO then setting the flag to 0, or by setting the RIE bit to 0.

TIE bit (Transmit Interrupt Enable)

The TIE bit enables or disables SCIn_TXI interrupt requests. SCIn_TXI interrupt requests are disabled by setting the TIE bit to 0.

Note: To switch the TIE bit value from 0 to 1 in FIFO mode, set the TIE and TE bits to 1 simultaneously or set the TIE bit to 1 when TE = 1. When TE = 0 in FIFO mode, setting the TIE bit to 1 is prohibited.

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

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
1:0	CKE[1:0]	Clock Enable 0 0: When SMR_SMCI.GM = 0: Disable output The SCKn pin is available for use as an I/O port if set up in the I/O port settings When SMR_SMCI.GM = 1: Fix output low 0 1: When SMR_SMCI.GM = 0: Output clock When SMR_SMCI.GM = 1: Output clock 1 0: When SMR_SMCI.GM = 0: Setting prohibited When SMR_SMCI.GM = 1: Fix output high 1 1: When SMR_SMCI.GM = 0: Setting prohibited When SMR_SMCI.GM = 1: Output clock	R/W ^{*1}
2	TEIE	Transmit End Interrupt Enable Set this bit to 0 in smart card interface mode	R/W
3	MPIE	Multi-Processor Interrupt Enable Set this bit to 0 in smart card interface mode	R/W
4	RE	Receive Enable 0: Disable serial reception 1: Enable serial reception	R/W ^{*2}
5	TE	Transmit Enable 0: Disable serial transmission 1: Enable serial transmission	R/W ^{*2}
6	RIE	Receive Interrupt Enable 0: Disable SCIn_RXI and SCIn_ERI interrupt requests 1: Enable SCIn_RXI and SCIn_ERI interrupt requests	R/W
7	TIE	Transmit Interrupt Enable 0: Disable SCIn_TXI interrupt requests 1: Enable SCIn_TXI interrupt requests	R/W

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written to TE and RE.

The SCR_SMCI register sets transmission and reception control, interrupt control, and clock source selection for transmission and reception.

For details on interrupt requests, see [section 32.10. 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 32.6.8. Clock Output Control](#).

TEIE bit (Transmit End Interrupt Enable)

Set the TEIE bit to 0 in smart card interface mode.

MPIE bit (Multi-Processor Interrupt Enable)

Set the MPIE bit to 0 in smart card interface mode.

RE bit (Receive Enable)

The RE bit enables or disables serial reception. When the RE bit is set to 1, serial reception starts by detecting the start bit. Set the reception format in the SMR_SMCI register before setting the RE bit to 1.

If reception is halted by setting the RE bit to 0, the ORER, FER, and PER flags in SSR_SMCI are not affected and the previous values are retained.

TE bit (Transmit Enable)

The TE bit enables or disables serial transmission. When the TE bit is set to 1, serial transmission is started by writing transmit data to TDR. Set the transmission format in the SMR_SMCI register before setting the TE bit to 1.

RIE bit (Receive Interrupt Enable)

The RIE bit enables or disables SCIn_RXI and SCIn_ERI interrupt requests.

SCIn_RXI and SCIn_ERI interrupt requests are disabled by setting the RIE bit to 0.

An SCIn_ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR_SMCI register, and then setting the flag to 0, or by setting the RIE bit to 0.

TIE bit (Transmit Interrupt Enable)

The TIE bit enables or disables SCIn_TXI interrupt requests. SCIn_TXI interrupt requests are disabled by setting the TIE bit to 0.

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

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT

Value after reset: 1 0 0 0 0 1 0 0

Bit	Symbol	Function	R/W
0	MPBT	Multi-Processor Bit Transfer Sets the value of the multi-processor bit in the transmission frame. 0: Data transmission cycle 1: ID transmission cycle	R/W
1	MPB	Multi-Processor Value of the multi-processor bit in the reception frame. 0: Data transmission cycle 1: ID transmission cycle	R
2	TEND	Transmit End Flag 0: A character is being transmitted 1: Character transfer is complete	R
3	PER	Parity Error Flag 0: No parity error occurred 1: Parity error occurred	R/(W) ^{*1}
4	FER	Framing Error Flag 0: No framing error occurred 1: Framing error occurred	R/(W) ^{*1}
5	ORER	Overrun Error Flag 0: No overrun error occurred 1: Overrun error occurred	R/(W) ^{*1}
6	RDRF	Receive Data Full Flag 0: No received data in RDR register 1: Received data in RDR register	R/(W) ^{*1}
7	TDRE	Transmit Data Empty Flag 0: Transmit data in TDR register 1: No transmit data in TDR register	R/(W) ^{*1}

Note 1. Only 0 can be written to clear the flag after reading 1.

The SSR register provides SCI status flags and transmission and reception multi-processor bits.

MPBT bit (Multi-Processor Bit Transfer)

The MPBT bit sets the value of the multi-processor bit in the transmit frame.

MPB bit (Multi-Processor)

The MPB bit holds the value of the multi-processor bit in the reception frame. This bit does not change when the SCR.RE bit is 0.

TEND flag (Transmit End Flag)

The TEND flag indicates completion of transmission.

[Setting conditions]

- When the SCR.TE bit is set to 0 (serial transmission is disabled) and the FCR.FM bit is set to 0 (non-FIFO selected). When the SCR.TE bit is set to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated on transmission of the tail-end bit of a character being transmitted.

[Clearing conditions]

- When transmit data is written to the TDR register while the SCR.TE bit is 1
- When 0 is written to TDRE after reading TDRE = 1 while the SCR.TE bit is 1

PER flag (Parity Error Flag)

The PER flag indicates that a parity error occurred during reception in asynchronous mode and the reception ended abnormally.

[Setting condition]

- When a parity error is detected during reception in asynchronous mode when the address match function is disabled (DCCR.DCME = 0).

Although receive data is transferred to the RDR register when the parity error occurs, no SCIn_RXI interrupt request occurs. When the PER flag is set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to PER after reading PER = 1. After writing 0 to the PER flag, read the PER flag to check that it is actually set to 0.

When the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

FER flag (Framing Error Flag)

The FER flag indicates that a framing error occurred during reception in asynchronous mode and the reception ended abnormally.

[Setting condition]

- When 0 is sampled as the stop bit during reception in asynchronous mode when the address match function is disabled (DCCR.DCME = 0).

In 2-stop-bit mode, only the first stop bit is checked. The second stop bit is not checked. Although receive data is transferred to the RDR register when the framing error occurs, no SCIn_RXI interrupt request occurs. When the FER flag is set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to FER after reading FER = 1. After writing 0 to the FER flag, read the FER flag to check that it is actually set to 0.

When the SCR.RE bit is set to 0 (serial reception is disabled), the FER flag is not affected and retains its previous value.

ORER flag (Overrun Error Flag)

The ORER flag indicates that an overrun error occurred during reception and the reception ended abnormally.

[Setting condition]

- When the next data is received before receive data that does not have a parity error and a framing error is read from the RDR register.

The data received before an overrun error occurred is saved in the RDR register, but data received after the error is lost. When the ORER flag is set to 1, receive data is not forwarded to the RDR register. In clock synchronous mode, serial transmission and reception are stopped.

[Clearing condition]

- When 0 is written to ORER after reading ORER = 1. After writing 0 to the ORER flag, read the ORER flag to check that it is actually set to 0.

When the SCR.RE bit is set to 0 (serial reception is disabled), the ORER flag is not affected and retains its previous value.

RDRF flag (Receive Data Full Flag)

The RDRF flag indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing conditions]

- When 0 is written to RDRF after reading RDRF = 1
- When data is forwarded from the RDR register

TDRE flag (Transmit Data Empty Flag)

The TDRE flag indicates the presence of transmit data in the TDR register.

[Setting conditions]

- When the SCR.TE bit is 0
- When data is transmitted from the TDR register to the TSR register

[Clearing conditions]

- When 0 is written to TDRE after reading TDRE = 1
- When the SCR.TE bit is 1 and data is written to the TDR register

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

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TDRE	RDRF	ORER	FER	PER	TEND	—	DR
Value after reset:	1	0	0	0	0	0	x	0

Bit	Symbol	Function	R/W
0	DR	Receive Data Ready Flag 0: Receiving is in progress, or no received data remains in FRDRHL after successfully completed reception (receive FIFO empty) 1: Next receive data is not received for a period after normal receiving is complete, when the amount of data stored in the FIFO is equal to or less than the receive triggering number	R/W ¹
1	—	The read value is undefined. The write value should be 1.	R/W
2	TEND	Transmit End Flag 0: A character is being transmitted 1: Character transfer is complete	R/W ¹
3	PER	Parity Error Flag 0: No parity error occurred 1: Parity error occurred	R/W ¹
4	FER	Framing Error Flag 0: No framing error occurred 1: Framing error occurred	R/W ¹

Bit	Symbol	Function	R/W
5	ORER	Overrun Error Flag 0: No overrun error occurred 1: Overrun error occurred	R/W ^{*1}
6	RDF	Receive FIFO Data Full Flag 0: The amount of receive data written in FRDRHL is less than the specified receive triggering number 1: The amount of receive data written in FRDRHL is equal to or greater than the specified receive triggering number	R/W ^{*1}
7	TDFE	Transmit FIFO Data Empty Flag 0: The amount of transmit data written in FTDRHL exceeds the specified transmit triggering number 1: The amount of transmit data written in FTDRHL is equal to or less than the specified transmit triggering number	R/W ^{*1}

Note 1. Only 0 can be written, to clear the flag after reading 1.

The SSR_FIFO register provides the SCI with FIFO mode status flags.

DR flag (Receive Data Ready Flag)

The DR flag indicates that the amount of data stored in the Receive FIFO Data Register (FRDRHL) falls below the specified receive triggering number, and that no next data is received after 15 ETUs (elementary time units) from the last stop bit in asynchronous mode. This flag is valid only in asynchronous mode, including multi-processor mode, when FIFO operation is selected.

In clock synchronous mode, the DR flag is not set to 1.

[Setting condition]

- When FRDRHL contains less data than the specified receive triggering number, and no next data is received after 15 ETUs^{*1} from the last stop bit, and the SSR_FIFO.FER and SSR_FIFO.PER flags are 0.

[Clearing conditions]

- When 1 is read from DR, after all received data is read
- When the FCR.FM bit is changed from 0 to 1

Note 1. This is equivalent to 1.5 frames in the 8-bit format with one stop bit.

The DR flag is only set to 1 when FIFO is selected in asynchronous mode, including multi-processor mode. It is not set to 1 in other operation modes.

TEND flag (Transmit End Flag)

The TEND flag indicates that FTDRHL does not contain valid data when transmitting the last bit of a serial character, so the transmission is halted.

[Setting condition]

- When FTDRHL does not contain transmit data when the last bit of a 1-byte serial character is transmitted.

[Clearing conditions]

- When transmit data is written to FTDRHL^{*1} while the SCR.TE bit is 1
- When 0 is written to TEND after 1 is read from TEND, when the SCR.TE bit is 1
- When the FCR.FM bit is changed from 0 to 1

Note 1. Do not use the TEND bit as a transmit end flag when the DTC writes data to FTDRHL in response to an SCIn_TXI interrupt request.

PER flag (Parity Error Flag)

The PER flag indicates whether there is a parity error in the data read from the FRDRHL register in asynchronous mode when the address match function is disabled (DCCR.DCME = 0).

[Setting condition]

- When data is received and a parity error is detected, when the address match function is disabled (DCCR.DCME = 0).

[Clearing condition]

- When 0 is written to PER after reading PER = 1.

The reception operation is continuous, and the receive data is stored in the FRDRHL register, even when a parity error occurs during reception.

When the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

FER flag (Framing Error Flag)

The FER flag indicates whether there is a framing error in the data read from the FRDRHL register in asynchronous mode when the address match function is disabled (DCCR.DCME = 0).

[Setting condition]

- When 0 is sampled as the stop bit during reception when the address match function is disabled (DCCR.DCME = 0).

[Clearing condition]

- When 0 is written to FER after reading FER = 1.

The reception operation is continuous, and the receive data is stored in the FRDRHL register, even when a framing error occurs during reception.

When the SCR.RE bit is set to 0 (serial reception is disabled), the FER flag is not affected and retains its previous value.

ORER flag (Overrun Error Flag)

The ORER flag indicates that the receive operation stopped abnormally because an overrun error occurred.

[Setting condition]

- When the next serial reception completes while the receive FIFO is full with 16-byte receive data.

[Clearing condition]

- When 0 is written to ORER after reading ORER = 1.

When the SCR.RE bit is set to 0 (serial reception is disabled), the ORER flag is not affected and retains its previous value.

RDF flag (Receive FIFO Data Full Flag)

The RDF flag indicates that receive data was transferred to the FRDRHL register, and the amount of data in FRDRHL is equal to or exceeds the specified receive triggering number. When RTRG is set to 0, the RDF flag is not set even when the amount of data in the receive FIFO is equal to 0.

[Setting condition]

- When the amount of receive data equal to or greater than the specified receive triggering number is stored in FRDRHL,^{*1} and the FIFO is not empty.

[Clearing conditions]

- When 0 is written to RDF after reading RDF = 1
- When FRDRHL is read by the DTC, but only when the block transfer is the last transmission
- When the setting and clearing conditions occur at the same time, the RDF bit is set to 0. After that, when the amount of data stored in the FRDRHL register is the same as or greater than the RTRG value, RDF is set to 1 after 1 PCLK.

Note 1. Because FRDRHL is a 16-stage FIFO register, the maximum amount of data that can be read when RDF is 1 is equivalent to the specified receive triggering number. If an attempt is made to read after all the data in FRDRHL is read, the data is undefined.

TDFF flag (Transmit FIFO Data Empty Flag)

The TDFF flag indicates that data is transferred from the FTDRHL register into the TSR register, the amount of data in FTDRHL is below the specified transmit triggering number, and writing of transmit data to FTDRHL is enabled.

[Setting conditions]

- When the TE bit in SCR is 0
- When the amount of transmit data written in FTDRHL is equal to or less than the specified transmit triggering number^{*1}

[Clearing conditions]

- When writing to FTDRHL is executed on the last transmission while the DTC is activated
- When 0 is written to the TDFE flag after reading TDFE = 1.^{*2}
The setting conditions are given priority when TE = 0. When the setting condition and clearing condition occur at the same time, the TDFE flag is set to 0. After that, when the amount of data stored in the FTDRHL register is equal to or less than the TTRG value, TDFE is set to 1 after 1 PCLK.

Note 1. Because the FTDRHL register is a 16-stage FIFO register, when the TDFE flag is 1, the maximum amount of data that can be written to the FTDRHL register is 16 minus FDR.T[4:0] bytes. If more data is written, data is discarded.

Note 2. Do not clear the TDFE flag during block transfer processing by the DTC.

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

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT

Value after reset: 1 0 0 0 0 1 0 0

Bit	Symbol	Function	R/W
0	MPBT	Multi-Processor Bit Transfer Set this bit to 0 in smart card interface mode	R/W
1	MPB	Multi-Processor Set this bit to 0 in smart card interface mode	R
2	TEND	Transmit End Flag 0: A character is being transmitted 1: Character transfer is complete	R
3	PER	Parity Error Flag 0: No parity error occurred 1: Parity error occurred	R/W ^{*1}
4	ERS	Error Signal Status Flag 0: No low error signal response 1: Low error signal response occurred	R/W ^{*1}
5	ORER	Overrun Error Flag 0: No overrun error occurred 1: Overrun error occurred	R/W ^{*1}
6	RDRF	Receive Data Full Flag 0: No received data in RDR register 1: Received data in RDR register	R/W ^{*1}
7	TDRE	Transmit Data Empty Flag 0: Transmit data in TDR register 1: No transmit data in TDR register	R/W ^{*1}

Note 1. Only 0 can be written, to clear the flag after reading 1.

The SSR_SMCI register provides the SCI with smart card interface mode status flags.

TEND flag (Transmit End Flag)

When there is no error signal from the receiving side, the TEND flag is set to 1 when more data for transfer is ready to be transferred to the TDR register.

[Setting conditions]

- When the SCR_SMCI.TE bit = 0 (serial transmission is disabled).
When the SCR_SMCI.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period elapses after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated.

The set timing is determined by the following register settings:

- When SMR_SMCI.GM = 0 and SMR_SMCI.BLK = 0, 12.5 ETUs after the start of transmission
- When SMR_SMCI.GM = 0 and SMR_SMCI.BLK = 1, 11.5 ETUs after the start of transmission
- When SMR_SMCI.GM = 1 and SMR_SMCI.BLK = 0, 11.0 ETUs after the start of transmission
- When SMR_SMCI.GM = 1 and SMR_SMCI.BLK = 1, 11.0 ETUs after the start of transmission

[Clearing conditions]

- When transmit data is written to the TDR register while the SCR_SMCI.TE bit is 1
- When 0 is written to TDRE after reading TDRE = 1 while the SCR_SMCI.TE bit is 1

PER flag (Parity Error Flag)

The PER flag indicates that a parity error occurred during reception in asynchronous mode and the reception ended abnormally.

[Setting condition]

- When a parity error is detected during reception. Although receive data is transferred to RDR when a parity error occurs, no SCIn_RXI interrupt request occurs. After the PER flag is set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1. After writing 0 to the PER flag, read the flag to check that it is actually set to 0.

When the RE bit in SCR_SMCI is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

ERS flag (Error Signal Status Flag)**[Setting condition]**

- When a low error signal is sampled.

[Clearing condition]

- When 0 is written to ERS after reading ERS = 1.

ORER flag (Overrun Error Flag)

The ORER flag indicates that an overrun error occurred during reception and the reception ended abnormally.

[Setting condition]

- When the next data is received before receive data that does not have a parity error is read from the RDR register. The data received before an overrun error occurred is saved in the RDR, but data received after the error is lost. When the ORER flag is set to 1, receive data is not forwarded to the RDR register.

[Clearing condition]

- When 0 is written to ORER after reading ORER = 1. After writing 0 to the ORER flag, read the flag to check that it is actually set to 0.

When the RE bit in SCR_SMCI is set to 0, the ORER flag is not affected and retains its previous value.

RDRF flag (Receive Data Full Flag)

The RDRF flag indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing conditions]

- When 0 is written to RDRF after reading RDRF = 1
- When data is forwarded from the RDR register

TDRE flag (Transmit Data Empty Flag)

The TDRE flag indicates the presence of transmit data in the TDR register.

[Setting conditions]

- When the SCR_SMCI.TE bit is 0
- When data is transmitted from the TDR register to the TSR register

[Clearing conditions]

- When 0 is written to TDRE after reading TDRE = 1
- When the SCR_SMCI.TE bit is 1 and data is written to the TDR register

32.2.16 SCMR : Smart Card Mode Register

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BCP2	—	—	CHR1	SDIR	SINV	—	SMIF

Value after reset: 1 1 1 1 0 0 1 0

Bit	Symbol	Function	R/W
0	SMIF	Smart Card Interface Mode Select 0: Non-smart card interface mode (asynchronous mode, clock synchronous mode, simple SPI mode, or simple IIC mode) 1: Smart card interface mode	R/W ^{*1}
1	—	This bit is read as 1. The write value should be 1.	R/W
2	SINV	Transmitted/Received Data Invert Set the SINV bit to 0 for operation in simple IIC mode. The SINV bit can be used in the following modes: • Smart card interface mode • Asynchronous mode (including multi-processor mode) • Clock synchronous mode • Simple SPI mode 0: TDR contents are transmitted as they are. Received data is stored as received in the RDR register. 1: TDR register contents are inverted before transmission. Receive data is stored in inverted form in the RDR register.	R/W ^{*1}

Bit	Symbol	Function	R/W
3	SDIR	Transmitted/Received Data Transfer Direction Set the SDIR bit to 1 for operation in simple IIC mode. The SDIR bit can be used in the following modes: • Smart card interface mode • Asynchronous mode (including multi-processor mode) • Clock synchronous mode • Simple SPI mode 0: Transfer LSB-first 1: Transfer MSB-first	R/W ^{*1}
4	CHR1	Character Length 1 Valid only in asynchronous mode. ^{*2} Selects the transmit/receive character length in combination with the SMR.CHR bit. 0: SMR.CHR = 0: Transmit/receive in 9-bit data length SMR.CHR = 1: Transmit/receive in 9-bit data length 1: SMR.CHR = 0: Transmit/receive in 8-bit data length (initial value) SMR.CHR = 1: Transmit/receive in 7-bit data length ^{*3}	R/W ^{*1}
6:5	—	These bits are read as 1. The write value should be 1.	R/W
7	BCP2	Base Clock Pulse 2 Selects the number of base clock cycles in combination with the SMR_SMCI.BCP[1:0] bits. Table 32.4 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 must be selected and the value of the MSB (bit [7]) in TDR cannot be transmitted.

The register selects the smart card interface and communication format.

SMIF bit (Smart Card Interface Mode Select)

Setting the SMIF bit to 1 selects smart card interface mode. Setting it to 0 selects all other modes:

- Asynchronous mode, including multi-processor mode
- Clock synchronous mode
- Simple SPI mode
- Simple IIC mode

SINV bit (Transmitted/Received Data Invert)

The SINV bit inverts the transmit and receive data logic level. It does not affect the logic level of the parity bit. To invert the parity bit, invert the PM bit in SMR or SMR_SMCI.

CHR1 bit (Character Length 1)

The CHR1 bit selects the data length of transmit and receive data in combination with the CHR bit in the SMR register. A fixed data length of 8 bits is used in modes other than asynchronous mode.

BCP2 bit (Base Clock Pulse 2)

The BCP2 bit selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR_SMCI.BCP[1:0] bits.

Table 32.4 Combinations of the SCMR.BCP2 and SMR_SMCI.BCP[1:0] bits (1 of 2)

SCMR.BCP2 bit	SMR_SMCI.BCP[1:0] bits	Number of base clock cycles for 1-bit transfer period
0	00b	93 clock cycles (S = 93) ^{*1}
0	01b	128 clock cycles (S = 128) ^{*1}
0	10b	186 clock cycles (S = 186) ^{*1}
0	11b	512 clock cycles (S = 512) ^{*1}
1	00b	32 clock cycles (S = 32) (Initial Value) ^{*1}
1	01b	64 clock cycles (S = 64) ^{*1}

Table 32.4 Combinations of the SCMR.BCP2 and SMR_SMCI.BCP[1:0] bits (2 of 2)

SCMR.BCP2 bit	SMR_SMCI.BCP[1:0] bits	Number of base clock cycles for 1-bit transfer period
1	10b	372 clock cycles (S = 372) ^{*1}
1	11b	256 clock cycles (S = 256) ^{*1}

Note 1. S is the value of S in BRR : Bit Rate Register.

32.2.17 BRR : Bit Rate Register

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
 SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x01

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

BRR is an 8-bit register that adjusts the bit rate.

As each SCI channel has independent baud rate generator control, different bit rates can be set for each channel. Table 32.5 shows the relationship between the setting (N) in the BRR and the bit rate (B) for normal asynchronous mode, multi-processor transfer, clock synchronous mode, smart card interface mode, simple SPI mode, and simple IIC mode.

The initial value of the BRR register is 0xFF. The BRR register can be read by the CPU, but it can be written to only when the TE and RE bits in SCR/SCR_SMCI are 0.

In the simple I2C mode, when selecting 16 clocks of the basic clock as one bit period by SEMR register, only BRR = 0x00 can be set

Table 32.5 Relationship between N setting in BRR and bit rate B

Mode	SEMR settings			BRR setting	Error
	BGDM bit	ABCS bit	ABCS E bit		
Asynchronous, multi-processor transfer	0	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	0	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	Don't care	Don't care	1	$N = \frac{PCLK \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{PCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI				$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	—
Smart card interface				$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$
Simple IIC ^{*1}	Don't care	0	Don't care	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	—
	Don't care	1	Don't care	N = 0x00	—

Note: B: Bit rate (bps)
 N: BRR setting for on-chip baud rate generator (0 ≤ N ≤ 255)
 PCLK: Operating frequency (MHz)

n and S: Determined by the SMR/SMR_SMCI and SCMR register settings as listed in [Table 32.7](#) and [Table 32.8](#).

Note 1. Adjust the bit rate so that the widths of high and low level of the SSCLn output in simple IIC mode satisfy the I²C bus standard.

Table 32.6 Calculating widths of SSCLn high and low levels

Mode	SSCLn	ABCS bit	Formula (result in seconds)
Simple IIC	Width at high level (minimum value)	0	$(N + 1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLK \times 10^6}$
	Width at low level (minimum value)		$(N + 1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLK \times 10^6}$
	Width at high level (minimum value)	1	$8 \times \frac{1}{PCLK \times 10^6}$
	Width at low level (minimum value)		$8 \times \frac{1}{PCLK \times 10^6}$

Table 32.7 Clock source settings

SMR or SMR_SMCI.CKS[1:0] bits setting	Clock source	n
CKS[1:0] bits		
00b	PCLK clock	0
01b	PCLK/4 clock	1
10b	PCLK/16 clock	2
11b	PCLK/64 clock	3

Table 32.8 Base clock settings in smart card interface mode

SCMR.BCP2 bit setting	SMR_SMCI.BCP[1:0] bits setting	Base clock cycles for 1-bit period	S
BCP2 bit	BCP[1:0] bits		
0	00b	93 clock cycles	93
0	01b	128 clock cycles	128
0	10b	186 clock cycles	186
0	11b	512 clock cycles	512
1	00b	32 clock cycles	32
1	01b	64 clock cycles	64
1	10b	372 clock cycles	372
1	11b	256 clock cycles	256

[Table 32.9](#) and [Table 32.10](#) list examples of BRR (N) settings in normal asynchronous mode. [Table 32.11](#) lists the maximum bit rate settable for each operating frequency. [Table 32.15](#) lists examples of BRR (N) settings in smart card interface mode. [Table 32.17](#) lists examples of lists examples of BRR (N) settings in simple I2C mode (N) settings in simple I2C 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 32.6.4. Receive Data Sampling Timing and Reception Margin](#). [Table 32.12](#) and [Table 32.14](#) list the maximum bit rates with external clock input.

When either the Asynchronous Mode Base Clock Select bit (ABCS) or the Baud Rate Generator Double-speed Mode Select bit (BGDM) in the Serial Extended Mode Register (SEMR) is set to 1 in asynchronous mode, the bit rate becomes twice the value listed in [Table 32.16](#). When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 32.9 Examples of BRR settings for different bit rates in asynchronous mode (1) (1 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	-0.26	2	177	-0.25	2	212	0.03	2	217	0.08

Table 32.9 Examples of BRR settings for different bit rates in asynchronous mode (1) (2 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
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

Table 32.9 Examples of BRR settings for different bit rates in asynchronous mode (1) (3 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	-0.17	3	70	0.03	3	75	0.48	3	79	-0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0.00	2	233	0.16	2	255	0.00
300	2	90	0.16	2	103	0.16	2	111	0.00	2	116	0.16	2	127	0.00
600	1	181	0.16	1	207	0.16	1	223	0.00	1	233	0.16	1	255	0.00
1200	1	90	0.16	1	103	0.16	1	111	0.00	1	116	0.16	1	127	0.00
2400	0	181	0.16	0	207	0.16	0	223	0.00	0	233	0.16	0	255	0.00
4800	0	90	0.16	0	103	0.16	0	111	0.00	0	116	0.16	0	127	0.00
9600	0	45	-0.93	0	51	0.16	0	55	0.00	0	58	-0.69	0	63	0.00
19200	0	22	-0.93	0	25	0.16	0	27	0.00	0	28	1.02	0	31	0.00
31250	0	13	0.00	0	15	0.00	0	16	1.20	0	17	0.00	0	19	-1.70
38400	—	—	—	0	12	0.16	0	13	0.00	0	14	-2.34	0	15	0.00

Note: In this example, SEMR.ABCS = 0, SEMR.ABCSE = 0, and SEMR.BGDM = 0.
 When either the ABCS or BGDM bit is set to 1, the bit rate doubles.
 When both ABCS and BGDM are set to 1, the bit rate increases four times.

Table 32.10 Examples of BRR settings for different bit rates in asynchronous mode (2) (1 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	20			25			30			33*1			40*1		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	-0.25	3	110	-0.02	3	132	0.13	3	145	0.33	3	177	-0.25
150	3	64	0.16	3	80	0.47	3	97	-0.35	3	106	0.39	3	129	0.16
300	2	129	0.16	2	162	-0.15	2	194	0.16	2	214	-0.07	3	64	0.16
600	2	64	0.16	2	80	0.47	2	97	-0.35	2	106	0.39	2	129	0.16
1200	1	129	0.16	1	162	-0.15	1	194	0.16	1	214	-0.07	2	64	0.16
2400	1	64	0.16	1	80	0.47	1	97	-0.35	1	106	0.39	1	129	0.16

Table 32.10 Examples of BRR settings for different bit rates in asynchronous mode (2) (2 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	20			25			30			33*1			40*1		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
4800	0	129	0.16	0	162	-0.15	0	194	0.16	0	214	-0.07	1	64	0.16
9600	0	64	0.16	0	80	0.47	0	97	-0.35	0	106	0.39	0	129	0.16
19200	0	32	-1.36	0	40	-0.76	0	48	-0.35	0	53	-0.54	0	64	0.16
31250	0	19	0.00	0	24	0.00	0	29	0.00	0	32	0.00	0	39	0.00
38400	0	15	1.73	0	19	1.73	0	23	1.73	0	26	-0.54	0	32	-1.36

Table 32.10 Examples of BRR settings for different bit rates in asynchronous mode (2) (3 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)					
	50*1			60*1		
	n	N	Error (%)	n	N	Error (%)
110	3	221	-0.02	—	—	—
150	3	162	-0.15	3	194	0.16
300	3	80	0.47	3	97	-0.35
600	2	162	-0.15	2	194	0.16
1200	2	80	0.47	2	97	-0.35
2400	1	162	-0.15	1	194	0.16
4800	1	80	0.47	1	97	-0.35
9600	0	162	-0.15	0	194	0.16
19200	0	80	0.47	0	97	-0.35
31250	0	49	0.00	0	59	0.00
38400	0	40	-0.76	0	48	-0.35

Note: In this example, SEMR.ABCS = 0, SEMR.ABCSE = 0, and SEMR.BGDM = 0. When either ABCS or BGDM bit is set to 1, the bit rate doubles. When both ABCS and BGDM bits are set to 1, the bit rate quadruples.

Note 1. SCI0, 1 only

Table 32.11 Maximum bit rate for each operating frequency in asynchronous mode (1 of 2)

PCLK (MHz)	SEMR register settings					Maximum bit rate (bps)	PCLK (MHz)	SEMR register 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	16	0	0	0	0	0	500000
		1	0	0	0	500000			1	0	0	0	1000000
	1	0	0	0	0	1000000		1	0	0	0	0	2000000
		1	0	0	0	1333333			1	0	0	0	2666666
	Don't care	Don't care	1	0	0	1333333		Don't care	Don't care	1	0	0	2666666
9.8304	0	0	0	0	0	307200	17.2032	0	0	0	0	0	537600
		1	0	0	0	614400			1	0	0	0	1075200
	1	0	0	0	0	1228800		1	0	0	0	0	2150400
		1	0	0	0	1638400			1	0	0	0	2867200
	Don't care	Don't care	1	0	0	1638400		Don't care	Don't care	1	0	0	2867200

Table 32.11 Maximum bit rate for each operating frequency in asynchronous mode (2 of 2)

PCLK (MHz)	SEMR register settings					Maximum bit rate (bps)	PCLK (MHz)	SEMR register settings					Maximum bit rate (bps)
	BGDM bit	ABCS bit	ABCSE bit	n	N			BGDM bit	ABCS bit	ABCSE bit	n	N	
10	0	0	0	0	0	312500	18	0	0	0	0	0	562500
		1	0	0	0	625000			1	0	0	0	1125000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1250000			1	0	0	0	2250000
	Don't care	Don't care	1	0	0	1666666		Don't care	Don't care	1	0	0	3000000
12	0	0	0	0	0	375000	19.6608	0	0	0	0	0	614400
		1	0	0	0	750000			1	0	0	0	1228800
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1500000			1	0	0	0	2457600
	Don't care	Don't care	1	0	0	2000000		Don't care	Don't care	1	0	0	3276800
12.288	0	0	0	0	0	384000	20	0	0	0	0	0	625000
		1	0	0	0	768000			1	0	0	0	1250000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1536000			1	0	0	0	2500000
	Don't care	Don't care	1	0	0	2048000		Don't care	Don't care	1	0	0	3333333
14	0	0	0	0	0	437500	25	0	0	0	0	0	781250
		1	0	0	0	875000			1	0	0	0	1562500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1750000			1	0	0	0	3125000
	Don't care	Don't care	1	0	0	2333333		Don't care	Don't care	1	0	0	4166666
30	0	0	0	0	0	937500	50 ^{*1}	0	0	0	0	0	1562500
		1	0	0	0	1875000			1	0	0	0	3125000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	3750000			1	0	0	0	6250000
	Don't care	Don't care	1	0	0	5000000		Don't care	Don't care	1	0	0	8333333
33 ^{*1}	0	0	0	0	0	1031250	60 ^{*1}	0	0	0	0	0	1875000
		1	0	0	0	2062500			1	0	0	0	3750000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	4125000			1	0	0	0	7500000
	Don't care	Don't care	1	0	0	5500000		Don't care	Don't care	1	0	0	10000000
40 ^{*1}	0	0	0	0	0	1250000							
		1	0	0	0	2500000							
	1	0	0	0	0								
		1	0	0	0	5000000							
	Don't care	Don't care	1	0	0	6666666							

Note 1. SCI0, 1 only

Table 32.12 Maximum bit rate with external clock input in asynchronous mode

PCLK (MHz)	External input clock (MHz)	Maximum bit rate (bps)	
		SEMR.ABCS bit = 0	SEMR.ABCS bit = 1
8	2.0000	125000	250000
9.8304	2.4576	153600	307200
10	2.5000	156250	312500
12	3.0000	187500	375000
12.288	3.0720	192000	384000
14	3.5000	218750	437500
16	4.0000	250000	500000
17.2032	4.3008	268800	537600
18	4.5000	281250	562500
19.6608	4.9152	307200	614400
20	5.0000	312500	625000
25	6.2500	390625	781250
30	7.5000	468750	937500
33 ^{*1}	8.2500	515625	1031250
40 ^{*1}	10.0000	625000	1250000
50 ^{*1}	12.5000	781250	1562500
60 ^{*1}	15.0000	937500	1875000

Note 1. SCI0, 1 only

Table 32.13 Examples of BRR settings for different bit rates in clock synchronous and simple SPI modes

Bit rate (bps)	Operating frequency PCLK (MHz)																			
	8		10		16		20		25		30		33 ^{*1}		40 ^{*1}		50 ^{*1}		60 ^{*1}	
	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110																				
250	3	124	—	—	3	249														
500	2	249	—	—	3	124	—	—			3	233								
1k	2	124	—	—	2	249	—	—	3	97	3	116	3	128	3	155	3	194	3	233
2.5k	1	199	1	249	2	99	2	124	2	155	2	187	2	205	2	249	3	77	3	93
5k	1	99	1	124	1	199	1	249	2	77	2	93	2	102	2	124	2	155	3	46
10k	0	199	0	249	1	99	1	124	1	155	1	187	1	205	1	249	2	77	2	93
25k	0	79	0	99	0	159	0	199	0	249	1	74	1	82	1	99	1	124	1	149
50k	0	39	0	49	0	79	0	99	0	124	0	149	0	164	1	49	1	61	1	74
100k	0	19	0	24	0	39	0	49	0	62	0	74	0	82	0	99	0	124	0	149
250k	0	7	0	9	0	15	0	19	0	24	0	29	0	32	0	39	0	49	0	59
500k	0	3	0	4	0	7	0	9	—	—	0	14	—	—	0	19	0	24	0	29
1M	0	1			0	3	0	4	—	—	—	—	—	—	0	9	—	—	0	14
2.5M			0	0 ^{*2}			0	1	—	—	0	2	—	—	0	3	0	4	0	5
5M							0	0 ^{*2}	—	—	—	—	—	—	0	1	—	—	0	2
7.5M											0	0 ^{*2}							0	1

Note: Space: Setting prohibited.
 —: Can be set, but an error will occur.

Note 1. SCI0, 1 only

Note 2. Continuous transmission or reception is not possible. After transmitting/receiving one frame of data, there is an interval of a 1-bit period before starting transmitting/receiving the next frame of data. That is, the output of the synchronization clock stops 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 32.14 Maximum bit rate with external clock input in clock synchronous and simple SPI modes

PCLK (MHz)	External input clock (MHz)	Maximum bit rate (Mbps)
8	1.3333	1.3333333
10	1.6667	1.6666667
12	2.0000	2.0000000
14	2.3333	2.3333333
16	2.6667	2.6666667
18	3.0000	3.0000000
20	3.3333	3.3333333
25	4.1667	4.1666667
30	5.0000	5.0000000
33 ^{*1}	5.5000	5.5000000
40 ^{*1}	6.6667	6.6666667
50 ^{*1}	8.3333	8.3333333
60 ^{*1}	10.0000	10.0000000

Note 1. SCI0, 1 only

Table 32.15 Examples of BRR settings for different bit rates in smart card interface mode, n = 0, S = 372 (1 of 4)

Bit rate (bps)	Operating frequency PCLK (MHz)											
	7.1424			10.00			10.7136			13.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	0	0.00	0	1	-30	0	1	-25	0	1	-8.99

Table 32.15 Examples of BRR settings for different bit rates in smart card interface mode, n = 0, S = 372 (2 of 4)

Bit rate (bps)	Operating frequency PCLK (MHz)											
	14.2848			16.00			18.00			20.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	0.00	0	1	12.01	0	2	-15.99	0	2	-6.66

Table 32.15 Examples of BRR settings for different bit rates in smart card interface mode, n = 0, S = 372 (3 of 4)

Bit rate (bps)	Operating frequency PCLK (MHz)											
	25.00			30.00			33.00 ^{*1}			40.00 ^{*1}		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	3	-12.49	0	3	5.01	0	4	-7.59	0	5	-6.66

Table 32.15 Examples of BRR settings for different bit rates in smart card interface mode, n = 0, S = 372 (4 of 4)

Bit rate (bps)	Operating frequency PCLK (MHz)											
	50.00 ^{*1}						60.00 ^{*1}					
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	6	0.01	0	7	5.01						

Note 1. SCI0, 1 only

Table 32.16 Maximum bit rate for each operating frequency in smart card interface mode, S = 32

PCLK (MHz)	Maximum bit rate (bps)	n	N
10.00	156250	0	0
10.7136	167400	0	0
13.00	203125	0	0
16.00	250000	0	0
18.00	281250	0	0
20.00	312500	0	0
25.00	390625	0	0
30.00	468750	0	0
33.00*1	515625	0	0
40.00*1	625000	0	0
50.00*1	781250	0	0
60.00*1	937500	0	0

Note 1. SCIO, 1 only

Table 32.17 Examples of BRR settings for different bit rates in simple I²C mode (1 of 2)

Bit rate (bps)	Operating frequency PCLK (MHz)											
	2				4				8			
	ABCS	n	N	Error (%)	ABCS	n	N	Error (%)	ABCS	n	N	Error (%)
10 k	0	0	5	4.2	0	0	11	4.2	0	0	24	0
25 k	0	0	2	-16.7	0	0	4	0	0	0	9	0
50 k	0	0	0	25	0	0	1	25	0	0	4	0
100 k*2	1	0	0	25	0	0	0	25	0	0	2	-16.7
125 k	1	0	0	0	0	0	0	0	0	0	1	0
250 k	—	—	—	—	1	0	0	0	0	0	0	0
350 k	—	—	—	—	—	—	—	—	—	—	—	—
400 k*2	—	—	—	—	—	—	—	—	—	—	—	—

Table 32.17 Examples of BRR settings for different bit rates in simple I²C mode (2 of 2)

Bit rate (bps)	Operating frequency PCLK (MHz)											
	16				32				64*1			
	ABCS	n	N	Error (%)	ABCS	n	N	Error (%)	ABCS	n	N	Error (%)
10 k	0	1	12	-3.8	0	1	24	0	0	1	49	0
25 k	0	1	4	0	0	1	9	0	0	0	79	0
50 k	0	1	2	-16.7	0	1	4	0	0	0	39	0
100 k*2	0	0	4	0	0	0	9	0	0	0	19	0
125 k	0	0	3	0	0	0	7	0	0	0	15	0
250 k	0	0	1	0	0	0	3	0	0	0	7	0
350 k	—	—	—	—	0	0	2	-4.8	0	0	5	-4.8
400 k*2	—	—	—	—	0	0	2	-16.7	0	0	4	0

Note 1. SCIO, 1 only.

Note 2. The bit rate of 100 kbps and 400 kbps indicate the set value at which the error is on the minus side.

Table 32.18 Minimum widths at high and low levels for SSCLn at different bit rates in simple I²C mode (1 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)											
	2				4				8			
	ABCS	n	N	Min. widths at SSCLn high/low levels (μs) [Number of cycles]	ABCS	n	N	Min. widths at SSCLn high/low levels (μs) [Number of cycles]	ABCS	n	N	Min. widths at SSCLn high/low levels (μs) [Number of cycles]
10 k	0	0	5	42.00/48.00 [84/96]	0	0	11	42.00/48.00 [168/192]	0	0	24	43.75/50.00 [350/400]
25 k	0	0	2	21.00/24.00 [42/48]	0	0	4	17.50/20.00 [70/80]	0	0	9	17.50/20.00 [140/160]
50 k	0	0	0	7.00/8.00 [14/16]	0	0	1	7.00/8.00 [28/32]	0	0	4	8.75/10.00 [70/80]
100 k	1	0	0	4.00/4.00 [8/8]	0	0	0	3.50/4.00 [14/16]	0	0	2	5.25/6.00 [42/48]
125 k	1	0	0	4.00/4.00 [8/8]	0	0	0	3.50/4.00 [14/16]	0	0	1	3.50/4.00 [28/32]
250 k	—	—	—	—	1	0	0	2.00/2.00 [8/8]	0	0	0	1.75/2.00 [14/16]
350 k	—	—	—	—	—	—	—	—	—	—	—	—
400 k	—	—	—	—	—	—	—	—	—	—	—	—

Table 32.18 Minimum widths at high and low levels for SSCLn at different bit rates in simple I²C mode (2 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)											
	16				32				64*1			
	ABCS	n	N	Min. widths at SSCLn high/low levels (μs) [Number of cycles]	ABCS	n	N	Min. widths at SSCLn high/low levels (μs) [Number of cycles]	ABCS	n	N	Min. widths at SSCLn high/low levels (μs) [Number of cycles]
10 k	0	1	12	45.50/52.00 [728/832]	0	1	24	43.75/50.00 [1400/1600]	0	1	49	44.75/50.00 [2800/3200]
25 k	0	1	4	17.50/20.00 [280/320]	0	1	9	17.50/20.00 [560/640]	0	0	79	17.50/20.00 [1120/1280]
50 k	0	1	2	10.50/12.00 [168/192]	0	1	4	8.75/10.00 [280/320]	0	0	39	8.75/10.00 [560/640]
100 k	0	0	4	4.38/5.00 [70/80]	0	1	2	5.25/6.00 [168/192]	0	0	19	4.38/5.00 [280/320]
125 k	0	0	3	3.50/4.00 [56/64]	0	0	7	3.50/4.00 [112/128]	0	0	15	3.50/4.00 [224/256]
250 k	0	0	1	1.75/2.00 [28/32]	0	0	3	1.75/2.00 [56/64]	0	0	7	1.75/2.00 [112/128]
350 k	—	—	—	—	—	0	2	1.31/1.50 [42/48]	0	0	5	1.31/1.50 [84/96]

Table 32.18 Minimum widths at high and low levels for SSCLn at different bit rates in simple I²C mode (3 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)											
	16				32				64*1			
	ABCS	n	N	Min. widths at SSCLn high/low levels (μs) [Number of cycles]	ABCS	n	N	Min. widths at SSCLn high/low levels (μs) [Number of cycles]	ABCS	n	N	Min. widths at SSCLn high/low levels (μs) [Number of cycles]
400 k	—	—	—	—	—	0	2	1.31/1.50 [42/48]	0	0	4	1.09/1.25 [70/80]

Note 1. SCI0, 1 only

32.2.18 MDDR : Modulation Duty Register

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
 SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x12

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

MDDR corrects the bit rate adjusted by the BRR register.

When the BRME bit in SEMR is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected using the settings in MDDR (M/256). [Table 32.19](#) shows the relationship between the MDDR setting (M) and the bit rate (B).

The initial value of MDDR is 0xFF. Bit [7] in this register is fixed to 1.

The CPU can read the MDDR register, but this register is only writable when the TE and RE bits in SCR/SCR_SMCI are 0.

Table 32.19 Relationship between MDDR setting (M) and bit rate (B) when bit rate modulation function is used

B: Bit rate (bps)

M: MDDR setting ($128 \leq \text{MDDR} \leq 256$)N: BRR setting for baud rate generator ($0 \leq N \leq 255$)

PCLK: Operating frequency (MHz)

n and S: Determined by the SMR/SMR_SMCI and SCMR register settings as listed in Table 32.7 and Table 32.8 in section 32.2.17. BRR: Bit Rate Register.

x: Don't care

Mode	SEMR settings			BRR setting	Error
	BGDM bit	ABCS bit	ABCSE bit		
Asynchronous multiprocessor transfer	0	0	0	$N = \frac{\text{PCLK} \times 10^6}{64 \times 2^{2n-1} \times \left(\frac{256}{M}\right) \times B} - 1$	$\text{Error}(\%) = \left\{ \frac{\text{PCLK} \times 10^6}{B \times 64 \times 2^{2n-1} \times \left(\frac{256}{M}\right) \times (N+1)} - 1 \right\} \times 100$
	1	0	0	$N = \frac{\text{PCLK} \times 10^6}{32 \times 2^{2n-1} \times \left(\frac{256}{M}\right) \times B} - 1$	$\text{Error}(\%) = \left\{ \frac{\text{PCLK} \times 10^6}{B \times 32 \times 2^{2n-1} \times \left(\frac{256}{M}\right) \times (N+1)} - 1 \right\} \times 100$
	0	1	0	$N = \frac{\text{PCLK} \times 10^6}{32 \times 2^{2n-1} \times \left(\frac{256}{M}\right) \times B} - 1$	$\text{Error}(\%) = \left\{ \frac{\text{PCLK} \times 10^6}{B \times 32 \times 2^{2n-1} \times \left(\frac{256}{M}\right) \times (N+1)} - 1 \right\} \times 100$
	1	1	0	$N = \frac{\text{PCLK} \times 10^6}{16 \times 2^{2n-1} \times \left(\frac{256}{M}\right) \times B} - 1$	$\text{Error}(\%) = \left\{ \frac{\text{PCLK} \times 10^6}{B \times 16 \times 2^{2n-1} \times \left(\frac{256}{M}\right) \times (N+1)} - 1 \right\} \times 100$
	x	x	1	$N = \frac{\text{PCLK} \times 10^6}{12 \times 2^{2n-1} \times \left(\frac{256}{M}\right) \times B} - 1$	$\text{Error}(\%) = \left\{ \frac{\text{PCLK} \times 10^6}{B \times 12 \times 2^{2n-1} \times \left(\frac{256}{M}\right) \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI*1				$N = \frac{\text{PCLK} \times 10^6}{8 \times 2^{2n-1} \times \left(\frac{256}{M}\right) \times B} - 1$	—
Smart card interface				$N = \frac{\text{PCLK} \times 10^6}{S \times 2^{2n+1} \times \left(\frac{256}{M}\right) \times B} - 1$	$\text{Error}(\%) = \left\{ \frac{\text{PCLK} \times 10^6}{B \times S \times 2^{2n+1} \times \left(\frac{256}{M}\right) \times (N+1)} - 1 \right\} \times 100$
Simple IIC*2				$N = \frac{\text{PCLK} \times 10^6}{64 \times 2^{2n-1} \times \left(\frac{256}{M}\right) \times B} - 1$	—

Note 1. Do not use this function in clock synchronous mode or in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).

Note 2. Adjust the bit rate so that the widths at high and low level of the SSCLn output in simple IIC mode satisfy the IIC standard.

Table 32.20 and Table 32.21 list examples of N settings in BRR and M settings in MDDR in normal asynchronous mode.

Table 32.20 Examples of the BRR and MDDR settings for different bit rates in asynchronous mode (1) (1 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	8					9.8304					10				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	5	236	0	0.03	0	7	(256)*1	0	0.00	0	10	173	1	-0.01
57600	0	3	236	0	0.03	0	4	240	0	0.00	0	4	236	0	0.03
115200	0	1	236	0	0.03	0	1	192	0	0.00	0	4	236	1	0.03
230400	0	0	236	0	0.03	0	0	192	0	0.00	0	1	189	1	0.14
460800	0	0	236	1	0.03	0	0	192	1	0.00	0	0	189	1	0.14

Table 32.20 Examples of the BRR and MDDR settings for different bit rates in asynchronous mode (1) (2 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	12					12.288					14				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	8	236	0	0.03	0	9	(256)*1	0	0.00	0	16	191	1	0.00
57600	0	5	236	0	0.03	0	4	192	0	0.00	0	13	236	1	0.03
115200	0	2	236	0	0.03	0	4	192	1	0.00	0	6	236	1	0.03
230400	0	2	236	1	0.03	0	2	230	1	-0.17	0	2	202	1	-0.11
460800	0	0	157	1	-0.18	0	0	154	1	-0.26	0	0	135	1	0.14

Table 32.20 Examples of the BRR and MDDR settings for different bit rates in asynchronous mode (1) (3 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	16					17.2032					18				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	11	236	0	0.03	0	13	(256)*1	0	0.00	0	18	166	1	-0.01
57600	0	7	236	0	0.03	0	6	192	0	0.00	0	18	249	1	-0.01
115200	0	3	236	0	0.03	0	6	192	1	0.00	0	8	236	1	0.03
230400	0	1	236	0	0.03	0	3	219	1	-0.20	0	1	210	0	0.14
460800	0	1	236	1	0.03	0	1	219	1	-0.20	0	0	210	0	0.14

Note 1. In this example, the ABCS and ABCSE bits in the SEMR register are 0. SEMR.BRME = 0 (M = 256) disables the bit rate modulation function.

Table 32.21 Examples of BRR and MDDR settings for different bit rates in asynchronous mode (2) (1 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	19.6608					20					25				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	15	(256)*1	0	0.00	0	10	173	0	-0.01	0	11	151	0	0.00
57600	0	9	240	0	0.00	0	9	236	0	0.03	0	7	151	0	0.00
115200	0	4	240	0	0.00	0	4	236	0	0.03	0	3	151	0	0.00
230400	0	1	192	0	0.00	0	4	236	1	0.03	0	1	151	0	0.00
460800	0	0	192	0	0.00	0	0	189	0	0.14	0	0	151	0	0.00

Table 32.21 Examples of BRR and MDDR settings for different bit rates in asynchronous mode (2) (2 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	30					33*1					40*1				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	36	194	1	0.01	0	14	143	0	0.01	0	21	173	0	-0.01
57600	0	10	173	0	-0.01	0	9	143	0	0.01	0	38	230	1	-0.01
115200	0	10	173	1	-0.01	0	4	143	0	0.01	0	9	236	0	0.03
230400	0	6	220	1	-0.09	0	4	143	1	0.01	0	4	236	0	0.03
460800	0	3	252	1	0.14	0	1	229	0	0.10	0	4	236	1	0.03

Table 32.21 Examples of BRR and MDDR settings for different bit rates in asynchronous mode (2) (3 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)									
	50 ^{*1}					60 ^{*1}				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	23	151	0	0.00	0	36	194	0	0.01
57600	0	15	151	0	0.00	0	21	173	0	-0.01
115200	0	7	151	0	0.00	0	10	173	0	-0.01
230400	0	3	151	0	0.00	0	10	173	1	-0.01
460800	0	1	151	0	0.00	0	6	220	1	-0.09

Note: In this example, the ABCS and ABCSE bits in the SEMR register are 0. SEMR.BRME = 0 (M = 256) disables the bit rate modulation function.

Note 1. SCI0, 1 only

32.2.19 SEMR : Serial Extended Mode Register

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
 SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x07

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RXDE SEL	BGDM	NFEN	ABCS	ABCS E	BRME	—	—

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	BRME ^{*2}	Bit Rate Modulation Enable 0: Disable bit rate modulation function 1: Enable bit rate modulation function	R/W ^{*1}
3	ABCSE	Asynchronous Mode Extended Base Clock Select 1 Valid only in asynchronous mode with SCR.CKE[1] = 0. 0: Clock cycles for 1-bit period determined by combination of the BGDM and ABCS bits in the SEMR register 1: Baud rate is 6 base clock cycles for 1-bit period	R/W ^{*1}
4	ABCS	Basic Clock Selection In details, see Table 32.22 .	R/W ^{*1}
5	NFEN	Digital Noise Filter Function Enable The NFEN bit must be 0 in all other modes. 0: In asynchronous mode: Disable noise cancellation function for RXDn input signal In simple IIC mode: Disable noise cancellation function for SSCLn and SSDAn input signals 1: In asynchronous mode: Enable noise cancellation function for RXDn input signal In simple IIC mode: Enable noise cancellation function for SSCLn and SSDAn input signals	R/W ^{*1}
6	BGDM	Baud Rate Generator Double-Speed Mode Select Valid only in asynchronous mode with SCR.CKE[1] = 0. 0: Output clock from baud rate generator with normal frequency 1: Output clock from baud rate generator with doubled frequency	R/W ^{*1}
7	RXDESEL	Asynchronous Start Bit Edge Detection Select Valid only in asynchronous mode. 0: Detect low level on RXDn pin as start bit 1: Detect falling edge of RXDn pin as start bit	R/W ^{*1}

Note 1. Writable only when the TE and RE bits in SCR/SCR_SMCI are 0 (both serial transmission and reception are disabled).

Note 2. In simple I2C mode, when 16 clocks are selected as one bit period by the basic clock selection bit, set BRME = 0.

The SEMR register selects the clock source for the 1-bit period in asynchronous mode.

BRME bit (Bit Rate Modulation Enable)

The BRME bit enables or disables the bit rate modulation function. The bit rate generated by the on-chip baud rate generator is evenly corrected when this function is enabled.

ABCSE bit (Asynchronous Mode Extended Base Clock Select 1)

The ABCSE bit sets the pulse number for the base clock in a 1-bit period to 6, and the double-frequency clock is output from the baud rate generator. When the bit rate is set to 6 while dividing the bus clock frequency, use this bit and set SMR.CKS[1:0] = 00b and BRR = 0.

Set it to "0" in modes other than asynchronous mode. Even in asynchronous mode, set it to "0" when using external clock.

ABCS bit (Basic Clock Selection)

Select the number of basic clocks in one bit period. See [Table 32.22](#) for details.

Table 32.22 Number of basic clocks in one bit period in each communication mode

Communication mode	ABCS	Content
Asynchronous mode	0	Select 16 basic clocks as 1 bit period
	1	Select 8 basic clocks as 1 bit period
Simple I2C mode	0	Select 32 basic clocks as 1 bit period
	1	Select 16 basic clocks as 1 bit period (only SCI3, 4)

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 SSDAn and SSCLn 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.

BGDM bit (Baud Rate Generator Double-Speed Mode Select)

The BGDM bit selects whether or not to double the base clock frequency output from the baud rate generator.

The BGDM bit is valid when the on-chip baud rate generator is selected as the clock source (SCR.CKE[1] = 0) in asynchronous mode (SMR.CM = 0). When external clock is selected (SCR.CKE[1] = 1), set it to 0. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved and the bit rate is doubled.

Set this bit to 0 in modes other than asynchronous mode.

RXDESEL bit (Asynchronous Start Bit Edge Detection Select)

The RXDESEL bit selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data reception operation depends on the setting of this bit. Set this bit to 1 when reception must be stopped while a break occurs or when reception must be started without keeping the RXDn pin input at the high level for the period of one data frame or longer after completion of the break.

Set this bit to 0 in modes other than asynchronous mode.

32.2.20 SNFR : Noise Filter Setting Register

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
 SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	NFCS[2:0]		

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	NFCS[2:0]	Noise Filter Clock Select In asynchronous mode, selects the standard setting for the base clock. In simple IIC mode, selects the standard settings for the clock source of the on-chip baud rate generator selected in the SMR.CKS[1:0] bits. 0 0 0: In asynchronous mode: Use clock signal divided by 1 with noise filter In simple IIC mode: Setting prohibited 0 0 1: In asynchronous mode: Setting prohibited In simple IIC mode: Use clock signal divided by 1 with noise filter 0 1 0: In asynchronous mode: Setting prohibited In simple IIC mode: Use clock signal divided by 2 with noise filter 0 1 1: In asynchronous mode: Setting prohibited In simple IIC mode: Use clock signal divided by 4 with noise filter 1 0 0: In asynchronous mode: Setting prohibited In simple IIC mode: Use clock signal divided by 8 with noise filter Others: Setting prohibited	R/W ^{*1}
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in SCR/SCR_SMCI are 0 (serial reception and transmission disabled).

The SNFR register sets the digital noise filter clock.

NFCS[2:0] bits (Noise Filter Clock Select)

The NFCS[2:0] bits select the sampling clock for the digital noise filter. To use the noise filter in asynchronous mode, set these bits to 000b. In simple IIC mode, when 32 clocks are selected as one bit period in the basic clock selection bits of the SEMR register, set the NFCS [2: 0] bits in the range from 001b to 100b. When any other value is selected for the basic clock selection bit, set the NFCS bit to 001b.

32.2.21 SIMR1 : IIC Mode Register 1

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
 SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x09

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IICDL[4:0]				—	—	IICM	

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	IICM	Simple IIC Mode Select 0: SCMR.SMIF = 0: Asynchronous mode (including multi-processor mode), clock synchronous mode, or simple SPI mode SCMR.SMIF = 1: Smart card interface mode 1: SCMR.SMIF = 0: Simple IIC mode SCMR.SMIF = 1: Setting prohibited	R/W ^{*1}
2:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7:3	IICDL[4:0]	SSDAn Delay Output Select SSDAn signal output delay in cycles of the clock signal from the on-chip baud rate generator. 0x00: No output delay Others: (IICDL - 1) to (IICDL) cycles	R/W ^{*1}

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (both serial transmission and reception are disabled).

SIMR1 selects simple IIC mode and the number of delay stages for the SSDAn output.

IICM bit (Simple IIC Mode Select)

In combination with the SCMR.SMIF bit, the IICM bit selects the operating mode.

IICDL[4:0] bits (SSDAn Delay Output Select)

The IICDL[4:0] bits specify an output delay on the SSDAn pin relative to the falling edge of the output on the SSCLn pin.

The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generator as the base. The signal obtained by frequency-dividing PCLK by the divisor set in SMR.CKS[1:0] is supplied as the clock signal from the on-chip baud rate generator. Set 00000b to IICDL[4:0] bits unless operation is in simple IIC mode. In simple IIC mode, set within the range shown in [Table 32.23](#).

Table 32.23 Settable value of IICDL[4: 0] bits in each communication mode

Communication mode	ABCS	Settable value of IICDL[4:0] bits
Other than simple IIC mode	Don't care	00000b
Simple IIC mode	0	00001b to 11111b
	1	00001b to 00100b

32.2.22 SIMR2 : IIC Mode Register 2

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	IICAC KT	—	—	—	IICCS C	IICINT M

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	IICINTM	IIC Interrupt Mode Select 0: Use ACK/NACK interrupts 1: Use reception and transmission interrupts	R/W ^{*1}
1	IICCS	Clock Synchronization 0: Do not synchronize with clock signal 1: Synchronize with clock signal	R/W ^{*1}
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	IICACKT	ACK Transmission Data 0: ACK transmission 1: NACK transmission and ACK/NACK reception	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (serial reception and transmission disabled).

SIMR2 selects how reception and transmission are controlled in simple IIC mode.

IICINTM bit (IIC Interrupt Mode Select)

The IICINTM bit selects the sources of interrupt requests in simple IIC mode.

IICCSC bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SSCLn clock signal is to be synchronized when the SSCLn pin is driven low because a wait was inserted by another other device.

The SSCLn clock signal is not synchronized if the IICCSC bit is 0. The SSCLn clock signal is generated according to the rate selected in the BRR register regardless of the level being input on the SSCLn pin.

Set the IICCSC bit to 1 except during debugging.

IICACKT bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set the IICACKT bit to 1 when ACK and NACK bits are received.

32.2.23 SIMR3 : IIC Mode Register 3

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
 SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x0B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IICSCLS[1:0]		IICSDAS[1:0]		IICSTIF	IICSTPREQ	IICRSTAREQ	IICSTAREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IICSTAREQ	Start Condition Generation 0: Do not generate start condition 1: Generate start condition*1 *3 *5 *6	R/W
1	IICRSTAREQ	Restart Condition Generation 0: Do not generate restart condition 1: Generate restart condition*2 *3 *5 *6	R/W
2	IICSTPREQ	Stop Condition Generation 0: Do not generate stop condition 1: Generate stop condition*2 *3 *5 *6	R/W
3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag 0: No requests are being made for generating conditions, or a condition is being generated 1: Generation of start, restart, or stop condition is complete. When 0 is written to IICSTIF, it is set to 0*4	R/W*4
5:4	IICSDAS[1:0]	SSDAn Output Select 0 0: Output serial data 0 1: Generate start, restart, or stop condition 1 0: Output low on SSDAn pin 1 1: Drive SSDAn pin to high-impedance state	R/W
7:6	IICSCLS[1:0]	SSCLn Output Select 0 0: Output serial clock 0 1: Generate start, restart, or stop condition 1 0: Output low on SSCLn pin 1 1: Drive SSCLn pin to high-impedance state	R/W

Note 1. Only generate a start condition after checking the bus state and confirming that the bus is free.

Note 2. Generate a restart or stop condition after checking the bus state and confirming that the bus is busy.

Note 3. Do not set more than one of the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.

Note 4. Write only 0. When 1 is written, the value is ignored.

Note 5. Execute the generation of a condition after the value of the IICSTIF flag is 0.

Note 6. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

The SIMR3 register is used to control the start, restart, and stop conditions in the simple I²C mode, and to hold the SSDAn and SSCLn pins at fixed levels.

IICSTAREQ bit (Start Condition Generation)

When a start condition is to be generated, set both IICSDAS[1:0] and IICSCLS[1:0] to 01b in addition to setting the IICSTAREQ bit to 1.

[Setting condition]

- On writing 1 to the bit.

[Clearing condition]

- On completion of start condition generation.

IICRSTAREQ bit (Restart Condition Generation)

When a restart condition is to be generated, set both IICSDAS[1:0] and IICSCLS[1:0] to 01b in addition to setting the IICRSTAREQ bit to 1.

[Setting condition]

- On writing 1 to the bit.

[Clearing condition]

- On completion of restart condition generation.

IICSTPREQ bit (Stop Condition Generation)

When a stop condition is to be generated, set both IICSDAS[1:0] and IICSCLS[1:0] to 01b in addition to setting the IICSTPREQ bit to 1.

[Setting condition]

- On writing 1 to the bit.

[Clearing condition]

- On completion of stop condition generation.

IICSTIF flag (Issuing of Start, Restart, or Stop Condition Completed Flag)

After generating a condition, the IICSTIF flag indicates that the condition generation is complete. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.

When the IICSTIF flag is 1 while an interrupt request is enabled by setting the SCR.TEIE bit, an STI request is output.

[Setting condition]

- On completion of a start, restart, or stop condition generation.

If the setting condition conflicts with any of the clearing conditions for the flag, the clearing condition takes precedence.

[Clearing conditions]

- On writing 0 to the bit. After writing 0 to the IICSTIF bit, read the bit to check that it is actually set to 0.
- On writing 0 to the SIMR1.IICM bit when operation is not in simple IIC mode.
- On writing 0 to the SCR.TE bit.

IICSDAS[1:0] bits (SSDAn Output Select)

The IICSDAS[1:0] bits control output from the SSDAn pin. Set IICSDAS[1:0] and IICSCLS[1:0] to the same value during normal operations.

IICSCLS[1:0] bits (SSCLn Output Select)

The IICSCLS[1:0] bits control output from the SSCLn pin. Set IICSDAS[1:0] and IICSCLS[1:0] to the same value during normal operations.

32.2.24 SISR : IIC Status Register

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
 SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	IICAC KR

Value after reset: 0 0 x x 0 x 0 0

Bit	Symbol	Function	R/W
0	IICACKR	ACK Reception Data Flag 0: ACK received 1: NACK received	R
1	—	This bit is read as 0.	R
2	—	The read value is undefined.	R
3	—	This bit is read as 0.	R
5:4	—	The read value is undefined.	R
7:6	—	These bits are read as 0.	R

SISR monitors the state in simple IIC mode.

IICACKR flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from the IICACKR flag. The IICACKR flag is updated on the rising edge of the SSCLn clock for the received ACK/NACK bit.

32.2.25 SPMR : SPI Mode Register

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
 SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x0D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CKPH	CKPOL	—	MFF	—	MSS	CTSE	SSE

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	SSE	SSn Pin Function Enable 0: Disable SSn pin function 1: Enable SSn pin function	R/W ¹
1	CTSE	CTS Enable 0: Disable CTS function (enable RTS output function) 1: Enable CTS function	R/W ¹
2	MSS	Master Slave Select 0: Transmit through TXDn pin and receive through RXDn pin (master mode) 1: Receive through TXDn pin and transmit through RXDn pin (slave mode)	R/W ¹
3	—	This bit is read as 0. The write value should be 0.	R/W
4	MFF	Mode Fault Flag 0: No mode fault error 1: Mode fault error	R/W ²
5	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6	CKPOL	Clock Polarity Select 0: Do not invert clock polarity 1: Invert clock polarity	R/W ^{*1}
7	CKPH	Clock Phase Select 0: Do not delay clock 1: Delay clock	R/W ^{*1}

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (both serial transmission and reception are disabled).

Note 2. Only 0 can be written to this bit, to clear the flag.

SPMR selects the extension settings in asynchronous and clock synchronous modes.

SSE bit (SSn Pin Function Enable)

Set the SSE bit to 1 to use the SSn pin to control transmission and reception in simple SPI mode. Set this bit to 0 in all other modes. In simple SPI mode, when master mode is selected (SCR.CKE[1:0] = 00b and SPMR.MSS = 0) and there is a single master, the SSn pin on the master side is not required to control reception and transmission. In such a case, set the SSE bit to 0. Do not set both the SSE and CTSE bits to 1. If this setting is made, operation is the same as that when these bits are set to 0.

CTSE bit (CTS Enable)

Set the CTSE bit to 1 if the SSn pin is to be used for inputting the CTS control signal to control transmission and reception. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, and simple IIC mode. Do not set both the CTSE and SSE bits to 1. If this setting is made, operation is the same as that when these bits are set to 0.

MSS bit (Master Slave Select)

The MSS bit selects master or slave operation in simple SPI mode. The functions of the TXDn and RXDn pins are reversed when this bit is set to 1, so that data is received through the TXDn pin and transmitted through the RXDn pin.

Set this bit to 0 in modes other than simple SPI mode.

MFF flag (Mode Fault Flag)

The MFF flag indicates mode fault errors. In a multi-master configuration, determine the mode fault error occurrence by reading this flag.

[Setting condition]

- When input on the SSn pin is low during master operation in simple SPI mode (SSE bit = 1 and MSS bit = 0).

[Clearing condition]

- On writing 0 to the bit after it is read as 1.

CKPOL bit (Clock Polarity Select)

The CKPOL bit selects the polarity of the clock signal output through the SCKn pin. See [Figure 32.69](#) for details. Set the CKPOL bit to 0 in all modes other than simple SPI mode and clock synchronous mode.

CKPH bit (Clock Phase Select)

The CKPH bit selects the phase of the clock signal output through the SCKn pin. See [Figure 32.69](#) for details. Set the CKPH bit to 0 in all modes other than simple SPI mode and clock synchronous mode.

32.2.26 FCR : FIFO Control Register

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)

Offset address: 0x14

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RSTRG[3:0]				RTRG[3:0]				TTRG[3:0]				DRES	TFRS T	RFRS T	FM
Value after reset:	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FM	FIFO Mode Select Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode. 0: Non-FIFO mode. Selects TDR/RDR or TDRHL/RDRHL for communication. 1: FIFO mode. Selects FTDRHL/FRDRHL for communication.	R/W ¹
1	RFRST	Receive FIFO Data Register Reset Valid only when FCR.FM = 1. 0: Do not reset FRDRHL 1: Reset FRDRHL	R/W
2	TFRST	Transmit FIFO Data Register Reset Valid only when FCR.FM = 1. 0: Do not reset FTDRHL 1: Reset FTDRHL	R/W
3	DRES	Receive Data Ready Error Select Selects the interrupt requested when detecting receive data ready. 0: Receive data full interrupt (SCIn_RXI) 1: Receive error interrupt (SCIn_ERI)	R/W
7:4	TTRG[3:0]	Transmit FIFO Data Trigger Number Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode. The trigger number is specified in the TTRG[3:0] bits.	R/W
11:8	RTRG[3:0]	Receive FIFO Data Trigger Number Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode. The trigger number is specified in the RTRG[3:0] bits.	R/W
15:12	RSTRG[3:0]	RTS Output Active Trigger Number Select Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode, when FCR.FM = 1, SPMR.CTSE = 0, and SPMR.SSE = 0. The trigger number is specified in the RSTRG[3:0] bits.	R/W

Note 1. Writable only when TE = 0 and RE = 0.

FCR selects FIFO mode, resets FTDRHL and FRDRHL, selects the FIFO data trigger number for transmission or reception, and selects the RTS output active trigger number.

FM bit (FIFO Mode Select)

When the FM bit is set to 1, FTDRHL and FRDRHL are selected for communication. When the FM bit is set to 0, TDR and RDR, or TDRHL and RDRHL are selected for communication.

RFRST bit (Receive FIFO Data Register Reset)

When the RFRST bit is set to 1, the FRDRHL register is reset and the received data count resets to 0. When 1 is written to the RFRST bit, it clears to 0 after 1 PCLK.

TFRST bit (Transmit FIFO Data Register Reset)

When the TFRST bit is set to 1, the FTDRHL register is reset and the transmit data count resets to 0. When 1 is written to the TFRST bit, it clears to 0 after 1 PCLK.

DRES bit (Receive Data Ready Error Select)

When detecting a receive data ready error, the selection can be made from an SCIn_RXI interrupt request or an SCIn_ERI interrupt request. When starting DTC or DMAC and reading from the FRDRH and FRDRL registers, set the DRES bit to 1.

TTRG[3:0] bits (Transmit FIFO Data Trigger Number)

The TDFE flag is set to 1 when the amount of transmit data in FTDRHL is equal to or less than the transmit triggering number specified in the TTRG[3:0] bits, and software can write data to FTDRHL. If SCR.TIE = 1, an SCIn_TXI interrupt request occurs.

RTRG[3:0] bits (Receive FIFO Data Trigger Number)

The RDF flag is set to 1 when the amount of receive data in FRDRHL is equal to or greater than the receive triggering number specified in the RTRG[3:0] bits, and software can read data from FRDRHL. If SCR.RIE = 1, an SCIn_RXI interrupt request occurs.

When RTRG[3:0] is 0, the RDF flag is not set even when the amount of data in the receive FIFO is equal to 0, and an SCIn_RXI interrupt does not occur.

RSTRG[3:0] bits (RTS Output Active Trigger Number Select)

When the amount of receive data stored in FRDRHL is equal to or greater than the receive triggering number specified in the RSTRG[3:0] bits, the RTS signal goes high.

When RSTRG[3:0] is 0, the RTS signal does not go high even when the amount of data in FRDRHL is equal to 0.

32.2.27 FDR : FIFO Data Count Register

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)

Offset address: 0x16

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

Bit	Symbol	Function	R/W
4:0	R[4:0]	Receive FIFO Data Count Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode, when FCR.FM = 1. Indicates the amount of receive data stored in FRDRHL.	R
7:5	—	These bits are read as 0.	R
12:8	T[4:0]	Transmit FIFO Data Count Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode, when FCR.FM = 1. Indicates the amount of non-transmitted data stored in FTDRHL.	R
15:13	—	These bits are read as 0.	R

The FDR register indicates the amount of data stored in FRDRHL and FTDRHL.

R[4:0] bits (Receive FIFO Data Count)

The R[4:0] bits indicate the amount of receive data stored in FRDRHL. 0x00 means no receive data, and 0x10 means that the maximum received data is stored in FRDRHL.

T[4:0] bits (Transmit FIFO Data Count)

The T[4:0] bits indicate the amount of non-transmitted data stored in FTDRHL. 0x00 means no transmit data, and 0x10 means that all (maximum amount) of the data to be transmitted is stored in FTDRHL.

32.2.28 LSR : Line Status Register

Base address: $SCIn = 0x4007\_0000 + 0x0020 \times n$ ($n = 0, 1$)

Offset address: 0x18

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

Bit	Symbol	Function	R/W
0	ORER	Overrun Error Flag Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode, and when FIFO is selected. 0: No overrun error occurred 1: Overrun error occurred	R ^{*1}
1	—	This bit is read as 0.	R
6:2	FNUM[4:0]	Framing Error Count Indicates the amount of data with a framing error in the receive data stored in FRDRHL.	R
7	—	This bit is read as 0.	R
12:8	PNUM[4:0]	Parity Error Count Indicates the amount of data with a parity error in the receive data stored in FRDRHL.	R
15:13	—	These bits are read as 0.	R

Note 1. Write 0 to SSR_FIFO.ORER to clear the flag.

The LSR register indicates the receive error status.

ORER flag (Overrun Error Flag)

The ORER flag reflects the value in SSR_FIFO.ORER.

FNUM[4:0] bits (Framing Error Count)

The FNUM[4:0] value indicates the amount of data with a framing error stored in the FRDRHL register.

PNUM[4:0] bits (Parity Error Count)

The PNUM[4:0] value indicates the amount of data with a parity error stored in the FRDRHL register.

32.2.29 CDR : Compare Match Data Register

Base address: $SCIn = 0x4007\_0000 + 0x0020 \times n$ ($n = 0, 1$)
 $SCIm = 0x4004\_1200 + 0x0020 \times m$ ($m = 2$ to 5, 9)

Offset address: 0x1A

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

Bit	Symbol	Function	R/W
8:0	CMPD[8:0]	Compare Match Data Holds compare data pattern for address match wakeup function.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

The CDR register sets the compare data for the address match function.

CMPD[8:0] bits (Compare Match Data)

The CMPD[8:0] bits set the data to be compared to receive data for the address match function, when the address match function is enabled (DCCR.DCME = 1).

Three bit lengths are available:

- CMPD[6:0] with 7-bit length
- CMPD[7:0] with 8-bit length
- CMPD[8:0] with 9-bit length

32.2.30 DCCR : Data Compare Match Control Register

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
 SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x13

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DCME	IDSEL	—	DFER	DPER	—	—	DCMF
Value after reset:	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DCMF	Data Compare Match Flag 0: Not matched 1: Matched	R/(W) ^{*1}
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	DPER	Data Compare Match Parity Error Flag 0: No parity error occurred 1: Parity error occurred	R/(W) ^{*1}
4	DFER	Data Compare Match Framing Error Flag 0: No framing error occurred 1: Framing error occurred	R/(W) ^{*1}
5	—	This bit is read as 0. The write value should be 0.	R/W
6	IDSEL	ID Frame Select Valid only in asynchronous mode, including multi-processor mode. 0: Always compare data regardless of the MPB bit value 1: Only compare data when MPB bit = 1 (ID frame)	R/W
7	DCME	Data Compare Match Enable Valid only in asynchronous mode, including multi-processor mode. 0: Disable address match function 1: Enable address match function	R/W

Note 1. Only 0 can be written, to clear the flag after reading 1.

The DCCR register controls the address match function.

DCMF flag (Data Compare Match Flag)

The DCMF flag indicates that the SCI detected a receive data match with the comparison data (CDR.CMPD).

[Setting condition]

- On match of the comparison data (CDR.CMPD) with the receive data when DCCR.DCME = 1.

[Clearing condition]

- When 0 is written after 1 is read from DCMF.

Clearing the SCR.RE bit to 0 does not affect the DCMF flag, which retains its previous value.

DPER flag (Data Compare Match Parity Error Flag)

The DPER flag indicates that a parity error occurred on address match detection (receive data match detection).

[Setting condition]

- When a parity error is detected in a frame in which an address match is detected.

[Clearing conditions]

- When 0 is written after 1 is read from DPER.

When the SCR.RE bit is set to 0 (serial reception is disabled), the DPER flag is not affected and retains its previous value.

DFER flag (Data Compare Match Framing Error Flag)

The DFER flag indicates that a framing error occurred on address match detection (receive data match detection).

[Setting conditions]

- When a stop bit of a frame in which an address match is detected is 0.
When in 2-stop-bit mode, only the first bit of the stop bits is checked for a value of 1 (the second stop bit is not checked).

[Clearing conditions]

- When 0 is written after 1 is read from DFER.

When the SCR.RE bit is set to 0 (serial reception is disabled), the DFER flag is not affected and retains its previous value.

IDSEL bit (ID Frame Select)

The IDSEL bit selects whether to compare data regardless of the MPB bit value or to compare data only when MPB = 1 (ID frame), when the address match function is enabled.

DCME bit (Data Compare Match Enable)

The DCME bit enables or disables the address match function (data compare match function).

If the SCI detects a match to the comparison data (CDR.CMPD) with the receive data, the DCME bit clears automatically, after which SCI operation mode is in normal receive mode. See [section 32.3.6. Address Match \(Receive Data Match Detection\) Function](#).

The write value must be 0 for all modes other than asynchronous mode.

32.2.31 SPTR : Serial Port Register

Base address: SCIn = 0x4007_0000 + 0x0020 × n (n = 0, 1)
SCIm = 0x4004_1200 + 0x0020 × m (m = 2 to 5, 9)

Offset address: 0x1C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SPB2I O	SPB2 DT	RXDM ON
Value after reset:	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	RXDMON	Serial Input Data Monitor Indicates the state of the RXD pin. 0: RXD terminal is the low level. 1: RXD terminals is the High level.	R
1	SPB2DT	Serial Port Break Data Select Selects the output level of the TXD pin when SCR.TE = 0. 0: Low level is output in TXD terminal.	R/W

Bit	Symbol	Function	R/W
		1: Output high level on TXD pin	
2	SPB2IO	Serial Port Break I/O Selects whether the value of SPB2DT is output to TXD pin. 0: Do not output value of SPB2DT bit on TXD pin 1: Output value of SPB2DT bit on TXD pin	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The SPTR register provides confirmation of the serial reception pin (RXD pin) status and sets the transmission pin status. This register can only be used in asynchronous mode.

The TXD pin status is determined by the combination of SCR.TE, SPTR.SPB2IO, and SPTR.SPB2DT settings, as shown in Table 32.24.

Table 32.24 TXD pin status

Value of SCR.TE	Value of SPTR.SPB2IO	Value of SPTR.SPB2DT	TXD pin status
0	0	—	Hi-Z (initial value)
0	1	0	Low level output
0	1	1	High level output
1	—	—	Serial transmit data is output

Note: —: Do not care.

Note: Use the SPTR register in asynchronous mode only. Using this register in any other mode is not guaranteed.

32.3 Operation in Asynchronous Mode

Figure 32.2 shows the general format for asynchronous serial communications. One frame consists of a start bit (low level), transmit or receive data, a parity bit, and stop bits (high level). In asynchronous serial communications, the communications line is usually held in the mark state (high level).

The SCI monitors the communications line. When the SCI detects a low, it regards that as a start bit and starts serial communication.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and receiver have a double-buffered structure in addition to FIFO mode, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

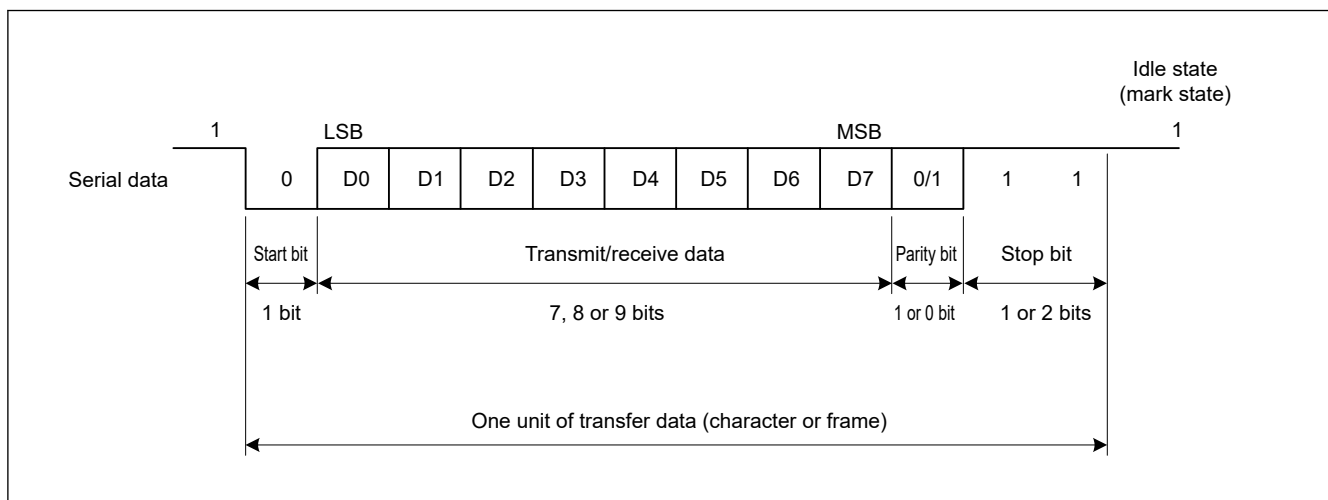


Figure 32.2 Data format in asynchronous serial communications with 8-bit data, parity bit, and 2 stop bits

32.3.1 Serial Data Transfer Format

Table 32.25 lists the serial data transfer formats that can be used in asynchronous mode. Any of 18 transfer formats can be selected with the SMR and SCMR settings. For details on the multi-processor function, see [section 32.4. Multi-Processor Communication Function](#).

Table 32.25 Serial transfer formats in asynchronous mode

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	0	ST	9-bit data									SP			
0	0	0	0	1	1	ST	9-bit data									SP	SP		
0	0	1	0	0	0	ST	9-bit data									P	SP		
0	0	1	0	1	1	ST	9-bit data									P	SP	SP	
1	0	0	0	0	0	ST	8-bit data								SP				
1	0	0	0	1	1	ST	8-bit data								SP	SP			
1	0	1	0	0	0	ST	8-bit data								P	SP			
1	0	1	0	1	1	ST	8-bit data								P	SP	SP		
1	1	0	0	0	0	ST	7-bit data							SP					
1	1	0	0	1	1	ST	7-bit data							SP	SP				
1	1	1	0	0	0	ST	7-bit data							P	SP				
1	1	1	0	1	1	ST	7-bit data							P	SP	SP			
0	0	—	1	0	0	ST	9-bit data									MPB	SP		
0	0	—	1	1	1	ST	9-bit data									MPB	SP	SP	
1	0	—	1	0	0	ST	8-bit data								MPB	SP			
1	0	—	1	1	1	ST	8-bit data								MPB	SP	SP		
1	1	—	1	0	0	ST	7-bit data							MPB	SP				
1	1	—	1	1	1	ST	7-bit data							MPB	SP	SP			

ST: Start bit
 SP: Stop bit
 P: Parity bit
 MPB: Multi-processor bit

32.3.2 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 32.3](#). The reception margin in asynchronous mode is determined by the following formula (1):

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100 [\%] \quad \dots \text{Formula (1)}$$

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 values of F = 0 and D = 0.5 in formula (1), the reception margin is determined using the following formula:

$$M = \{0.5 - 1/(2 \times 16)\} \times 100 (\%) = 46.875 \%$$

This represents the computed value. Renesas recommends a margin of 20% to 30% in system design.

Note 1. In this example, the SEMR.ABCS bit is 0 and the SEMR.ABCSE is 0. When the ABCS bit is 1 and the ABCSE bit is 0, a frequency of 8 times the bit rate is used as a base clock, and receive data is sampled on the rising edge of the 4th pulse of the base clock.

When the ABCSE bit is 1, a sextuple frequency of a bit rate is used as a base clock, and receive data is sampled on the rising edge of the 3rd pulse of the base clock.

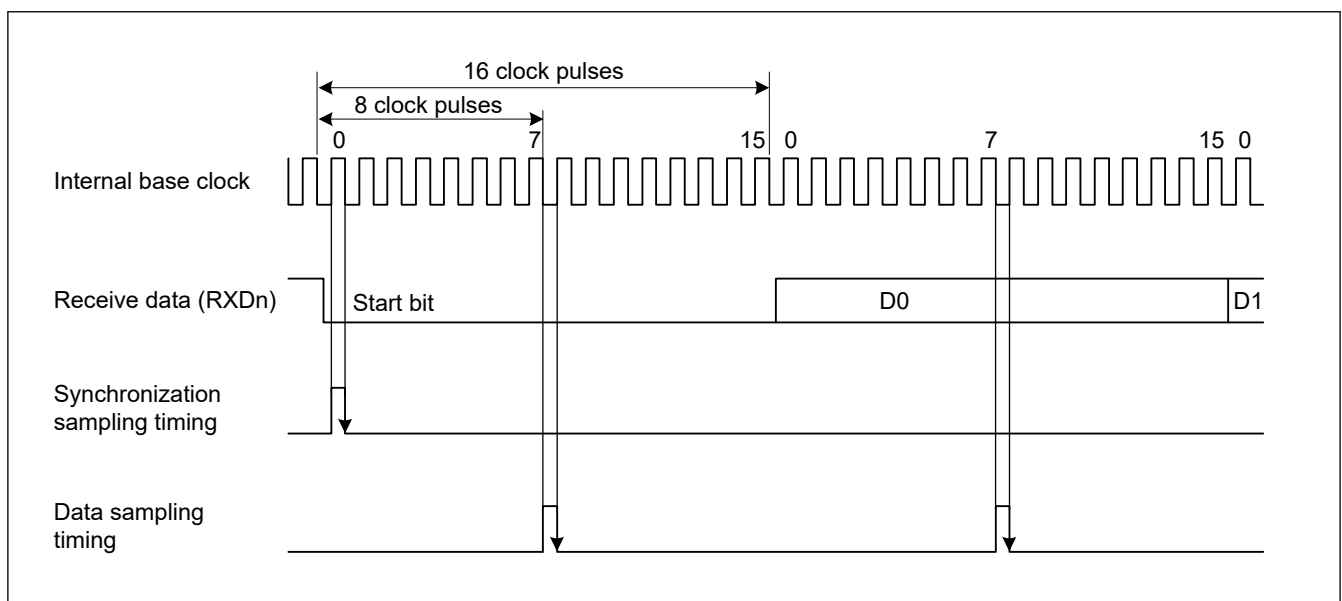


Figure 32.3 Receive data sampling timing in asynchronous mode

32.3.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input to the SCKn pin can be selected as the transfer clock of the SCI, based on the SMR.CM and SCR.CKE[1:0] settings.

When an external clock is input to the SCKn pin, the clock frequency must be 16 times the bit rate (when SEMR.ABCS = 0) or 8 times the bit rate (when SEMR.ABCS = 1).

When the SCI uses its internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is configured so that the rising edge of the clock is in the middle of the transmit data, as shown in [Figure 32.4](#).

When clock output is enabled, the clock is output after setting the SCR.TE or SCR.RE bit to 1.

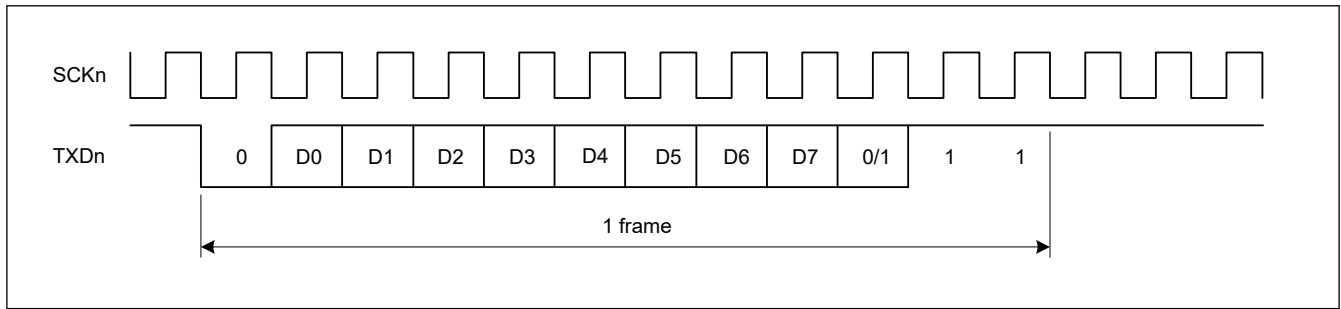


Figure 32.4 Phase relationship between output clock and transmit data in asynchronous mode when SMR.CHR = 0, PE = 1, MP = 0, and STOP = 1

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

When the SEMR.ABCS bit is set to 1 and eight pulses of the base clock for a 1-bit period is selected, the SCI operates on the bit rate twice that of when ABCS is set to 0. When the SEMR.BGDM bit is set to 1, the cycle of the base clock is half and the bit rate is double that of when BGDM is set to 0. When the SCR.CKE[1] bit is set to 0 and the on-chip baud rate generator is selected, setting the ABCS and BGDM bits to 1 allows the SCI to operate at a bit rate four times that when the ABCS and BGDM bits are set to 0. When the SEMR.ABCSE bit is set to 1, the number of basic clock pulses is 6 during a period of 1 bit, and the SCI operates at a bit rate 16/3 times that when SEMR.ABCS = 0, SEMR.BGDM = 0, and SEMR.ABCSE = 0.

As shown by Formula (1) in [section 32.3.2. Receive Data Sampling Timing and Reception Margin in Asynchronous Mode](#), the reception margin decreases when the SEMR.ABCS or SEMR.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.

32.3.5 CTS and RTS Functions

The CTS function uses input on the CTSn pin in transmission control. Setting the SPMR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, placing a low level on the CTSn pin causes transmission to start.

Driving the CTSn pin high while transmission is in progress does not affect transmission of the current frame.

In the RTS function, which uses output on the RTSn pin, a low level is output when reception becomes possible. Conditions for output of the low and high levels are shown in this section.

[Conditions for low level output]

Satisfaction of all conditions are listed in this section.

Non-FIFO selected

- The value of the SCR.RE bit is 1
- Reception is not in progress
- There is no received data yet to be read
- The ORER, FER, and PER flags in the SSR register are all 0

FIFO selected

- The value of the SCR.RE bit is 1
- The amount of receive data written in FRDRHL is equal to or less than the setting value of FCRH.RSTRG[3:0]
- The ORER flag in the SSR_FIFO register (ORER in FRDRH) is 0

[Condition for high level output]

- The conditions for low-level output are not satisfied

32.3.6 Address Match (Receive Data Match Detection) Function

The address match function can be used only in asynchronous mode.

If the DCCR.DCME bit is set to 1, when one frame of data is received, the SCI compares that received data with the data set in CDR.CMPD. If the SCI detects a match to the comparison data (CDR.CMPD^{*3}) with the received data, the SCI can issue the SCIn_RXI interrupt request.

If the SMR.MP bit is set to 0, comparison occurs only for valid data in receive format. In multi-processor mode (SMR.MP bit = 1), if the DCCR.IDSEL bit is set to 1, receive data where the MPB bit is 1 is subject to comparison for address match and receive data where the MPB bit is 0 is always treated as a non-match.

If the DCCR.IDSEL bit is set to 0, SCI performs address match detection regardless of the MPB bit value of the received data.

Until SCI detects a match to the comparison data (CDR.CMPD^{*3}) with receive data, received data is skipped (discarded), and the SCI cannot detect a parity error or framing error.

When SCI detects a match, the DCCR.DCME bit is automatically cleared, and the DCCR.DCMF flag is set to 1. If the DCCR.IDSEL bit is set to 1, the SCR.MPIE bit is automatically cleared. If DCCR.IDSEL is set to 0, the value of the SCR.MPIE bit is retained. If the SCR.RIE bit is set to 1, the SCI issues an SCIn_RXI interrupt request.

If the SCI detects a framing error in the receive data for which a match is detected, the DCCR.DFER flag is set to 1, and if the SCI detects a parity error in that frame, the DCCR.DPER flag is set to 1. The compared receive data is not stored in the RDR register^{*1}, and SSR.RDRF remains 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.

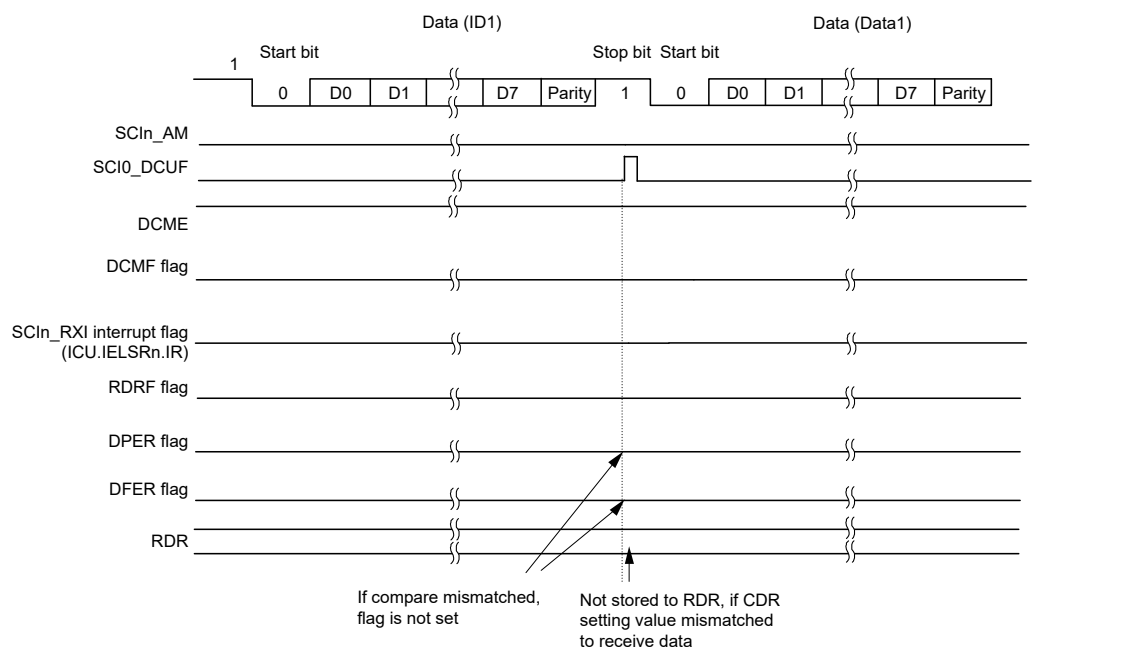
When the DCCR.DFER or DCCR.DPER flag is set, the address match is not performed. Before enabling the address match function, set the DCCR.DFER and DCCR.DPER flags to 0.

Examples of the address match function are shown in [Figure 32.5](#) and [Figure 32.6](#).

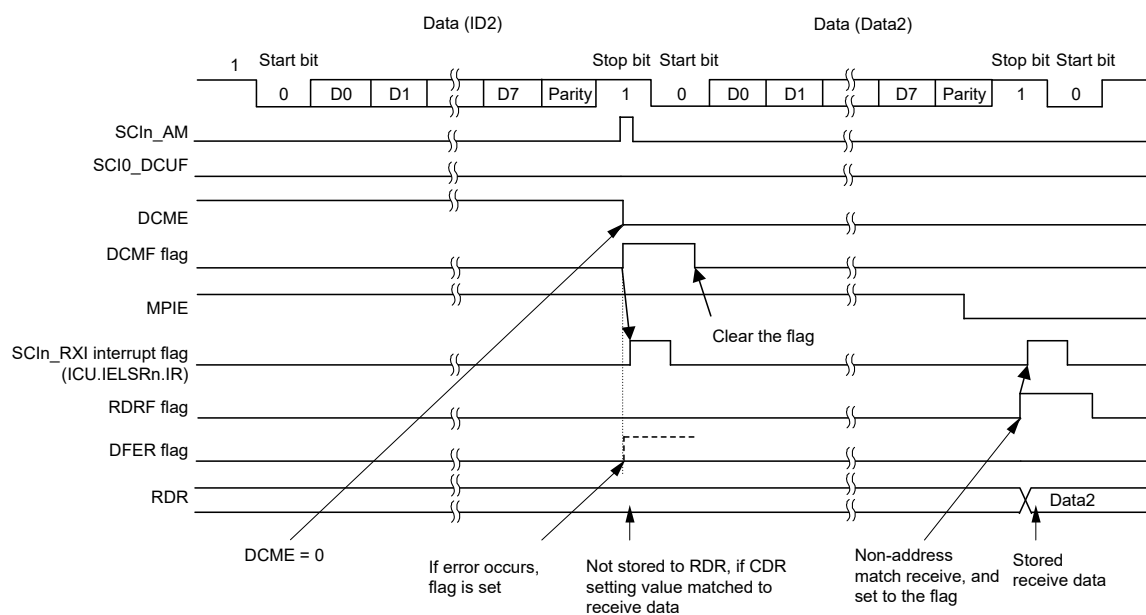
Note 1. When FCR.FM = 1, this refers to the FRDRHL register.

Note 2. When FCR.FM = 1, this refers to the SSR_FIFO.RDF flag.

Note 3. This comparative target can select one length of 3 types: CMPD[6:0] with 7-bit length, CMPD[7:0] with 8-bit length, and CMPD[8:0] with 9-bit length.



(a) Example of compare mismatched between receive data and CDR (8-bit length/parity/non multi-processor mode)



(b) Example of compare matched between receive data and CDR (8-bit length/parity/non multi-processor mode)

Figure 32.5 Example of address match (1) normal mode

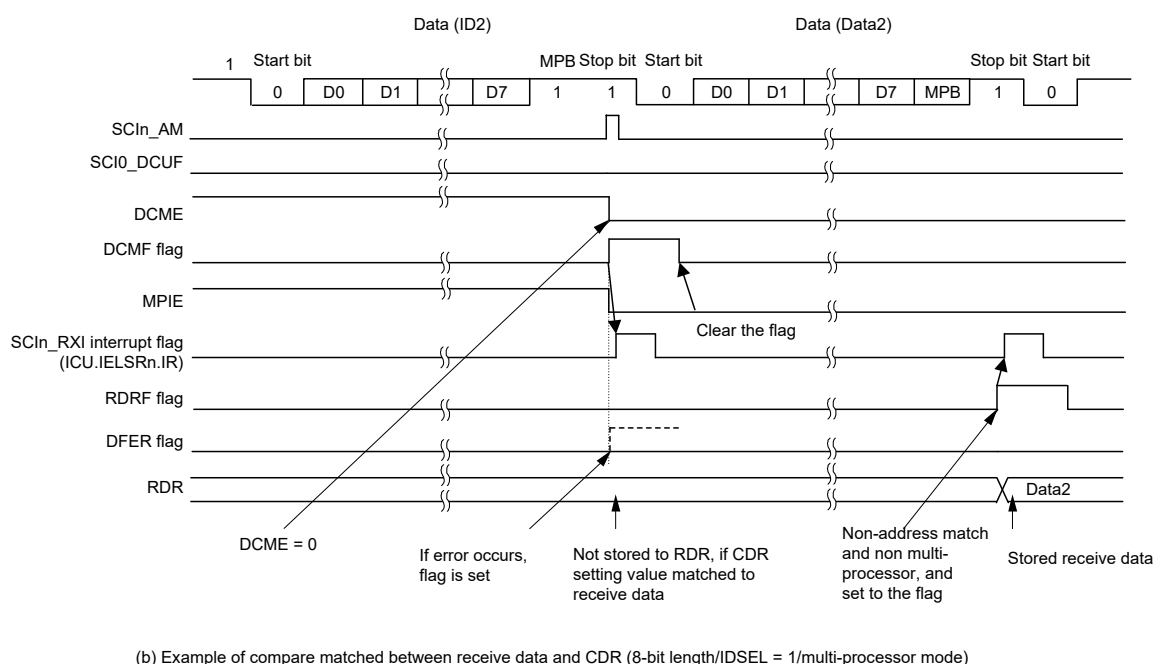
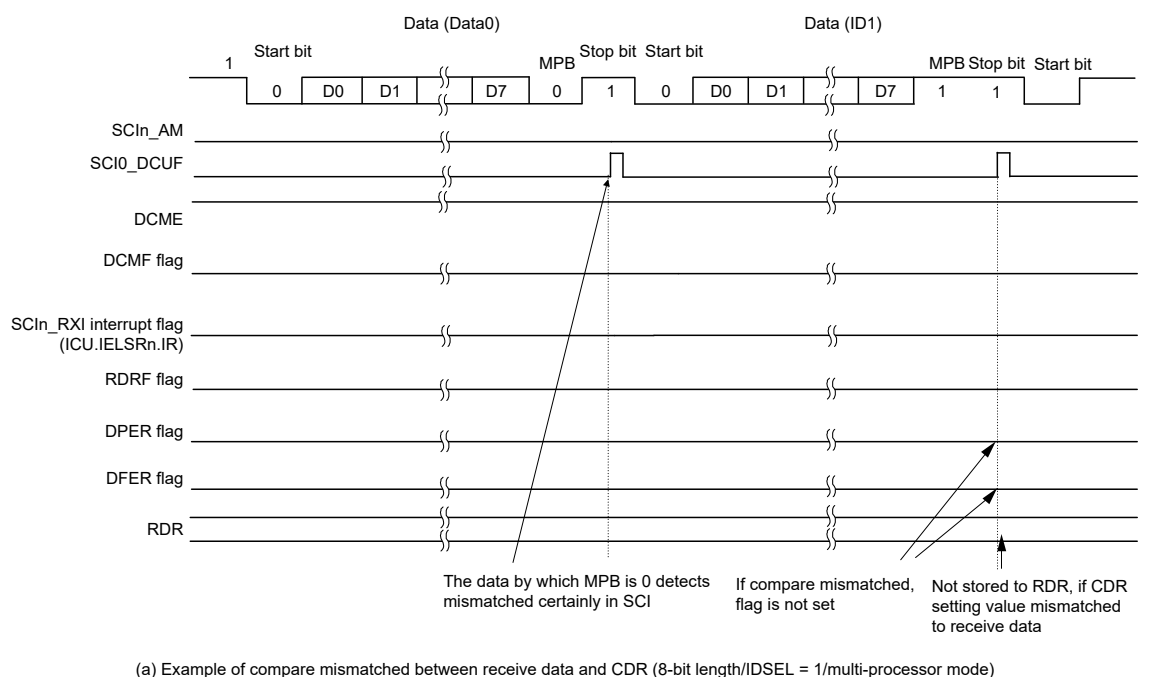


Figure 32.6 Example of address match (2) multi-processor mode

32.3.7 SCI Initialization in Asynchronous Mode

Before transmitting and receiving data, start by writing the initial value 0x00 to the SCR register, then continue through the SCI initialization procedure (select non-FIFO or FIFO) shown in [Table 32.26](#) and [Table 32.27](#). Whenever the operating mode or transfer format is to be changed, the SCR register must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied during initialization.

Note: Setting the SCR.RE bit to 0 initializes neither the ORER, FER, RDRF, RDF, PER, and DR flags in SSR/SSR_FIFO nor RDR and RDRHL. When the TE bit is set to 0, the TEND flag for the selected FIFO buffer is not initialized.

Note: In non-FIFO mode, switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of an SCIn_TXI interrupt request.

Table 32.26 Example flow of SCI initialization in asynchronous mode with non-FIFO selected

No.	Step Name	Description
1	Start initialization	
2	Set the SCR.TIE, RIE, TE, RE, and TEIE bits to 0	
3	Set the FCR.FM bit to 0	Set the FCR.FM bit to 0.
4	Set the SCR.CKE[1:0] bits	Set the clock selection in SCR. When the clock output is selected in asynchronous mode, the clock is output immediately after SCR settings are made.
5	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits to 0.	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits to 0. Step 5 can be skipped if the values have not been changed from the initial values.
6	Set the data transmission/reception format in SMR, SCMR, and SEMR	Set data transmission/reception format in SMR, SCMR, and SEMR.
7	Set a value in BRR	Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.
8	Set a value in MDDR	Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is set to 0 or an external clock is used.
9	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
10	Set the SCR.TE or RE bit to 1, and set the SCR.TIE and RIE bits	Set the SCR.TE or RE bit to 1. Also set the SCR.TIE and RIE bits. Setting the TE and RE bits allows TXDn and RXDn to be used.
11	Initialization completion	

Table 32.27 Example flow of SCI initialization in asynchronous mode with FIFO selected (1 of 2)

No.	Step Name	Description
1	Start initialization	
2	Set the SCR.TIE, RIE, TE, RE, and TEIE bits to 0	
3	Set the FCR.FM, TFRST, and RFRST bits to 1. Set the FCR.TTRG[3:0], RTRG[3:0], and RSTRG[3:0] bits.	Set the FCR.FM, TFRST, and RFRST bits to 1 (FIFO mode enabled, transmit/receive FIFOs empty). Set the FCR.TTRG[3:0], RTRG[3:0], and RSTRG[3:0] bits.
4	Set the SCR.CKE[1:0] bits	Set the clock selection in SCR. When the clock output is selected in asynchronous mode, the clock is output immediately after SCR settings are made.
5	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits to 0.	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits to 0. Step 5 can be skipped if the values have not been changed from the initial values.
6	Set the data transmission/reception format in SMR, SCMR, and SEMR	Set data transmission/reception format in SMR, SCMR, and SEMR.
7	Set a value in BRR	Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.
8	Set a value in MDDR	Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is set to 0 or an external clock is used.

Table 32.27 Example flow of SCI initialization in asynchronous mode with FIFO selected (2 of 2)

No.	Step Name	Description
9	Set the FCR.TFRST and RFRST bits to 0	Set the FCR.TFRST and RFRST bits to 0.
10	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
11	Set the SCR.TE or RE bit to 1, and set the SCR.TIE and RIE bits	Set the SCR.TE or RE bit to 1. Also set the SCR.TIE and RIE bits. Setting the TE and RE bits allows TXDn and RXDn to be used.
12	Initialization completion	

32.3.8 Serial Data Transmission in Asynchronous Mode

(1) Non-FIFO selected

Figure 32.7, Figure 32.8, and Figure 32.9 show examples of serial transmission in asynchronous mode.

In serial transmission, the SCI operates as described in this section. When the SCR.TE bit is set to 1, the high level is output to TXD for one frame.

- The SCI transfers data from the TDR^{*1} register to the TSR register when data is written to TDR^{*1} in the SCIn_TXI interrupt handling routine.
The SCIn_TXI interrupt request at the beginning of transmission is generated when the SCR.TE and SCR.TIE bits are set to 1 simultaneously by a single instruction.
- Transmission starts after the SPMR.CTSE bit is set to 0 (CTS function is disabled) or a low level on the CTSn pin causes data transfer from the TDR^{*1} register to the TSR register. If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to the TDR^{*1} register in the SCIn_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 (SCIn_TXI interrupt requests are disabled) and the SCR.TEIE bit to 1 (an SCIn_TEI interrupt request is enabled) after the last of the data to be transmitted is written to the TDR^{*1} register from the handling routine for SCIn_TXI requests.
- 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
- The SCI checks for update of the TDR register on output of the stop bit.
- When the TDR register is updated, setting the SPMR.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn pin causes transfer of the next transmit data from the TDR^{*1} register to the TSR register and transmission of the stop bit, after which serial transmission of the next frame starts.
- If the TDR register is not updated, the SSR.TEND flag is set to 1, the stop bit is sent, and the mark state is entered, in which 1 is output. If the SCR.TEIE bit is 1, the SSR.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated.

Note 1. The TDRHL register when 9-bit data length is selected.

Figure 32.7, Figure 32.8, and Figure 32.9 show examples of serial transmission in asynchronous mode.

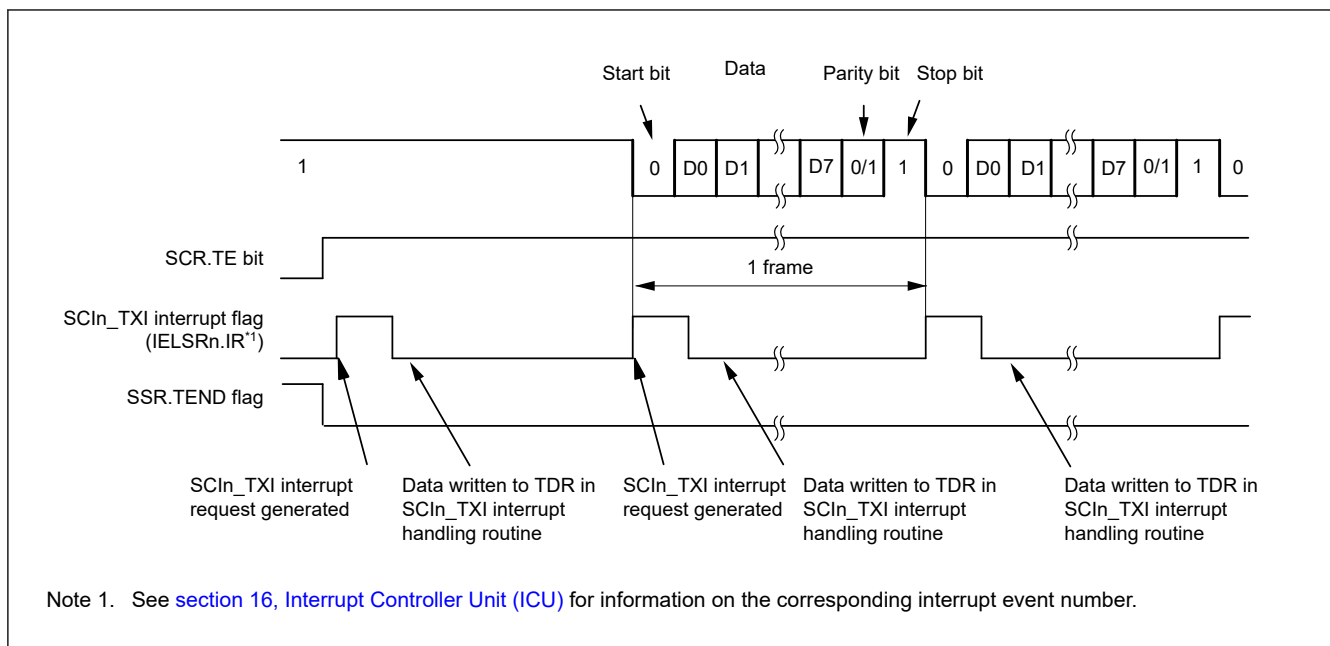


Figure 32.7 Example operation for serial transmission in asynchronous mode (1) with 8-bit data, parity bit, 1 stop bit, CTS function not used, and at the beginning of transmission

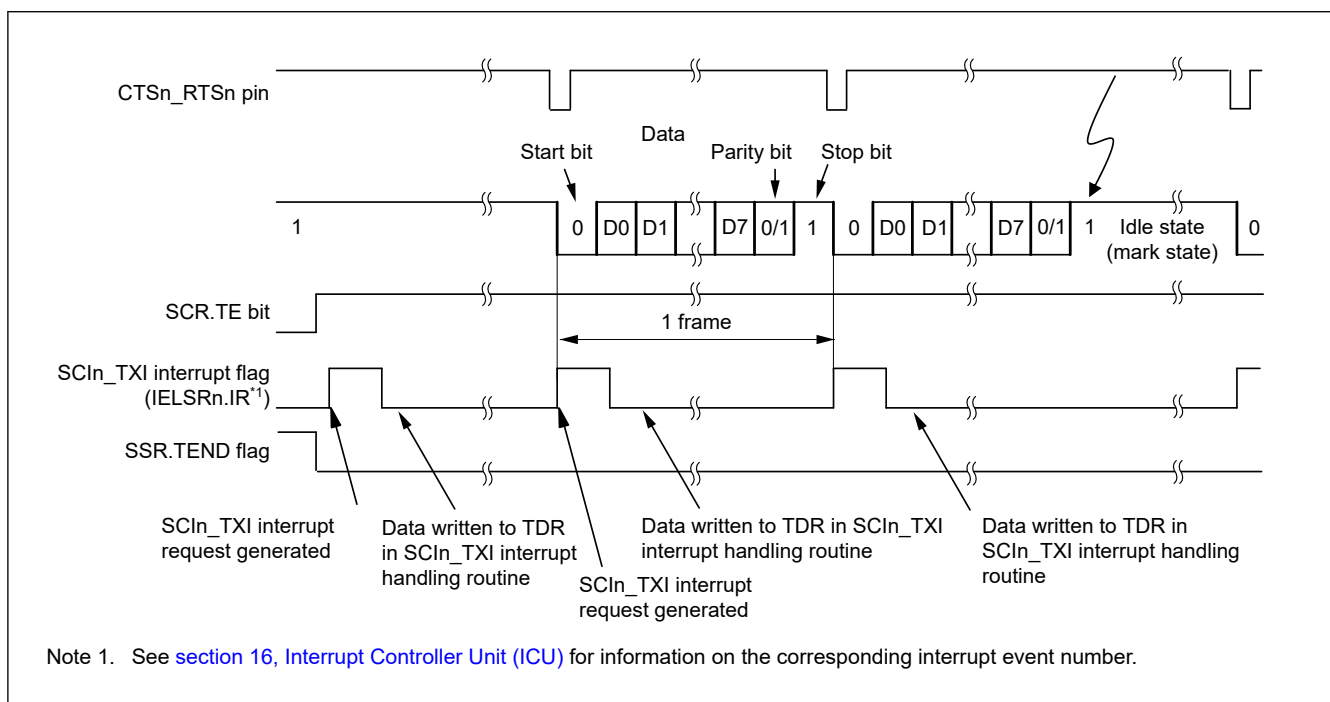


Figure 32.8 Example operation for serial transmission in asynchronous mode (2) with 8-bit data, parity bit, one stop bit, CTS function used, and at the beginning of transmission

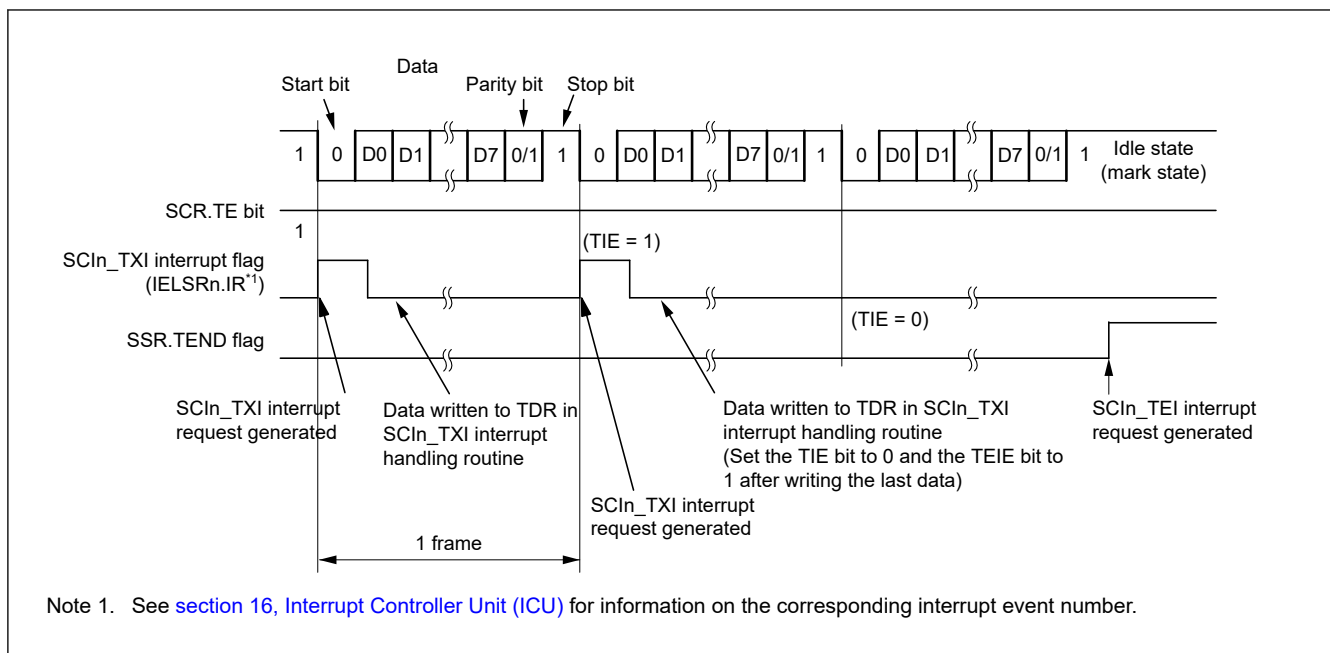


Figure 32.9 Example operation for serial transmission in asynchronous mode (3) with 8-bit data, parity bit, one stop bit, CTS function not used, and from the middle of transmission until transmission completion

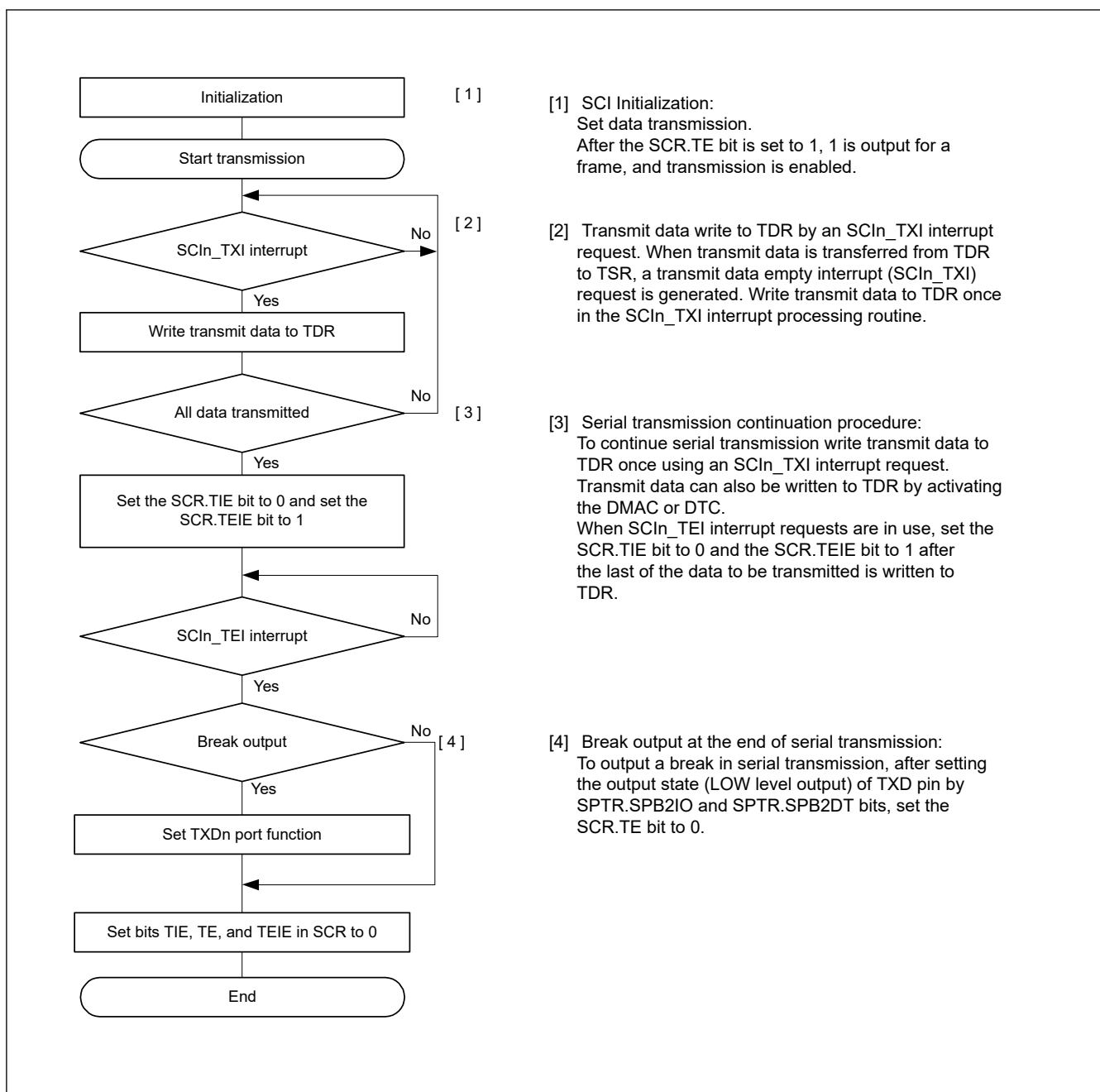


Figure 32.10 Example flow of serial transmission in asynchronous mode with non-FIFO selected

(2) FIFO selected

Figure 32.11 shows an example of a data format that is written to FTDRH and FTDRL in asynchronous mode.

Data corresponding to the data length is set to FTDRH and FTDRL. Write 0 for unused bits. Write in order from FTDRH to FTDRL.

Data Length	Register Setting		Transmit data in FTDRH, FTDRL															
			FTDRL															
	SCMR. CHR1	SMR. CHR	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
8 bits	1	1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
9 bits	0	Don't care	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Note: —: Invalid. The write value should be 0.

Figure 32.11 Data format written to FTDRH and FTDRL with FIFO selected

In serial transmission, the SCI operates as described in this section. When the TE bit is set to 1, the high level is output to TXD for one frame (preamble).

1. The SCI transfers data from the FTDRL^{*1} register to the TSR register when data is written to FTDRL^{*1} in the SCIn_TXI interrupt handling routine. The amount of data that can be written to FTDRL is 16 minus FDR.T[4:0] bytes. The SCIn_TXI interrupt request at the beginning of transmission is generated when the SCR.TE and SCR.TIE bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the SPMR.CTSE bit is set to 0 (CTS function is disabled) or a low level on the CTSn pin causes data transfer from the FTDRL^{*1} register to the TSR register. When the amount of transmit data written in FTDRL is equal to or less than the specified transmit triggering number, SSR_FIFO.TDFE is set to 1. If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to FTDRL^{*1} in the SCIn_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 (SCIn_TXI interrupt requests are disabled) and the SCR.TEIE bit to 1 (an SCIn_TEI interrupt request is enabled) after the last of the data to be transmitted is written to the FTDRL^{*1}^{*2} register from the handling routine for SCIn_TXI requests.
3. Data is sent from the TXDn pin in the following order:
 - Start bit
 - Transmit data
 - Parity bit or multi-processor bit (can be omitted depending on the format)
 - Stop bit
4. On output of the stop bit, the SCI checks whether non-transmitted data remains in the FTDRL^{*3} register.
5. When data is set to FTDRL^{*3}, setting the SPMR.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn pin causes transfer of the next transmit data from FTDRL^{*1} to TSR and transmission of the stop bit, after which serial transmission of the next frame starts.
6. If data is not set in FTDRL^{*3}, the TEND flag in SSR_FIFO is set to 1, the stop bit is sent, and the mark state is entered in which 1 is output. If the SCR.TEIE bit is 1, the SSR_FIFO.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated.

Note 1. Write data not to FTDRL but to the FTDRH and FTDRL registers.

Note 2. Write data in order from FTDRH to FTDRL when 9-bit data length is selected.

Note 3. The SCI only checks for update to the FTDRL register and not the FTDRH register when 9-bit data length is selected.

Figure 32.12 shows an example flow of serial transmission in asynchronous mode with FIFO selected.

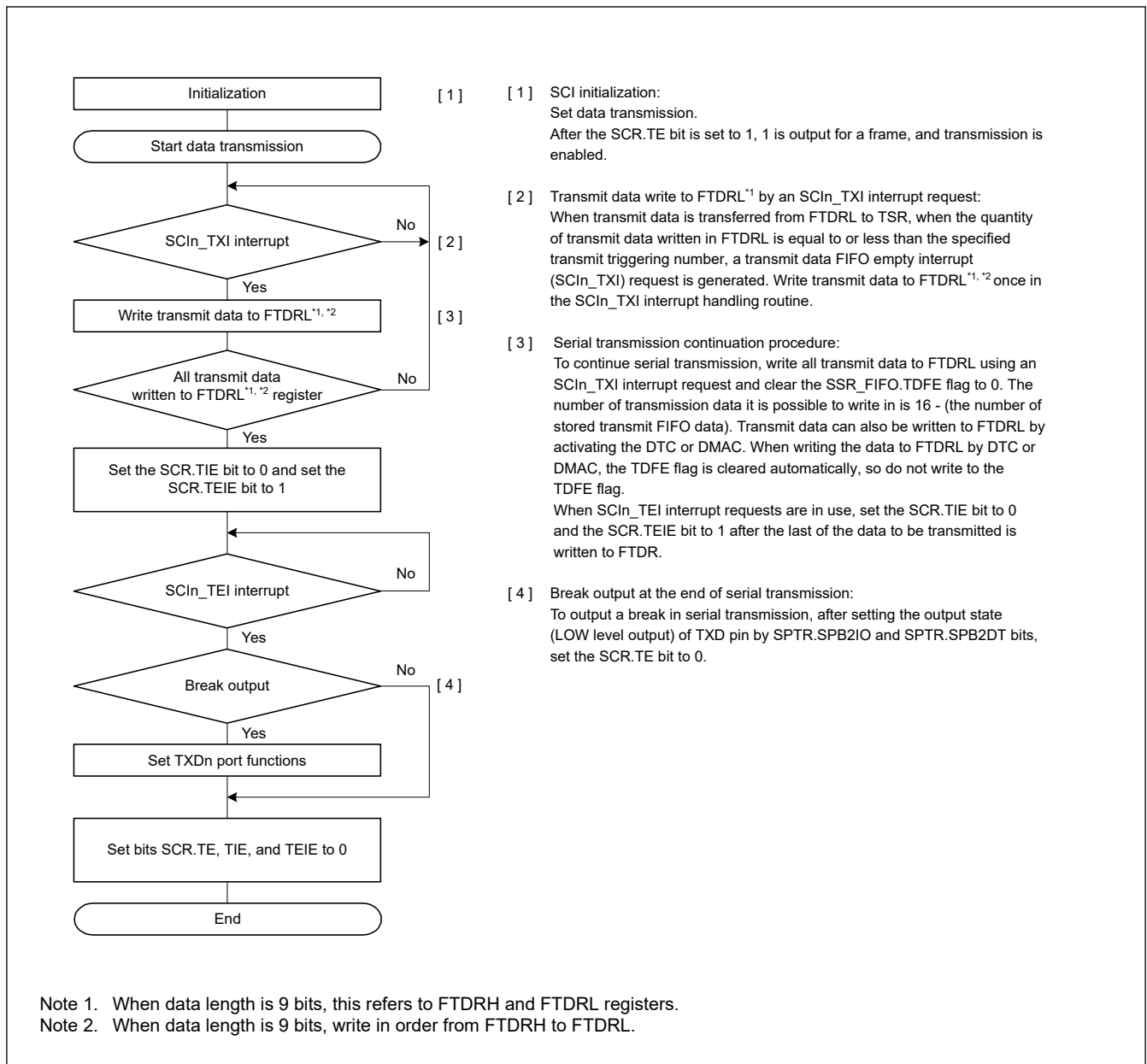


Figure 32.12 Example flow of serial transmission in asynchronous mode with FIFO selected

32.3.9 Serial Data Reception in Asynchronous Mode

(1) Non-FIFO selected

Figure 32.13 and Figure 32.14 show an example of the operation for serial data reception in asynchronous mode.

In serial data reception, the SCI operates as follows:

1. When the value of the SCR.RE bit becomes 1, the output signal on the RTSn pin goes low.
2. The SCI monitors the communications line and when it detects a start bit, the SCI performs internal synchronization, stores receive data in RSR.
3. If the multi-processor communication function is enabled (SMR.MP = 1), see [section 32.4.2. Multi-Processor Serial Data Reception](#). If the address match function (data compare match function) is enabled (DCCR.DCME = 1), the SCI cannot detect a parity or framing error as receive data are skipped (discarded) until the SCI detects a match between the receive data and comparison data (CDR.CMPD^{*1}).

4. If the SCI detects an address match, the DCCR.DCME bit is automatically cleared, the DCCR.DCMF flag becomes 1, and an SCIn_AM interrupt^{*2} request is generated. To enable the generation of an SCIn_RXI interrupt request, set the SCR.RIE bit to 1. The compared receive data are not stored in the RDR register^{*3}. The SSR.RDRF flag remains 0.
5. If the SCI detects a framing error in the receive data for which an address match is detected, the DCCR.DFER flag is set to 1, and if the SCI detects a parity error in that frame, the DCCR.DPER flag becomes 1. To enable the generation of an SCIn_ERI interrupt request, set the SCR.RIE bit to 1.
6. If a framing or a parity error is detected (the DCCR.DFER flag or DCCR.DPER flag is 1) in the SCIn_AM interrupt handling routine, set the DCCR.DFER and DCCR.DPER flags to 0 and set the DCCR.DCME bit to 1 to enable the address match function again. If neither a framing nor a parity error has been detected (the DCCR.DFER and DCCR.DPER flags are both 0), set the DCCR.DCMF flag to 0. See [Figure 32.5](#).
7. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the RDR^{*3} register.
8. If a parity error is detected, the SSR.PER flag is set to 1 and receive data is transferred to the RDR^{*3} register. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated.
9. If a framing error is detected, the SSR.FER flag is set to 1 and receive data is transferred to the RDR^{*3} register. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated.
10. When reception finishes successfully, receive data is transferred to the RDR^{*3} register. If the SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in the SCIn_RXI interrupt handling routine before reception of the next receive data is complete. Reading the received data that was transferred to the RDR register causes the RTSn pin to output low.

Note 1. This scope of comparison is selectable as one of three lengths: CMPD[6:0] is for 7-bit length, CMPD[7:0] is for 8-bit length, and CMPD[8:0] is for 9-bit length.

Note 2. As no interrupt enable bit is assigned to the SCIn_AM interrupt, an interrupt request is generated by setting the DCCR.DCMF to 1.

Note 3. Only read data in the RDRHL register when 9-bit data length is selected.

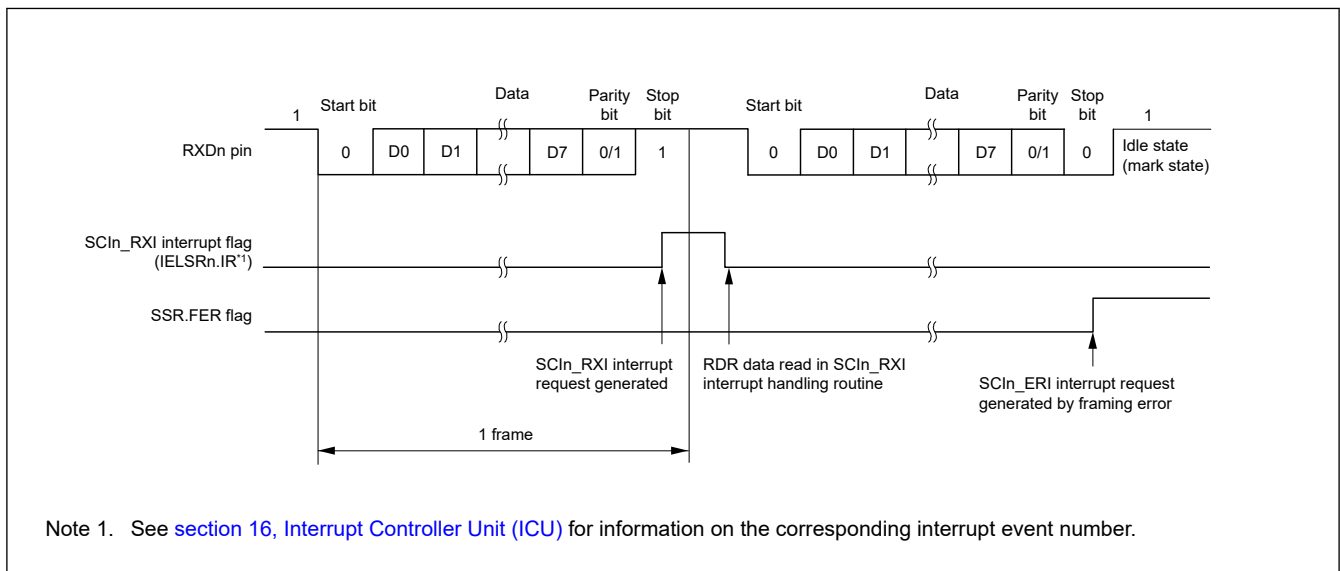


Figure 32.13 Example of SCI operation for serial reception in asynchronous mode (1) when the RTS function is not used, and with 8-bit data, parity bit, and 1 stop bit

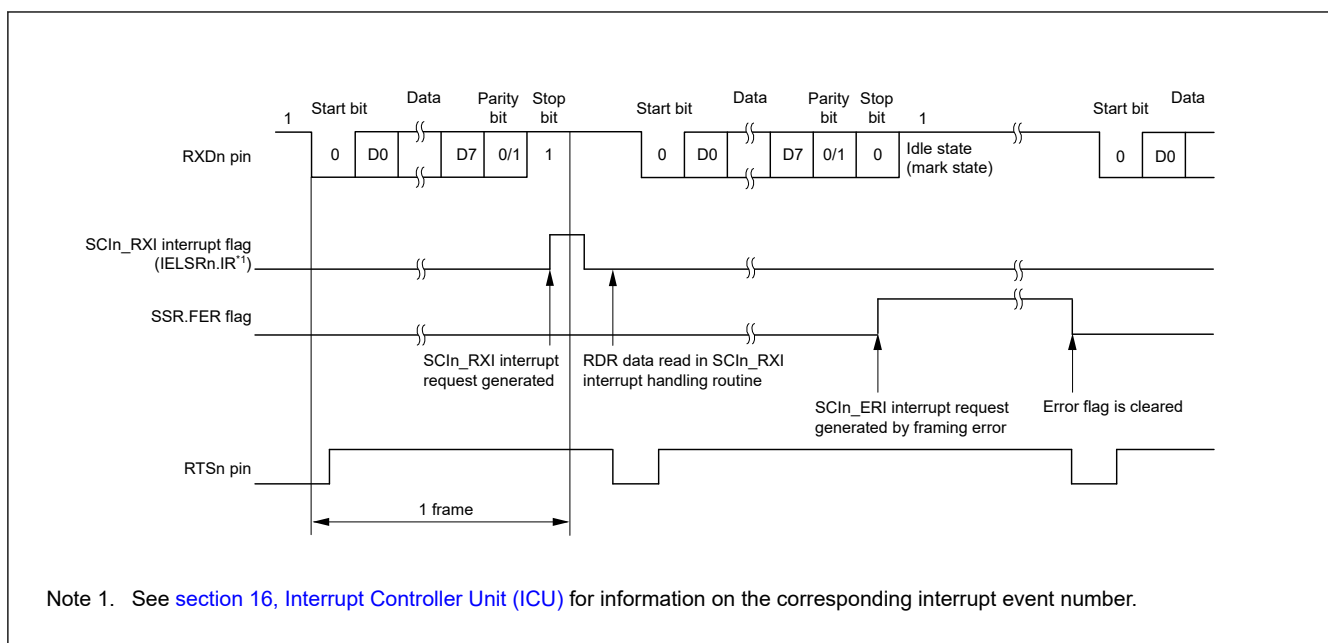


Figure 32.14 Example of SCI operation for serial reception in asynchronous mode (2) when RTS function is used, and with 8-bit data, parity bit, and 1 stop bit

[Table 32.28](#) lists the states of the flags in the SSR register and receive data handling when a receive error is detected.

If a receive error is detected, an SCIn_ERI interrupt request is generated but an SCIn_RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER bits to 0 before resuming reception. In addition, be sure to read the RDR or RDRHL register during overrun error processing. When a reception is forced to terminate by setting the SCR.RE bit to 0 during operation, read the RDR or RDRHL register because received data that is not yet read might be left in the RDR or RDRHL.

[Figure 32.15](#) and [Figure 32.16](#) show example flows of serial data reception.

Table 32.28 Flags in SSR Status Register and receive data handling

Flags in the SSR Status Register			Receive data	Receive error type
ORER	FER	PER		
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR ^{*1}	Framing error
0	0	1	Transferred to RDR ^{*1}	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR ^{*1}	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error

Note 1. Only read data in the RDRHL register when 9-bit data length is selected.

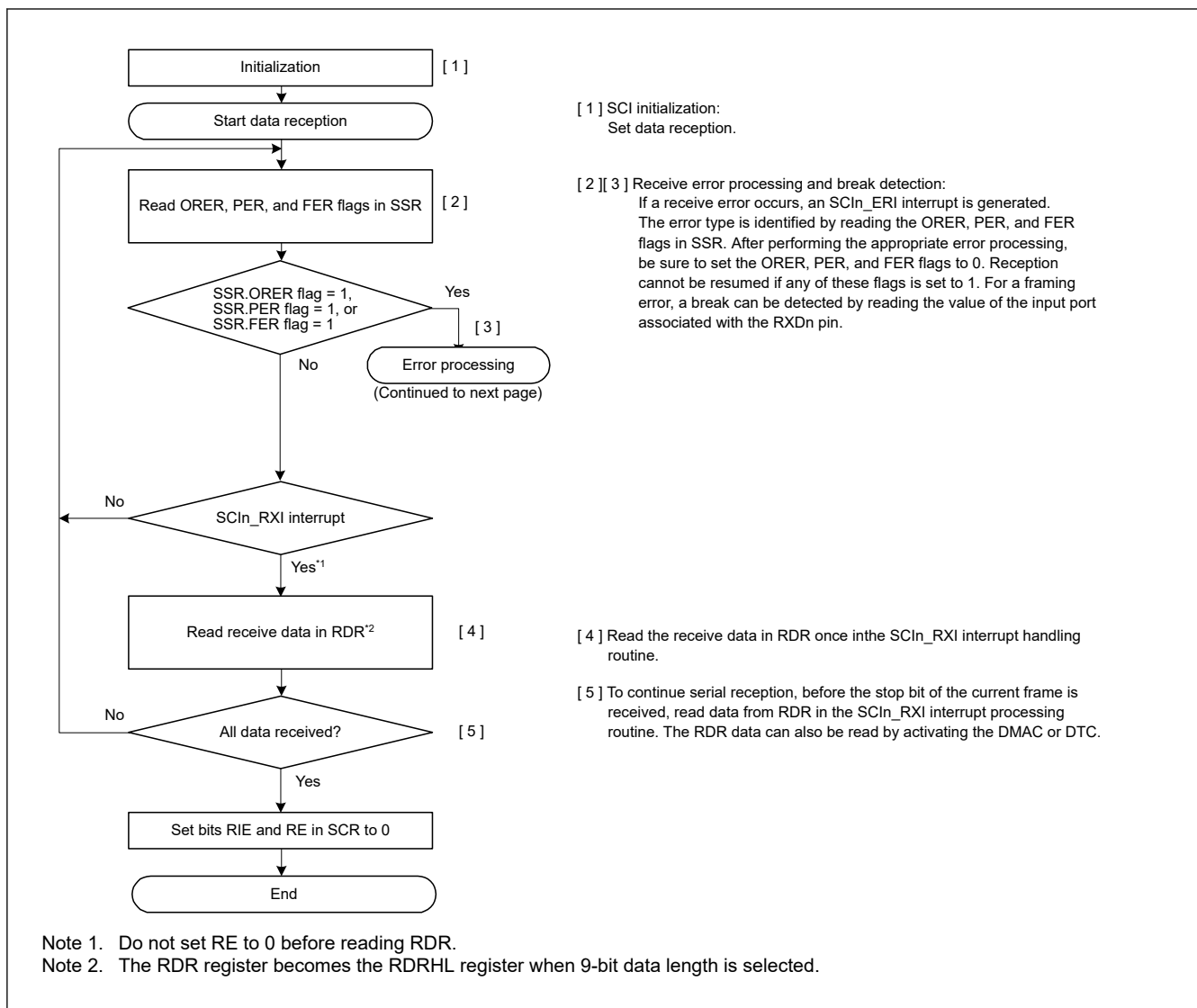


Figure 32.15 Example flow of serial reception in asynchronous mode with non-FIFO selected and Address Matching Disabled (1)

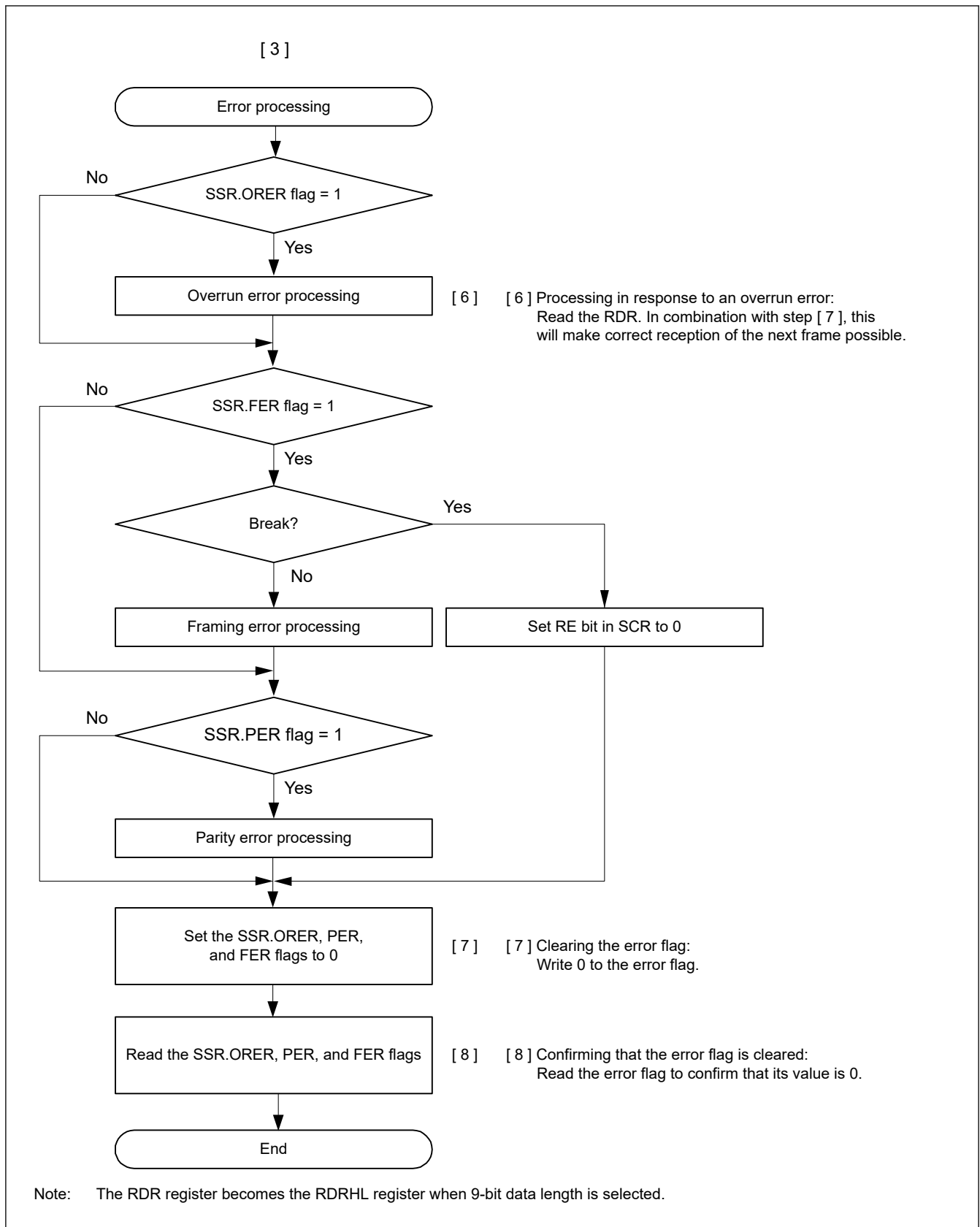
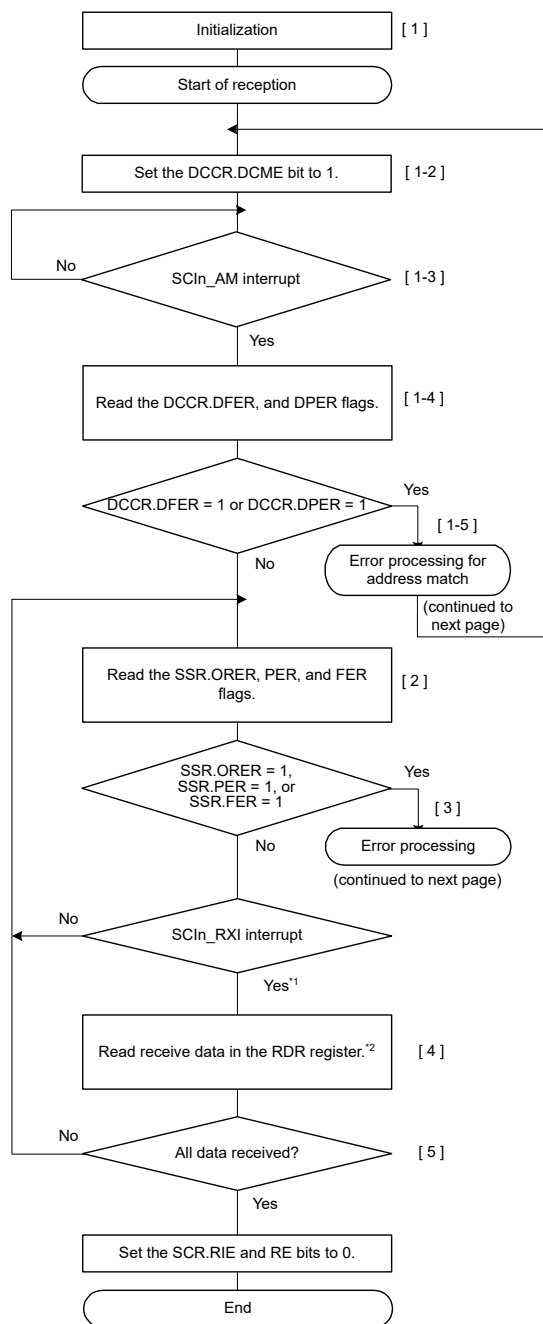


Figure 32.16 Example flow of serial reception in asynchronous mode with non-FIFO selected and Address Matching Disabled (2)



[1] SCI initialization:
Set data reception.

[1-2] Address match cycle:
Set the DCCR.DCME bit to 1 to enable address matching.

[1-3] Comparing receive data and data for comparison (CDR.CMPD):
Compare receive data and data for comparison (CDR.CMPD).
The SCI cannot detect a parity or framing error as receive data are skipped (discarded) until the SCI detects a match between the receive data and data for comparison.
If the SCI detects an address match, the DCCR.DCME bit is cleared, the DCCR.DCMF flag becomes 1, and an SCIn_AM interrupt request is issued.

[1-4] [1-5] Reception error processing:
An error is identified by reading the DFER or DPER flag in DCCR in the SCIn_AM interrupt handling routine.
After appropriate handling of the error, set both the DFER and DPER flags to 0.
If either of these flags is set to 1, address matching cannot be resumed.

[2] [3] Reception error processing and break detection:
When a reception error occurs, the SCIn_ERI interrupt is generated.
The error is identified by reading the ORER, PER, and FER flags in the SSR.
After appropriate handling of the error, set all of the ORER, PER, and FER flags to 0. If any of these flags is set to 1, reception cannot be resumed.
If a framing error occurs, a break can be detected by reading the value of the input port bit corresponding to the RXDn pin.

[4] Read the receive data in the RDR register from within the SCIn_RXI interrupt handling routine.

[5] Procedure for continuing serial reception:
To continue serial reception, read the value of the RDR register in the SCIn_RXI interrupt handling routine before the stop bit is received.
The value of the RDR register can also be read by activating the DMAC or DTC.

Note 1. Be sure to set the RE bit to 0 before reading the RDR register.

Note 2. The RDRHL register is used instead of the RDR register when the 9-bit data length is selected.

Figure 32.17 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO not Selected and Address Matching Enabled) (1)

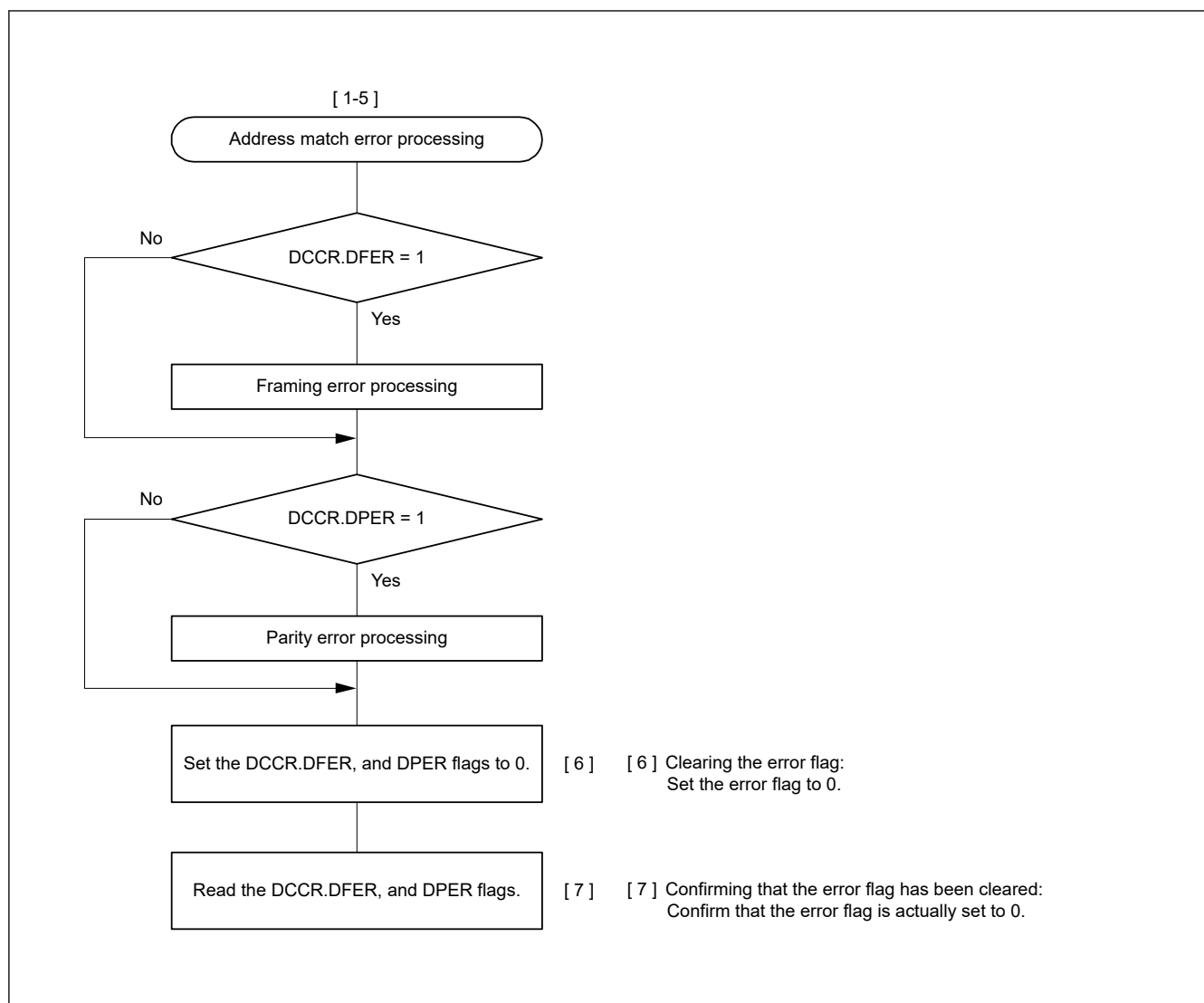


Figure 32.18 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO not Selected and Address Matching Enabled) (2)

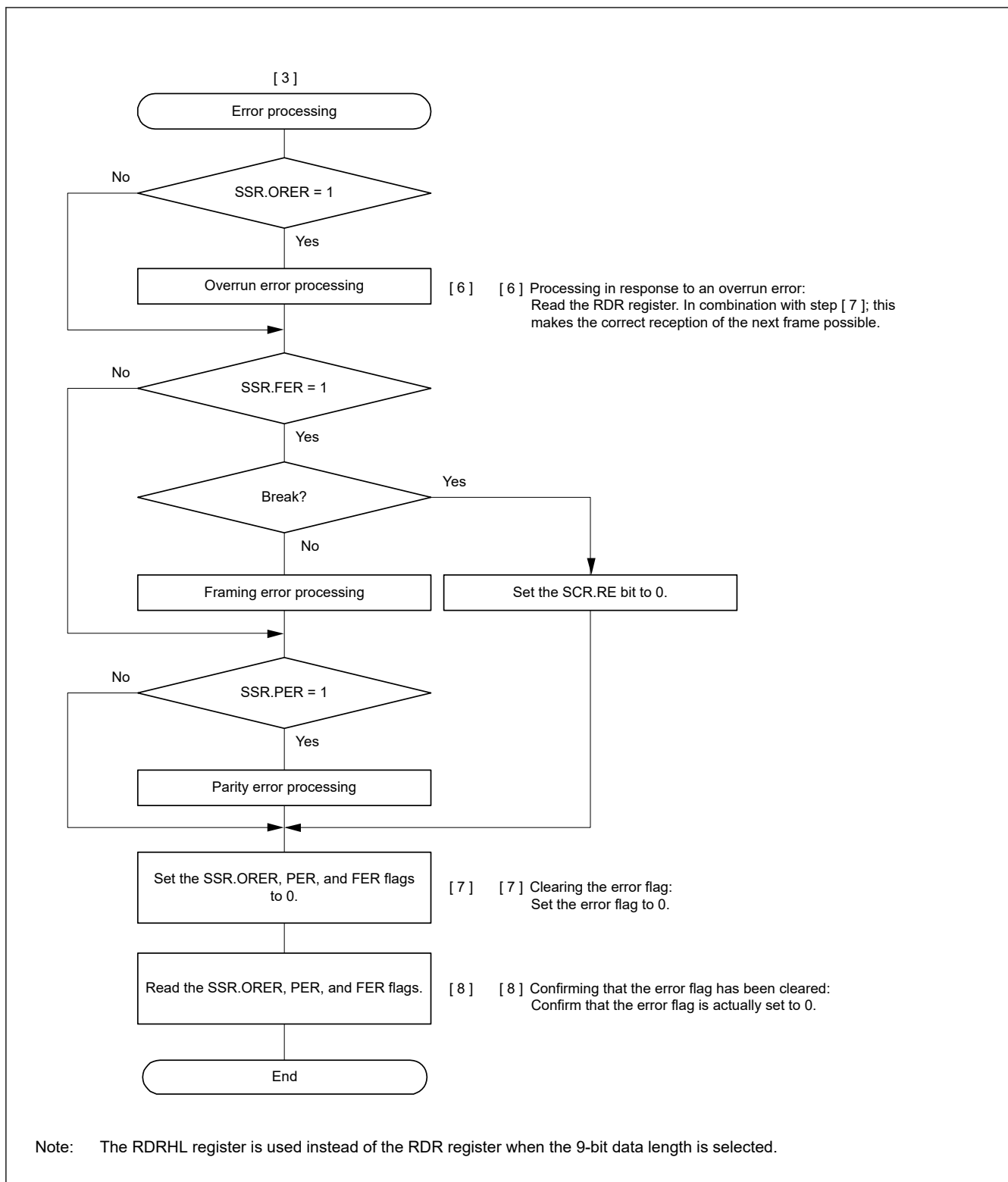


Figure 32.19 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO not Selected and Address Matching Enabled) (3)

(2) FIFO selected

Figure 32.20 shows an example of a data format that is written to FRDRH register and FRDRL register in asynchronous mode.

In asynchronous mode, 0 is written to the MPB bit in the FRDRH register. Data that corresponds to the data length is written to FRDRH and FRDRL. Unused bits are written as 0. Read in order from FRDRH to FRDRL. If software reads FRDRL, the

SCI updates FER, PER, and receive data (RDAT[8:0]) in the FRDRL register with the next data. The flags RDF, ORER, and DR in the FRDRH register always reflect the associated flags in the SSR_FIFO register.

Data Length	Register Setting		Receive data in FRDRH, FRDRL															
	SCMR. CHR1	SMR. CHR	FRDRHL															
			FRDRH								FRDRL							
			b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	0	—	RDF	ORER	FER	PER	DR	0	0	0							
8 bits	1	1	—	RDF	ORER	FER	PER	DR	0	0								
9 bits	0	Don't care	—	RDF	ORER	FER	PER	DR	0									

Note: 0 is always read for MPB bit (FRDRH[1])
 When data length is 7 bits, 0 is always read for FRDRH[0] and FRDRL[7]
 When data length is 8 bits, 0 is always read for FRDRH[0]
 FRDRH[7] bit is read as an indefinite value

Figure 32.20 Data format stored in FRDRH and FRDRL with FIFO selected

In serial data reception, the SCI operates as follows:

1. When the value of the SCR.RE bit becomes 1, the output signal on the RTSn pin goes low.
2. The SCI monitors the communications line and, when it detects a start bit, the SCI performs internal synchronization, stores receive data in the RSR register.
3. If the multi-processor communications function is enabled (SMR.MP = 1), see [section 32.4.2. Multi-Processor Serial Data Reception](#). If the address match function (data compare match function) is enabled (DCCR.DCME = 1), the SCI cannot detect a parity or framing error as receive data are skipped (discarded) until the SCI detects a match between the receive data and the data for comparison (CDR.CMPD^{*1}).
4. If the SCI detects an address match, the DCCR.DCME bit is automatically cleared, the DCCR.DCMF flag becomes 1, and an SCIn_AM interrupt^{*2} request is generated. To enable the generation of an SCIn_RXI interrupt request, set the SCR.RIE bit to 1. The compared receive data are not stored in the RDR register^{*3}. The SSR.RDRF flag remains 0.
5. If the SCI detects a framing error in the receive data for which an address match was detected, the DCCR.DFER flag is set to 1, and if the SCI detects a parity error in that frame, the DCCR.DPER flag becomes 1. To enable the generation of an SCIn_ERI interrupt request, set the SCR.RIE bit to 1.
6. If a framing or a parity error is detected (the DCCR.DFER flag or DCCR.DPER flag is 1) in the SCIn_AM interrupt handling routine, set the DCCR.DFER and DCCR.DPER flags to 0 and set the DCCR.DCME bit to 1 to enable the address match function again. If neither a framing nor a parity error has been detected (the DCCR.DFER and DCCR.DPER flags are 0), set the DCCR.DCMF flag to 0. See [Figure 32.5](#).
7. If an overrun error occurs during normal communications, the SSR_FIFO.ORER flag is set to 1. If the SCR.RIE bit in SCR is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the FRDRL^{*3} register.
8. If a parity error is detected, the PER flag and receive data are transferred to the FRDRL^{*3} register. If the SCR.RIE bit is set to 1, an SCIn_ERI interrupt request is generated.
9. If a framing error is detected, the FER flag and receive data are transferred to the FRDRL^{*3} register. If the SCR.RIE bit is set to 1, an SCIn_ERI interrupt request is generated.
10. After a framing error is detected and when SCI detects that the continuous receive data is for one frame, reception stops.
11. When the amount of data stored in the FRDRL register falls below the specified receive triggering number, and the next data is not received after 15 ETUs from the last stop bit in asynchronous mode, the SSR_FIFO.DR flag is set to 1. When the SCR.RIE bit is 1 and the FCR.DRES bit is 0, the SCI generates an SCIn_RXI interrupt request. When the FCR.DRES bit is 1, SCI generates an SCIn_ERI interrupt request.
12. When reception finishes successfully, receive data is transferred to the FRDRL^{*3} register. The RDF bit is set to 1 when the amount of receive data written to FRDRHL is equal to or greater than the specified receive triggering number. If the

SCR.RIE bit in SCR is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the FRDRL^{*4} register in the SCIn_RXI interrupt handling routine, before an overrun error occurs. If the received data that is transferred to FRDRL^{*5} is less than the RTS trigger number, the RTSn pin outputs low.

- Note 1. One of three lengths is selected for the target for comparison: CMPD[6:0] is for 7-bit length, CMPD[7:0] is for 8-bit length, and CMPD[8:0] is for 9-bit length.
- Note 2. As no interrupt enable bit is assigned to the SCIn_AM interrupt, an interrupt request is generated by setting the DCCR.DCMF to 1.
- Note 3. Only read data in the FRDRH and FRDRL registers when 9-bit data length is selected.
- Note 4. Read data in order from FRDRH to FRDRL when 9-bit data length is selected.
- Note 5. The SCI only checks for update to the FRDRL register and not to the FRDRH register when 9-bit data length is selected.

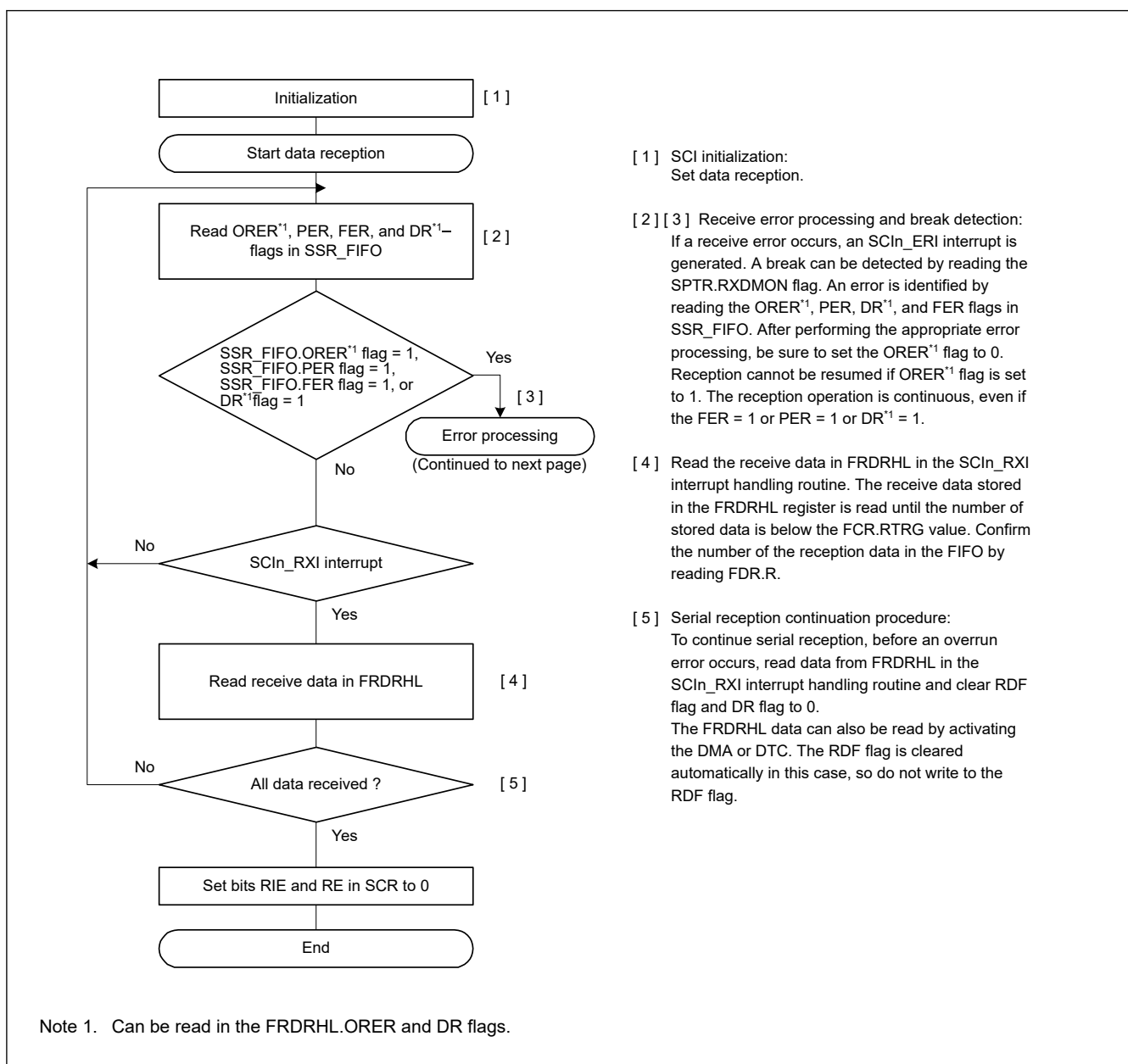


Figure 32.21 Example flow of serial reception in asynchronous mode with FIFO selected and Address Matching Enabled (1)

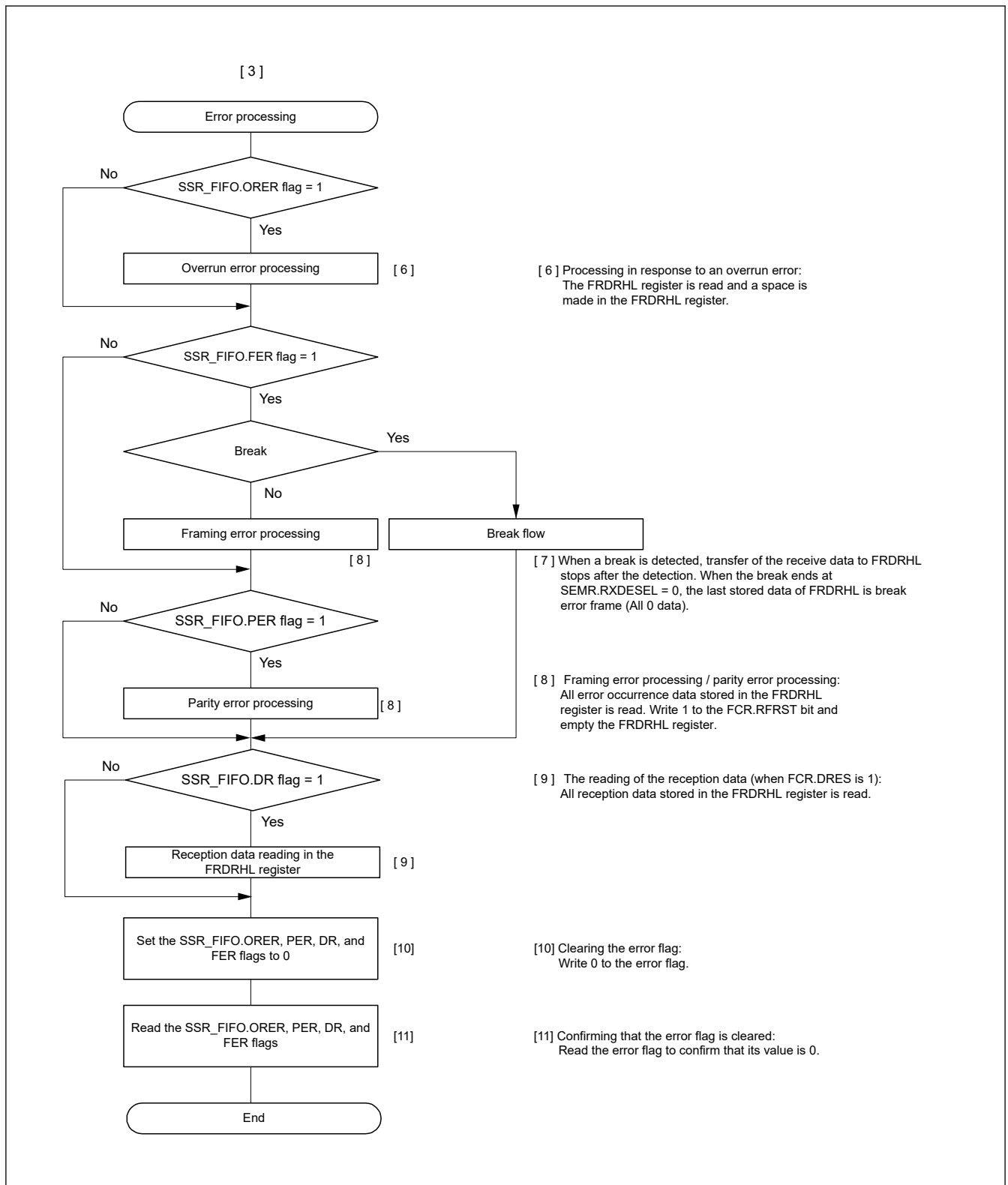
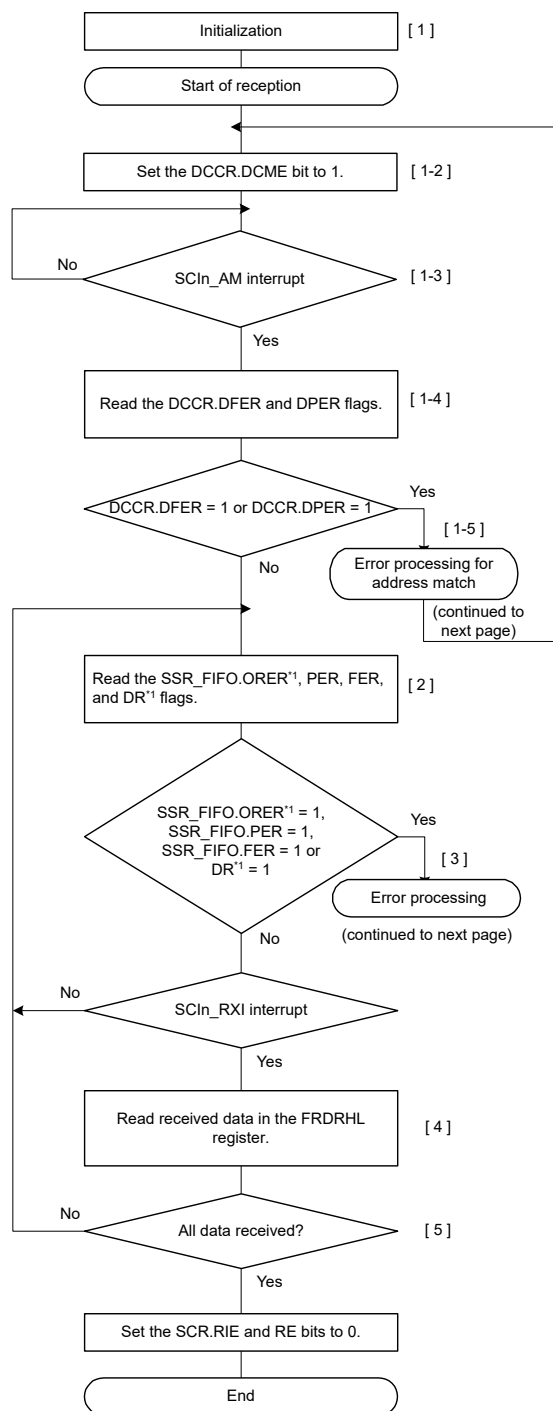


Figure 32.22 Example flow of serial reception in asynchronous mode with FIFO selected Address Matching Disabled (2)



[1] SCI initialization:
Set data reception.

[1-2] Address match cycle:
Set the DCCR.DCME bit to 1 to enable address matching.

[1-3] Comparing received data and data for comparison (CDR.CMPD):
Compare receive data and comparison data.
The SCI cannot detect a parity or framing error as received data are skipped (discarded) until the SCI detects a match between the received data and data for comparison.
If the SCI detects an address match, the DCCR.DCME bit is cleared, the DCCR.DCMF flag becomes 1, and an SCIn_AM interrupt request is issued.

[1-4] [1-5] Reception error processing:
An error is identified by reading the DFER or DPER flag in DCCR from within the SCIn_AM interrupt handling routine.
After appropriate handling of the error, set both the DFER and DPER flags to 0.
If either of these flags is set to 1, address matching cannot be resumed.

[2] [3] Reception error processing and break detection:
When a reception error occurs, an SCIn_ERI interrupt is generated.
A break is identified by reading SPTR.RXDMON.
The error is identified by reading each among the SSR_FIFO.ORER*1, PER, FER, and DR*1 flags.
After appropriate handling of the error, set the ORER*1 flag to 0.
If the ORER*1 flag remains 1, reception cannot be resumed.
The reception operation is continuous, even if the FER = 1, PER = 1 or DR*1 = 1.

[4] Reading the received data in the FRDRHL register from within the SCIn_RXI interrupt handling routine:
The received data stored in the FRDRHL register are read until the amount of stored data is less than the value of the FCR.RTRG[3:0] bits.
Confirm the number of received data in the FIFO with the value of the FDR.R[4:0] bits.

[5] Serial reception continuation procedure:
To continue with serial reception, read data from the FRDRHL register from within the SCIn_RXI interrupt handling routine before an overrun error occurs, and set the RDF and DR flags to 0. The data in the FRDRHL register can also be read by activating the DMAC or DTC. In this case, do not write to the RDF flag because the flag is automatically cleared.

Note 1. The FRDRHL.ORER and DR flags can also be read.

Figure 32.23 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO Selected and Address Matching Enabled) (1)

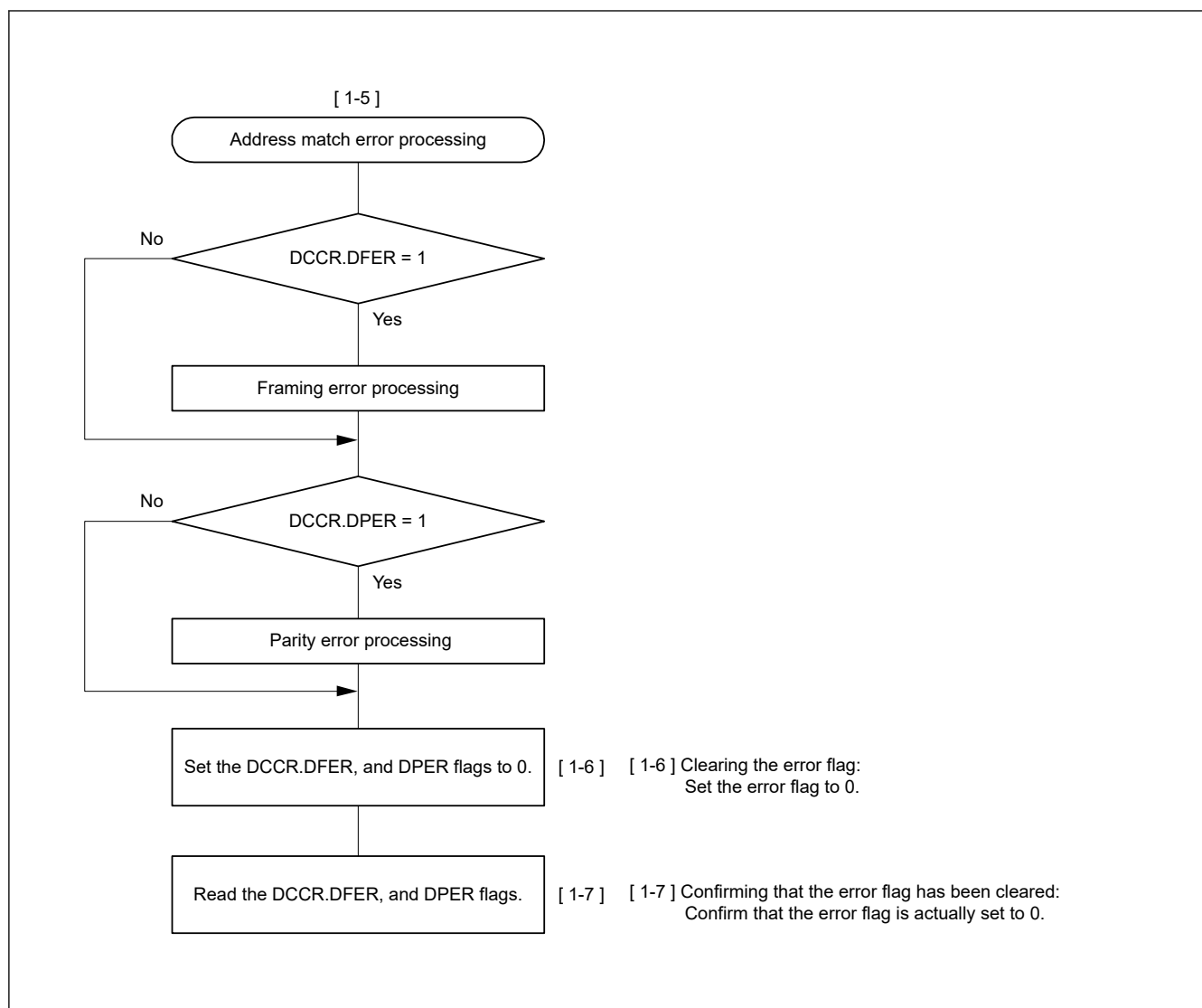


Figure 32.24 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO Selected and Address Matching Enabled) (2)

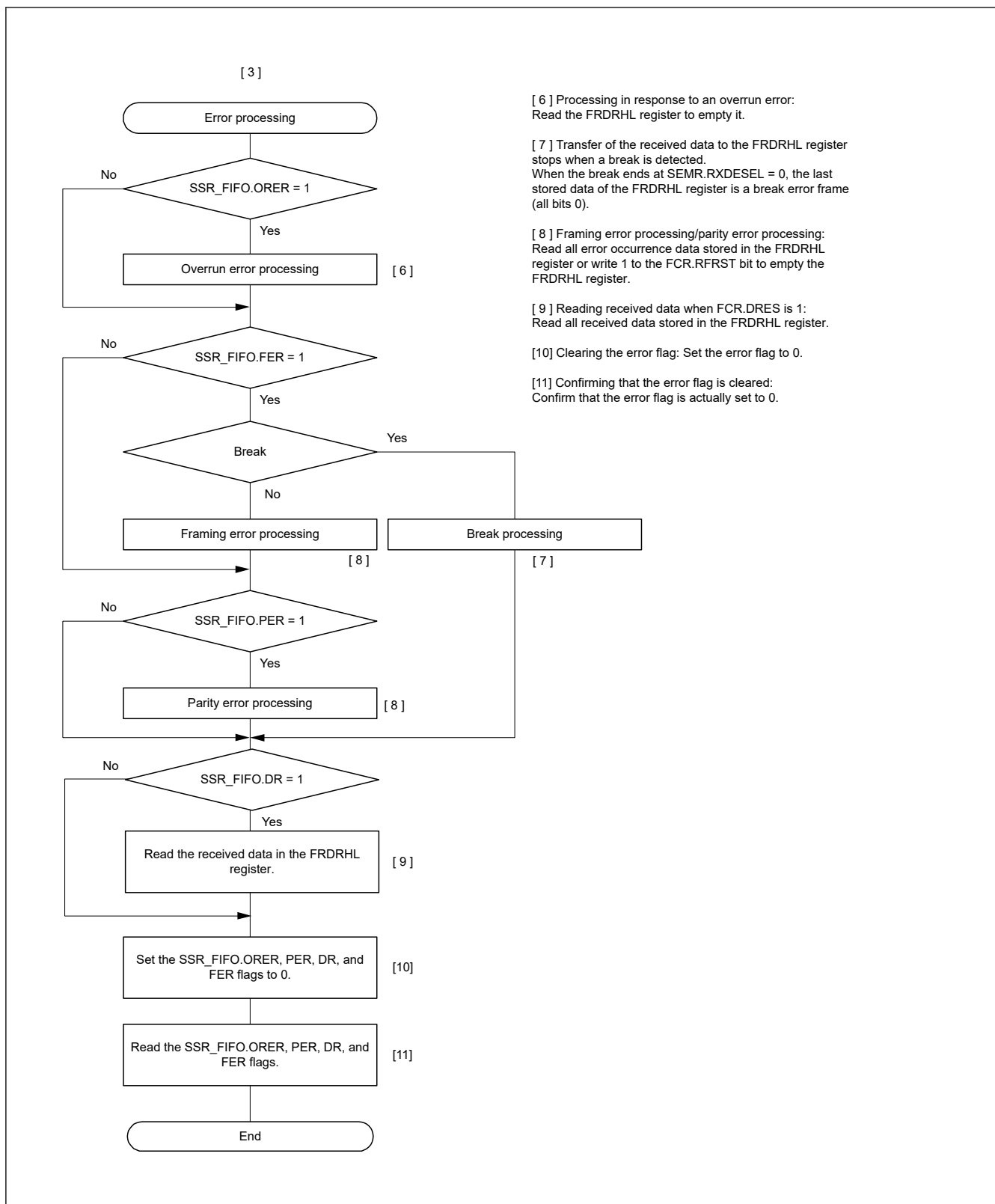


Figure 32.25 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO Selected and Address Matching Enabled) (3)

32.4 Multi-Processor Communication Function

The multi-processor communication function enables the SCI to transmit and receive data between multiple processors by sharing an asynchronous serial communication line that has an added multi-processor bit. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID

transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station.

The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle:

- When the multi-processor bit is set to 1, the transmission cycle is the ID transmission cycle
- When the multi-processor bit is set to 0, the transmission cycle is the data transmission cycle

Figure 32.26 shows an example of communication between processors using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits communication data in which the multi-processor bit set to 0 is added to the transmit data. After receiving communication data with the multi-processor bit set to 1, the receiving station compares the received ID with the ID of the receiving station itself. If the two match, the receiving station receives communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until it receives data in which the multi-processor bit is set to 1.

(1) Non-FIFO selected

To support this function, the SCI provides the SCR.MPIE bit. When the MPIE bit is set to 1, the following operations are disabled until the reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from the RSR register to the RDR register (the RDRHL register when 9-bit data length is selected)
- Detection of a receive error
- Setting of the respective RDRF, ORER, and FER status flags in the SSR register

When the SCI receives a character in which the multi-processor bit is set to 1, the SSR.MPBT bit is set to 1 and the SCR.MPIE bit is automatically cleared, returning the SCI to normal reception operation. If the SCR.RIE bit is set to 1, an SCIn_RXI interrupt is generated.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in normal asynchronous mode. The clock used for the multi-processor communication is the same as the clock used in normal asynchronous mode.

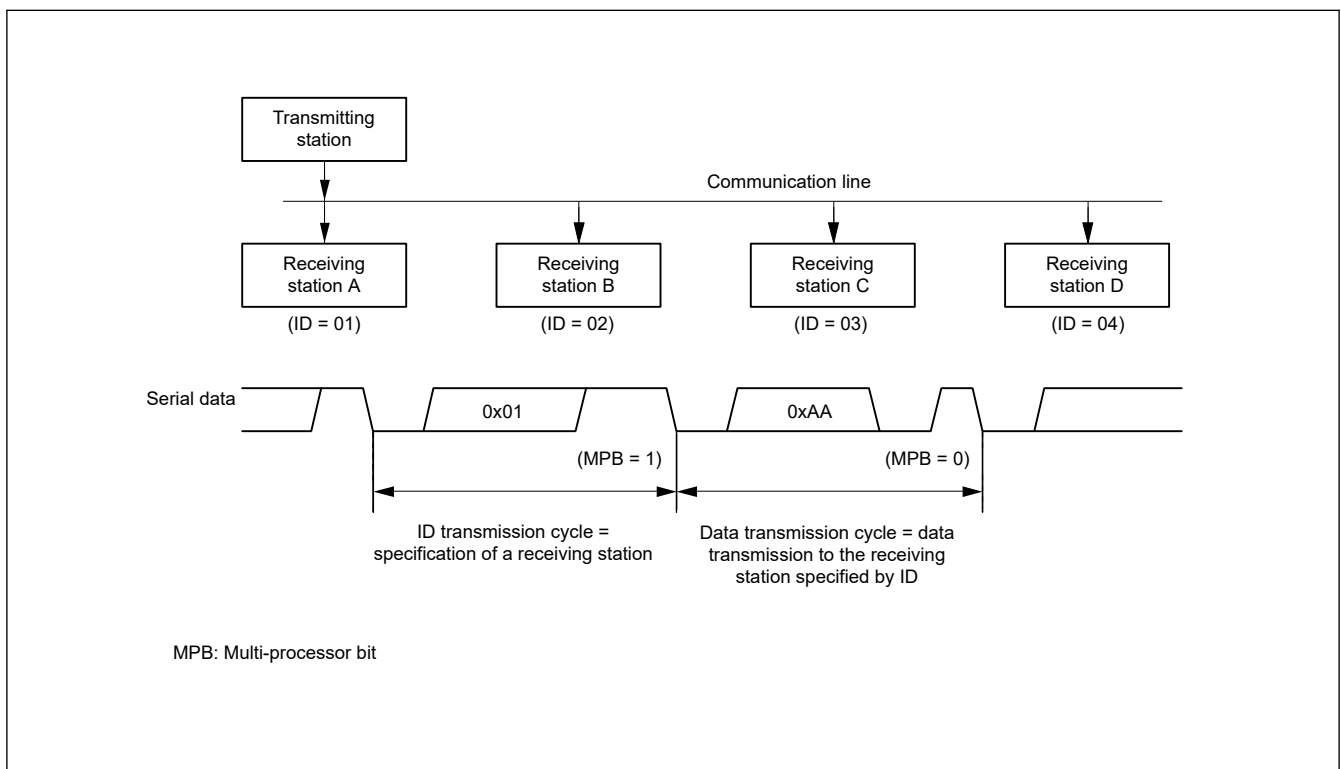


Figure 32.26 Example of communication using multi-processor format with transmission of data 0xAA to receiving station A

(2) FIFO selected

For data transmission, software must write data to FTDRHL.MPBT that corresponds to transmit data in FTDRHL.TDAT. For data reception, the multi-processor bit that is part of the receive data is written to FTDRHL.MPB and receive data is written to FRDRL.

When the MPIE bit is set to 1, the following operations are disabled until reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from the RSR register to the FRDRHL register
- Detection of a receive error
- Break
- Setting of the respective RDE, ORER, and FER status flags in the SSR_FIFO register

When the SCI receives an 8-bit character in which the multi-processor bit is set to 1, the FTDRHL.MPB bit is set to 1 and receive data is written to FRDRHL.RDAT. The SCR.MPIE bit is automatically cleared, returning the SCI to normal reception operation. If the SCR.RIE bit is set to 1, an SCIn_RXI interrupt is generated.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in normal asynchronous mode with non-FIFO selected.

32.4.1 Multi-Processor Serial Data Transmission

(1) Non-FIFO selected

[Figure 32.27](#) shows an example flow of multi-processor data transmission. In the ID transmission cycle, the ID must be transmitted with the SSR.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as operations in asynchronous mode. Write the values in the order of the FTDRH register then the FTDRL register.

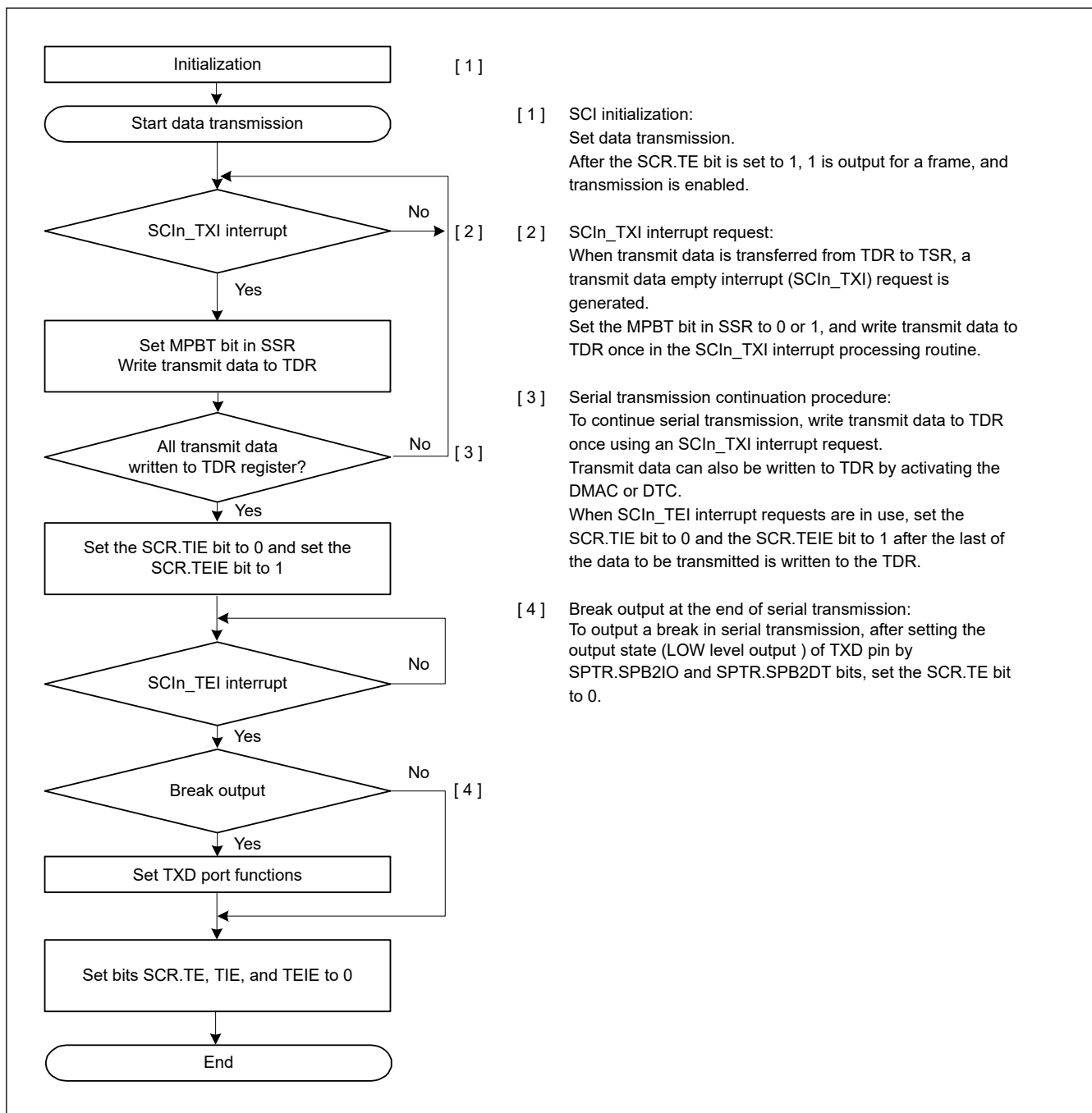


Figure 32.27 Example flow of multi-processor serial transmission with non-FIFO selected

(2) FIFO selected

Figure 32.28 shows an example of data format that is written to FTDRH and FTDRL in multi-processor mode. The FTDRH.MPBT bit is set to 1. Data is set to FTDRH and FTDRL with the correct data length. Write 0 for unused bits. Write in order from FTDRH to FTDRL.

Data Length	Register Setting		Transmit data in FTDRH, FTDRL															
			FTDRHL															
	SCMR. CHR1	SMR. CHR	FTDRH								FTDRL							
			b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	0	—	—	—	—	—	—	MPBT	—	—							
8 bits	1	1	—	—	—	—	—	—	MPBT	—								
9 bits	0	Don't care	—	—	—	—	—	—	MPBT									

Note: —: Invalid. The write value should be 0.

Figure 32.28 Data format written to FTDRH and FTDRL in multi-processor mode with FIFO selected

Figure 32.29 shows an example flow of multi-processor serial transmission with FIFO selected. In the ID transmission cycle, the ID must be transmitted with the FTDRH.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as operations in asynchronous mode with non-FIFO selected.

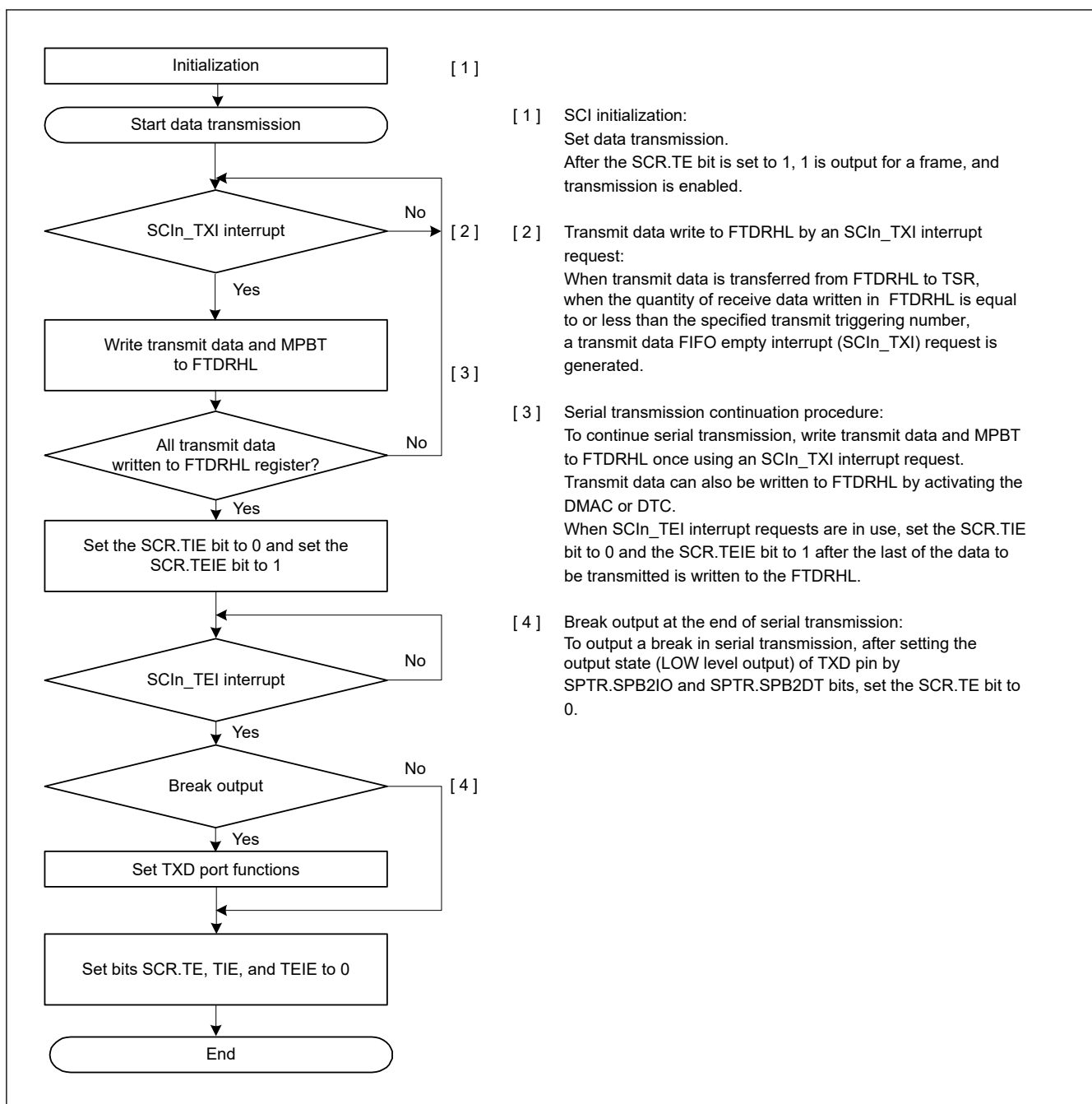


Figure 32.29 Example flow of serial transmission in multi-processor mode with FIFO selected

32.4.2 Multi-Processor Serial Data Reception

(1) Non-FIFO selected

Figure 32.31 and Figure 32.32 are example flows of multi-processor serial reception. When the SCR.MPIE bit is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to the RDR register (the RDRHL register when 9-bit data length is selected), and the SCIn_RXI interrupt request is generated. The rest of the operations are the same as operations in asynchronous mode. Read the order from FRDRH to FRDRL.

Figure 32.30 shows an example operation for data reception.

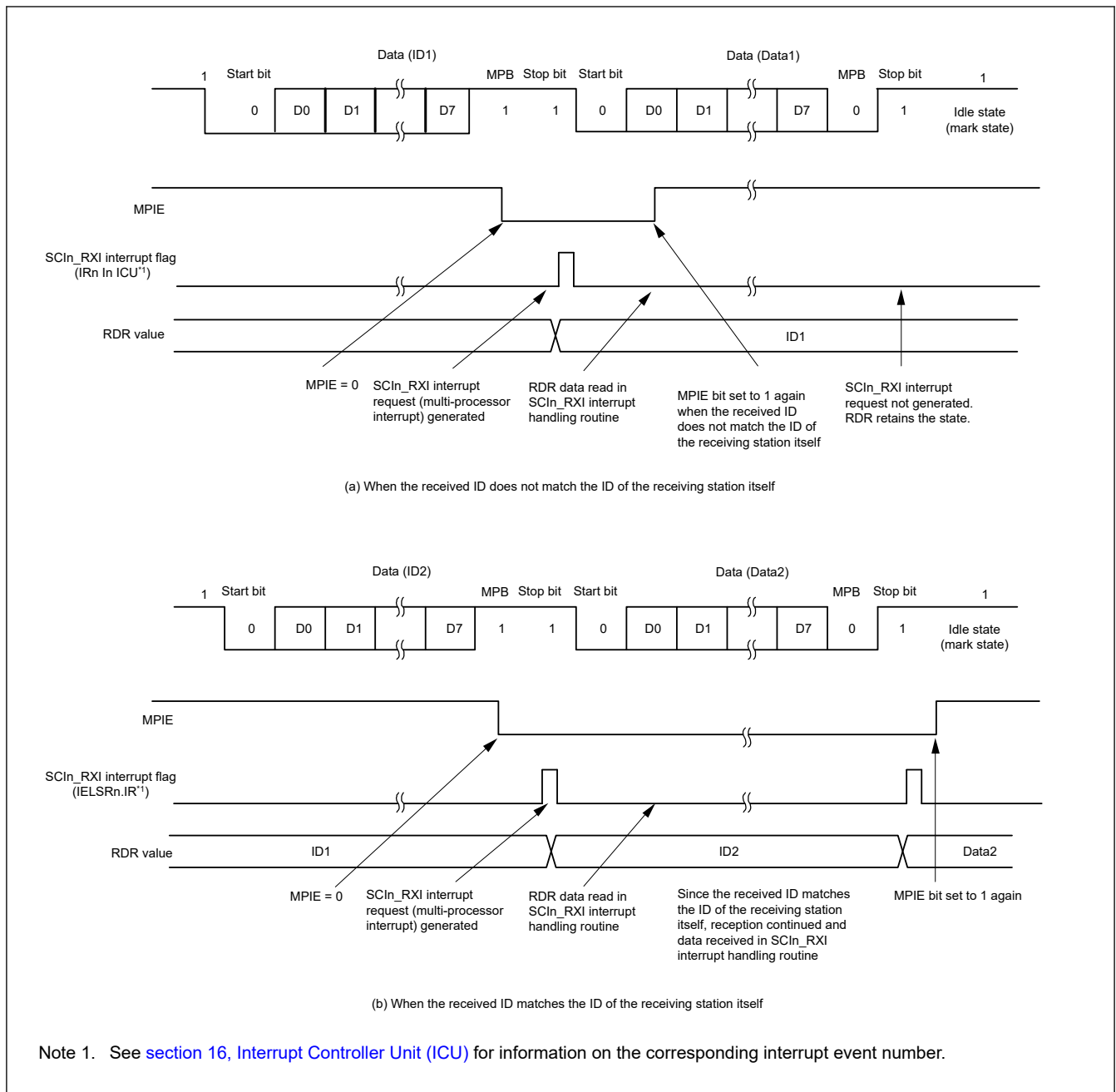


Figure 32.30 Example of SCI reception with 8-bit data, multi-processor bit, and 1 stop bit

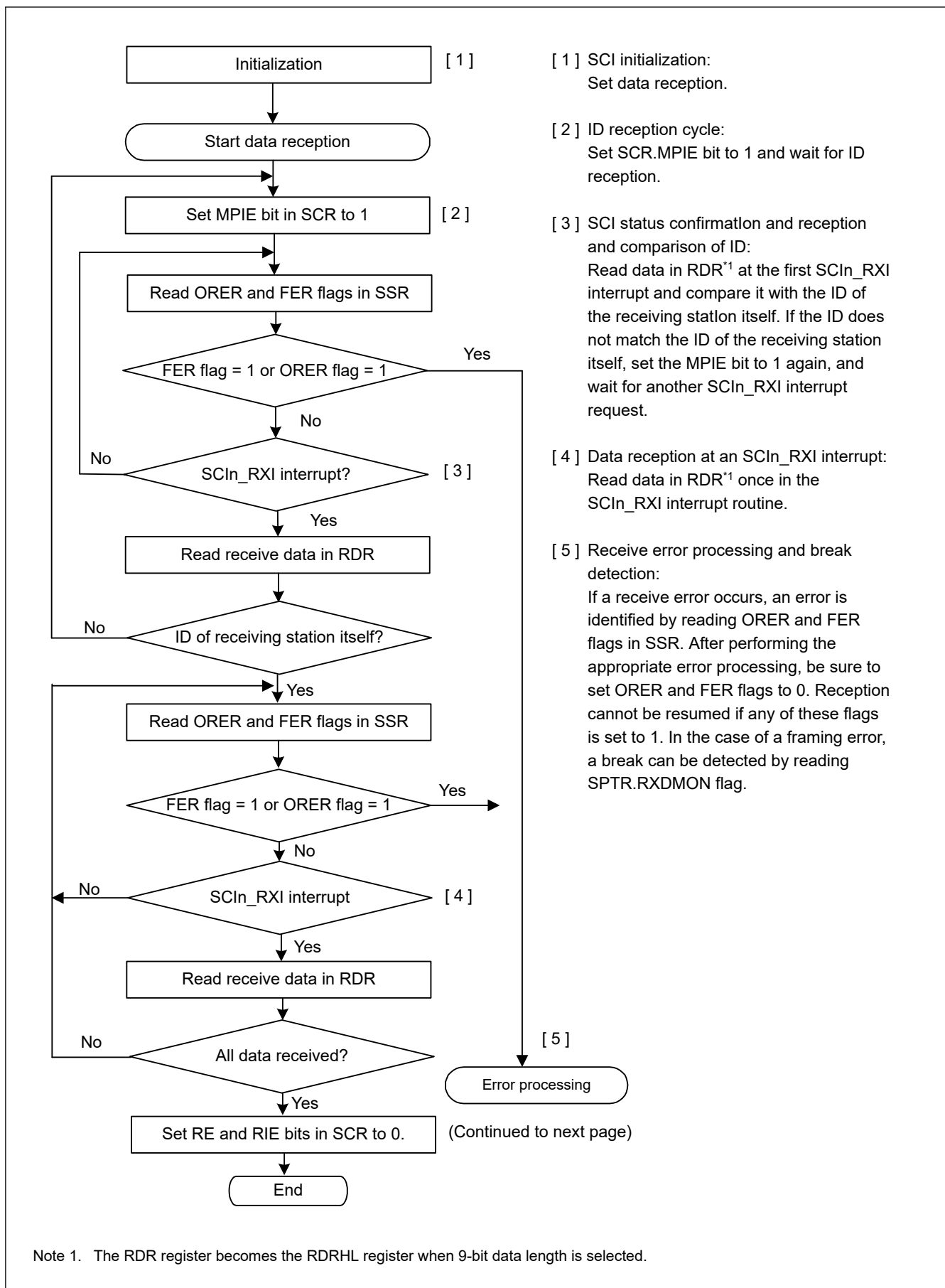


Figure 32.31 Example flow of multi-processor serial reception with non-FIFO selected (1)

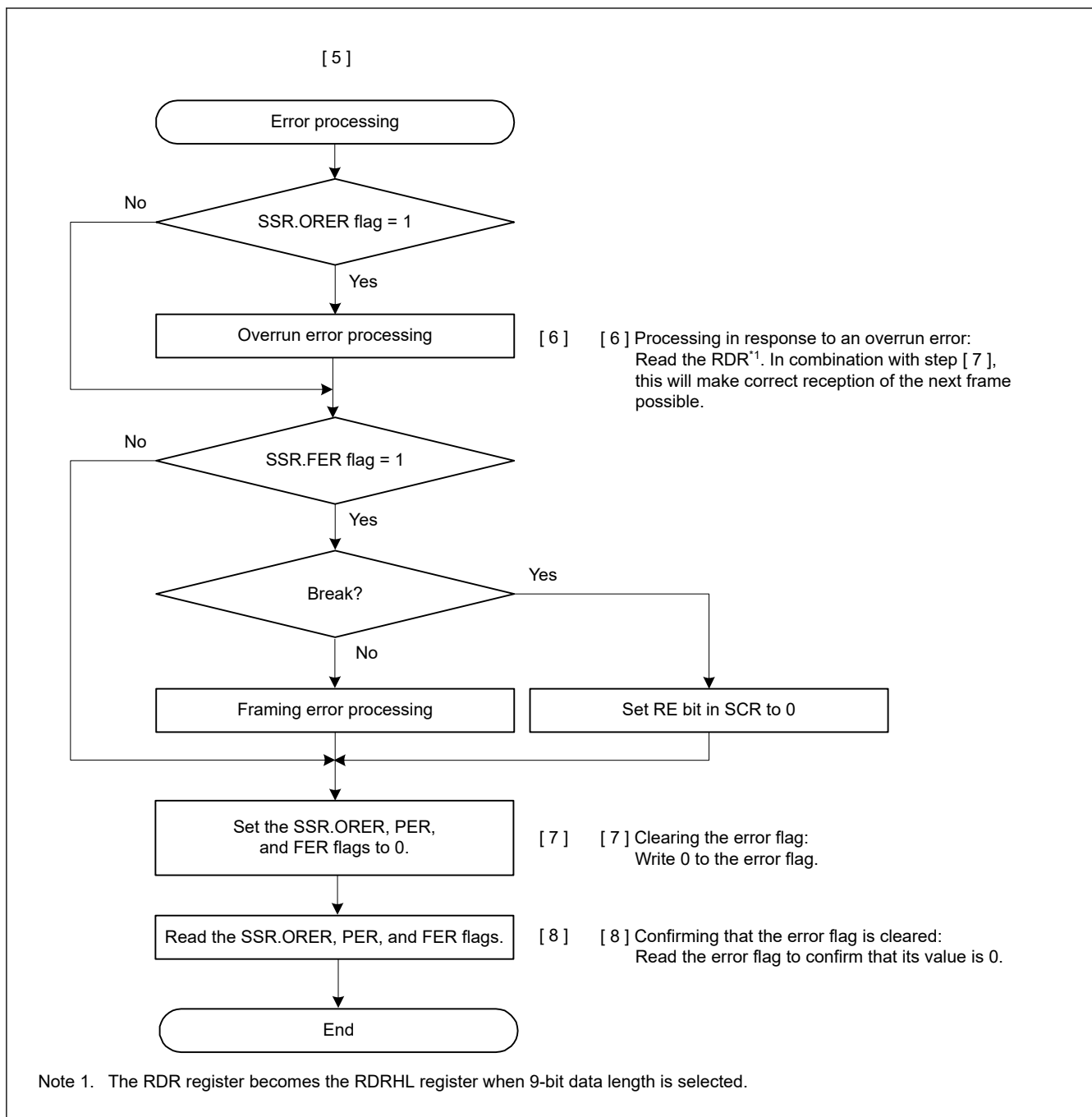


Figure 32.32 Example flow of multi-processor serial reception with non-FIFO selected (2)

(2) FIFO selected

Figure 32.33 shows an example of a data format that is written to FRDRH and FRDRL in multi-processor mode.

In multi-processor mode, the MPB value that is a part of the receive data is written to the FRDRH.MPB bit. A value of 0 is written to the FRDRH.PER flag. Data is written to FRDRH and FRDRL with the correct data length. Unused bits are written with 0. Read in order from FRDRH to FRDRL. When software reads the FRDRL register, the SCI updates FER, MPB, and receive data (RDAT[8:0]) in FRDRL with the next data. The RDF, ORER and DR flags in the FRDRH register always reflect the associated flags in the SSR_FIFO register.

Data Length	Register Setting		Receive data in FRDRH, FRDRL															
			FRDRHL															
	SCMR. CHR1	SMR. CHR	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	0	—	RDF	ORER	FER	0	DR	MPB	0	0							
8 bits	1	1	—	RDF	ORER	FER	0	DR	MPB	0								
9 bits	0	Don't care	—	RDF	ORER	FER	0	DR	MPB									

Note: When data length is 7 bits, 0 is always read for FRDRH[0] and FRDRL[7]
 When data length is 8 bits, 0 is always read for FRDRH[0]
 FRDRHL[15] bit is read as an indefinite value

Figure 32.33 Data format stored in FRDRH and FRDRL in multi-processor mode with FIFO selected

Figure 32.34 shows an example flow of multi-processor data reception with FIFO selected. When the SCR.MPIE is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data, MPB and associated errors are transferred to the FRDRHL register. The SCR.MPIE bit is automatically cleared and normal reception continues.

If a framing error occurs and the SSR_FIFO.FER flag is set to 1, the SCI continues data reception. The rest of the operations are the same as operations in asynchronous mode with non-FIFO selected.

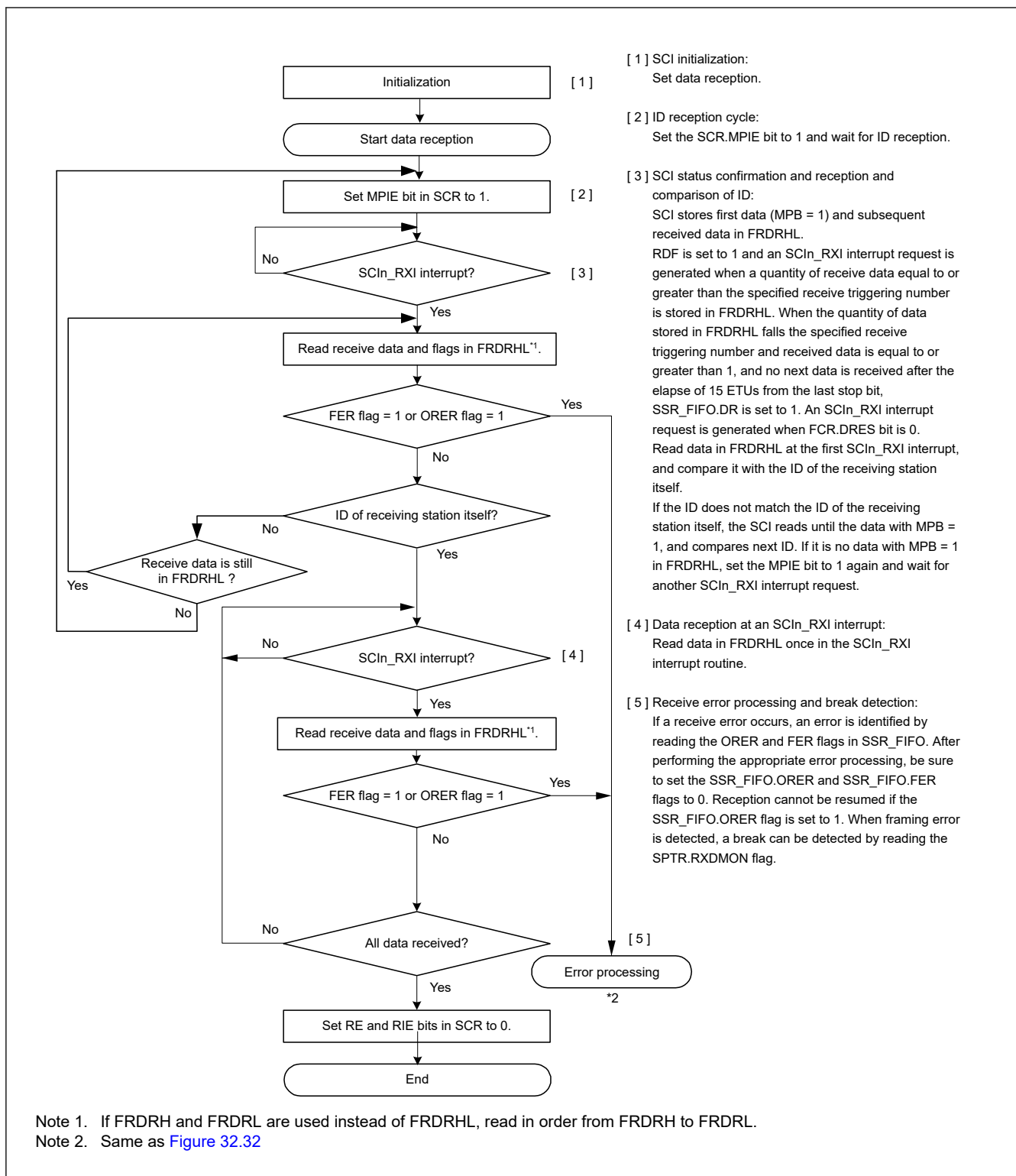


Figure 32.34 Example flow of serial reception in multi-processor mode with FIFO selected

32.5 Operation in Clock Synchronous Mode

Figure 32.35 shows the data format for clock synchronous serial data communications.

In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In clock synchronous mode, no parity bit can be added.

In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the

transmission line holds the last bit as output state. When the SPMR.CKPH bit is 1 in slave mode, the transmission line holds the first bit output state.

Within the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by using a shared clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

However, it is not possible to perform continuous transfer in the fastest bit rate setting (BRR[7:0] = 0x00 and SMR.CKS[1:0] = 00b). Therefore, when the FIFO is selected, this setting (BRR[7:0] = 0x00 and SMR.CKS[1:0] = 00b) is not available.

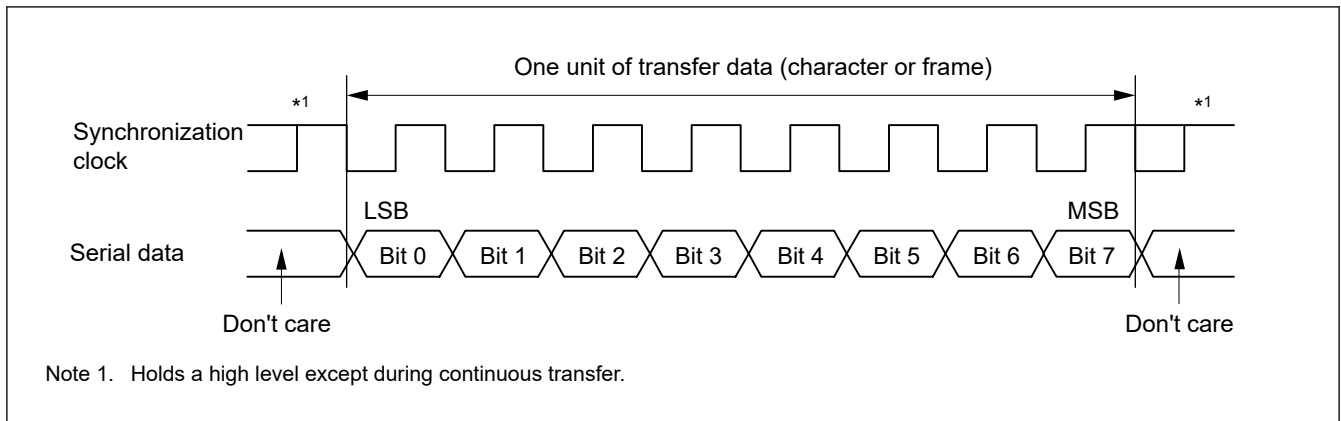


Figure 32.35 Data format in clock synchronous serial communications with LSB-first order

32.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCKn pin can be selected based on the SCR.CKE[1:0] setting.

When the SCI operates on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character. When no transfer is performed, the clock is held high. However, when only data reception is performed while the CTS function is disabled, the synchronization clock output starts when the SCR.RE bit set to 1. The synchronization clock stops when it goes high*1 and an overrun error occurs or the SCR.RE bit is set to 0.

When only data reception is performed and the CTS function is enabled, the clock output does not start when the SCR.RE bit set to 1 and the CTSn pin input is high. The synchronization clock output starts when the SCR.RE bit is set to 1 and the CTSn pin input is low. Following that, when the CTSn pin input is high on completion of the frame reception, the synchronization clock output stops when it goes high. If the CTSn pin input continues to be low, the synchronization clock stops when it goes high*1 and an overrun error occurs or the SCR.RE bit is set to 0.

Note 1. The signal is held high while (SPMR.CKPH = 0 and SPMR.CKPOL = 1) or (SPMR.CKPH = 1 and SPMR.CKPOL = 1). It is held low while (SPMR.CKPH = 0 and SPMR.CKPOL = 1) or (SPMR.CKPH = 1 and SPMR.CKPOL = 0).

32.5.2 CTS and RTS Functions

In the CTS function, the CTSn 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 pin low causes data reception or transmission to start.

Setting the CTSn 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 RTSn pin output is used to request the start of data reception or transmission when the clock source is an external synchronizing clock. The RTSn output goes low when serial communication becomes possible. Conditions for output of the RTSn low and high are shown as follows:

[Conditions for low output]

Satisfaction of all the following conditions:

Non-FIFO selected when all of the following conditions are satisfied

- The value of the SCR.RE bit or the SCR.TE bit is 1
- Neither transmission nor reception is in progress
- There is no received data available to be read when the SCR.RE bit is 1
- Transmit data is written when the SCR.TE bit is 1 and SCR.CKE[1] bit is 0
- Data is available for transmission in the TSR register when SCR.TE bit is 1 and SCR.CKE[1] bit is 1
- The SSR.ORER flag is 0

FIFO selected when all of the following conditions are satisfied

- The value of the SCR.RE bit or the SCR.TE bit is 1
- Neither transmission nor reception is in progress
- The amount of receive data written in FRDRHL is less than the setting value of FCRH.RSTRG[3:0] when SCR.RE = 1
- Data that has not been transmitted is available in FTDRHL when SCR.TE bit is 1 and SCR.CKE[1] bit is 0
- Data is available for transmission in the TSR register when SCR.TE bit is 1 and SCR.CKE[1] bit is 1
- The SSR_FIFO.ORER flag is 0

[Condition for high output]

- The conditions for low output are not satisfied

32.5.3 SCI Initialization in Clock Synchronous Mode

Before transmitting and receiving data, start by writing the initial value 0x00 to the SCR register, then continue through the SCI initialization procedure given in the sections describing non-FIFO and FIFO selection in [section 32.5.2. CTS and RTS Functions](#). Anytime the operating mode or transfer format is to be changed, the SCR register must be initialized before the change can be made.

Note: Setting the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in SSR/SSR_FIFO nor the RDR register. When the TE bit is set to 0, the TEND flag for the selected FIFO buffer is not initialized.

Note: In non-FIFO mode, switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 when the SCR.TIE bit is 1 generates an SCIn_TXI interrupt request.

Table 32.29 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected (1 of 2)

No.	Step Name	Description
1	Start initialization	
2	Set the SCR.TIE, RIE, TE, RE, and TEIE bits to 0	
3	Set the FCR.FM bit to 0	Set the FCR.FM bit to 0.
4	Set the SCR.CKE[1:0] bits	Set the clock selection in SCR.
5	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits.	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits. Step 5 can be skipped if the values have not been changed from the initial values.
6	Set the data transmission/reception format in SMR, SCMR, and SEMR	Set data transmission/reception format in SMR, SCMR, and SEMR.
7	Set a value in BRR	Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.
8	Set a value in MDDR	Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is set to 0 or an external clock is used.
9	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

Table 32.29 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected (2 of 2)

No.	Step Name	Description
10	Set the SCR.TE or RE bit to 1, and set the SCR.TIE and RIE bits	Set the SCR.TE or RE bit to 1. Also set the SCR.TIE and RIE bits. Setting the TE and RE bits allows TXDn and RXDn to be used.
11	Initialization completion	

Note: In simultaneous transmit and receive operations, the TE and RE bits in SCR must both be set to 0 or set to 1 simultaneously

Table 32.30 Example flow of SCI initialization in clock synchronous mode with FIFO selected

No.	Step Name	Description
1	Start initialization	
2	Set the SCR.TIE, RIE, TE, RE, and TEIE bits to 0	
3	Set the FCR.FM, TFRST, and RFRST bits to 1. Set the FCR.TTRG[3:0], RTRG[3:0], and RSTRG[3:0] bits.	Set the FCR.FM, TFRST, and RFRST bits to 1 (FIFO mode enabled, transmit/receive FIFOs empty). Set the FCR.TTRG[3:0], RTRG[3:0], and RSTRG[3:0] bits.
4	Set the SCR.CKE[1:0] bits	Set the clock selection in SCR.
5	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits.	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits. Step 5 can be skipped if the values have not been changed from the initial values.
6	Set the data transmission/reception format in SMR, SCMR, and SEMR	Set data transmission/reception format in SMR, SCMR, and SEMR.
7	Set a value in BRR	Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.
8	Set a value in MDDR	Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is set to 0 or an external clock is used.
9	Set the FCR.TFRST and RFRST bits to 0	Set the FCR.TFRST and RFRST bits to 0.
10	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
11	Set the SCR.TE or RE bit to 1, and set the SCR.TIE and RIE bits	Set the SCR.TE or RE bit to 1. Also set the SCR.TIE and RIE bits. Setting the TE and RE bits allows TXDn and RXDn to be used.
12	Initialization completion	

Note: In simultaneous transmit and receive operations, the TE and RE bits in SCR must both be set to 0 or set to 1 simultaneously.

32.5.4 Serial Data Transmission in Clock Synchronous Mode

(1) Non-FIFO selected

Figure 32.36, Figure 32.37, and Figure 32.38 show examples of serial transmission in clock synchronous mode.

In serial data transmission, the SCI operates as follows:

1. The SCI transfers data from the TDR register to the TSR register when data is written to TDR in the SCIn_TXI interrupt handling routine. The SCIn_TXI interrupt request at the beginning of transmission is generated when the TE bit is set to 1 but only after the TIE bit in the SCR is also set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. After transferring data from TDR to TSR, the SCI starts transmission. When the SCR.TIE bit is set to 1, an SCIn_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to TDR in the SCIn_TXI interrupt handling routine before transmission of the current transmit data finishes. When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the TDR register from the handling routine for SCIn_TXI requests.

3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the input CTS signal is low when the SPMR.CTSE bit is 1.
4. The SCI checks for update to the TDR register on output of the last bit.
5. When the TDR register is updated, the next transmit data is transferred from TDR to TSR, and serial transmission of the next frame starts.
6. If TDR is not updated, the SSR.TEND flag is set to 1. The TXDn pin retains the output state of the last bit. If the SCR.TEIE bit is 1, an SCIn_TEI interrupt request is generated and the SCKn pin is held high.

Figure 32.36, Figure 32.37, and Figure 32.38 show examples of serial data transmission.

Transmission does not start while a receive error flag (ORER, FER, or PER in SSR) is set to 1. Always set the receive error flags to 0 before starting transmission.

Note: Setting the SCR.RE bit to 0 does not clear the receive error flags.

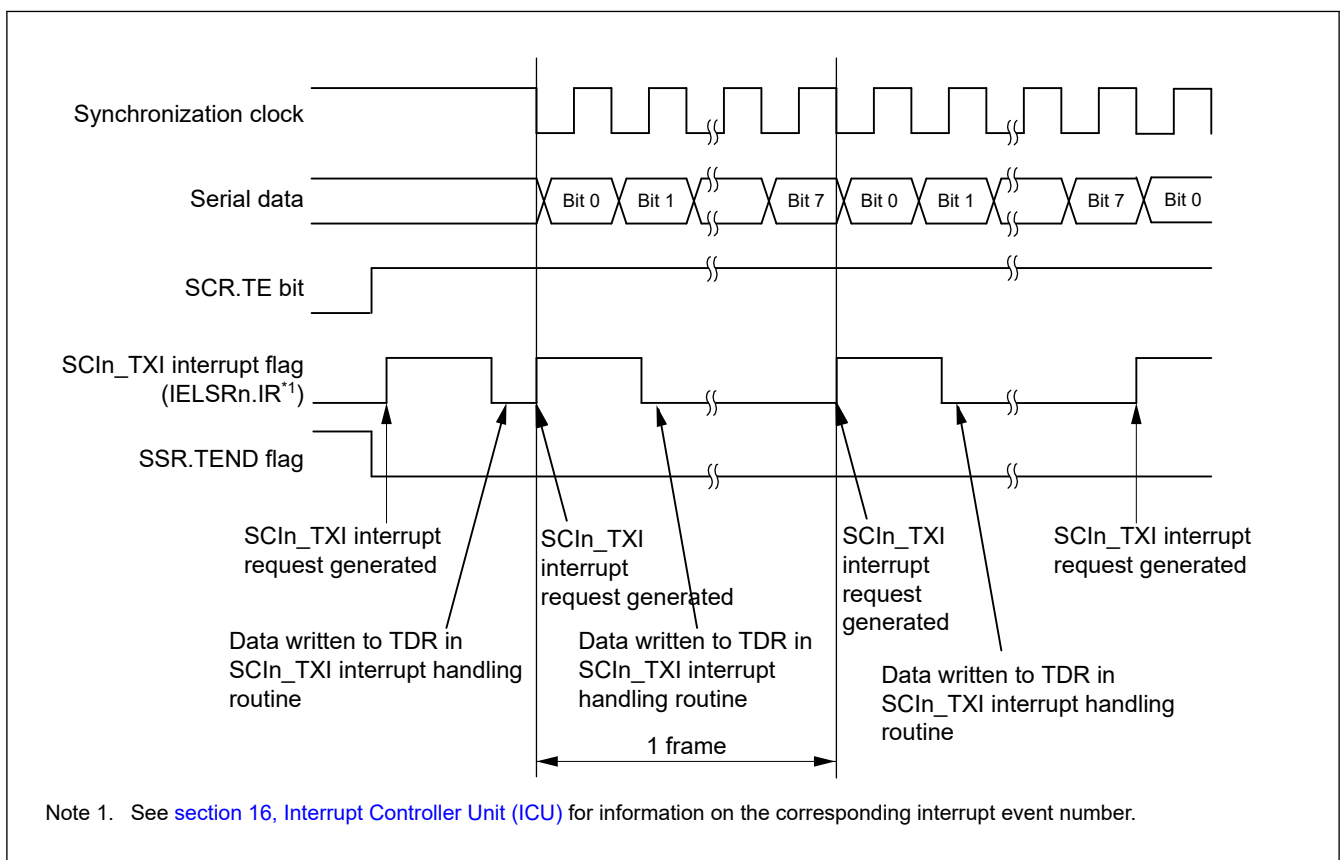
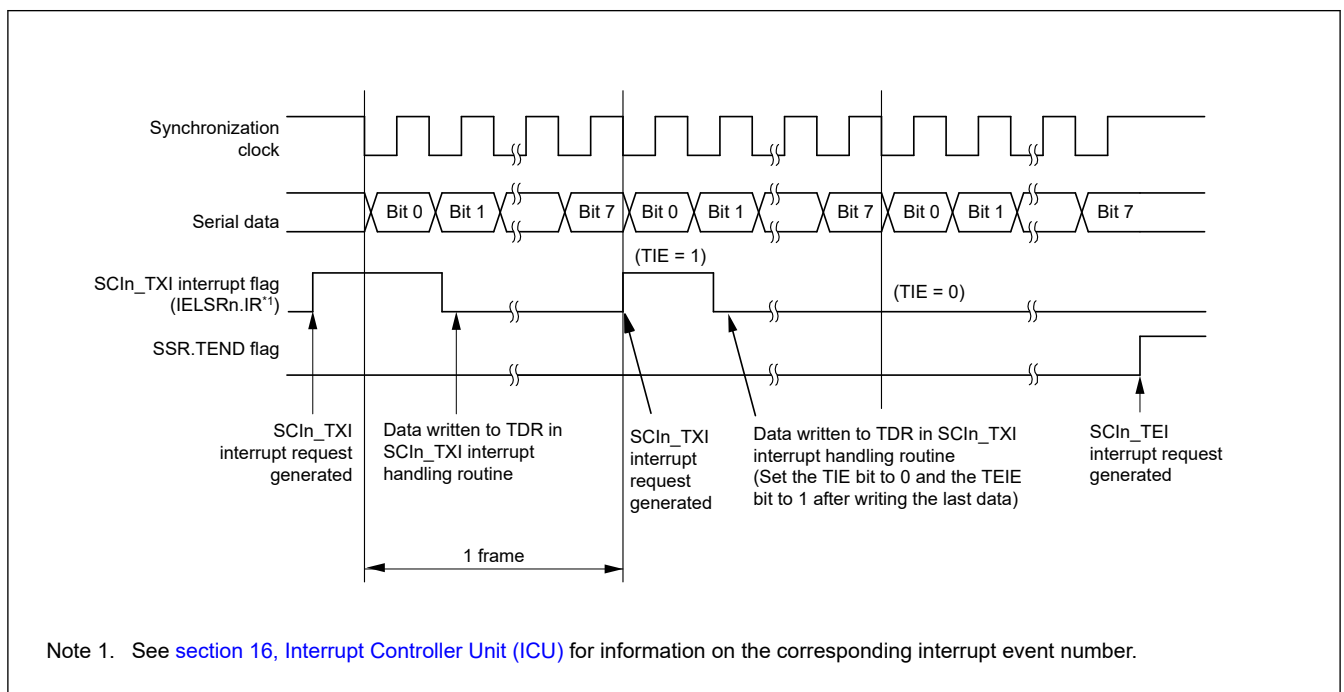
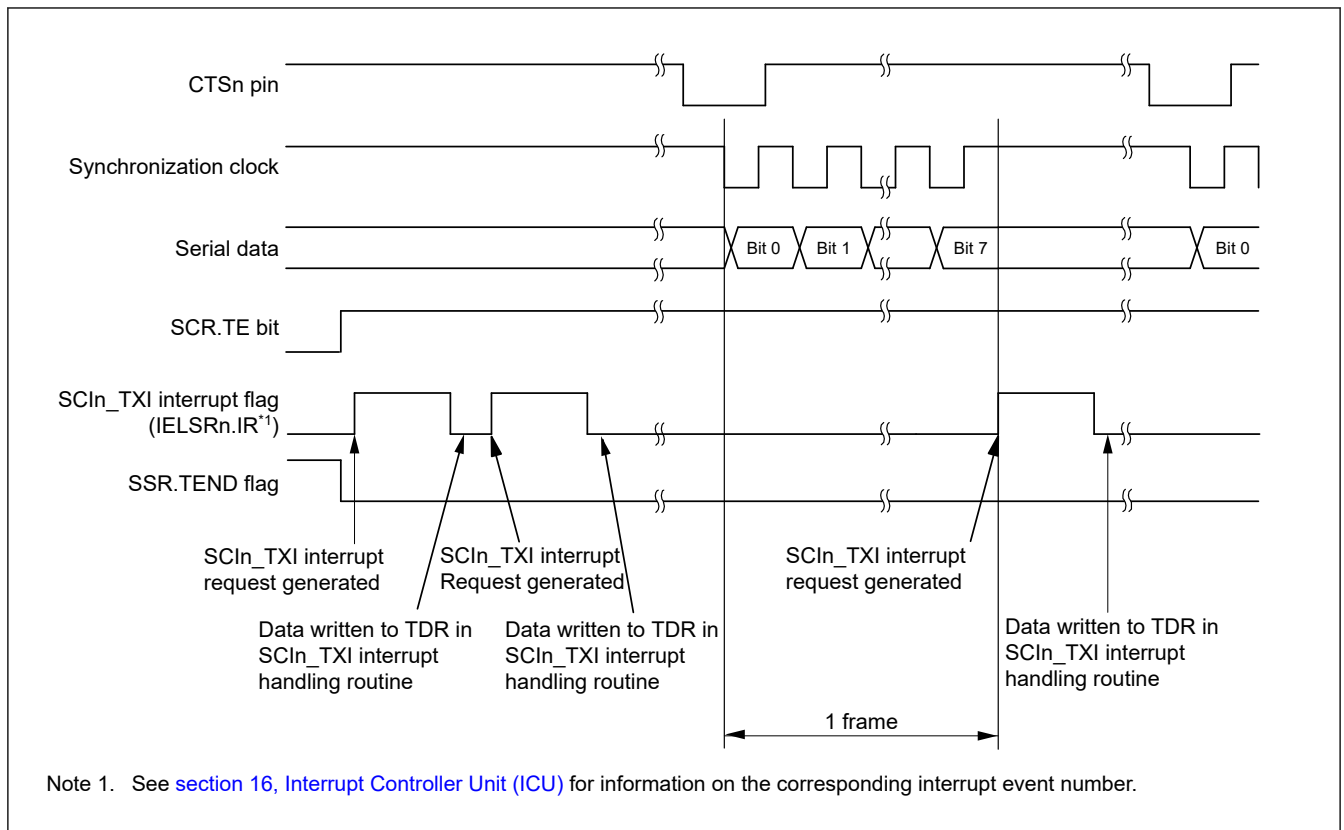


Figure 32.36 Example of serial data transmission in clock synchronous mode when the CTS function is not used at the beginning of transmission



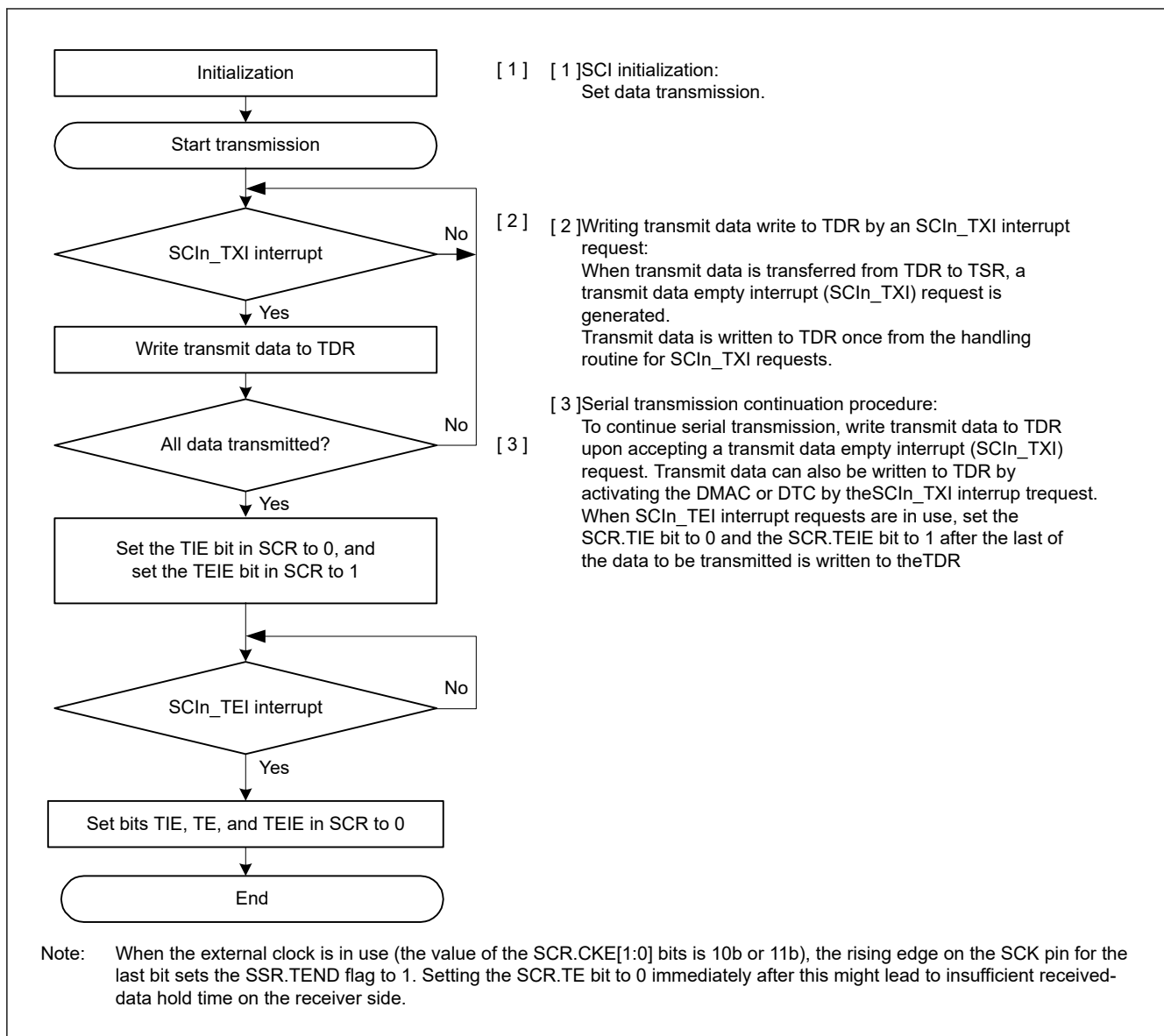


Figure 32.39 Example flow of serial transmission in clock synchronous mode with non-FIFO selected

(2) FIFO selected

Figure 32.40 shows an example of serial transmission in clock synchronous mode with FIFO selected.

In serial data transmission, the SCI operates as follows:

1. The SCI transfers data from the FTDRL^{*1} register to the TSR register when data is written to FTDRL^{*1} in the SCIn_TXI interrupt handling routine. The amount of data that can be written to FTDRL is 16 minus FDR.T[4:0] bytes. The SCIn_TXI interrupt request at the beginning of transmission is generated when the SCR.TE bit is set to 1 but only after the SCR.TIE bit is also set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. After transferring data from FTDRL to TSR, the SCI starts transmission. When the amount of transmit data written in FTDRL is equal to or less than the specified transmit triggering number, the SSR_FIFO.TDFE is set to 1. When the SCR.TIE bit is set to 1, an SCIn_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to FTDRL in the SCIn_TXI interrupt handling routine before transmission of the current transmit data has finished. When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the FTDRL from the handling routine for SCIn_TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the input CTS signal is low when the SPMR.CTSE bit is 1.

4. The SCI checks whether non-transmitted data remains in FTDRL on output of the stop bit.
5. When FTDRL is updated, the next transmit data is transferred from FTDRL to TSR and serial transmission of the next frame starts.
6. If FTDRL is not updated, the SSR_FIFO.TEND flag is set to 1. The TXDn pin retains the output state of the last bit. If the SCR.TEIE bit is 1, an SCIn_TEI interrupt request is generated and the SCKn pin is held high.

Note 1. In clock synchronous mode, FTDRH is not used.

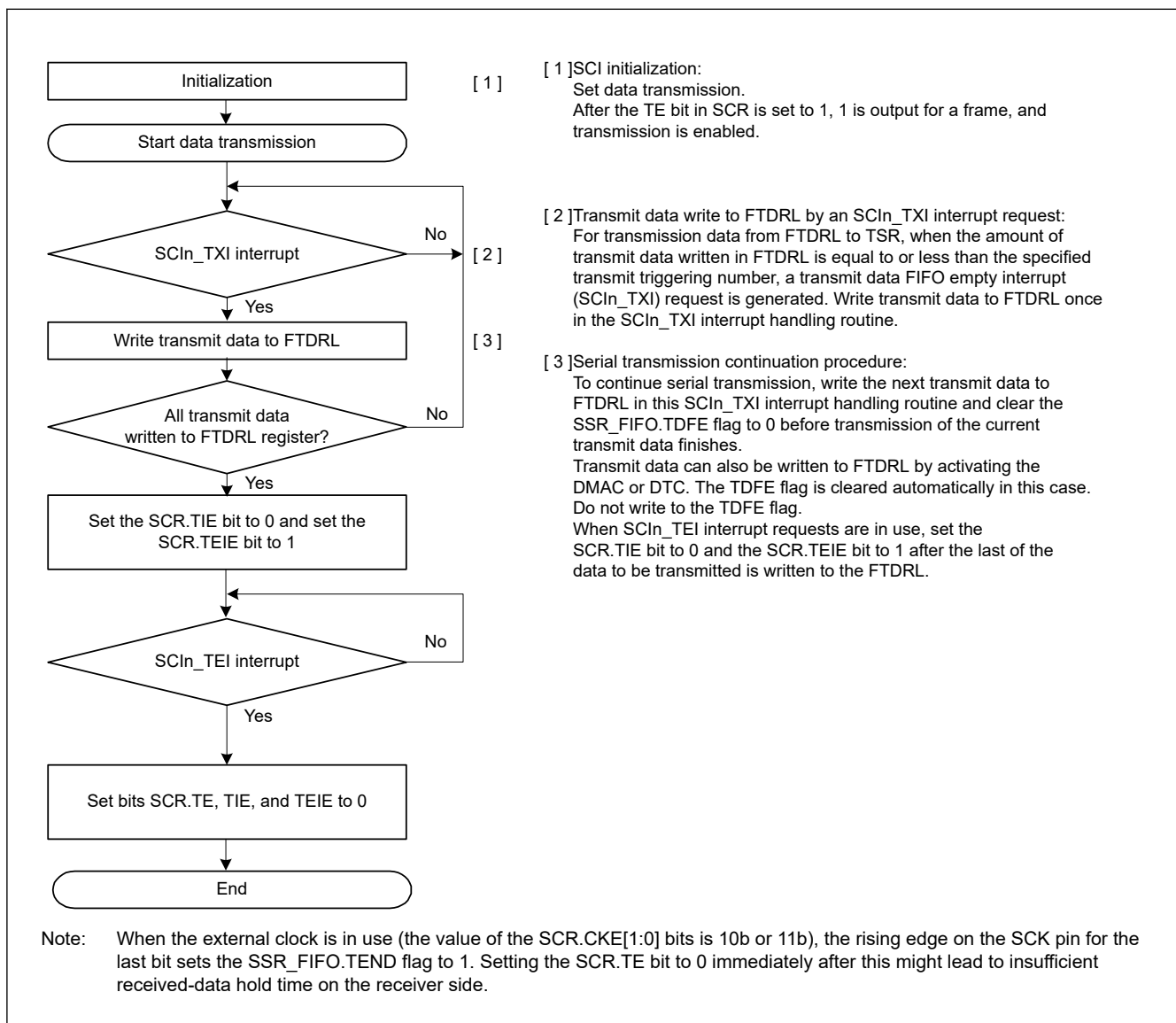


Figure 32.40 Example flow of serial transmission in clock synchronous mode with FIFO selected

32.5.5 Serial Data Reception in Clock Synchronous Mode

(1) Non-FIFO selected

Figure 32.41 and Figure 32.42 show examples of SCI operation for serial reception in clock synchronous mode.

In serial data reception, the SCI operates as follows:

1. When the value of the SCR.RE bit becomes 1, the RTSn pin goes low.
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.

3. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. When reception completes successfully, receive data is transferred to the RDR register. If the SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the received data transferred to the RDR register in the SCIn_RXI interrupt handling routine before reception of the next receive data completes. Reading the received data that is transferred to RDR causes the RTSn pin to output low.

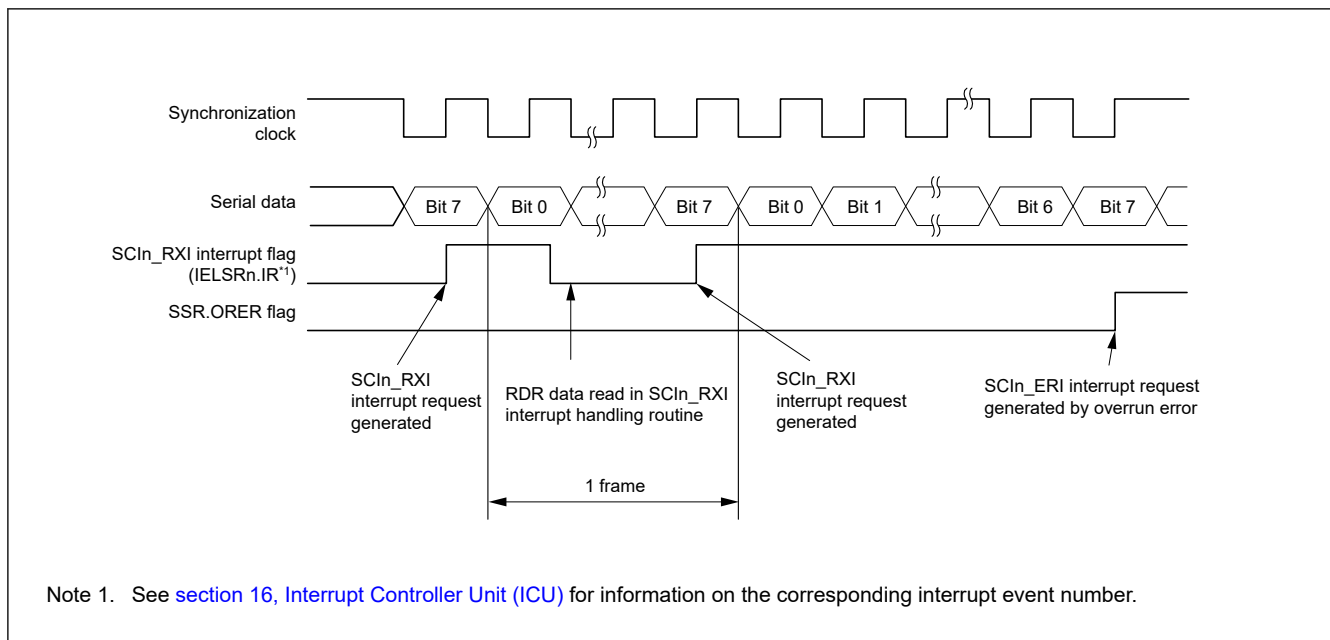


Figure 32.41 Example operation for serial reception in clock synchronous mode (1) when the RTS function is not used

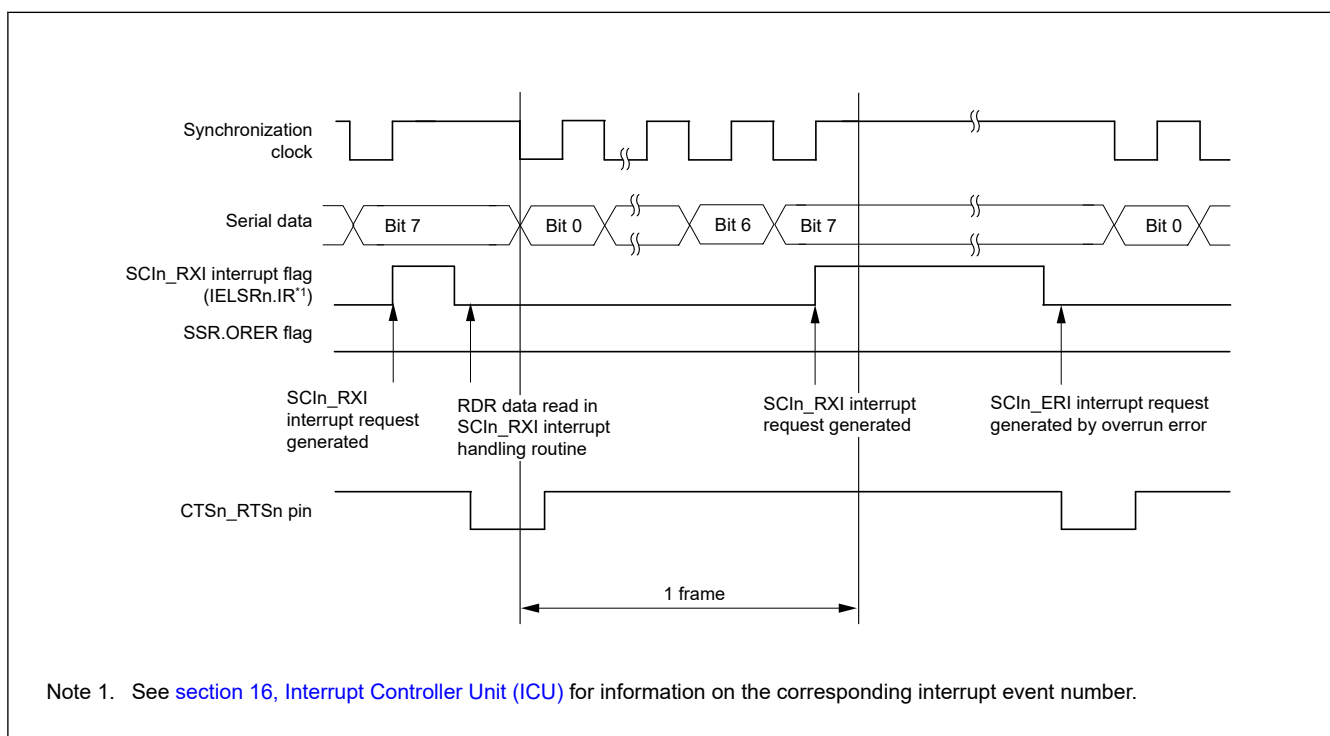


Figure 32.42 Example operation for serial reception in clock synchronous mode (2) when RTS function is used

Data transfer cannot resume while the receive error flag is 1. Therefore, clear the ORER, FER, and PER flags in the SSR register to 0 before resuming data reception. Additionally, always read the RDR register during overrun error processing.

When a data reception is forced to terminate by a 0 write to the SCR.RE bit during operation, read the RDR register because received data that is not yet read might be left in the RDR register.

Figure 32.43 shows an example flow of serial data reception.

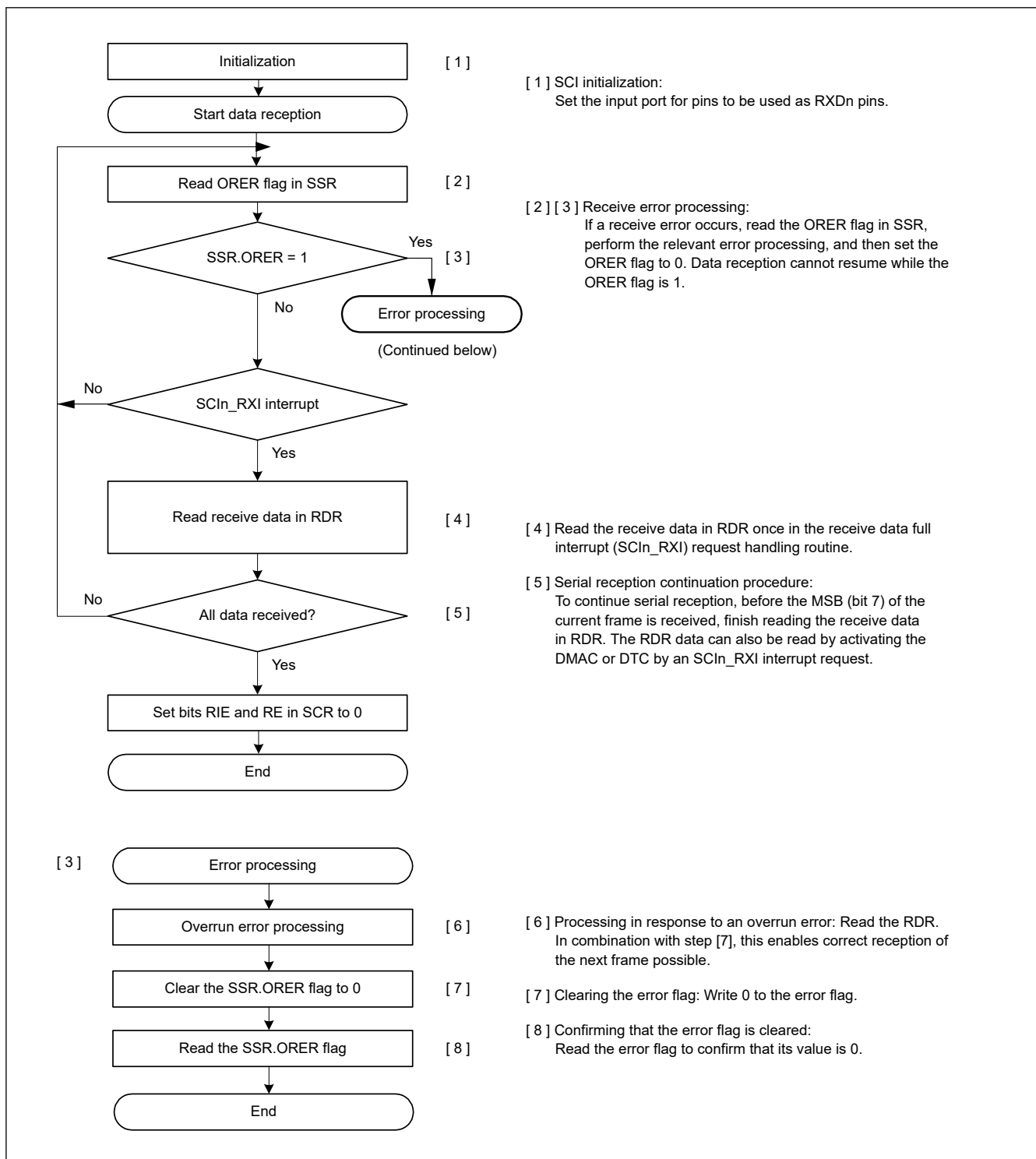


Figure 32.43 Example flow of serial reception in clock synchronous mode with non-FIFO selected

(2) FIFO selected

Figure 32.44 shows an example of serial reception in clock synchronous mode with FIFO selected.

In serial data reception, the SCI operates as follows:

1. When the value of the SCR.RE bit becomes 1, the RTSn pin goes low.

2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.
3. If an overrun error occurs, the SSR_FIFO.Over flag is set to 1. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated. Received data is not transferred to the FRDRL^{*1} register.
4. When data reception completes successfully, the receive data is transferred to the FRDRL^{*1} register. The RDF flag is set to 1 when the amount of the receive data stored in FRDRL is equal to or greater than the specified receive triggering number. If the SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous data reception is enabled by reading the receive data transferred to FRDRL^{*2} in the SCIn_RXI interrupt handling routine before an overrun error occurs. If the amount of received data that is transferred to FRDRL is less than the RTS trigger number, the CTSn_RTSn pin goes low.

Note 1. In clock synchronous mode, FRDRH is not used.

Note 2. Read data in order from FRDRH to FRDRL when RDF and Over are read with receive data.

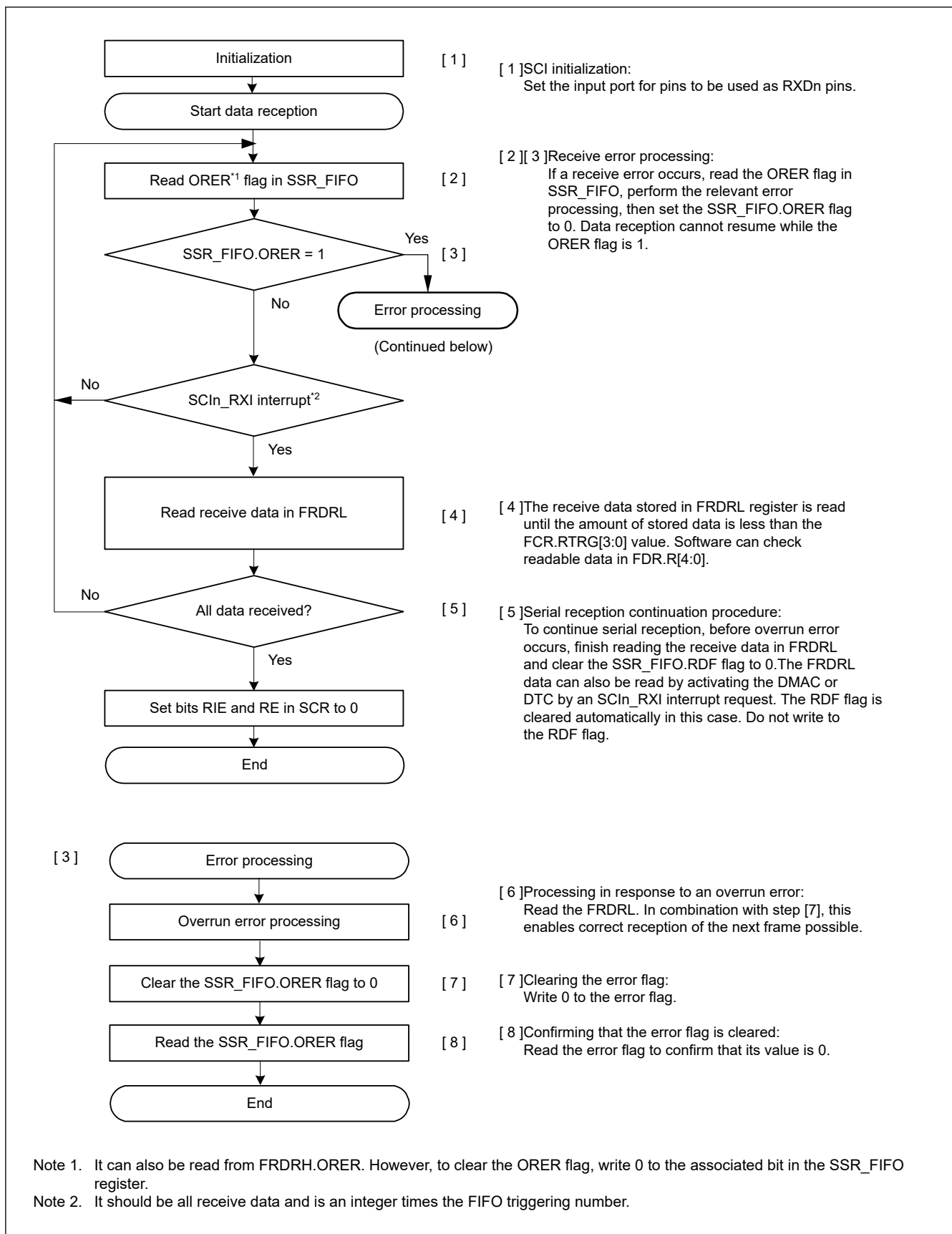


Figure 32.44 Example flow of serial reception in clock synchronous mode with FIFO selected

32.5.6 Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode

(1) Non-FIFO selected

[Figure 32.45](#) shows an example flow of simultaneous serial transmit and receive operations in clock synchronous mode. After initializing the SCI, use the following procedure for simultaneous serial data transmit and receive operations.

To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the data transmission by verifying that the SSR.TEND flag is set to 1.
2. Initialize the SCR register, and then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the data reception.
2. Set the RIE and RE bits to 0, and then check that the receive error flag ORER in the SSR register is 0.
3. Set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.

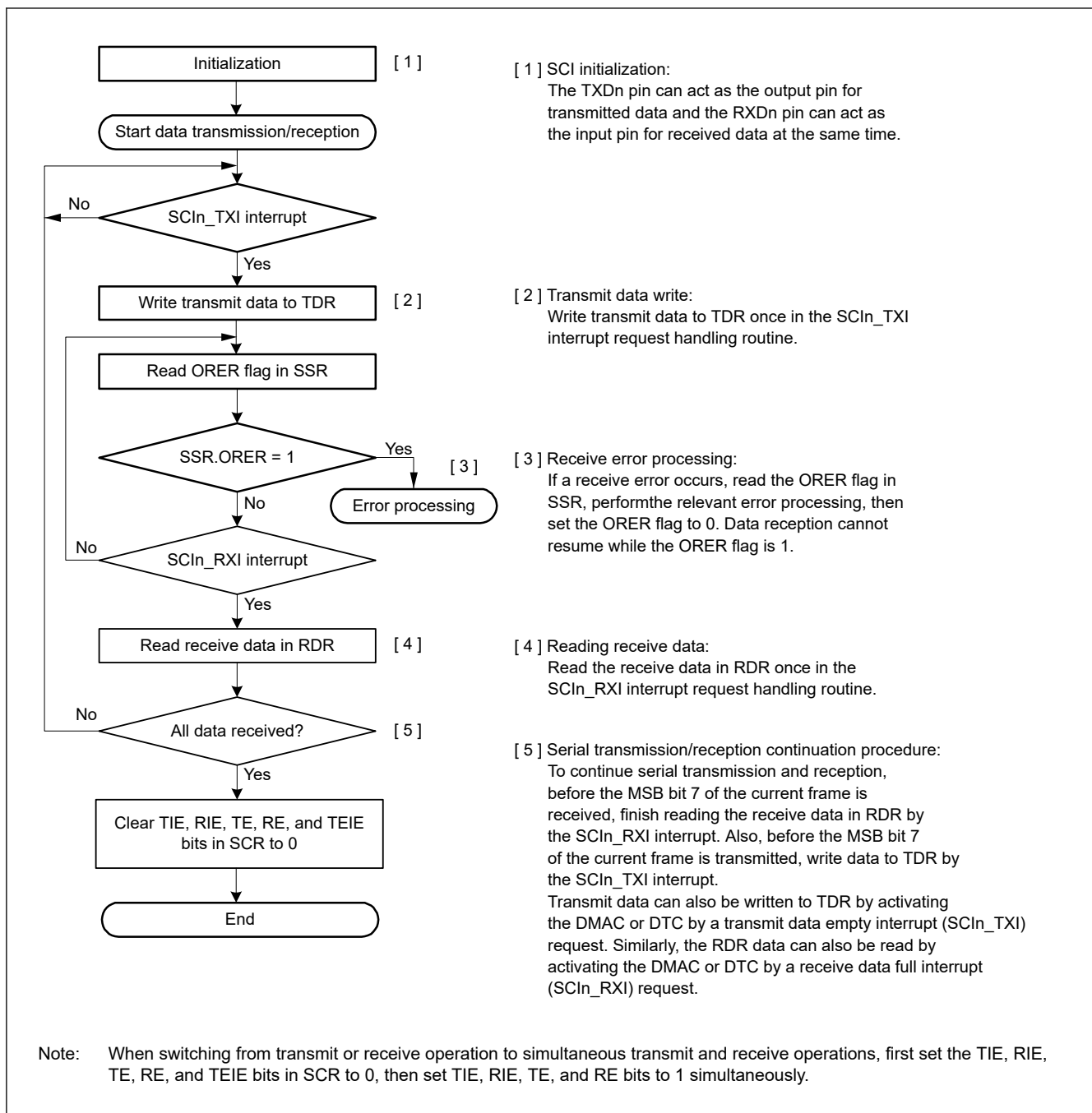


Figure 32.45 Example flow of simultaneous serial transmission and reception in clock synchronous mode with non-FIFO selected

(2) FIFO selected

Figure 32.46 shows an example flow of simultaneous serial transmit and receive operations in clock synchronous mode with FIFO selected.

After initializing the SCI, use the following procedure for simultaneous serial data transmit and receive operations.

To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the transmission by verifying that the SSR_FIFO.TEND flag is set to 1.
2. Initialize the SCR register, then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the reception.

2. Set the RIE and RE bits to 0.
3. Check that the receive error flags ORER in the SSR_FIFO register are 0, and then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.

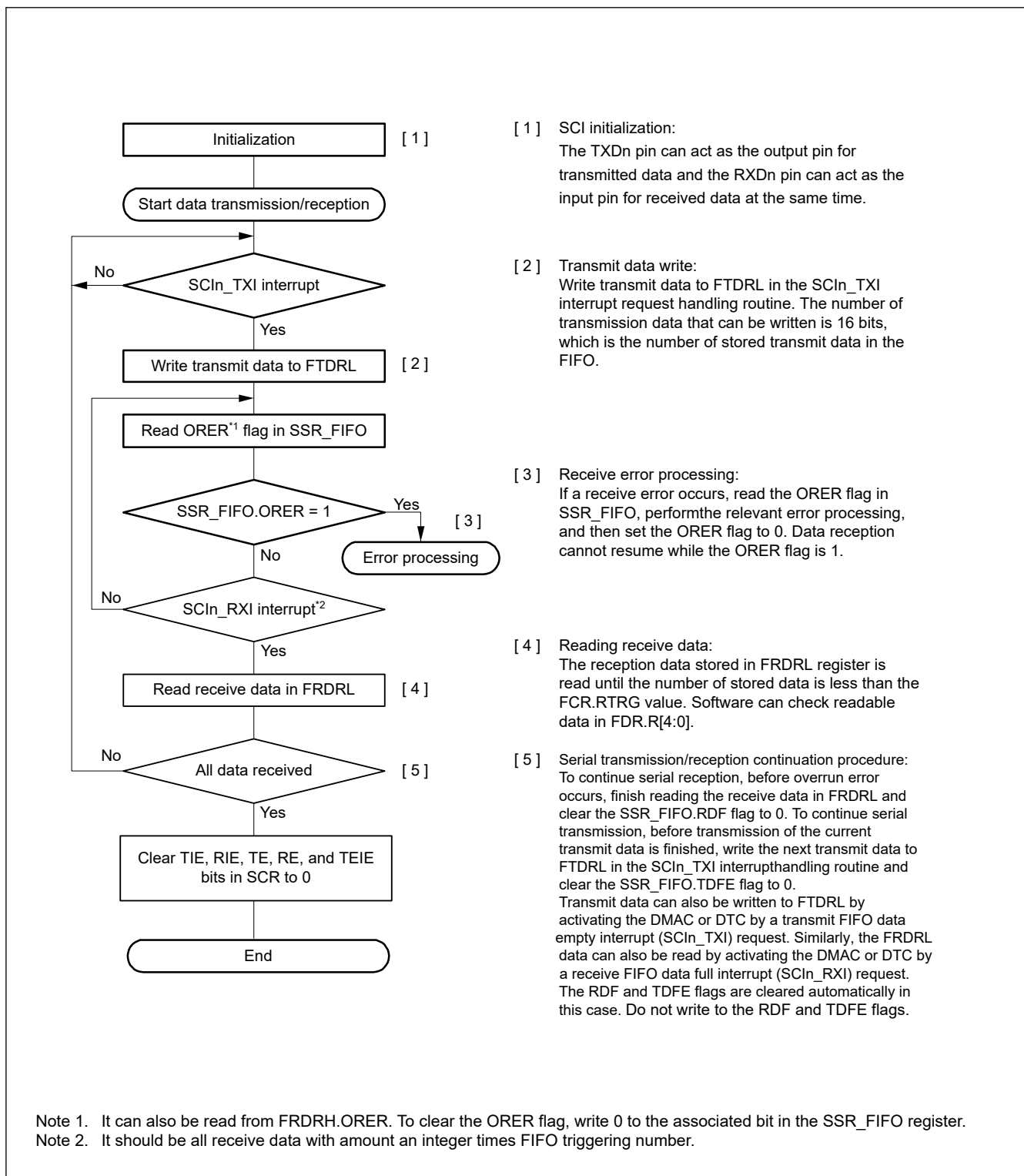


Figure 32.46 Example flow of simultaneous serial transmission and reception in clock synchronous mode with FIFO selected

32.6 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.

32.6.1 Example Connection

Figure 32.47 shows an example connection between a smart card (IC card) and the MCU. As shown in Figure 32.47, because the MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.

Setting the SCR_SMCI.TE and SCR_SMCI.RE bits to 1 with an IC card disconnected enables closed-loop transmission or reception, allowing self-diagnosis. To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card.

An output port of the MCU can be used to output a reset signal.

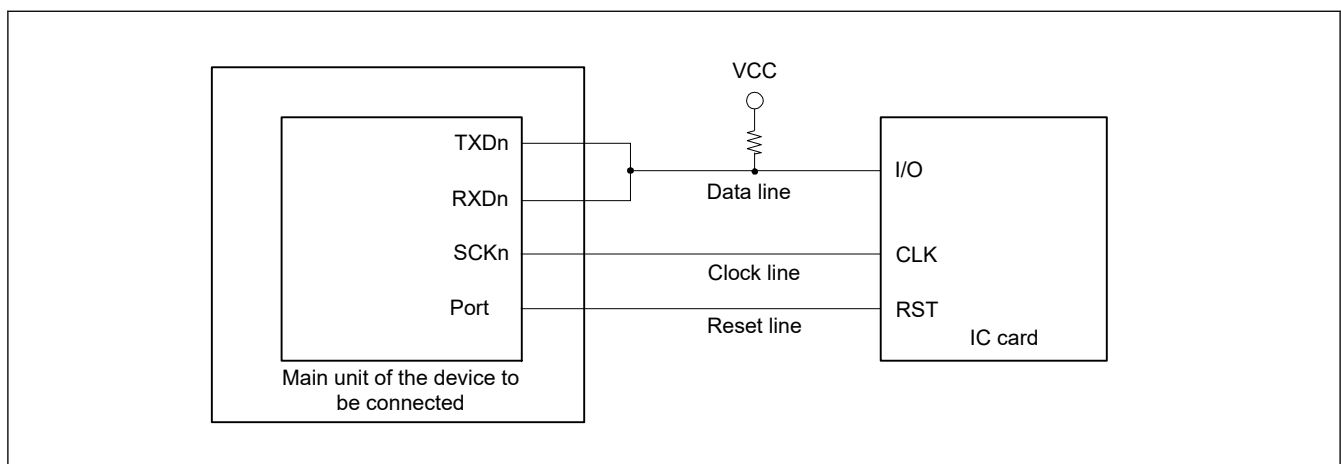


Figure 32.47 Example connection with a smart card (IC card)

32.6.2 Data Format (Except in Block Transfer Mode)

Figure 32.48 shows the data transfer formats in smart card interface mode:

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 ETUs (elementary time unit – the time required for transferring 1 bit) is set as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low error signal is output for 1 ETU after 10.5 ETUs elapse from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 ETUs.

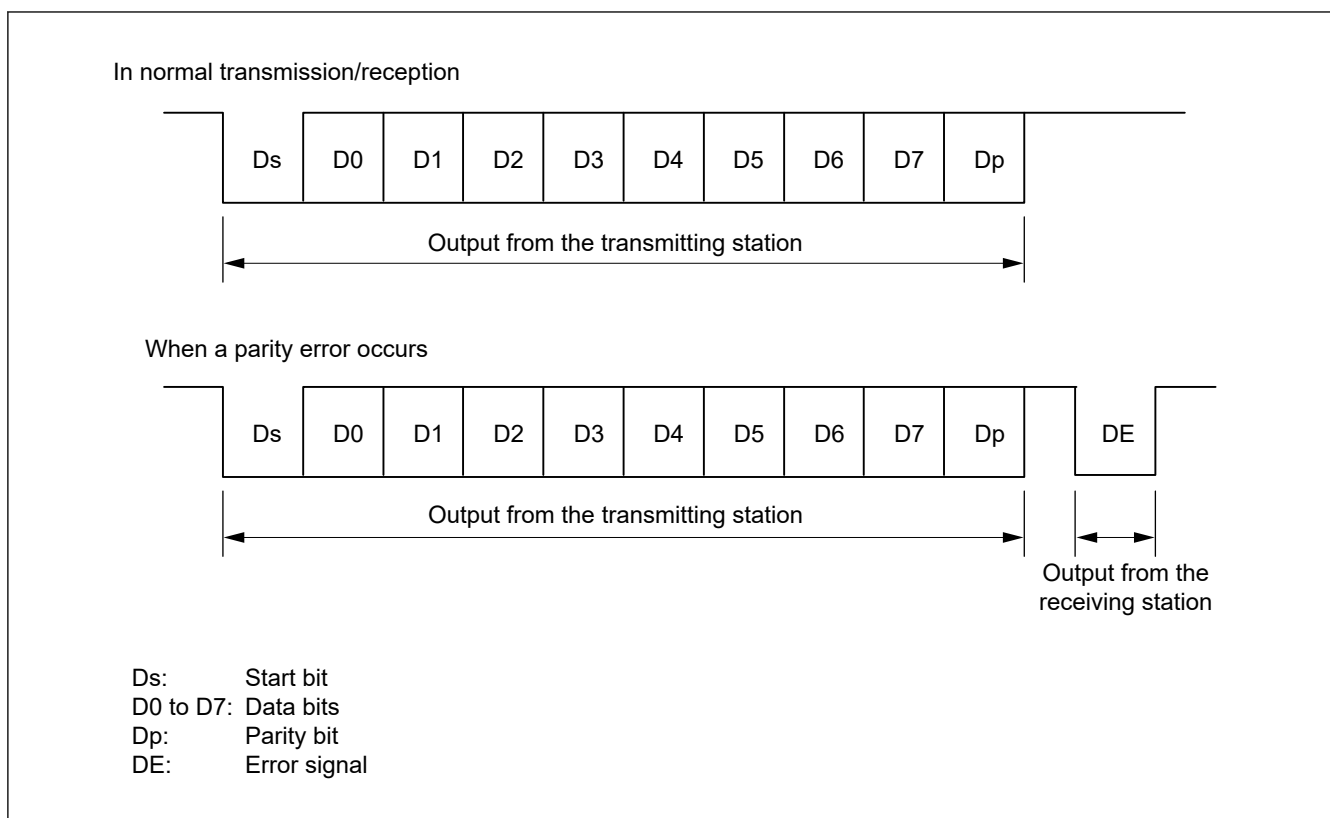


Figure 32.48 Data formats in smart card interface mode

For communications with IC cards of the direct convention type and inverse convention type, follow the procedures in this section.

(1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 indicate the Z and A states, respectively, and data is transferred with LSB-first for the start character, as shown in Figure 32.49. Therefore, data in the start character in the figure is 0x3B.

When using the direct convention type, write 0 to both the SCMR.SDIR and SCMR.SINV bits. Write 0 to the SMR_SMCI.PM bit to use even parity, which is prescribed by the smart card standard.

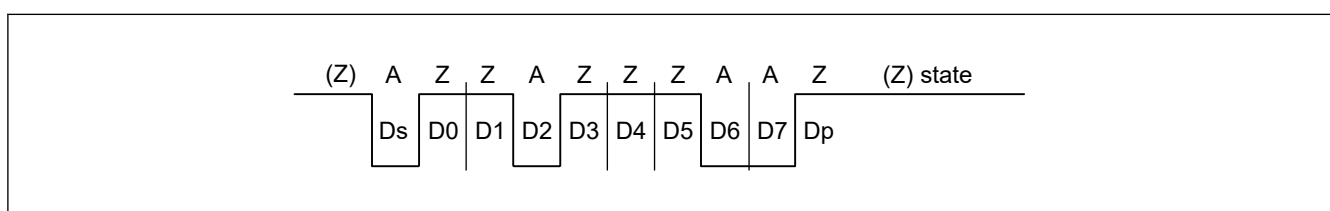


Figure 32.49 Direct convention with SDIR in SCMR = 0, SINV in SCMR = 0, and PM in SMR_SMCI = 0

(2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 indicate the A and Z states, respectively, and data is transferred with MSB-first for the start character, as shown in Figure 32.50. Therefore, data in the start character in the figure is 0x3F.

When using the inverse convention type, write 1 to both the SCMR.SDIR and SCMR.SINV bits. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to the Z state. Because the SINV bit of the MCU only inverts data bits D7 to D0, write 1 to the PM bit in SMR_SMCI to invert the parity bit for both transmission and reception.

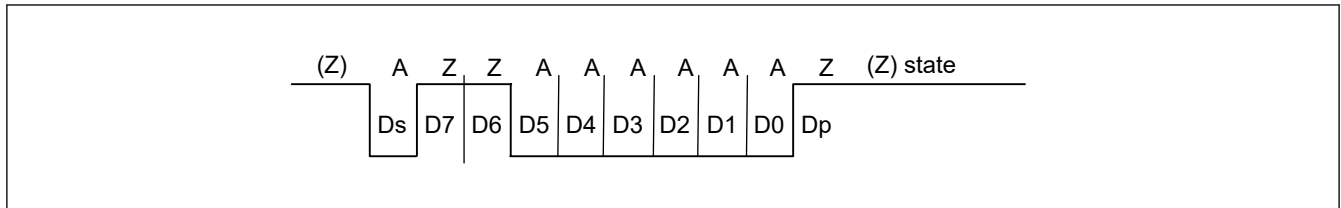


Figure 32.50 Inverse convention with SDIR in SCMR = 1, SINV in SCMR = 1, and PM in SMR_SMCI = 1

32.6.3 Block Transfer Mode

Block transfer mode differs from normal smart card interface mode as follows:

- Even if a parity error is detected during reception, no error signal is output. Because the PER flag in SSR_SMCI is set by error detection, clear the PER flag before receiving the parity bit of the next frame.
- During transmission, at least 1 ETU is set as a guard time from the end of the parity bit until the start of the next frame
- Because the same data is not retransmitted, the TEND flag in SSR_SMCI is set to 11.5 ETUs after transmission starts
- In block transfer mode, the ERS flag in SSR_SMCI indicates the error signal status as in normal smart card interface mode, but the flag is read as 0 because no error signal is transferred

32.6.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the on-chip baud rate generator can be used as a transfer clock in smart card interface mode.

In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate set up in the SCMR.BCP2 and the SMR_SMCI.BCP[1:0] bits. The frequency is always 16 times the bit rate in normal asynchronous mode.

For data reception, the falling edge of the start bit is sampled with the base clock to perform internal synchronization.

Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 32.51. The reception margin is determined by the following formula:

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100 \text{ [%]}$$

M: Reception margin (%)

N: Ratio of bit rate to clock (N = 32, 64, 372, 256)

D: Duty cycle of clock (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the specified formula, the reception margin is determined using the following formula:

$$M = \{0.5 - 1/(2 \times 372)\} \times 100 \text{ [%]} = 49.866 \text{ %}$$

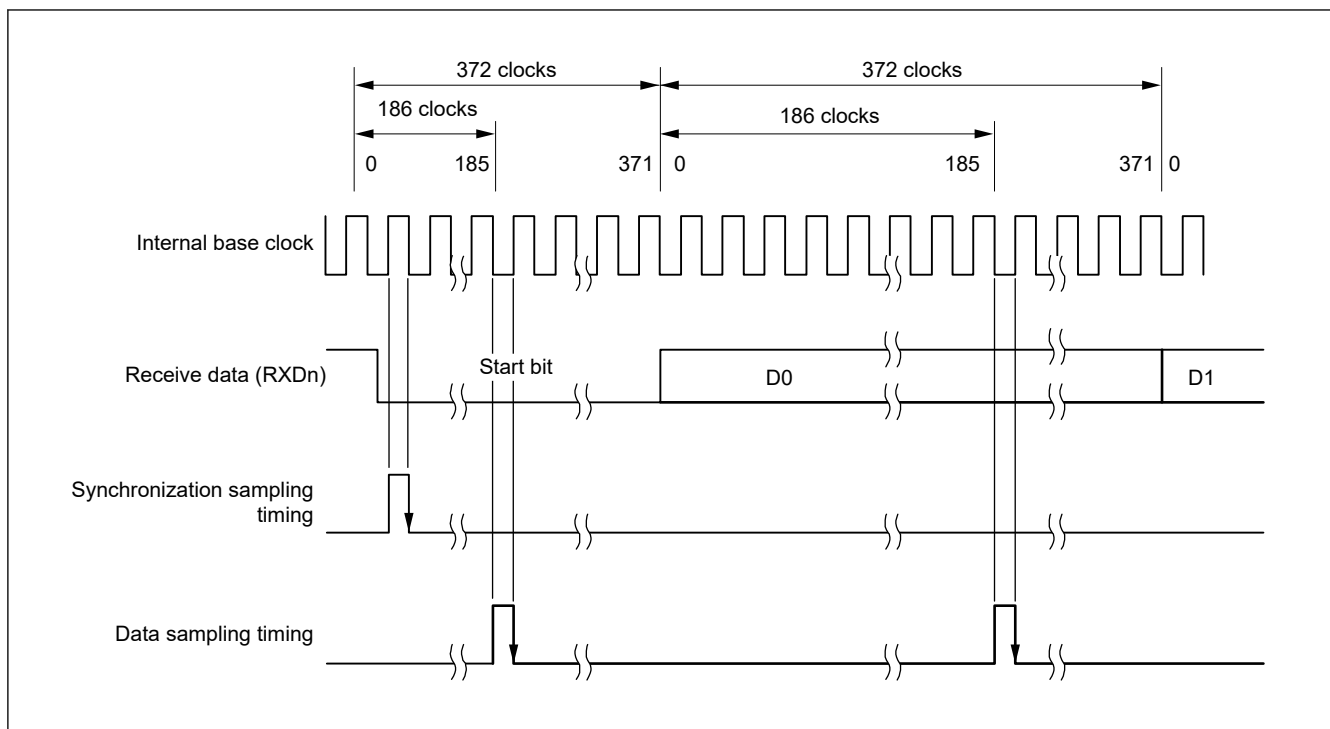


Figure 32.51 Receive data sampling timing in smart card interface mode when the clock frequency is 372 times the bit rate

32.6.5 SCI Initialization (Smart Card Interface Mode)

Before transmitting and receiving data, write the initial value 0x00 in the SCR_SMCI register and initialize the SCI following the example flow shown in [Table 32.31](#).

Always set the initial value in the TIE, RIE, TE, RE, TEIE bits in the SCR_SMCI register before switching from transmission to reception mode or from reception to transmission mode. When SCR_SMCI.RE is set to 0, the RDR register is not initialized.

To change from reception mode to transmission mode, first check that reception has completed, then initialize the SCI. At the end of initialization, set SCR_SMCI.TE = 1 and SCR_SMCI.RE = 0. Reception completion can be verified by reading the SCIn_RXI request, ORER, or PER flag in SSR_SMCI.

To change transmission mode to reception mode, first check that transmission has completed, then initialize the SCI. At the end of initialization, set SCR_SMCI.TE = 0 and SCR_SMCI.RE = 1. Transmission completion can be verified by reading the TEND flag in SSR_SMCI.

Table 32.31 Example flow of SCI initialization in smart card interface mode (1 of 2)

No.	Step Name	Description
1	Start initialization	
2	Set SCR_SMCI.TIE, RIE, TE, RE, TEIE, and CKE[1:0] to 0	
3	Set I/O port functions	Set the I/O ports so that the necessary pin functions can be enabled among the TXDn, RXDn, and SCKn pins.
4	Set SSR_SMCI.ORER, ERS, PER to 0	Set the SSR_SMCI.ORER, ERS, and PER flags in to 0. After reading the SSR_SMCI register, write 0 to the target flags.
5	Set the SIMR1.IICM bit to 0 Set the SPMR.CKPH and CKPOL bits to 0	Set the SIMR1.IICM bit to 0, and set the SPMR.CKPH and CKPOL bits to 0. Skip this step when the initial values are not changed.
6	Set SMR_SMCI.GM, BLK, PM, BCP[1:0], CKS[1:0], and set SMR_SMCI.PE to 1	Set the operation mode and the transmission or reception format in SMR_SMCI.

Table 32.31 Example flow of SCI initialization in smart card interface mode (2 of 2)

No.	Step Name	Description
7	Set SCMR.BCP2, SDIR, SINV Set SCMR.SMIF to 1	Set the transmission or reception format in SCMR.
8	Set SEMR.BRME and SEMR.RXDESEL to 0	Set SEMR.BRME and SEMR.RXDESEL to 0.
9	Set a value in BRR	Write the value for the bit rate in BRR.
10	Set a value in the MDDR	Write the value obtained by correcting a bit rate error in the MDDR register. This step is not required if the bit rate adjustment function is not used.
11	Set a value in SCR_SMCI.CKE[1:0]	Set the SCR_SMCI.CKE[1:0] bits. When the CKE[0] bit is set to 0, the clock is output from the SCKn pin.
12	Set SCR_SMCI.TE or RE to 1, and set SCR_SMCI.TIE, RIE	Set the TE or RE bit in SCR_SMCI to 1, then set the TIE and RIE bits in SCR_SMCI. Do not simultaneously set the TE and RE bits to 1 if self-diagnosis is not used.
13	Initialization completed	

32.6.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode) is different from that in non-smart card interface mode, in that an error signal is sampled and data can be re-transmitted in smart card mode. [Figure 32.52](#) shows the data re-transfer operation during transmission.

1. When an error signal from the receiver end is sampled after 1-frame data is transmitted, the SSR_SMCI.ERS flag is set to 1. If the SCR_SMCI.RIE bit is 1, an SCIn_ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.
2. For a frame in which an error signal is received, the SSR_SMCI.TEND flag is not set. Data is re-transferred from TDR to TSR, allowing automatic data retransmission.
3. If no error signal is returned from the receiver, the ERS flag is not set to 1.
4. In this case, the SCI determines that transmission of 1-frame data, including the re-transfer, is complete, and the TEND flag is set. If the SCR_SMCI.TIE bit is 1, an SCIn_TXI interrupt request is generated. Write transmit data to the TDR to start transmission of the next data.

[Figure 32.54](#) shows an example flow of serial transmission. All the processing steps are automatically performed using an SCIn_TXI interrupt request to activate the DTC or DMAC.

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 or DMAC is activated by an SCIn_TXI interrupt request if the SCIn_TXI interrupt request is previously specified as a source of DTC or DMAC activation, allowing the transfer of transmit data. The TEND flag is automatically set to 0 when the DTC or DMAC 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 or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission when an error occurs. Because the ERS flag is not automatically cleared, set the RIE bit to 1 before enabling an SCIn_ERI interrupt request to be generated if an error occurs, and clear the ERS flag to 0.

When transmitting or receiving data using the DTC or DMAC, always enable the DTC or DMAC before making the SCI settings.

For DTC or DMAC settings, see [section 20, Data Transfer Controller \(DTC\)](#), [section 19, DMA Controller \(DMAC\)](#).

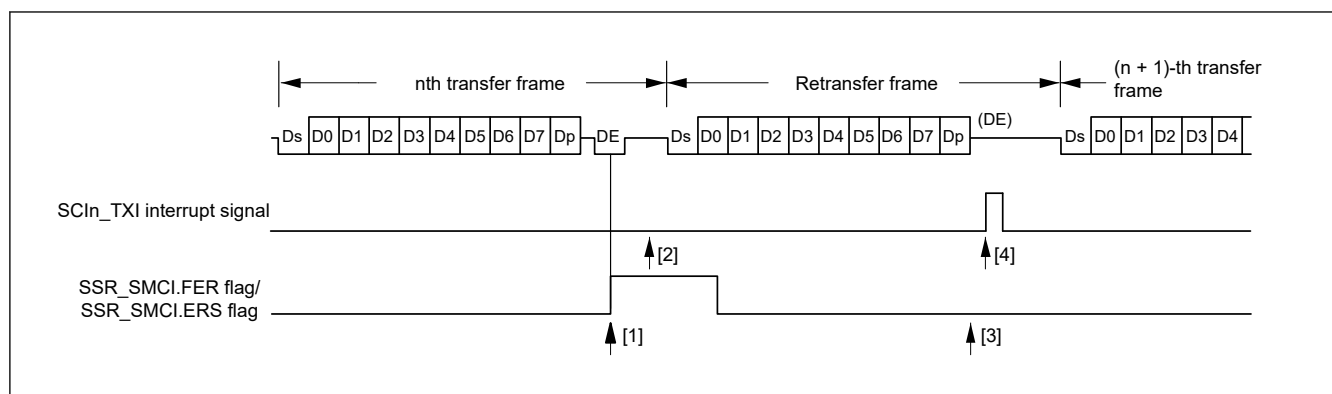


Figure 32.52 Data re-transfer operation in smart card interface transmission mode

The SSR_SMCI.TEND flag is set at different timings depending on the SMR_SMCI.GM bit setting. [Figure 32.53](#) shows the TEND flag generation timing.

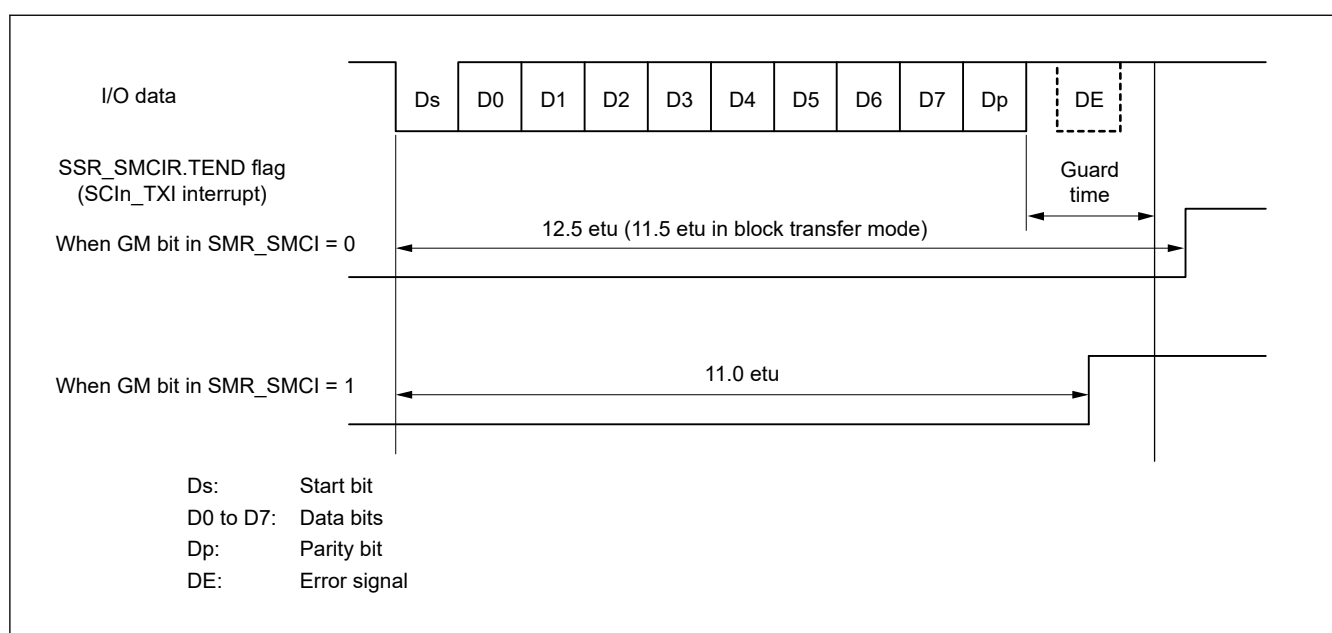


Figure 32.53 SSR.TEND flag generation timing during transmission

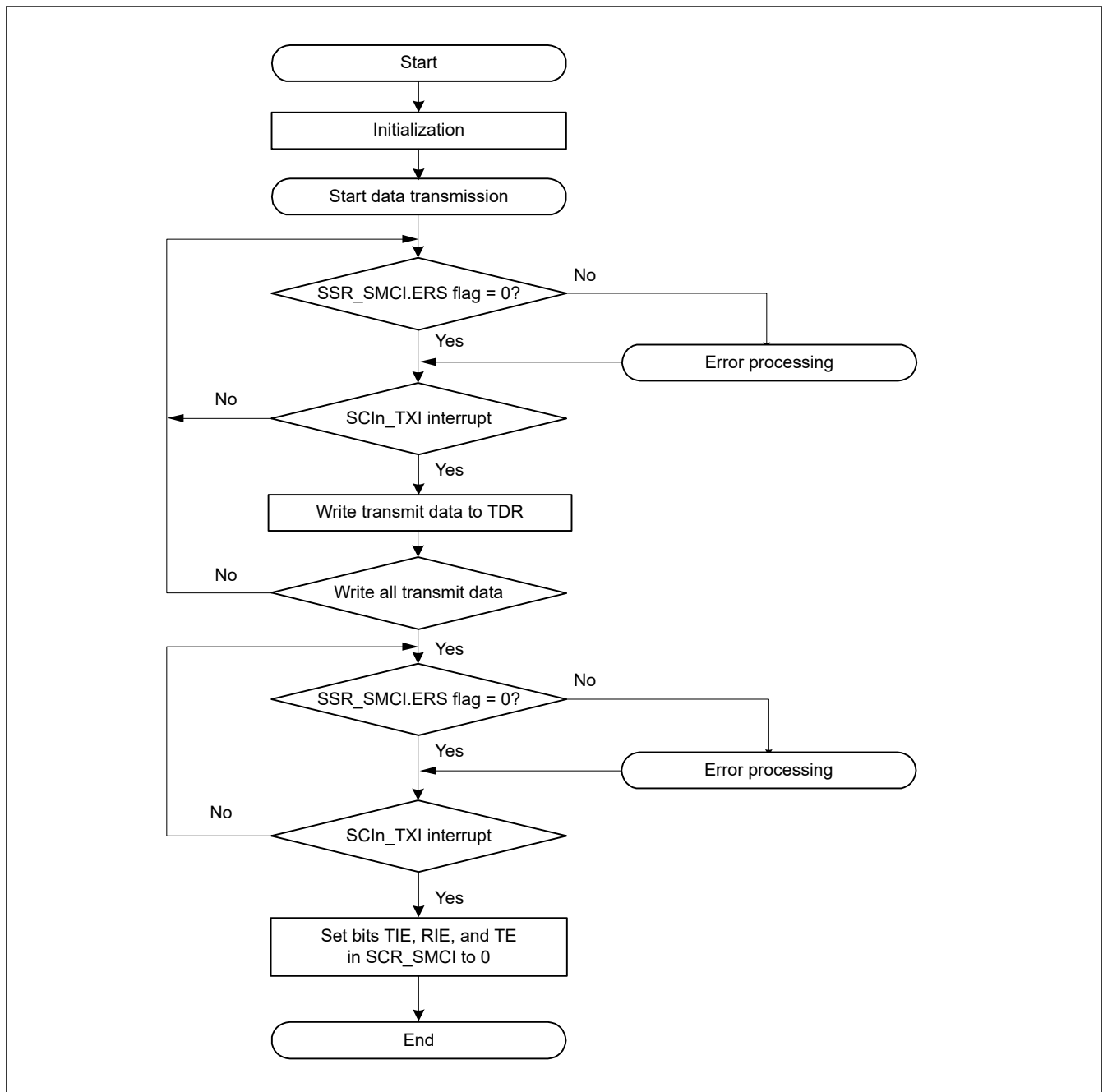


Figure 32.54 Example flow of smart card interface transmission

32.6.7 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 32.55](#) shows the data re-transfer operation in reception mode.

1. If a parity error is detected in the receive data, the SSR_SMCI.PER flag is set to 1. When the SCR_SMCI.RIE bit is 1, an SCIn_ERI interrupt request is generated. Clear the PER flag to 0 before the next parity bit is sampled.
2. For a frame in which a parity error is detected, no SCIn_RXI interrupt is generated.
3. When no parity error is detected, the SCR_SMCI.PER flag is not set to 1.
4. In this case, data is determined to be received successfully. When the SCR_SMCI.RIE bit is 1, an SCIn_RXI interrupt request is generated.

[Figure 32.56](#) 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 or DMAC.

In reception, setting the RIE bit to 1 allows an SCIn_RXI interrupt request to be generated. The DTC or DMAC is activated by an SCIn_RXI interrupt request if the SCIn_RXI interrupt request is previously specified as a source of DTC or DMAC 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 or DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC or DMAC is transferred.

If a parity error occurs and the PER flag is set to 1 during reception, the receive data is transferred to RDR, allowing the data to be read.

When a reception is forced to terminate by setting SCR_SMCI.RE to 0 during operation, read the RDR register because the received data that is not yet read might be left in the RDR.

Note: For operations in block transfer mode, see [section 32.3.9. Serial Data Reception in Asynchronous Mode](#).

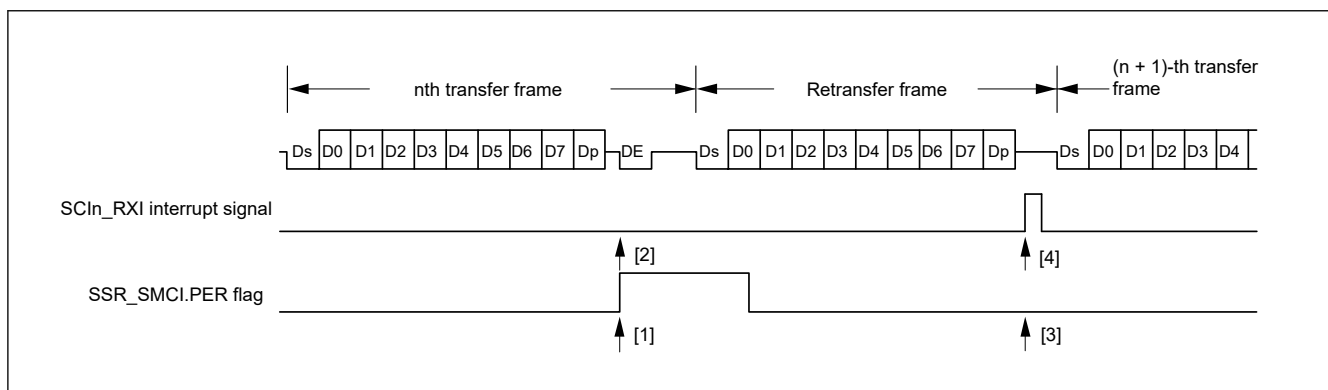


Figure 32.55 Data re-transfer operation in smart card interface reception mode

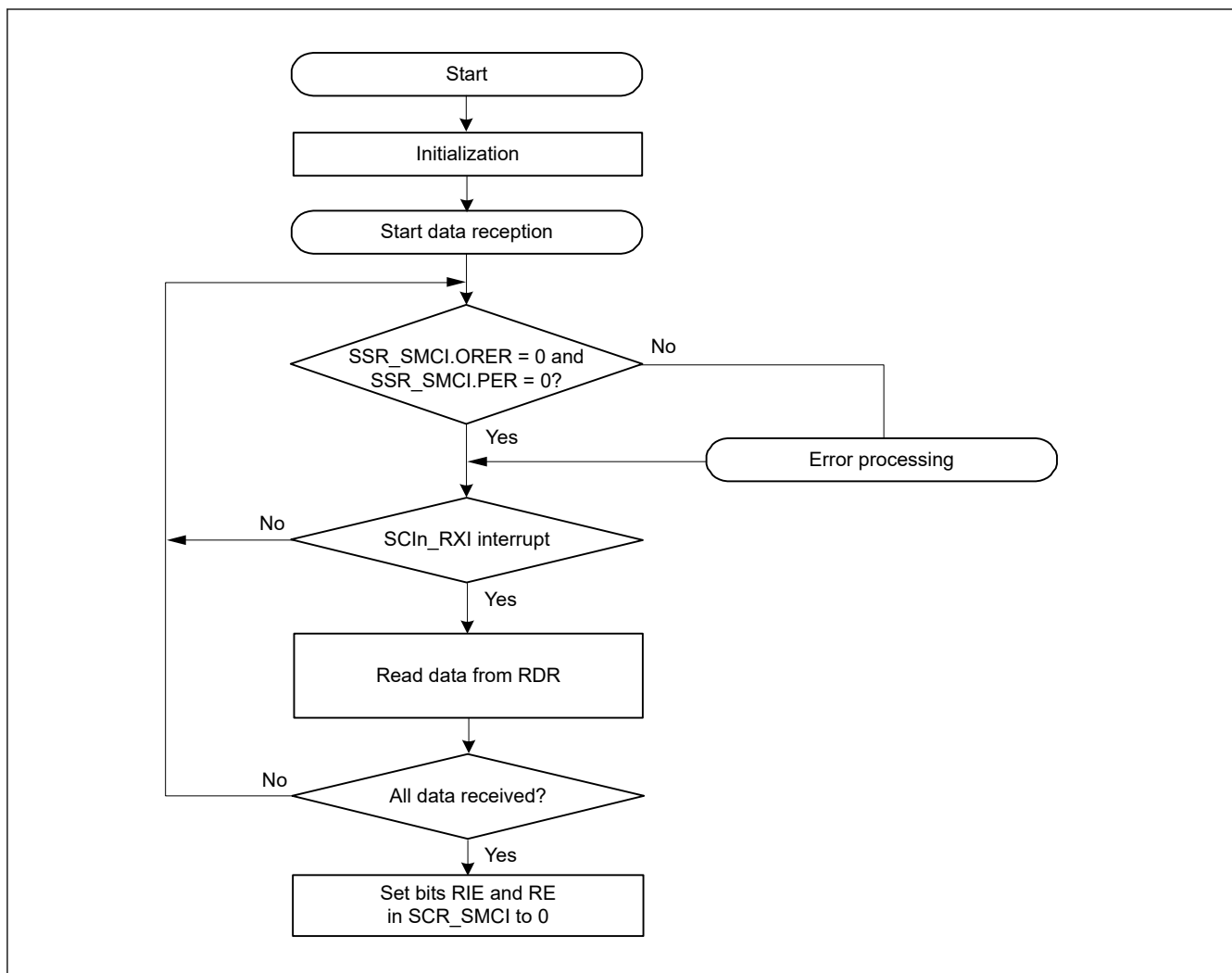


Figure 32.56 Example flow of smart card interface reception

32.6.8 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 32.2.12. SCR_SMCI : Serial Control Register for Smart Card Interface Mode \(SCMR.SMIF = 1\)](#). When setting the clock output, the base clock described in [section 32.6.4. Receive Data Sampling Timing and Reception Margin](#).

[Figure 32.57](#) shows an example timing for the clock output control when the CKE[1] bit in SCR_SMCI is set to 0 and the CKE[0] bit in SCR_SMCI is controlled.

When the GM bit in SMR_SMCI is 0, output control by the CKE[0] bit in SCR_SMCI is immediately reflected on the SCK pin, so there is a possibility that pulses with an unintended width may 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.

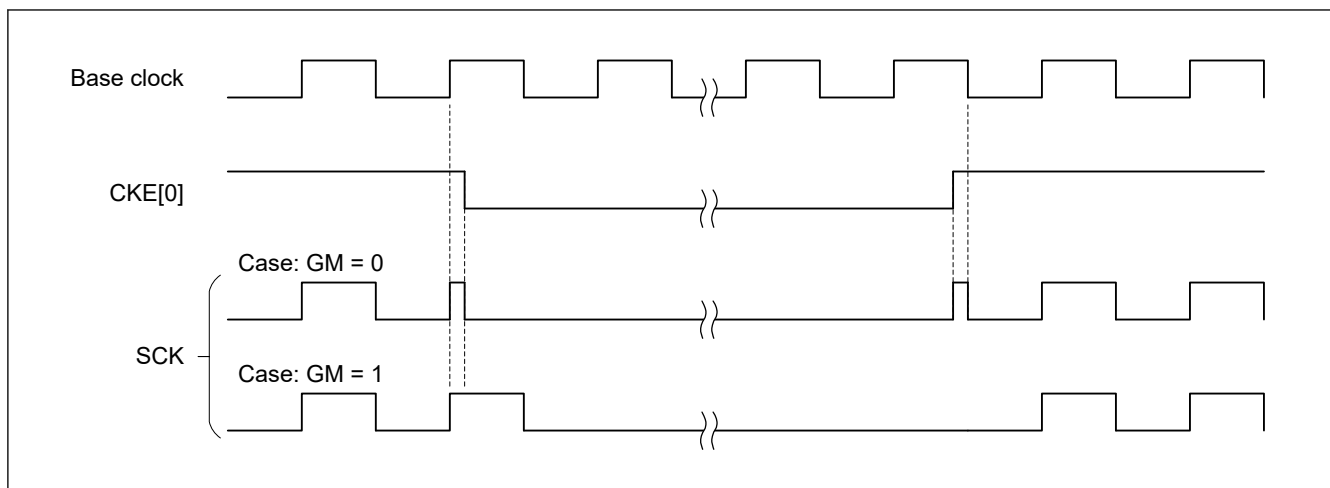
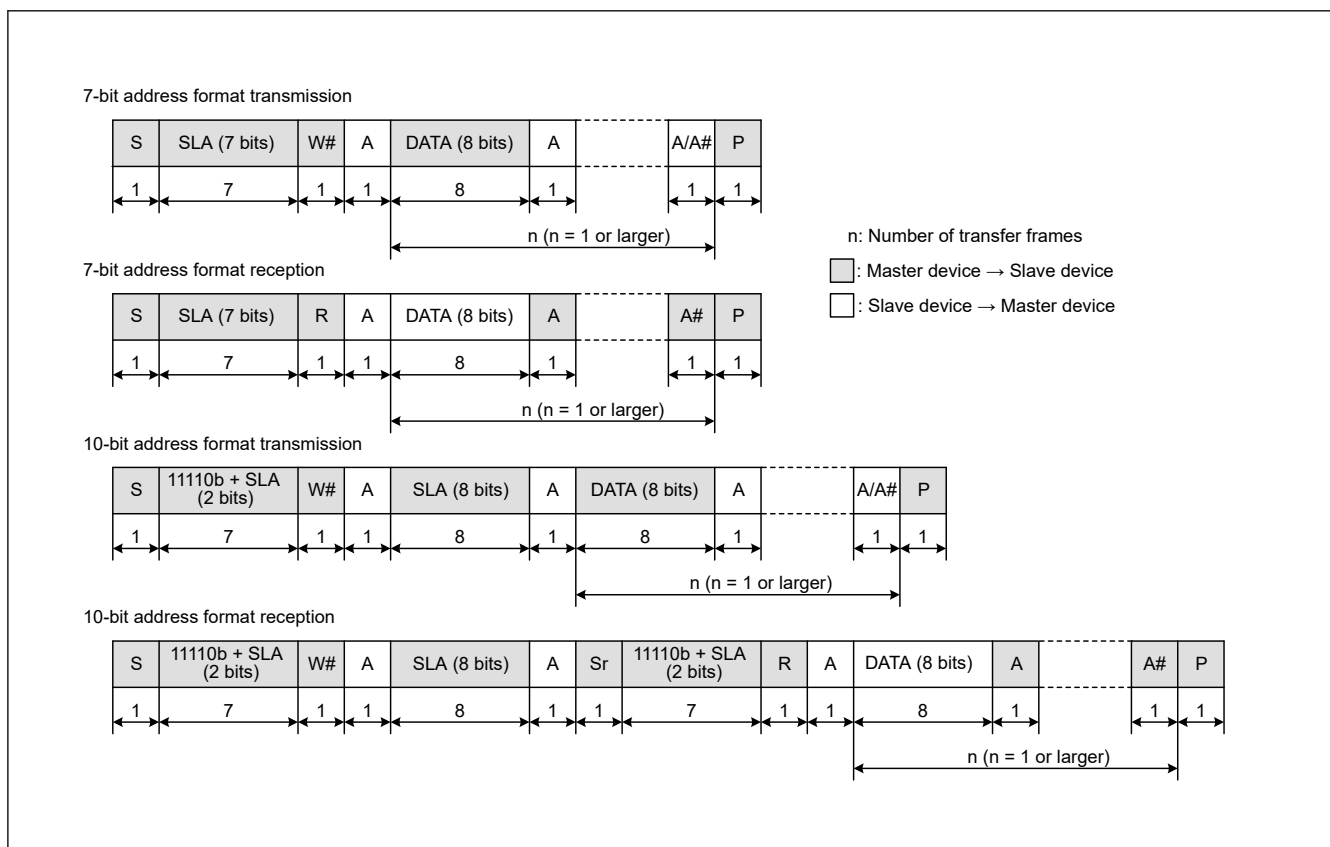


Figure 32.57 Clock Output timing

32.7 Operation in Simple IIC Mode

Simple IIC mode format is composed of 8 data bits and an acknowledge bit. By continuing into a slave-address frame after a start condition or restart condition, a master device can specify a slave device as the partner for communications. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied. The 8 data bits in all frames are transmitted in order from the MSB.

The I²C bus format and timing of the I²C bus are shown in Figure 32.58 and Figure 32.59.

Figure 32.58 I²C bus format

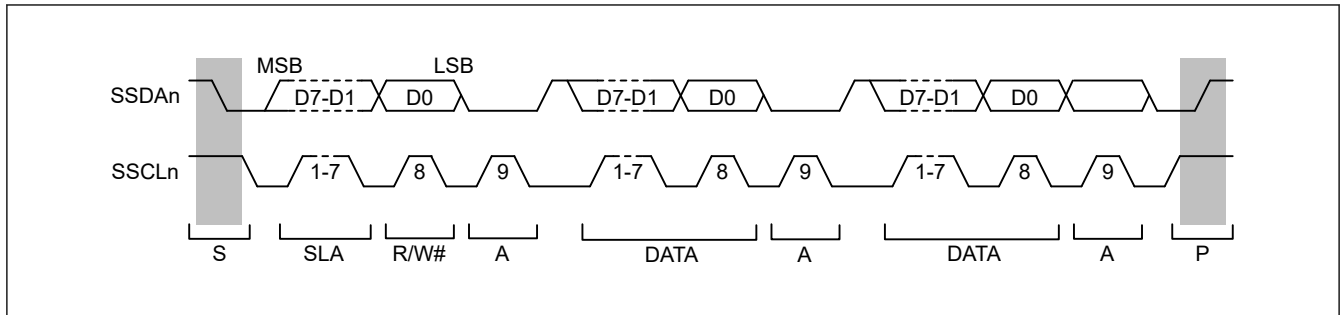


Figure 32.59 I²C bus timing when SLA is 7 bits

- **S:** Indicates a start condition, when the master device changes the level on the SSDAn line from high to low while the SSCLn 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 SSDAn line from high to low while the SSCLn 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 SSDAn line from low to high while the SSCLn line is high

32.7.1 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 SSDAn line falls (from the high level to the low level) and the SSCLn 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 SSCLn line falls (from the high level to the low level), the IICSTAREQ bit in SIMR3 is set to 0, and a start-condition generated interrupt is output

Writing 1 to the IICRSTAREQ bit in SIMR3 causes the generation of a restart condition. The generation of a restart condition proceeds through the following operations:

- The SSDAn line is released and the SSCLn line is kept at the low level
- The period at low level for the SSCLn line is set as half of a bit period at the bit rate determined by the BRR setting
- The SSCLn line is released (transition from the low to the high level)
- When a high level is detected on the SSCLn 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 SSDAn line falls (from the high level to the low level)
- The hold time for the restart condition is set as half of a bit period at the bit rate determined by the BRR setting
- The level on the SSCLn line falls (from the high level to the low level), the SIMR3.IICRSTAREQ bit is set to 0, and a restart-condition generated interrupt is output

Writing 1 to the SIMR3.IICSTPREQ bit causes the generation of a stop condition. The generation of a stop condition proceeds through the following operations:

- The level on the SSDAn line falls (from the high level to the low level) and the SSCLn line is kept at the low level
- The period at low level for the SSCLn line is set as half of a bit period at the bit rate determined by the BRR setting
- The SSCLn line is released (transition from the low to the high level)

- When a high level is detected on the SSCLn 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 SSDAn line is released (transition from the low to the high level), the SIMR3.IICSTPREQ bit is set to 0, and a stop-condition generated interrupt is output

Figure 32.60 shows the timing of operations in the generation of start, restart, and stop conditions.

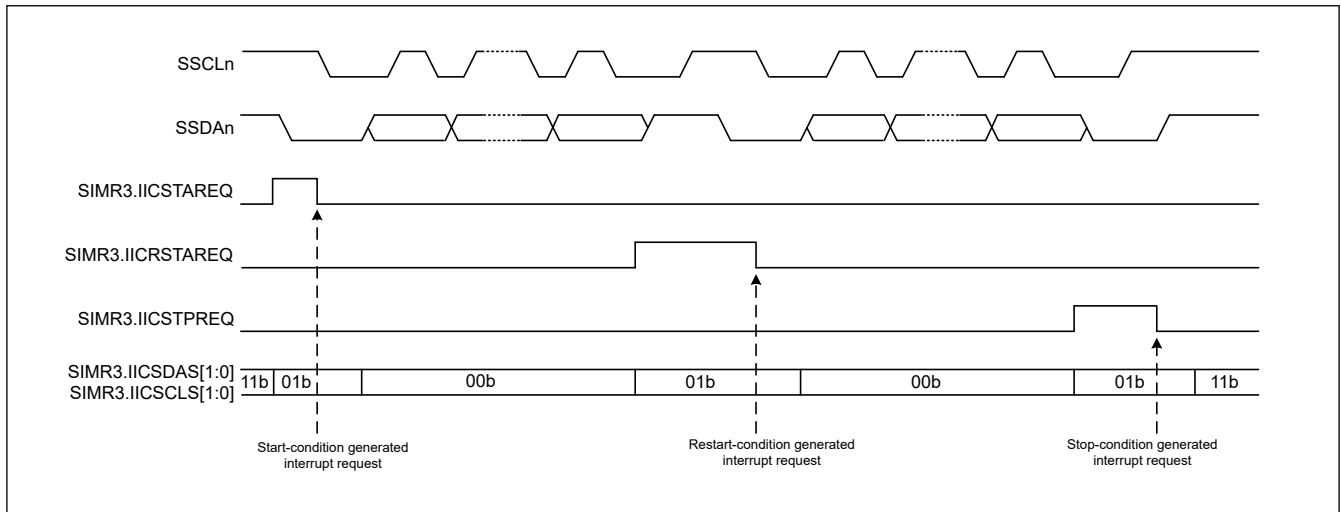


Figure 32.60 Timing of operations in generation of start, restart, and stop conditions

32.7.2 Clock Synchronization

The SSCLn line can be driven low if a wait is inserted by a slave device at the other side of the transfer. Setting the SIMR2.IICCSC bit to 1 applies control to obtain synchronization when a difference arises between the levels of the internal SSCLn clock signal and the level being input on the SSCLn pin.

When the SIMR2.IICCSC bit is set to 1, the level of the internal SSCLn clock signal changes from low to high. Counting to determine the period at a high level stops while the low level is being input on the SSCLn pin. Counting to determine the period at a high level starts after the transition of the input on the SSCLn pin to the high level.

The interval from this time until counting to determine the period at high level starts on the transition of the SSCLn pin to the high level, is the total of the delay of SSCLn output, delay for noise filtering of the input on the SSCLn pin (2 or 3 cycles of sampling clock for the noise filter), and delay for internal processing (1 or 2 cycles of PCLK). The period at high level of the internal SSCLn clock is extended even when other devices do not place the low level on the SSCLn 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 SSCLn pin and the internal SSCLn clock. If the SIMR2.IICCSC bit is 0, synchronization with the internal SSCLn 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 SSCLn 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 SSCLn 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 32.61 shows an example operation for synchronizing the clocks.

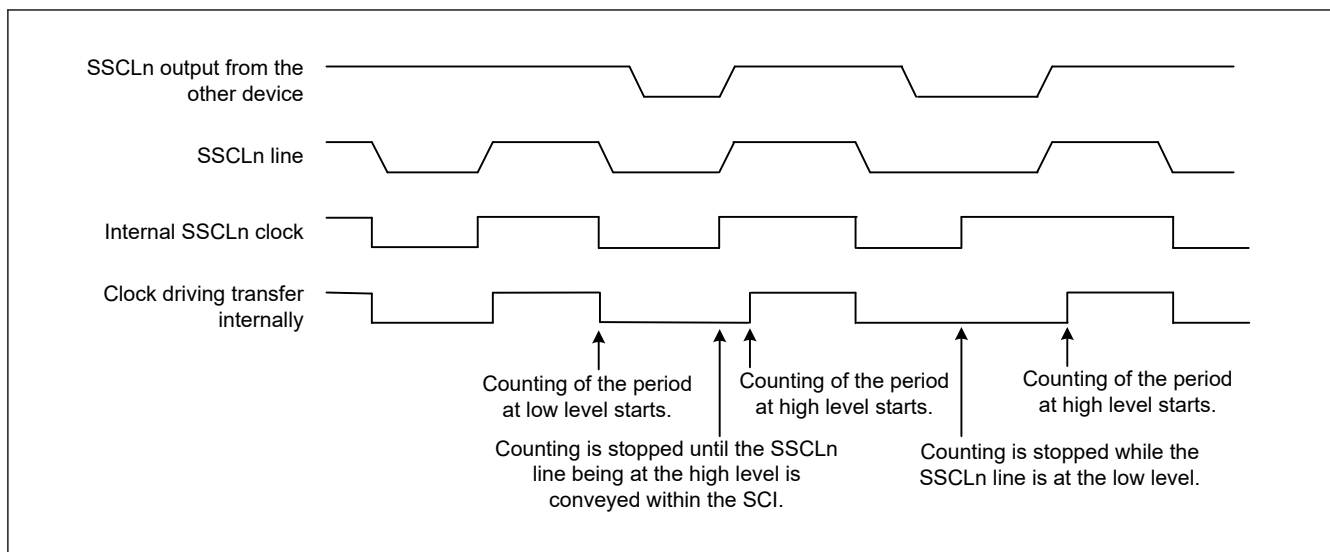


Figure 32.61 Example operations for clock synchronization

32.7.3 SSDAn Output Delay

The SIMR1.IICDL[4:0] bits can be used to set a delay for output on the SSDAn pin relative to falling edges of output on the SSCLn pin. Delay settings from 0 to 31 are selectable, representing periods of the corresponding numbers of cycles of the clock signal from the on-chip baud rate generator (derived by frequency-dividing the base clock, PCLK, by the divisor selected in the SMR.CKS[1:0] bits). A delay for output on the SSDAn pin applies to the start condition/restart condition/stop condition signal, 8-bit transmit data, and acknowledge bit.

If the SSDAn output delay is shorter than the time for the level on the SSCLn pin to fall, the change of the output on the SSDAn pin starts while the output level on the SSCLn pin is falling, creating a possibility of erroneous operation for slave devices. Ensure that settings for the delay of output on the SSDAn pin specify times greater than the time output on the SSCLn pin takes to fall (300 ns for IIC in normal mode and fast mode).

Figure 32.62 shows the timing of delays in SSDAn output.

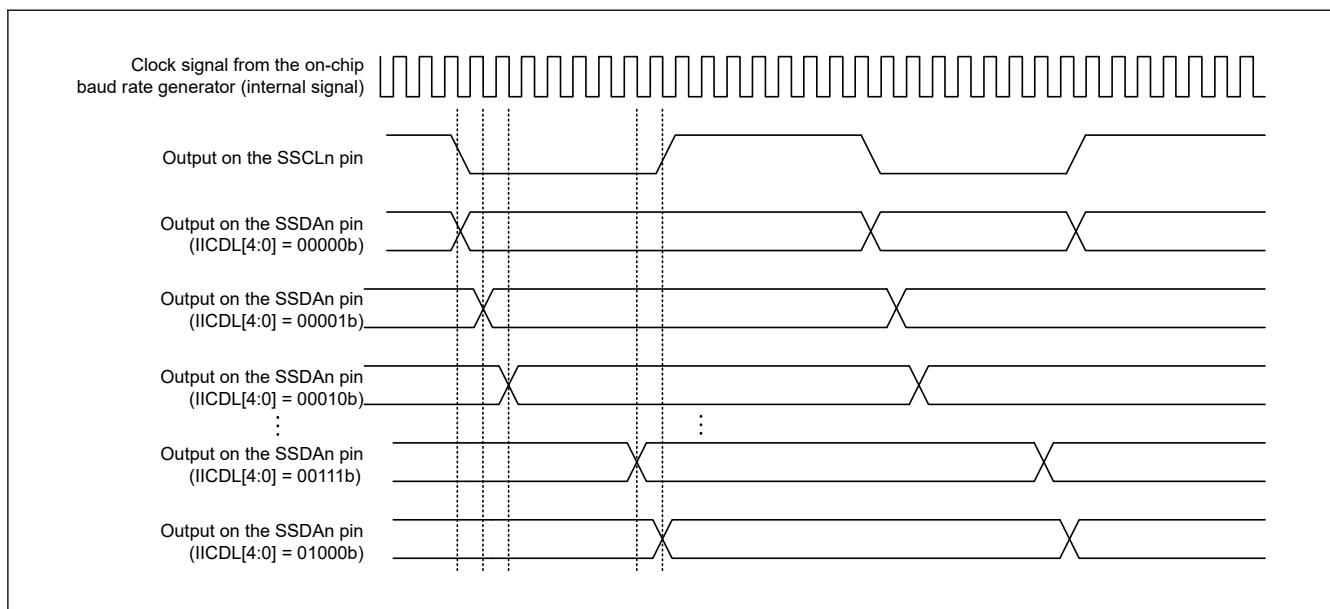


Figure 32.62 Timing of delays in SSDAn output

32.7.4 SCI Initialization in Simple IIC Mode

Before transferring data, write the initial value 0x00 to SCR and initialize the interface following the example shown in Table 32.32.

Before making any changes to the operating mode or transfer format, be sure to set SCR to its initial value. In simple IIC mode, the open-drain setting for the communication ports should be made on the port side.

Table 32.32 Example flow of SCI initialization in simple IIC mode

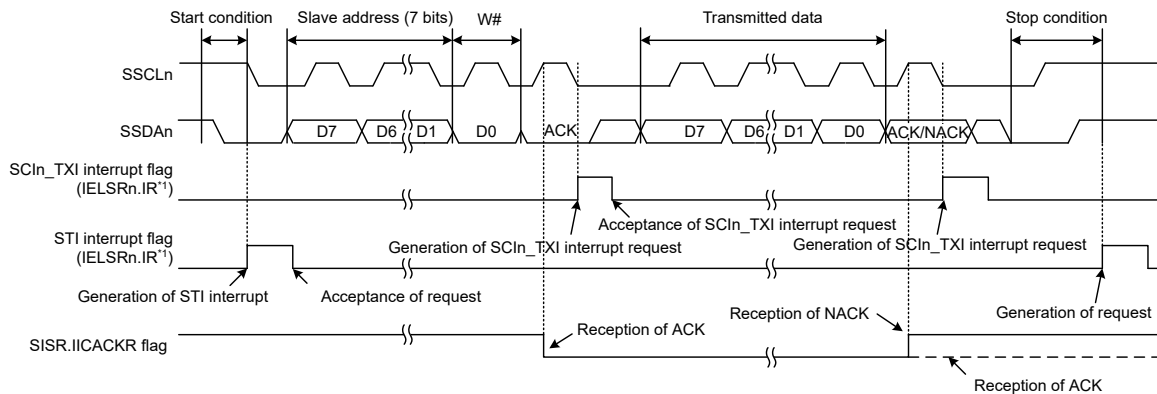
No.	Step Name	Description
1	Start of initialization	
2	Set the TIE, RIE, TE, RE, TEIE and CKE[1:0] bits in SCR to 0	
3	Set the I/O port functions	Set the I/O port to allow use (on N-channel open-drain output pins) of the SSCLn and SSDAn pin functions.
4	Set the IICSDAS[1:0] and IICSCLS[1:0] bits in SIMR3 to 11b	Place the SSCLn and SSDAn pins in the high-impedance state until a start condition is to be generated.
5	Set up the transfer or reception format in SMR and SCMR	Set the format for transmission and reception in SMR and SCMR. In SMR, set the CKS[1:0] bits to the target value and set the other bits to 0. In SCMR, set the SDIR bit to 1 and the SINV and SMIF bits to 0.
6	Set the value in BRR	Write the value for the targeted bit rate to BRR.
7	Set a value in MDDR	Write the value obtained by correcting a bit rate error in MDDR. This step is not required if the BRME bit in SEMR is set to 0.
8	Set the values in SEMR, SNFR, SIMR1, SIMR2, and SPMR	Set the values in SEMR, SNFR, SIMR1, SIMR2, and SPMR. Set the NFEN and BRME bits in SEMR. In SNFR, set the NFCS[2:0] bits. In SIMR1, set the IICM bit to 1 and the IICDL[4:0] bits as required. In SIMR2, set the IICACKT and IICCS bits to 1 and the IICINTM bits as required. In SPMR, set all the bits to 0.
9	Set the SCR.RE and TE bit to 1 and set the SCR.TIE, RIE and TEIE bits	Set the RE and TE bits in the SCR to 1. Then, set the SCR.TIE, RIE, and TEIE bits (for transmission and when the SIMR2.IICINTM bit is 1, set the RIE bit to 0). Setting the TE and RE bits to 1 enables the SSCLn and SSDAn pin functions.
10	Start of transmission or reception	

32.7.5 Operation in Master Transmission in Simple IIC Mode

Figure 32.63 and Figure 32.64 show examples of master transmission and Figure 32.65 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 32.37 for more information on the STI interrupt.

When 10-bit slave addresses are in use, steps [3] and [4] in Figure 32.65 are repeated twice.

In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the SCIn_TXI interrupt request generation timing during clock synchronous transmission.



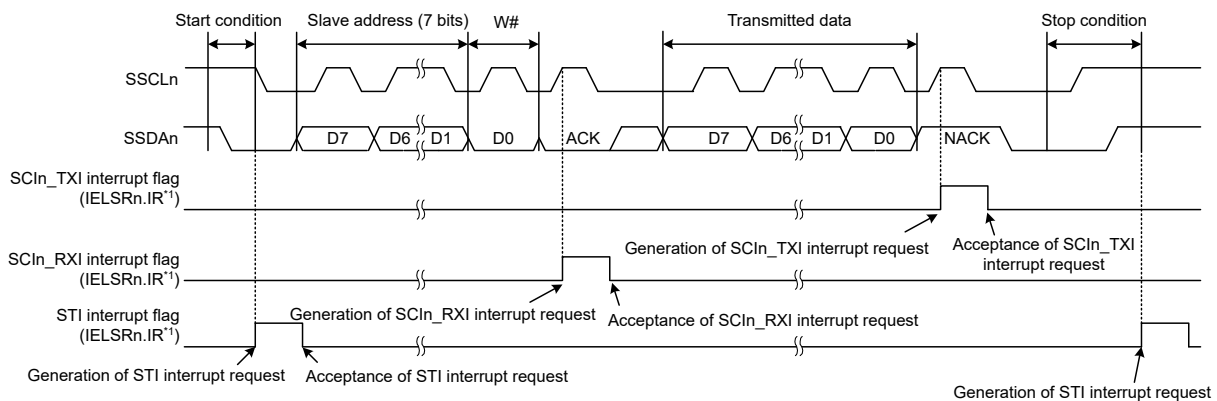
Note 1. See [section 16, Interrupt Controller Unit \(ICU\)](#) for information on the corresponding interrupt event number.

Figure 32.63 Example 1 of operations for master transmission in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts

When the SIMR2.IICINTM bit is set to 0 (use ACK/NACK interrupts) during master transmission, the DTC or DMAC is activated by the ACK interrupt as the trigger and required number of data bytes are transmitted. When the NACK is received, error processing such as transmission stop and retransmission is performed using the NACK interrupt as the trigger.

To restart communication for some reason after writing data in the TDR register, use the following procedure:

1. Set the TE and RE bits in the SCR register to 0 to stop communication.
2. Set 0xF0 in the SIMR3 register, release the I²C bus, and clear the generation of a condition.
3. If the RDRF flag in the SSR register is set, clear it.
4. Set the TE and RE bits in the SCR register to 1 and start the next communication.



Note 1. See [section 16, Interrupt Controller Unit \(ICU\)](#) for information on the corresponding interrupt event number.

Figure 32.64 Example 2 of operations for master transmission in simple IIC mode with 7-bit slave addresses, ACK interrupts, and NACK interrupts

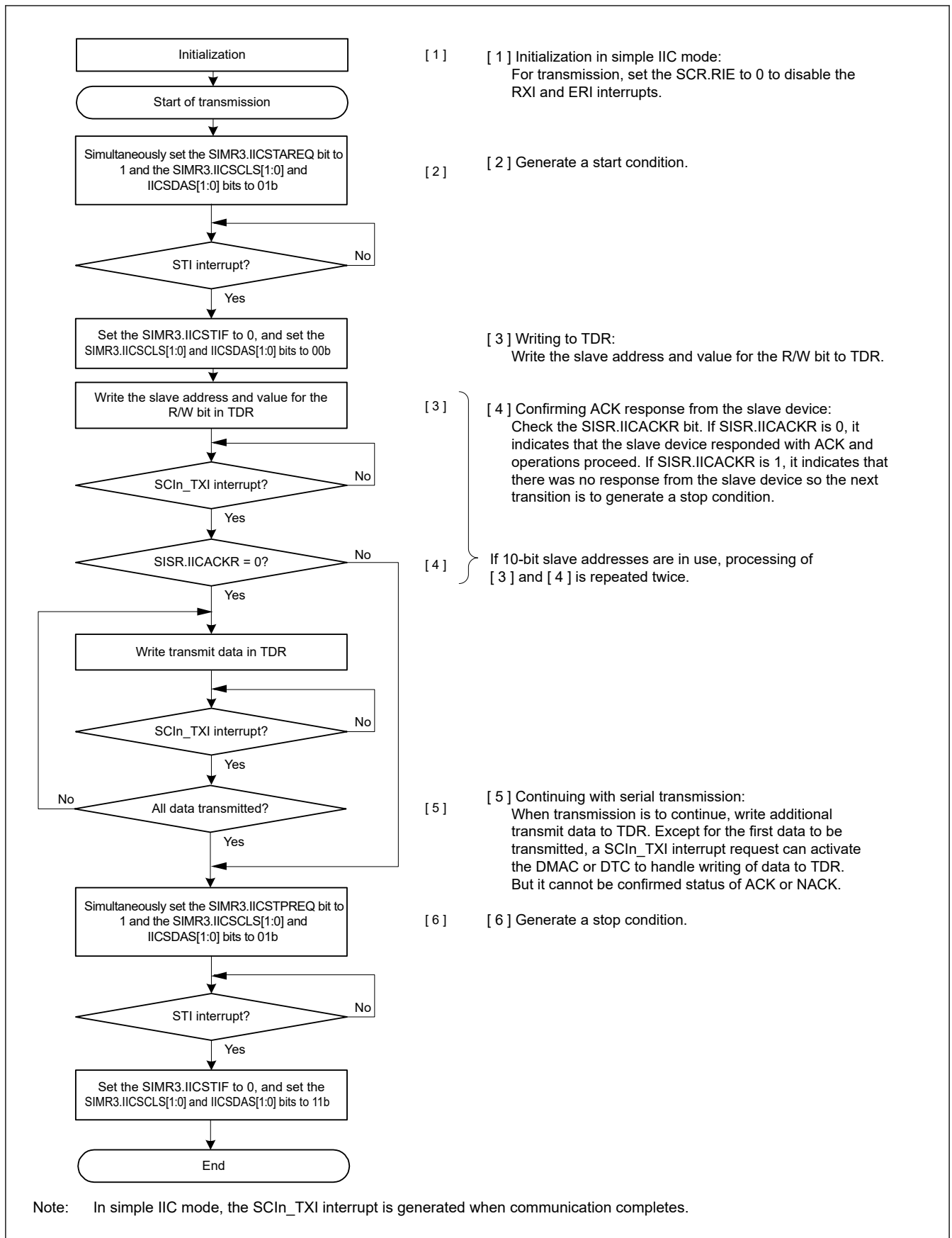


Figure 32.65 Example flow of master transmission in simple IIC mode with transmission interrupts and reception interrupts

32.7.6 Master Reception in Simple IIC Mode

Figure 32.66 shows an example operation in simple IIC mode master reception and Figure 32.67 shows an example flow of master reception.

The value of the SIMR2.IICINTM bit is assumed to be 1 (use reception and transmission interrupts).

In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the SCIn_TXI interrupt request generation timing during clock synchronous transmission.

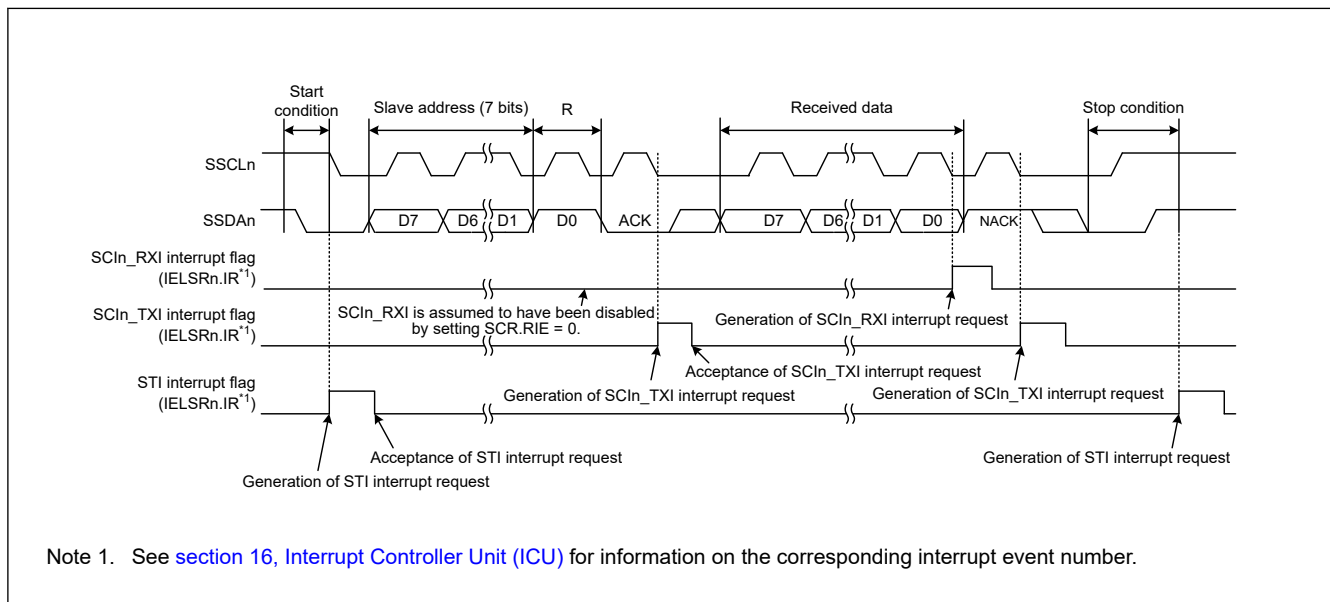
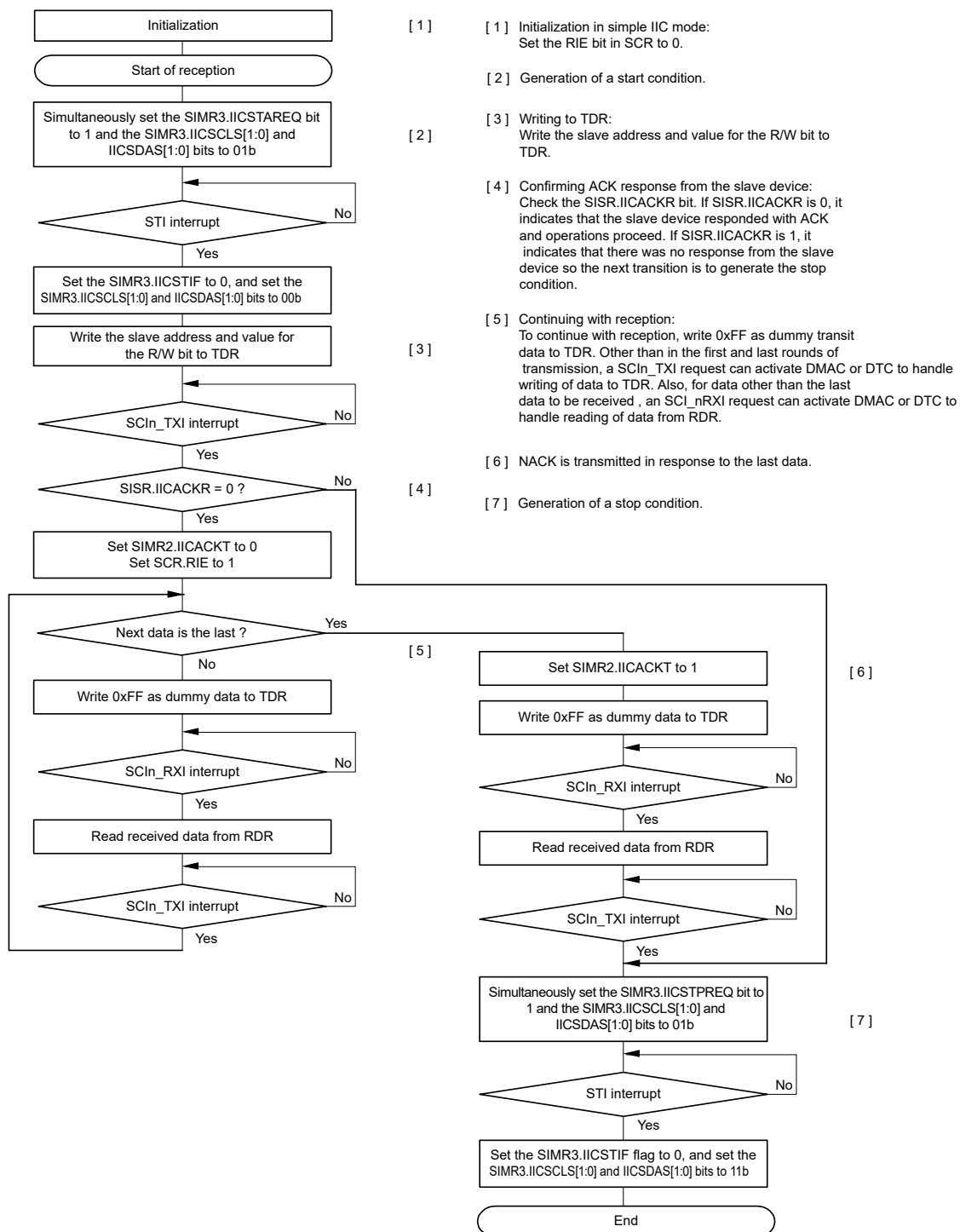


Figure 32.66 Example operations for master reception in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts



Note: In simple IIC mode, the SCIn TXI interrupt request is generated when communication is complete.

Figure 32.67 Example flow of master reception in simple IIC mode with transmission interrupts and reception interrupts

32.8 Operation in Simple SPI Mode

As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Using the settings for clock synchronous mode (SCMR.SMIF = 0, SIMR1.IICM = 0, SMR.CM = 1) and setting the SPMR.SSE bit to 1 places the SCI in simple SPI mode. However, the SSn pin function on the master side is not required for connection of the device used as the master in simple SPI mode when the configuration only has a single master. Therefore, set the SPMR.SSE bit to 0 in such cases.

Figure 32.68 shows an example of connections for simple SPI mode. Control a general port pin to produce the SSn output signal from the master.

In simple SPI mode, data is transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended. The data can be inverted by setting the SCMR.SINV bit to 1.

Because the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a shared clock signal. Additionally, because both the transmitter and receiver have a buffered structure, writing the next transmit data while transmission is in progress and reading previously received data while reception is in progress are both possible. This enables continuous transfer.

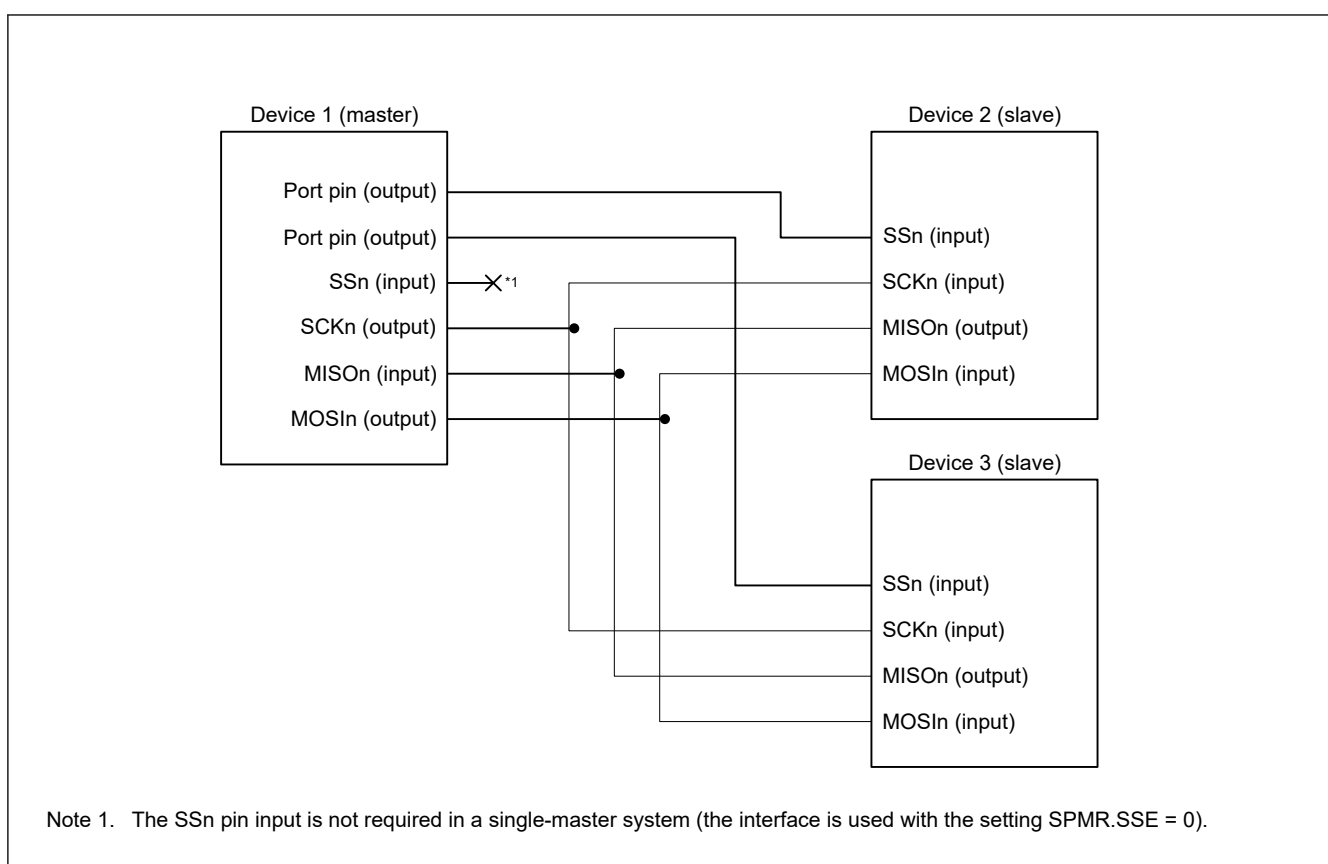


Figure 32.68 Example connections using simple SPI mode in single master mode with SPMR.SSE bit = 0

32.8.1 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 32.33 lists the relationship between the pin states, mode, and level on the SSn pin.

Table 32.33 States of pins by mode and input level on SSn pin (1 of 2)

Mode	Input on SSn pin	State of MOSIn pin	State of MISOIn pin	State of SCKn pin
Master mode* ¹	High level (transfer can proceed)	Output for data transmission* ²	Input for received data	Clock output* ³
	Low level (transfer cannot proceed)	High-impedance	Input for received data (but disabled)	High-impedance

Table 32.33 States of pins by mode and input level on SSn pin (2 of 2)

Mode	Input on SSn pin	State of MOSIn pin	State of MISO pin	State of SCKn pin
Slave mode	High level (transfer cannot proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer can proceed)	Input for received data	Output for data transmission	Clock input

Note 1. When there is only a single master (SPMR.SSE = 0), transfer is possible regardless of the input level on the SSn pin. This is equivalent to input of a high level on the SSn pin.

Note 2. The MOSIn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE bit = 0).

Note 3. The SCKn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE and RE bits = 00b) in a multi-master configuration (SPMR.SSE = 1).

32.8.2 SS Function in Master Mode

Setting the CKE[1:0] bits in the SCR to 00b and the MSS bit in the SPMR to 0 selects master operation. The SSn pin is not used in single-master configurations (SPMR.SSE = 0), so transmission or reception can proceed regardless of the value of the SSn pin.

When the level on the SSn pin is high in a multi-master configuration (SPMR.SSE = 1), a master device outputs clock signals from the SCKn pin before starting transmission or reception to indicate that there are no other masters or another master is performing reception or transmission.

When the level on the SSn pin is low in a multi-master configuration (SPMR.SSE = 1), there are other masters, and a transmission or reception is in progress. The MOSIn output and SCKn pins are placed in the high-impedance state and starting transmission or reception is not possible. In addition, the value of the SPMR.MFF bit is 1, indicating a mode fault error. In a multi-master configuration, start error processing by reading SPMR.MFF flag. If a mode fault error occurs while transmission or reception is in progress, transmission or reception 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.

32.8.3 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 SSn pin is high, the MISO output pin is in the high-impedance state and clock input through the SCKn pin is ignored. When the SSn pin is low, clock input through the SCKn pin is valid and transmission or reception can proceed.

If the input on the SSn pin changes from low to high during transmission or reception, the MISO output pin is placed in the high-impedance state. Meanwhile, the internal processing for transmission or reception continues at the rate of the clock input through the SCKn pin until processing for the character being transmitted or received is complete, after which it stops, and the appropriate interrupt (SCI_n_TXI, SCI_n_RXI, or SCI_n_TEI) is generated.

32.8.4 Relationship between Clock and Transmit/Receive Data

The CKPOL and CKPH bits in the SPMR register can be used to set up the clock for use in transmission and reception in four different ways. The relation between the clock signal and the transmission and reception of data is shown in [Figure 32.69](#). The relation is the same for both master and slave operation. This is the same as when the level on the SSn pin is high.

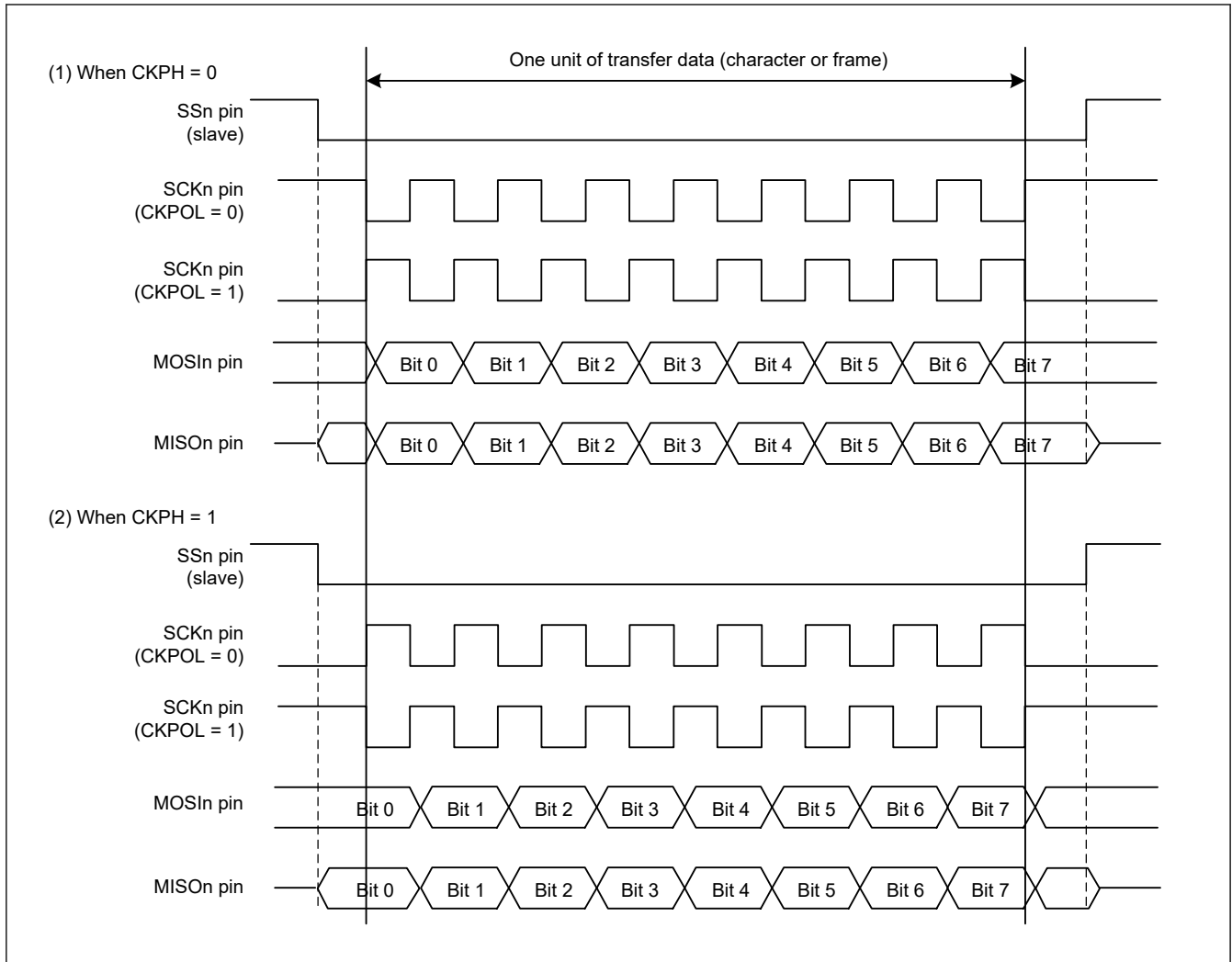


Figure 32.69 Relation between clock signal and transmit or receive data in simple SPI mode

32.8.5 SCI Initialization in Simple SPI Mode

Initialization in simple SPI mode is the same as in clock synchronous mode. See [section 32.5.3. SCI Initialization in Clock Synchronous Mode](#) for an example initialization flow. The CKPOL and CKPH bits in the SPMR register must be set to ensure that the clock signal is suitable for both master and slave devices.

Always initialize the SCR register before making any changes to the operating mode or transfer format.

Note: Only the RE bit is set to 0. The SSR. ORER, FER, PER, and RDR flags are not initialized.

Changing the value of the TE bit from 1 to 0 or from 0 to 1 when the TIE bit in the SCR register is 1 at the same time, leads to the generation of a transmit data empty interrupt (SCIn_TXI).

32.8.6 Transmission and Reception of Serial Data in Simple SPI Mode

In master operation, ensure that the SSn pin of the slave device on the other side of the transfer is at the low level before starting the transfer and at the high level on completion of the transfer. Otherwise, the procedures are the same as in clock synchronous mode.

32.9 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 the PCLK is selected in the CKS[1:0] bits in SMR/SMR_SMCI.

Figure 32.70 shows an example where the PCLK is selected in the CKS[1:0] bits in SMR/SMR_SMCI, the BRR bit is set to 0, and the MDDR is set to 160 in asynchronous mode. In this example, the cycle of the base clock is evenly corrected ($256/160$) and the bit rate is also corrected ($160/256$).

Note: Enabling an internal clock causes bias, and expansion and contraction are generated in the pulse width of the internal base clock.

Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).

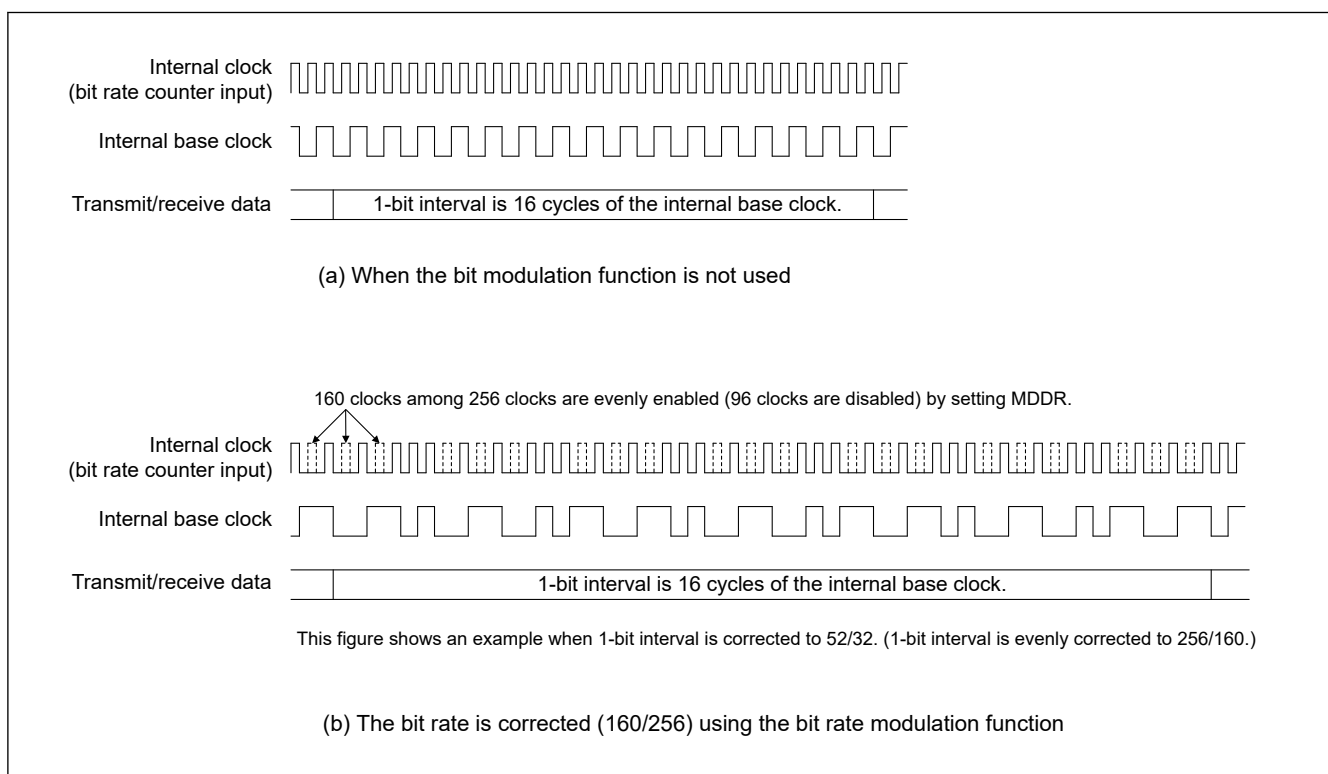


Figure 32.70 Example internal base clock when bit rate modulation function is used

32.10 Interrupt Sources

32.10.1 Buffer Operation for SCIn_TXI and SCIn_RXI Interrupts (non-FIFO selected)

If the conditions for an SCIn_TXI and SCIn_RXI interrupt are satisfied while the interrupt status flag in the ICU is 1, the ICU does not output the interrupt request but saves it internally with a capacity for retention of one request per source.

When the interrupt status flag in the ICU is set to 0, the interrupt request retained within the ICU is output. The internally retained interrupt request is automatically discarded when the actual interrupt is output. Clearing of the associated interrupt enable bit (the TIE or RIE bit in the SCR/SCR_SMCI) can also be used to discard an internally retained interrupt request.

32.10.2 Buffer Operation for SCIn_TXI and SCIn_RXI Interrupts (FIFO selected)

When an interrupt status flag in the ICU is set to 1, the SCIn_TXI and SCIn_RXI interrupts do not output interrupt requests to the ICU. When an interrupt status flag of the ICU is set to 0, and if the conditions for an SCIn_TXI and SCIn_RXI interrupts are satisfied, an interrupt request is generated.

32.10.3 Interrupts in Asynchronous, Clock Synchronous, and Simple SPI Modes

(1) Non-FIFO selected

Table 32.34 lists interrupt sources in asynchronous mode, clock synchronous mode, and simple SPI mode. A different interrupt vector can be assigned to each interrupt source. Individual interrupt sources can be enabled or disabled with the enable bits in the SCR register.

If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated when transmit data is transferred from the TDR or TDRHL register*¹ to the TSR register. An SCIn_TXI interrupt request can also be generated by using a single instruction to set the SCR.TE and SCR.TIE bits to 1 at the same time. An SCIn_TXI interrupt request can activate the DTC or DMAC to handle data transfer.

An SCIn_TXI interrupt request is not generated by setting the SCR.TE bit to 1 when SCR.TIE is 0 or by setting the SCR.TIE bit to 1 when the SCR.TE is 1.*²

When new data is not written by the time of transmission of the last bit of the current transmit data and SCR.TEIE is 1, the SSR.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated. Additionally, when SCR.TE is 1, the SSR.TEND flag retains the value 1 until more transmit data is written to the TDR or TDRHL register*¹, and setting SCR.TEIE to 1 leads to the generation of an SCIn_TEI interrupt request.

Writing data to the TDR or TDRHL register*¹ leads to clearing of the SSR.TEND flag and, after a certain time, discarding of the SCIn_TEI interrupt request.

If the SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated when received data is stored in the RDR register. An SCIn_RXI interrupt request can activate the DTC or DMAC to handle data transfer.

Setting any of the SSR.ORER, FER, PER flags to 1 when the SCR.RIE bit is 1 leads to the generation of an SCIn_ERI interrupt request. An SCIn_RXI interrupt request is not generated in this case. Clearing all three flags (ORER, FER, PER) leads to discarding of the SCIn_ERI interrupt request.

Note 1. When asynchronous mode and 9-bit data length are selected.

Note 2. To temporarily prohibit SCIn_TXI interrupts on transmission of the last of the data when a new round of transmission is to be started, after handling the transmission-completed interrupt, control activation of the interrupt by using the interrupt request enable bit in the ICU rather than using the SCR.TIE bit. This approach can prevent the suppression of SCIn_TXI interrupt requests in the transfer of new data.

(2) FIFO selected

Table 32.35 lists interrupt sources in FIFO selected mode.

If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated when the stored amount of data in the FTDRL register becomes the threshold value indicated in FCR.TTRG or below. An SCIn_TXI interrupt request can also be generated by using a single instruction to set the SCR.TIE and SCR.TE bits to 1 simultaneously or by setting SCR.TIE to 1 when SCR.TE is 1.

An SCIn_TXI interrupt request is not generated by setting SCR.TE to 1 when SCR.TIE is 0.

If SCR.TEIE is 1 and if the next data is not written to the FTDRL register by the time the last bit of the transmit data is sent, the SSR_FIFO.TEND flag is set to 1 and the SCIn_TEI interrupt request is generated.

If SCR.RIE is 1, the SCIn_RXI interrupt request is generated when the stored amount of data in the FRDRL register is equal to or greater than the threshold value indicated in FCR.RTRG. When RTRG is 0, an SCIn_RXI interrupt does not occur even when the amount of data in the receive FIFO is equal to 0.

If the SCR.RIE bit is 1, when the SSR_FIFO.ORER flag is set to 1 or data with a framing error or a parity error is stored in the FRDRL register, the SCIn_ERI interrupt request is generated. When the amount of data stored in the FRDRL register is at the threshold value or above, the SCIn_RXI interrupt request is also generated. The SCIn_ERI interrupt request can be canceled, in which case SSR_FIFO.ORER, FER, and PER flags are all cleared.

Table 32.34 SCI interrupt sources with non-FIFO selected (1 of 2)

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_ERI (n = 0 to 9)	Receive error* ¹	SSR.ORER, SSR.FER, SSR.PER, DCCR.DFER, DCCR.DPER	SCR.RIE	Not possible
SCIn_RXI (n = 0 to 9)	Receive data full	SSR.RDRF	SCR.RIE	Possible
	Address match	DCCR.DCMF	SCR.RIE	Possible
SCIn_AM (n = 0 to 9)	Address match	DCCR.DCMF	—	Not possible
SCIn_TXI (n = 0 to 9)	Transmit data empty	SSR.TDRE	SCR.TIE	Possible

Table 32.34 SCI interrupt sources with non-FIFO selected (2 of 2)

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_TEI (n = 0 to 9)	Transmit end	SSR.TEND	SCR.TEIE	Not possible

Note 1. The interrupt flag is only ORER when in clock synchronous and simple SPI mode.

Table 32.35 SCI interrupt sources with FIFO selected

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_ERI (n = 0 to 9)	Receive error* ¹	SSR_FIFO.ORER, SSR_FIFO.FER, SSR_FIFO.PER, DCCR.DFER, DCCR.DPER	SCR.RIE	Not possible
		SSR_FIFO.DR (when FCR.DRES = 1)	SCR.RIE	Not possible
SCIn_RXI (n = 0 to 9)	Receive data full	SSR_FIFO.RDF	SCR.RIE	Possible
	Receive data ready	SSR_FIFO.DR (when FCR.DRES = 0)	SCR.RIE	Possible
	Address match	DCCR.DCMF	SCR.RIE	Possible
SCIn_AM (n = 0 to 9)	Address match	DCCR.DCMF	—	Not possible
SCIn_TXI (n = 0 to 9)	Transmit data empty	SSR_FIFO.TDFE	SCR.TIE	Possible
SCIn_TEI (n = 0 to 9)	Transmit end	SSR_FIFO.TEND	SCR.TEIE	Not possible

Note 1. The interrupt flag is only ORER when in clock synchronous and simple SPI mode.

32.10.4 Interrupts in Smart Card Interface Mode

[Table 32.36](#) lists interrupt sources in smart card interface mode. A transmit end interrupt (SCIn_TEI) request and an address match (SCIn_AM) request cannot be used in this mode.

Table 32.36 SCI Interrupt sources

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_ERI (n = 0 to 9)	Receive error or error signal detection	SSR_SMCI.ORER, SSR_SMCI.PER, SSR_SMCI.ERS	SCR_SMCI.RIE	Not possible
SCIn_RXI (n = 0 to 9)	Receive data full	SSR_SMCI.RDRF	SCR_SMCI.RIE	Possible
SCIn_TXI (n = 0 to 9)	Transmit data empty	SSR_SMCI.TEND	SCR_SMCI.TIE	Possible

Data transmission or reception using the DTC or DMAC is also possible in smart card interface mode, similar to normal SCI mode. In transmission, when the SSR_SMCI.TEND flag is set to 1, an SCIn_TXI interrupt request is generated. This SCIn_TXI interrupt request activates the DTC or DMAC, allowing transfer of transmit data if the SCIn_TXI request is previously specified as a source of DTC or DMAC activation. The TEND flag is automatically set to 0 when the DTC or DMAC 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 or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission after an error occurrence. However, the SSR_SMCI.ERS flag is not automatically set to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the SCR_SMCI.RIE bit to 1 to enable an SCIn_ERI interrupt request to be generated at error occurrence.

When transmitting or receiving data using the DTC or DMAC, always enable the DTC or DMAC before making the SCI settings. For DTC or DMAC settings, see [section 20, Data Transfer Controller \(DTC\)](#), [section 19, DMA Controller \(DMAC\)](#).

In reception, an SCIn_RXI interrupt request is generated when receive data is set to the RDR register. This SCIn_RXI interrupt request activates the DTC or DMAC, allowing transfer of the receive data if the SCIn_RXI request is previously specified as a source of DTC or DMAC activation. If an error occurs, the error flag is set. Therefore, the DTC or DMAC is not activated and an SCIn_ERI interrupt request is issued to the CPU instead. The error flag must be cleared.

32.10.5 Interrupts in Simple IIC Mode

Table 32.37 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 or DMAC 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 SSCLn signal for the 8th bit. If SCIn_RXI is previously set up as an activation source for the DTC or DMAC, the SCIn_RXI request activates the DTC or DMAC to handle transfer of the received data.
- An SCIn_TXI request is generated on the falling edge of the SSCLn signal for the 9th bit (acknowledge bit). If SCIn_TXI is previously set up as an activation source for the DTC or DMAC, the SCIn_TXI request activates the DTC or DMAC 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 SSDAn pin is low on the rising edge of the SSCLn signal for the 9th bit (acknowledge bit)
- An SCIn_TXI request (NACK detection) is generated if the input on the SSDAn pin is high on the rising edge of the SSCLn signal for the 9th bit (acknowledge bit)
- If SCIn_RXI is previously set up as an activation source for the DTC or DMAC, the SCIn_RXI request activates the DTC or DMAC to handle transfer of the received data.

If the DTC or DMAC is used for data transfer in reception or transmission, always set up and enable the DTC or DMAC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in SIMR3 are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 32.37 SCI interrupt sources

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_RXI (n = 0 to 9)	Reception, ACK detection	—	SCMR.RIE	Possible ^{*1}
SCIn_TXI (n = 0 to 9)	Transmission, NACK detection	—	SCMR.TIE	Possible
SCIn_TEI(STIn) (n = 0 to 9)	Completion of generation of a start, restart, or stop condition	SIMR3.IICSTIF	SCMR.TEIE	Not possible

Note 1. Activation of the DTC or DMAC is only possible when the SIMR2.IICINTM bit is 1 (use reception and transmission interrupts)

32.11 Event Linking

By using interrupt request signals as event signals, the SCIn can provide linked operation through the ELC for modules selected in advance.

Event signals can be output regardless of the values of the associated interrupt request enable bits.

(1) Error event output (receive error or error signal detected) (SCIn_ERI, n = 2)

- 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

(2) Receive data full event output (SCIn_RXI, n = 2)

- Indicates that ACK is detected if the SIMR2.IICINTM bit is 0 in simple IIC mode
- Indicates that the 8th-bit SSCLn falling edge is detected if the SIMR2.IICINTM bit is 1 in simple IIC mode
- When the SIMR2.IICINTM bit is 1 during master transmission in simple IIC mode, set the ELC so that receive data full events are not used

Non-FIFO selected

- Indicates that received data is set in the Receive Data Register (RDR or RDRHL).

FIFO selected

- Using this event output is prohibited.

(3) Transmit data empty event output (SCIn_TXI, n = 2)

- Indicates that the SCR/SCR_SMCI.TE bit is changed from 0 to 1
- Indicates that transmission is complete in smart card interface mode
- Indicates that NACK is detected if the SIMR2.IICINTM bit is 0 in simple IIC mode
- Indicates that the 9th-bit SSCLn falling edge is detected if the SIMR2.IICINTM bit is 1 in simple IIC mode

Non-FIFO selected

- Indicates that transmit data is transferred from the Transmit Data Register (TDR or TDRHL) to the Transmit Shift Register (TSR).

FIFO selected

- Using this event output is prohibited.

(4) Transmit end event output (SCIn_TEI, n = 0 to 9)

- Indicates the completion of transmission
- Indicates that the starting condition, resumption condition, or termination condition is generated in simple IIC mode

Note: When FIFO is selected, using this event output is prohibited

32.12 Address Non-match Event Output (SCI0_DCUF)

SCI0_DCUF indicates the non-match of comparison data (CDR.CMPD) with receive data that is one frame of the data that is received when DCCR.DCME is set to 1 in asynchronous mode, including multi-processor mode. This event can be used for Snooze end request only. In detail, see [section 13, Power-Saving Functions](#).

32.13 Noise Cancellation Function

[Figure 32.71](#) shows the configuration of the noise filter used for noise cancellation. The noise filter consists of a 2-stage flip-flop circuit and a match detection circuit. When the input signals of the noise filter and the output signals of the 2-stage flip-flop circuits completely match, the matched level is conveyed as an internal signal. Unless otherwise matched, the previous value is retained. When the same level is retained for 3 cycles or longer on the sampling clock of the noise filter, it is considered as a valid receive signal. A change in pulse for 3 cycles or shorter is considered as noise, not as a receive signal.

When SEMR.ABCS = 0 and SEMR.ABCSE = 0, the cycle is 1/16 of a 1-bit period.

When SEMR.ABCS = 1 and SEMR.ABCSE = 0, the cycle is 1/8 of a 1-bit period.

When SEMR.ABCSE = 1, the cycle is 1/6 of a 1-bit period.

In asynchronous mode, the noise cancellation function can be applied to the receive signal input to the RXDn pin. The receive level of the RXDn is taken in the flip-flop circuit of the noise filter on the base clock of asynchronous mode.

In simple IIC mode, this function can be used for each input on SSDAn and SSCLn. The sampling clock is selected from four baud rate generator settings (1, 2, 4, and PCLK divided by 4) in SNFR.NFCS.

If the base clock is stopped once with the noise filter enabled and then the base clock input is restarted again, the noise filter operation resumes from the state where the clock was stopped. When SCR.TE and SCR.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, the function determines that a level match is detected and the result is conveyed as an internal signal. When the level being input corresponds to 0, the initial output of the noise filter is retained until the level matches in three consecutive sampling cycles.

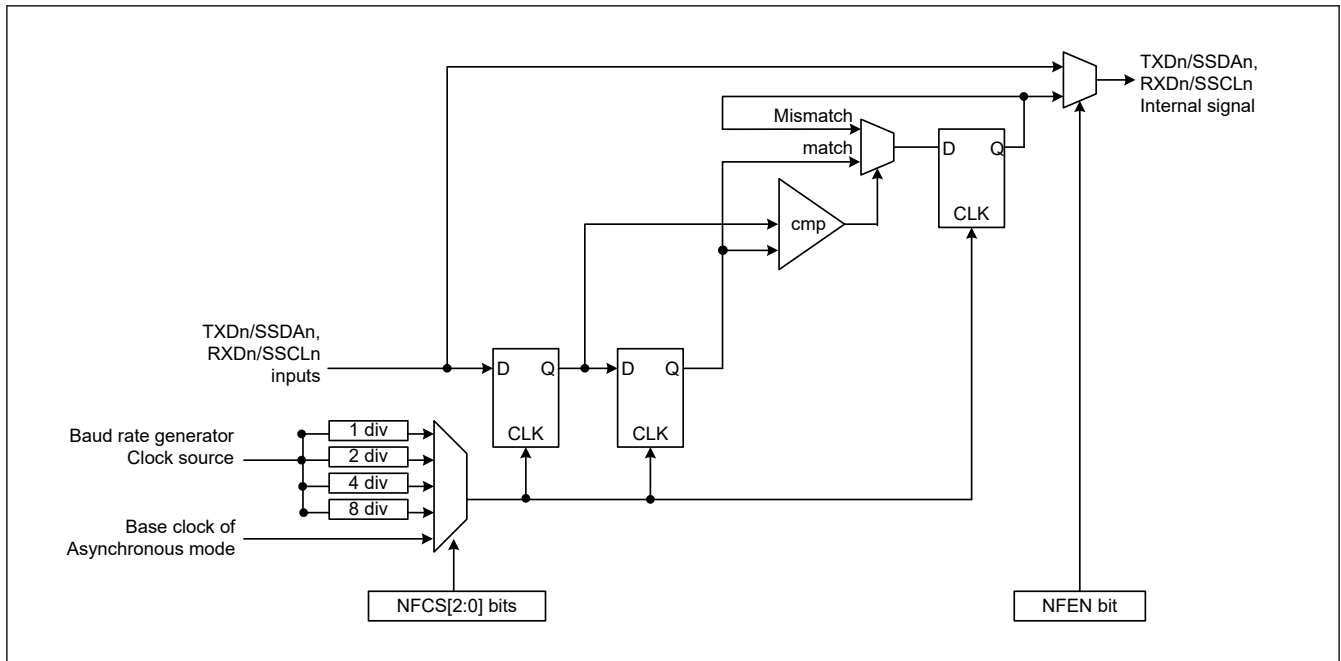


Figure 32.71 Digital noise filter circuit block diagram

32.14 Usage Notes

32.14.1 Settings for the Module-Stop Function

The Module Stop Control Register B (MSTPCRB) can enable or disable SCI operation. The SCI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

32.14.2 SCI Operation during Low Power State

(1) Transmission

When setting the module to the stopped state or in transitions to Software Standby, stop operations (by setting the TIE, TE, and TEIE bits in the SCR/SCR_SMCI to 0) after switching the TXDn pin to the general I/O port pin function. When setting I/O port as an SCI connection, the SPTR register can control the state of the TXDn pin. Setting the TE bit to 0 initializes the TSR register and the TEND bit in the SSR/SSR_SMCI is initialized to 1 with non-FIFO selected, and the value is retained, with FIFO selected. Depending on the port settings and SPTR register settings, output pins might output the level before a transition to the low-power state is made after release from the module-stopped state or Software Standby mode. When transitions to these states are made during transmission, the transmitted data becomes indeterminate.

To transmit data in the same transmission mode after cancellation of the low-power state:

1. Set the TE bit to 1.
2. Read SSR/SSR_FIFO/SSR_SMCI.
3. Write data to TDR sequentially to start data transmission.

To transmit data with a different transmission mode, initialize the SCI first.

[Figure 32.72](#) shows an example flow of transition to Software Standby mode during transmission. [Figure 32.73](#) and [Figure 32.74](#) show the port pin states during transition to Software Standby mode.

Before specifying the module-stop state or making a transition to Software Standby mode from the transmission mode using DTC or DMAC transfer, stop the transmit operations (TE = 0). To start transmission after cancellation using the DTC or DMAC, set the TE bit to 1. The SCIn_TXI interrupt flag is set to 1 and transmission starts using the DTC or DMAC.

(2) Reception

When address match function is not used as wake-up 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 32.75 shows an example flow of transition to Software Standby mode during reception.

When address match function is used as wake-up condition

Before specifying the module-stop state or making a transition to Software Standby mode:

1. Set the operations after cancellation of the low power state.
2. Set CDR.CMPD and DCCR.DCME to 1.
3. Set the receive operations (RE = 1 in SCR/SCR_SMCI).
4. Set the module-stop state or Software Standby mode.

When SCI transfers to low power mode, if the receive data pin (RXD) is at the low level, set SEMR.RXDESEL = 0.

When setting SEMR.RXDESEL = 1, there is a possibility that a start bit (falling edge of RXD pin) cannot be detected on release of the low power mode.

Figure 32.76 shows an example flow of transition to Software Standby mode during reception with address match.

When using SCI0 in Snooze mode

When using SCI0 in Snooze mode, some restrictions apply, including maximum bit rates. For details, see [section 13, Power-Saving Functions](#).

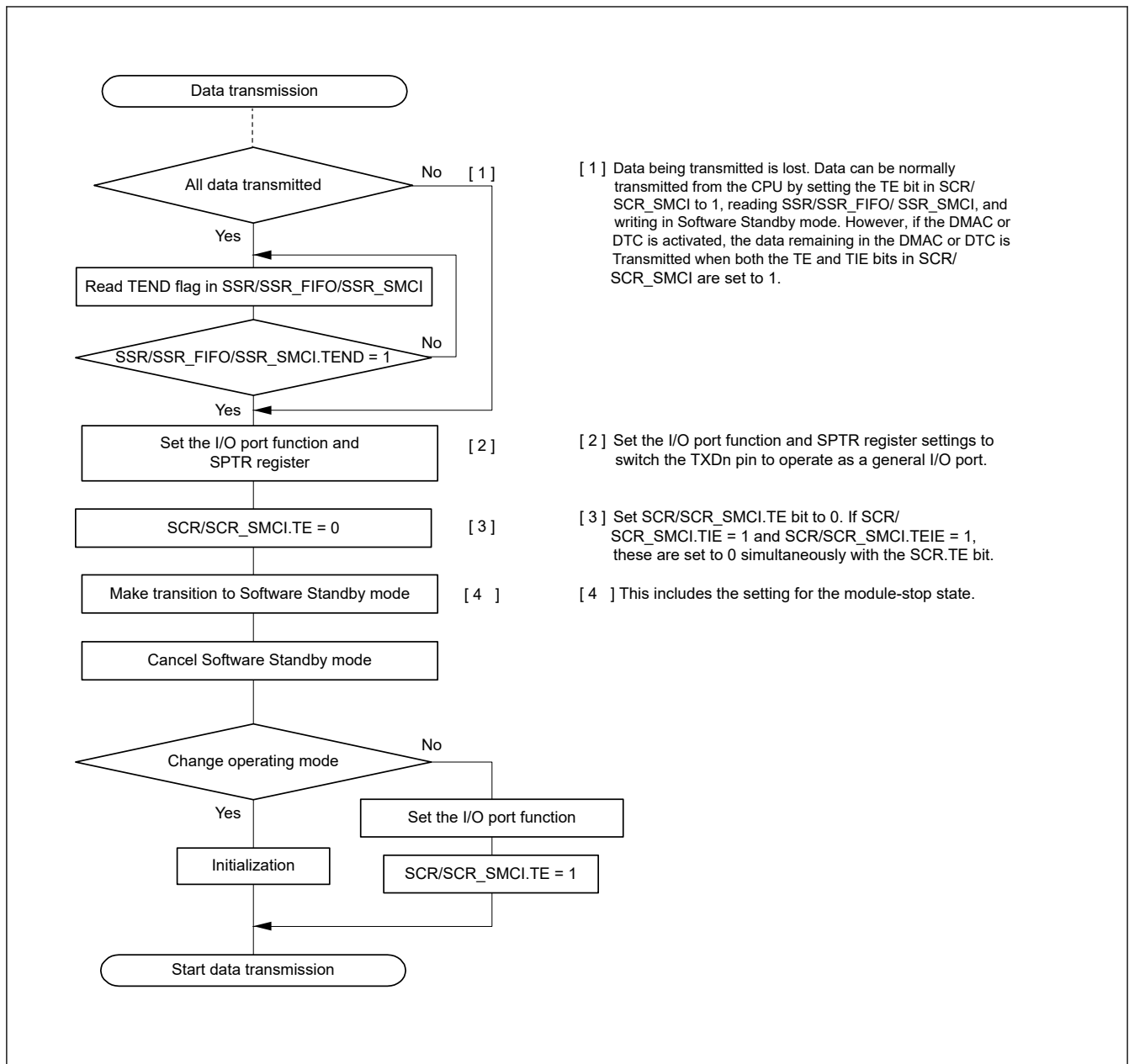


Figure 32.72 Example flow of transition to Software Standby mode during transmission

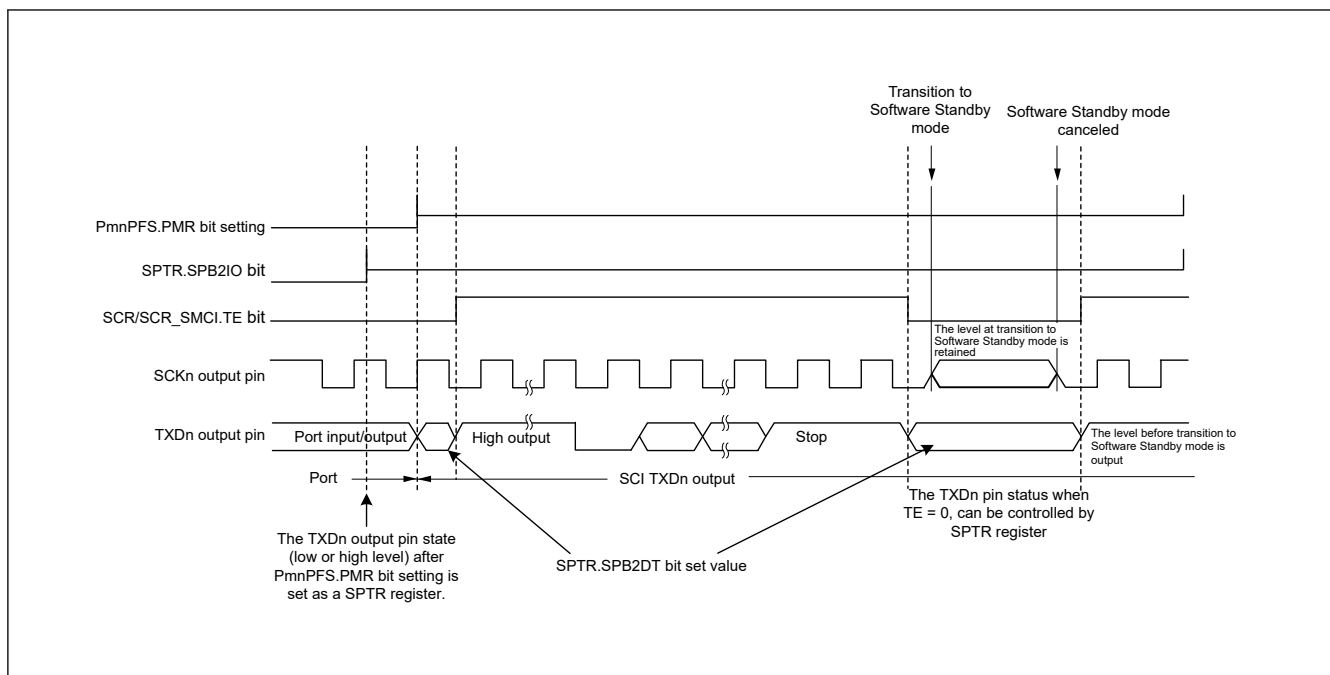


Figure 32.73 Port pin states during transition to Software Standby mode with internal clock and asynchronous transmission

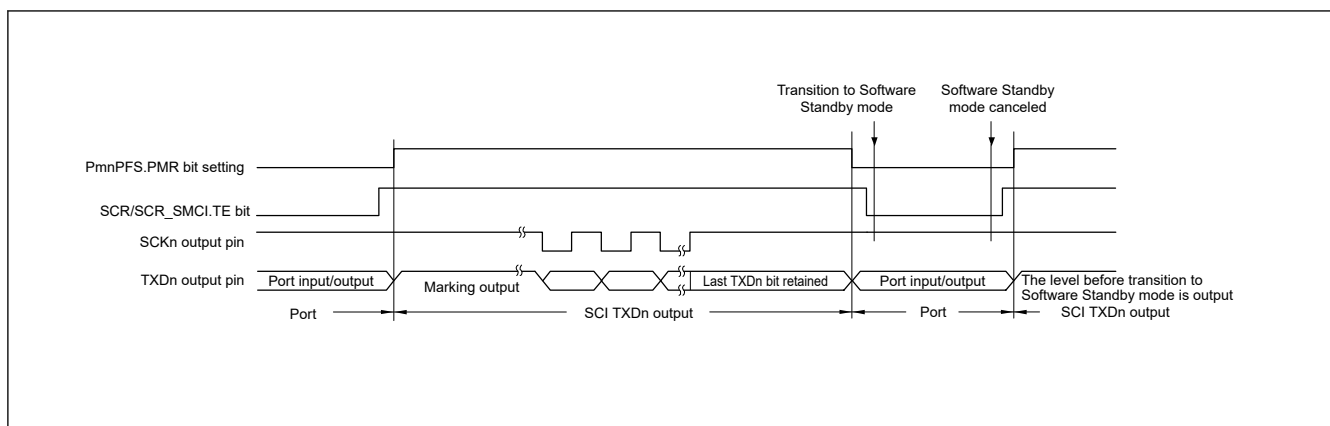
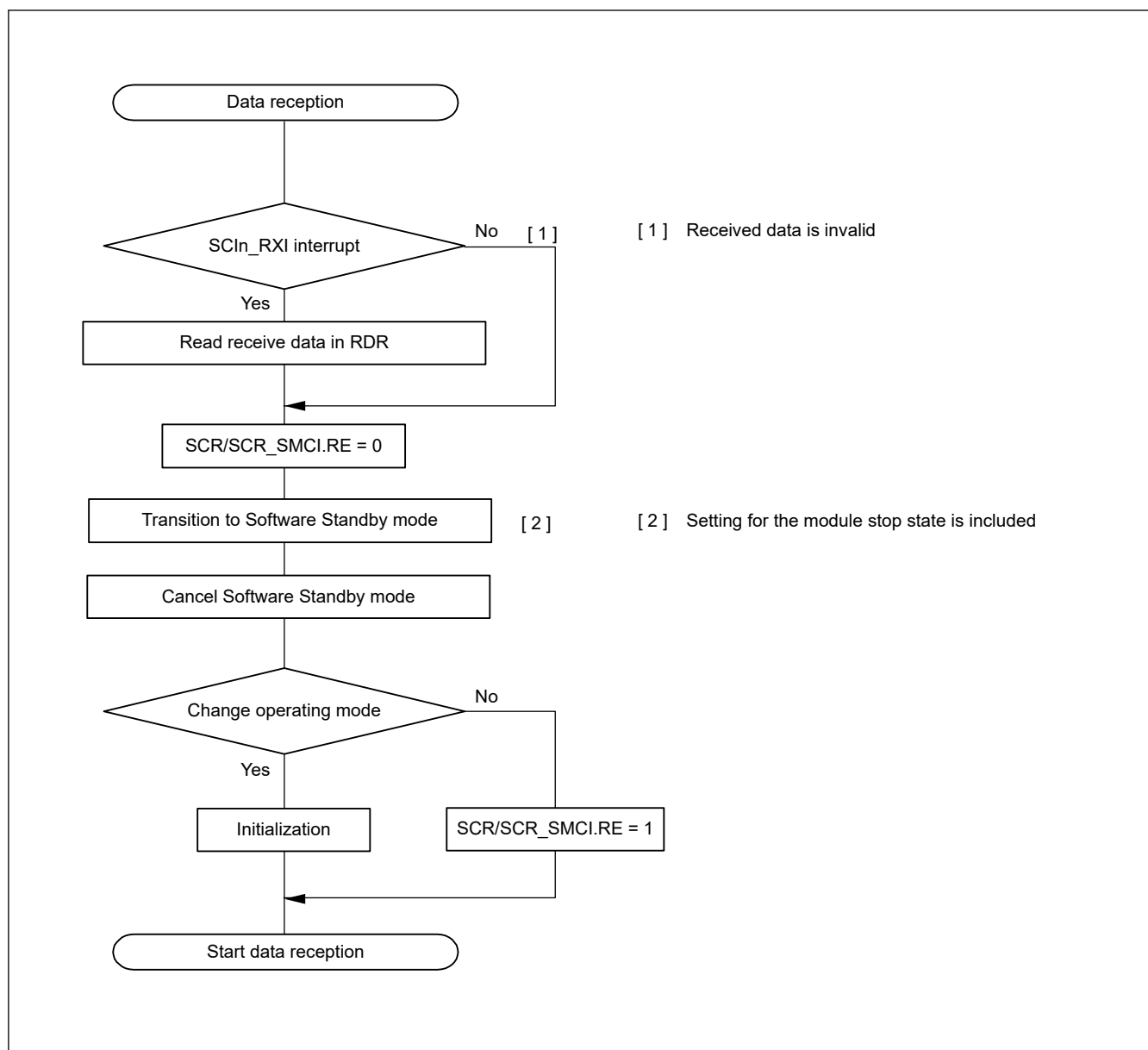


Figure 32.74 Port pin states during transition to Software Standby mode with internal clock and clock synchronous transmission

**Figure 32.75** Example flow of transition to Software Standby mode during reception

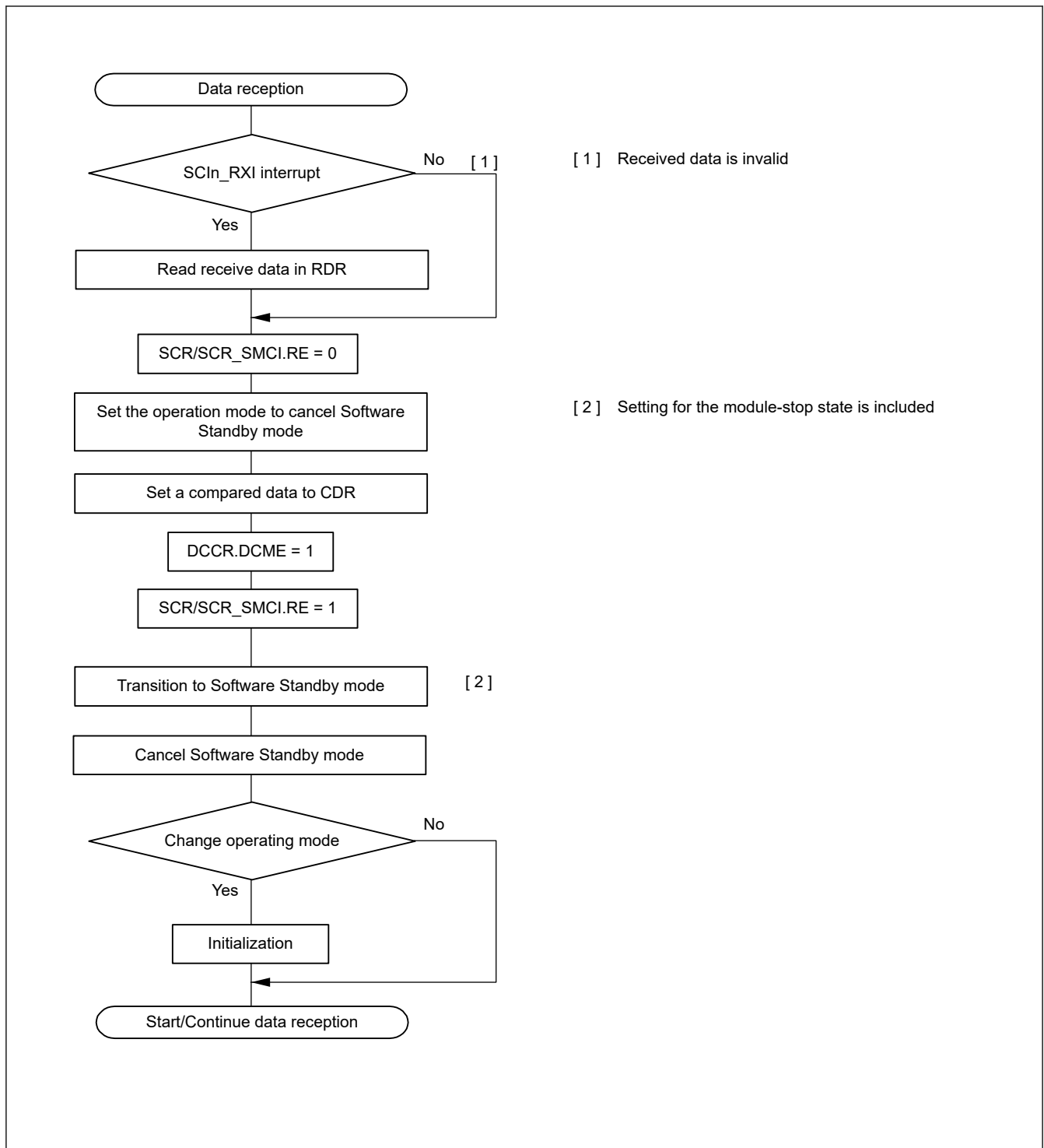


Figure 32.76 Example flow of transition to Software Standby mode during reception with address match

32.14.3 Break Detection and Processing

(1) Non-FIFO selected

When a framing error is detected, a break can be detected by reading the RXDn pin value directly. In a break, the input from the RXDn pin becomes all 0s, and the SSR.FER flag is set to 1 to indicate a framing error, and the SSR.PER flag might also be set to 1 to indicate a parity error. The SCI continues the receive operation even after a break is received. Therefore, even if the FER flag is 0, indicating that no framing error occurred, it is set to 1 again. When the SEMR.RXDESEL bit is 1, the SCI sets the SSR.FER flag to 1 and stops receiving operations until a start bit of the next data frame is detected. If the SSR.FER flag is set to 0, the SSR.FER flag retains 0 during the break.

When the RXDn pin is set to 1 and the break ends, detecting the beginning of the start bit on the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

(2) FIFO selected

After a framing error is detected and when the SCI detects that continuous receive data is 0 for 1 frame, reception stops. When a framing error is detected, a break can be detected by reading the SPTR.RXDMON flag value. After the RXD signal is in high and the break is finished, data reception to the FRDRHL register resumes.

32.14.4 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.

32.14.5 Receive Error Flags and Transmit Operation in Clock Synchronous Mode and Simple SPI Mode

Transmission cannot start when a receive error flag (ORER) in SSR/SSR_FIFO is set to 1, even when data is written to TDR or FTDRL^{*1}. Always set the receive error flags to 0 before starting transmission.

Note: The receive error flags cannot be set to 0 when the RE bit in SCR/SCR_SMCI is set to 0 (serial reception is disabled).

Note 1. Do not use the FTDRH register in simple SPI mode.

32.14.6 Restrictions on Clock Synchronous Transmission in Clock Synchronous Mode and Simple SPI Mode

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of transmission

Wait at least the following time from writing transmit data to TDR to the start of the external clock input:

1 PCLK cycle + data output delay time for the slave (t_{DO}) + setup time for the master (t_{SU}). See [Figure 32.77](#).

(2) Continuous transmission

Write the next transmit data to TDR or TDRHL before the falling edge of the transmit clock for bit [7]. See [Figure 32.77](#).

When updating TDR after bit [7] has started to transmit, update TDR while the synchronization clock is in the low-level period, and set the high-level width of the transmit clock (bit [7]) to 4 PCLK cycles or longer. See [Figure 32.77](#).

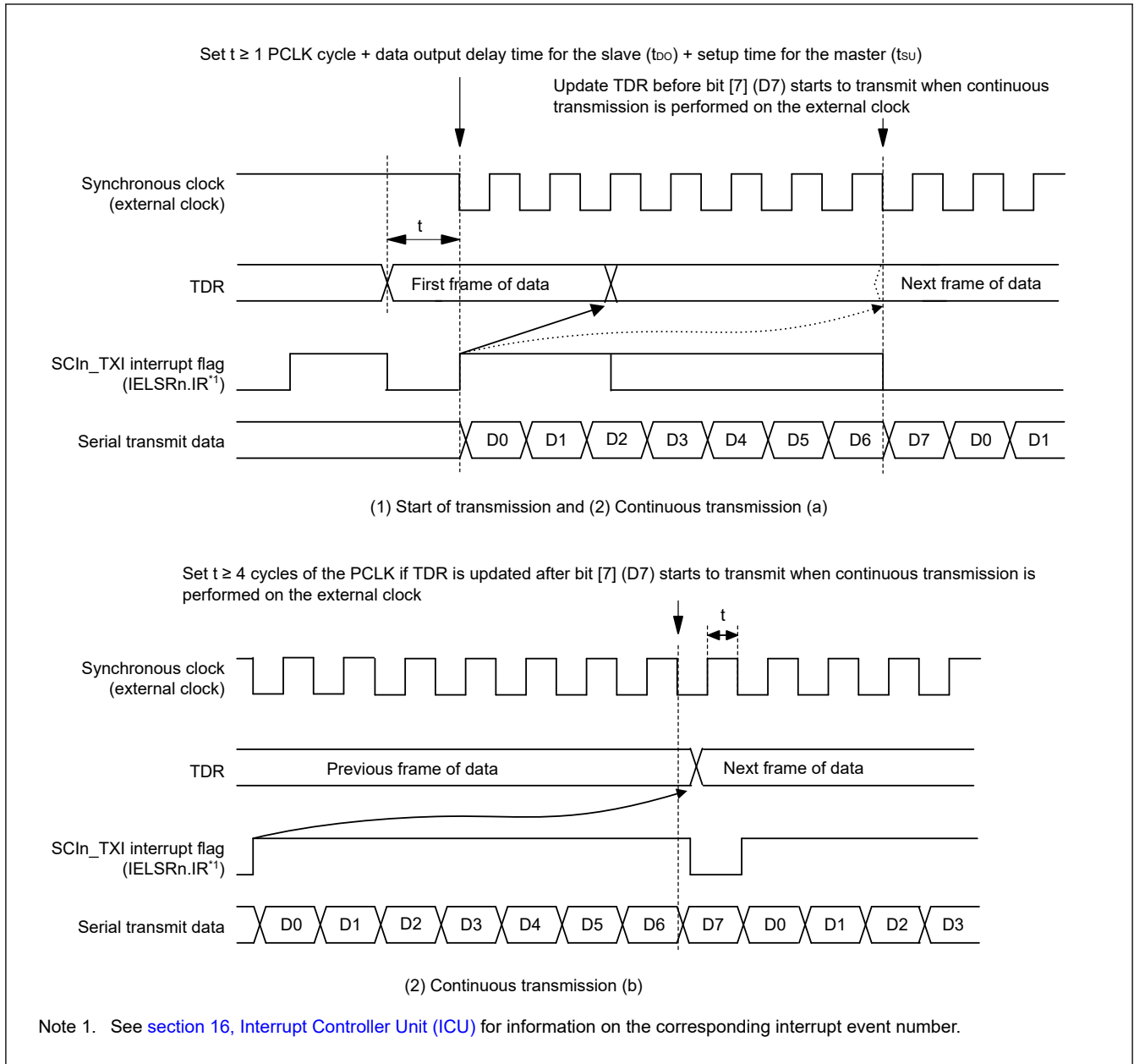


Figure 32.77 Restraints on use of external clock in clock synchronous transmission

32.14.7 Restrictions on Using DTC or DMAC

During transmission or reception operations using the DTC or DMAC, do not set transfer data for the DTC or DMAC.

(1) Writing data to TDR (FTDRHL)

Non-FIFO selected

Data can be written to TDR and TDRHL. However, if new data is written to TDR or TDRHL when transmit data remains in TDR or TDRHL, the previous data in TDR and TDRHL is lost because it was not transferred to TSR yet. When using DTC or DMAC, always write transmit data to TDR or TDRHL in the SCIn_TXI interrupt request handling routine.

FIFO selected

It is possible to write data to the FTDRH and FTDRHL 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 or DMAC to read RDR and RDRHL, always set the receive data full interrupt (SCIn_RXI) as the activation source of the relevant SCI.

32.14.8 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IELSRn.IR flag) in the ICU is 1, follow the procedure in this section to clear interrupt requests before permitting operations (by setting the SCR/SCR_SMCI.TE or SCR/SCR_SMCI.RE bit to 1). For details on the interrupt status flag, see [section 16, Interrupt Controller Unit \(ICU\)](#).

1. Confirm that transfer has stopped (the SCR/SCR_SMCI.TE or SCR/SCR_SMCI.RE bit is 0)
2. Set the associated interrupt enable bit (SCR/SCR_SMCI.TIE or SCR/SCR_SMCI.RIE) to 0
3. Read the associated interrupt enable bit (SCR/SCR_SMCI.TIE or SCR/SCR_SMCI.RIE bit) to check that it actually becomes 0
4. Set the interrupt status flag, IELSRn.IR, in the ICU to 0

32.14.9 External Clock Input in Clock Synchronous Mode and Simple SPI Mode

In clock synchronous mode and simple SPI mode, the external clock SCKn must be input as follows:

High-pulse period, low-pulse period = 2 PCLK cycles or more, period = 6 PCLK cycles or more.

32.14.10 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 32.78](#). If the TE and RE bits in the SCR register become 0 before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Additionally, an SCIn_RXI interrupt might lead to the input signal on the SSn pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master configuration, the SCKn pin output goes to high-impedance while the input on the SSn pin is at the low level if a mode fault error occurs while a character is being transferred, stopping supply of the clock signal to the connected slave. Reset the connected slave to avoid misaligned bits when transfer is restarted.

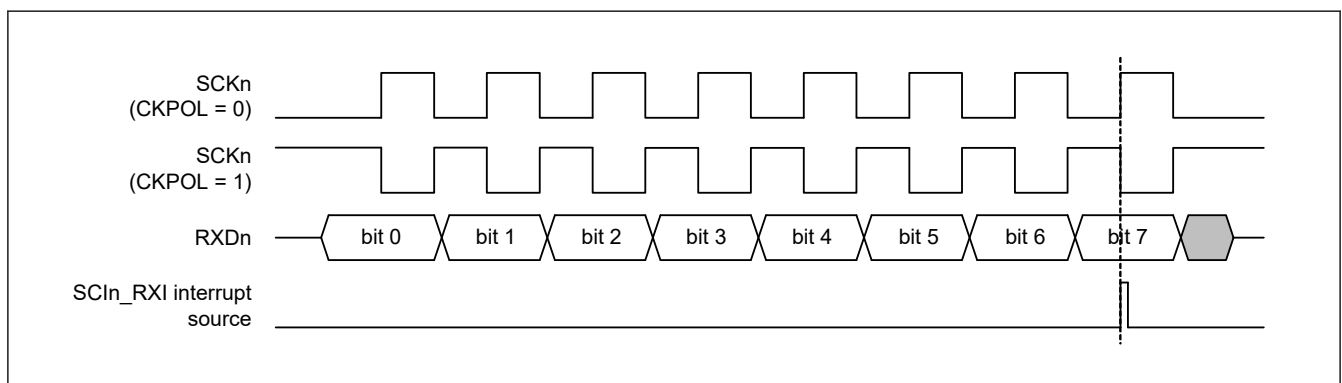


Figure 32.78 Timing of SCIn_RXI interrupt in simple SPI mode with clock delay

(2) Slave mode

- Wait at least the following time from writing transmit data in the TDR register to the start of the external clock input.
1 PCLK cycle + data output delay for the slave (t_{DO}) + setup time for the master (t_{SU})

Also wait at least 5 PCLK cycles from the input of the low level on the SSn pin to the start of the external clock input.

- Provide an external clock signal to the master the same as the data length for transfer
- Control the input on the SSn pin before the start and after the end of data transfer
- When the input level on the SSn pin is to be changed from low to high while a character is being transferred, set the TE and RE bits in the SCR register to 0 and, after restoring the settings, restart transfer of the first byte

(3) Higher communication speed

SCI3 and SCI4 can increase the simple I2C communication speed by setting SEMR.ABCS = 1.

- The setting range of SIMR1.IICDL[4: 0] is 00001b to 00100b (0 to 4 cycles).
- When SCMR.SMIF = 1, SMR.CKS[1: 0] can only be set to 00b (no PCLK division).
- BRR register can only be set to 0x00.
- The SEMR.BRME bit can only be set to 0b (bit modulation function disabled).
- The SNFR.NFCS[2: 0] bits can only be set to 001b (using 1 division).

32.14.11 Notes on Transmitt Enable bit (SCR.TE)

In initial register value, when SCR.TE bit is “0”, the terminal as “TXDn” outputs high impedance.

So please make sure that the TXDn line won't be high impedance by the following one of ways.

1. The pull-up resistance is connected to the TXDn line.
2. Before SCR.TE bit is “0”, the function of the terminal is changed to general-purpose input port or output port. And after SCR.TE bit is “1”, the function of the terminal is changed to “TXDn”.
3. In asynchronous mode, you can set SPTR and decided level of TXDn terminal during SCR.TE is “0”.

In the Simple SPI mode slave operation, the RXDn terminal operates in the same way as the above TXDn terminal, so please deal with 1 or 2 in the same way. (3 can not be used.)

32.14.12 Note on Stopping Reception When Using the RTS Function in Asynchronous Mode

One clock cycle of PCLK is required for the time from setting the SCR.RE bit to 0 to stopping the RTS signal generator in asynchronous mode.

When reading the RDR (or RDRL) register after setting the SCR.RE bit to 0, confirm that the RE bit has been set to 0 before reading the RDR (or RDRL) register to prevent these two processes from being performed consecutively.

33. IrDA Interface (IrDA)

33.1 Overview

The IrDA (Infrared Data Association) interface sends and receives IrDA data communication waveforms in association with SCI1 based on the IrDA standard 1.0.

Enabling the IrDA function by setting the IRE bit in the IRCR register allows encoding and decoding the TXD1 and RXD1 signals of SCI1 to the waveforms conforming to the IrDA standard 1.0 (IRTXD1 and IRRXD1 pins). Connecting the waveforms to an infrared transmitter/receiver implements infrared data communication that conforms to the IrDA standard 1.0 system.

With the IrDA standard 1.0 system, data transfer can be started at 9600 bps and the transfer rate can be changed whenever necessary. Because the IrDA interface cannot change the transfer rate automatically, the transfer rate should be changed through software.

Figure 33.1 shows the association between the IrDA interface and SCI1.

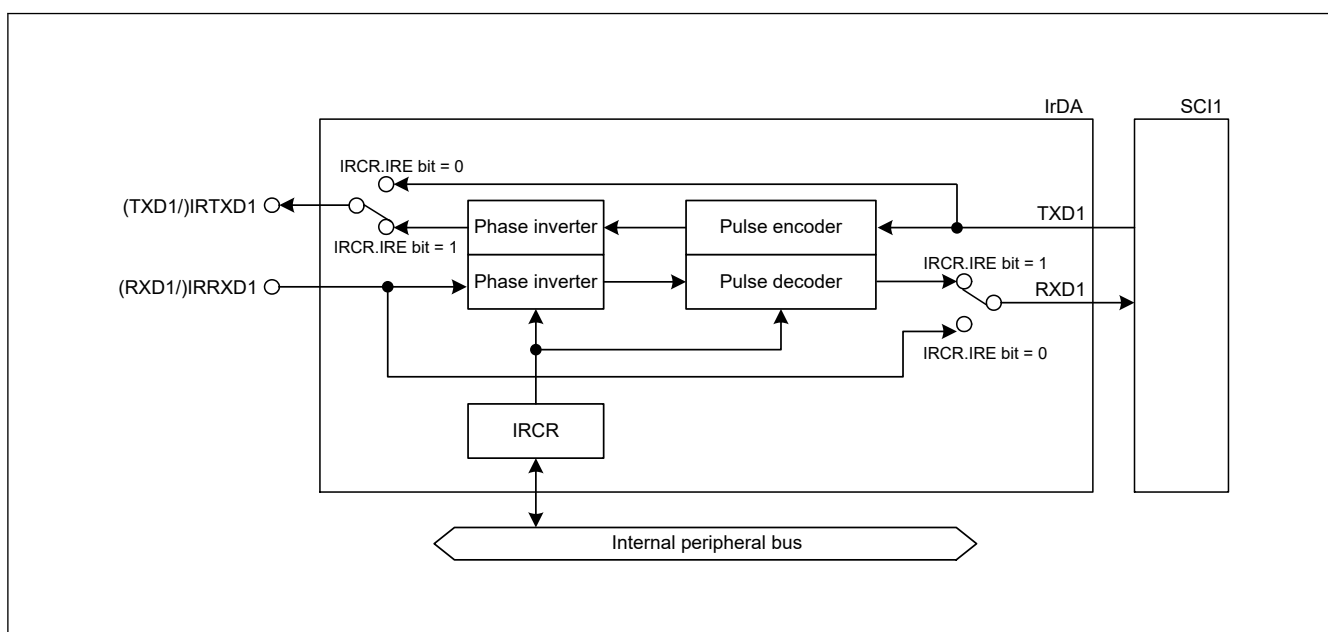


Figure 33.1 Association between the IrDA interface and SCI1

Table 33.1 IrDA I/O pins

Pin name	I/O	Description
(TXD1)/IRTXD1	Output	Transmitted data
(RXD1)/IRRXD1	Input	Received data

33.2 Register Descriptions

33.2.1 IRCR : IrDA Control Register

Base address: IRDA = 0x4007_0F00

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IRE	—	—	—	IRTXI NV	IRRXI NV	—	—

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	IRRXINV	IRRXD Polarity Switching 0: Use input from IRRXD pin as received data as is 1: Use input from IRRXD pin as received data after the polarity is inverted	R/W
3	IRTXINV	IRTXD Polarity Switching 0: Output transmitted data to IRTXD pin as is 1: Output transmitted data to IRTXD pin after the polarity is inverted	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	IRE	IrDA Enable 0: Use serial input/output pins for normal serial communication 1: Use serial input/output pins for IrDA data communication	R/W

IRRXINV bit (IRRXD Polarity Switching)

The IRRXINV bit inverts the logic level of the input from IRRXD pin.

IRTXINV bit (IRTXD Polarity Switching)

The IRTXINV bit inverts the logic level of the output to IRTXD pin.

IRE bit (IrDA Enable)

The IRE bit configures the I/O pins for normal communication mode or IrDA data communication mode.

33.3 Operation**33.3.1 IrDA Interface Setup Procedure**

Use the following procedure to set up the IrDA interface for operation:

1. Set the associated pins to IRTXD1 and IRRXD1 in the Port mn Pin Function Select Register (PmnPFS.PSEL[4:0] = 00101b) of the I/O port function.
2. Specify the peripheral function in the Port mn Pin Function Select Register (PmnPFS.PMR = 1) of the I/O port function.
3. Specify the IrDA function in the IRCR register.
4. Set the SCI1-related registers of the Serial Communications Interface (SCI).

33.3.2 Transmission

During transmission, the signals output from the SCI1 (UART frames) are converted to the IR frame data through the IrDA interface, see [Figure 33.2](#). When the IRCR.IRTXINV bit is 0 and serial data is 0, high-level pulses with 3/16 the width of the bit cycle (1-bit width period) are output (initial setting). The IrDA standard requires that the minimum high-level pulse width must be 1.41 μ s and the maximum high-level pulse width must be $(3/16 + 2.5\%) \times \text{bit cycle}$ or $(3/16 \times \text{bit cycle}) + 1.08 \mu$ s. When the serial data is 1, no pulses are output.

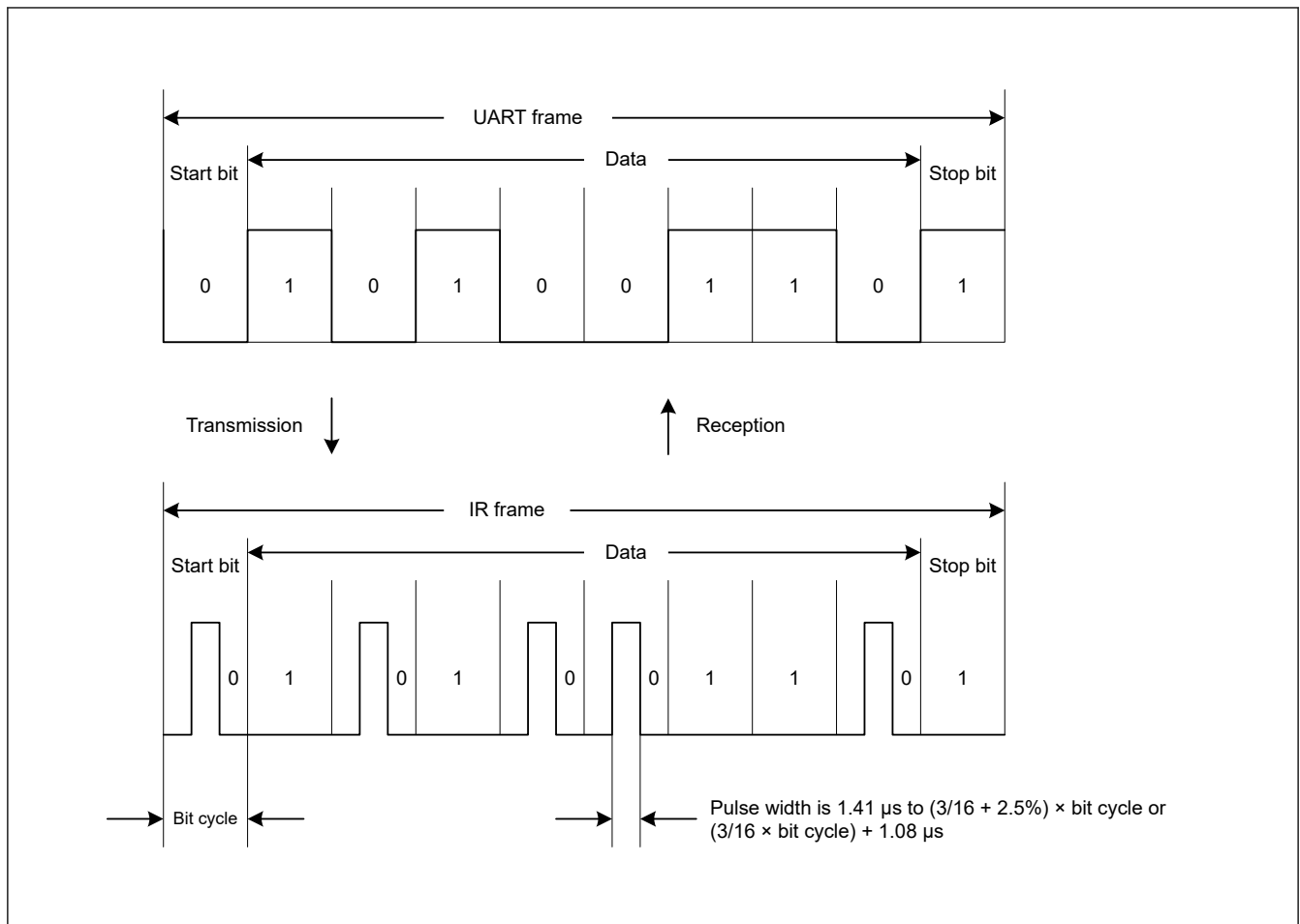


Figure 33.2 Transmission and reception through the IrDA interface

33.3.3 Reception

During reception, the IR frame data is converted to the UART frame data through the IrDA interface and is input to the SCI1. Low-level data is input to the SCI1 when the IRRXINV bit is 0 and a high-level pulse is detected. High-level data is input to the SCI1 when no pulse is detected for a 1-bit period.

33.4 Usage Notes

33.4.1 Settings for the Module-Stop Function

The Module Stop Control Register B (MSTPCRB) can enable or disable IrDA operation. The IrDA is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

33.4.2 Asynchronous Reference Clock for SCI1

The IrDA receives a clock with frequency 16 times the bit rate from the SCI1 and operates in conjunction with the SCI1. When using the IrDA, set the SCI1.SEMR.ABCS bit to 0.

34. I²C Bus Interface (IIC)

34.1 Overview

The I²C bus interface (IIC) has 2 channels. The IIC module conforms with and provides a subset of the NXP I²C (Inter-Integrated Circuit) bus interface functions.

Table 34.1 lists the IIC specifications, Figure 34.1 shows a block diagram, and Figure 34.2 shows an example of I/O pin connections to external circuits, with an I²C bus configuration. Table 34.2 lists the I/O pins.

Table 34.1 IIC specifications (1 of 2)

Parameter	Specifications
Communications format	- I²C-bus format or SMBus format - Master or slave mode selectable - Automatic securing of the setup times, hold times, and bus-free times for the transfer rate
Transfer rate	Fast-mode supported, up to 400 kbps
SCL clock	For master operation, the duty cycle of the SCL clock is selectable in the range from 4% to 96%
Issuing and detecting conditions	- Start, restart, and stop conditions are automatically generated - Start conditions (including restart conditions) and stop conditions are detectable
Slave address	- Configurable for up to three different slave addresses 7- and 10-bit address formats supported, including simultaneous use - General call addresses, device ID addresses, and SMBus host addresses detectable
Acknowledgment	- For transmission, automatic loading of the acknowledge bit Transfer of the next transmit data can be automatically suspended on detection of a not-acknowledge bit. - For reception, automatic transmission of the acknowledge bit If a wait between the 8th and 9th clock cycles is selected, the software can control the value in the acknowledge field in response to the received value.
Wait function	During reception, the following wait periods are available by holding the SCL clock low: - Waiting between the eighth and ninth clock cycles - Waiting between the ninth clock cycle and the 1st clock cycle of the next transfer
SDA output delay function	Output timing of transmitted data, including the acknowledge bit, can be delayed
Arbitration	- For multi-master operation: - SCL clock synchronization is possible when conflict occurs with the SCL signal from another master - When issuing the start condition creates conflict on the bus, loss of arbitration is detected by testing for a mismatch between the internal signal for the SDA line and the level on the SDA line - In master operation, loss of arbitration is detected by testing for non-matching between the signal on the SDA line and the internal signal for the SDA line - Loss of arbitration because the start condition occurs while the bus is busy is detectable, to prevent the issuing of double start conditions - Loss of arbitration is detectable on transfer of a not-acknowledge bit because the internal signal for the SDA line and the level on the SDA line do not match - Loss of arbitration because a mismatch of internal and line levels for data is detectable in slave transmission
Timeout function	Internal detection of long-interval stops of the SCL clock
Noise cancellation	- Digital noise filters for both the SCL and SDA signals - Programmable window for noise cancellation by the filters
Interrupt sources	- Transfer error or event occurrence (arbitration-lost, NACK, timeout, start or restart condition, or stop condition) - Receive data full, including matching with a slave address - Transmit data empty, including matching with a slave address - Transmit end
Module-stop function	Module-stop state can be set
IIC operating modes	- Master transmit - Master receive - Slave transmit - Slave receive

Table 34.1 IIC specifications (2 of 2)

Parameter	Specifications
Event link function (output)	- Transfer error or event occurrence (arbitration-lost, NACK, timeout, start or restart condition, or stop condition) - Receive data full, including matching with a slave address - Transmit data empty, including matching with a slave address - Transmit end

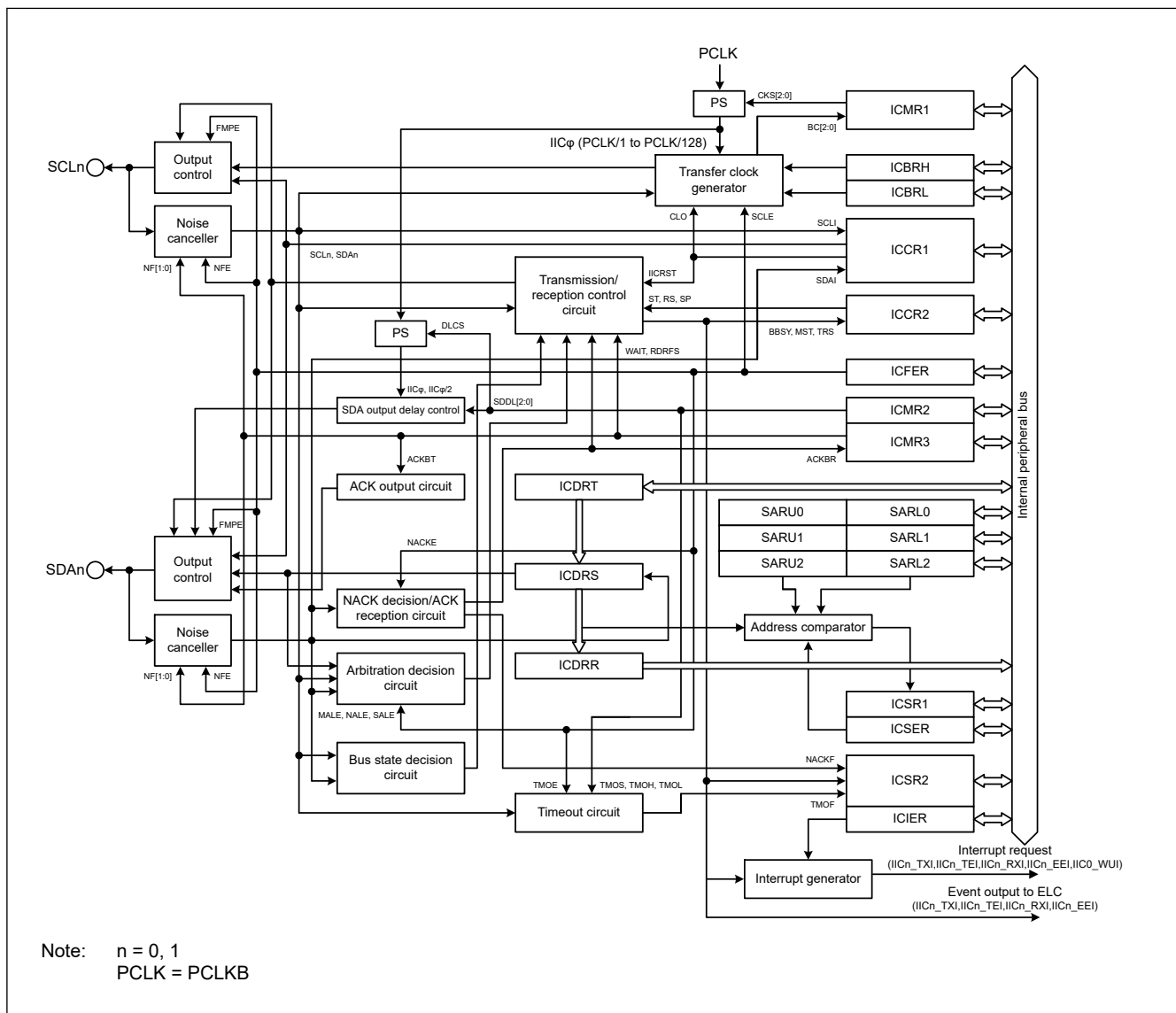


Figure 34.1 IIC block diagram

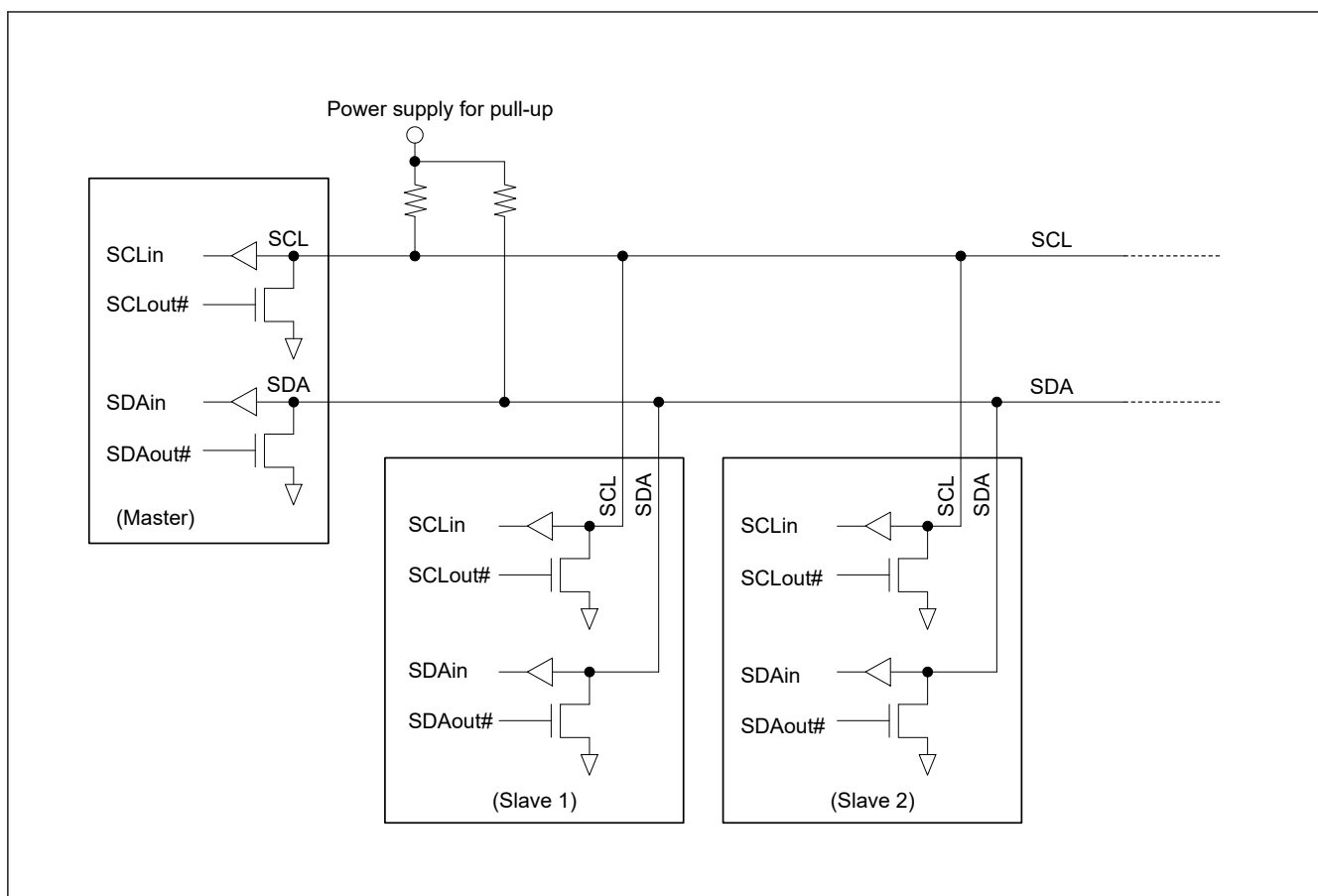


Figure 34.2 I/O pin connection to an external circuit (I²C bus configuration example)

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 34.2 IIC I/O pins

Channel	Pin name	I/O	Function
IICn	SCLn	I/O	IICn serial clock I/O pin
	SDAn	I/O	IICn serial data I/O pin

Note: n = 0, 1

34.2 Register Descriptions

34.2.1 ICCR1 : I²C Bus Control Register 1

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ICE	IICRS T	CLO	SOWP	SCLO	SDAO	SCLI	SDAI

Value after reset: 0 0 0 1 1 1 1 1

Bit	Symbol	Function	R/W
0	SDAI	SDA Line Monitor 0: SDAn line is low 1: SDAn line is high	R

Bit	Symbol	Function	R/W
1	SCLI	SCL Line Monitor 0: SCLn line is low 1: SCLn line is high	R
2	SDAO	SDA Output Control/Monitor 0: Read: IIC drives SDAn pin low Write: IIC drives SDAn pin low 1: Read: IIC releases SDAn pin Write: IIC releases SDAn pin	R/W
3	SCLO	SCL Output Control/Monitor Use an external pull-up resistor to drive the signal high. 0: Read: IIC drives SCLn pin low Write: IIC drives SCLn pin low 1: Read: IIC releases SCLn pin Write: IIC releases SCLn pin	R/W
4	SOWP	SCLO/SDAO Write Protect This bit is read as 1. 0: Write enable SCLO and SDAO bits 1: Write protect SCLO and SDAO bits	W
5	CLO	Extra SCL Clock Cycle Output This bit clears automatically after 1 clock cycle is output. 0: Do not output extra SCL clock cycle (default) 1: Output extra SCL clock cycle	R/W
6	IICRST	I ² C Bus Interface Internal Reset This setting clears the bit counter and the SCLn/SDAn output latch. 0: Release IIC reset or internal reset 1: Initiate IIC reset or internal reset	R/W
7	ICE	I ² C Bus Interface Enable Used in combination with the IICRST bit to select either IIC or internal reset. 0: Disable (SCLn and SDAn pins in inactive state) 1: Enable (SCLn and SDAn pins in active state)	R/W

SDAO bit (SDA Output Control/Monitor) and SCLO bit (SCL Output Control/Monitor)

The SDAO and SCLO bits directly control the 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, transmission, or reception. Operation after rewriting under these conditions is not guaranteed. When reading these bits, the state of signals output from the IIC can be read.

CLO bit (Extra SCL Clock Cycle Output)

The CLO bit allows output of an extra SCL clock cycle for debugging or error processing. Normally, set this bit to 0. Setting the bit to 1 in a normal communication state causes a communication error. For details on this function, see [section 34.11.2. Extra SCL Clock Cycle Output Function](#).

IICRST bit (I²C Bus Interface Internal Reset)

The IICRST bit initiates an internal state reset of the IIC. Setting this bit to 1 initiates an IIC reset or internal reset. Whether an IIC reset or internal reset is initiated is determined by the settings of this bit in combination with the ICE bit. [Table 34.3](#) lists the IIC resets.

The IIC reset initializes all registers except ICCR1.ICE and ICCR1.IICRST bits and internal states of the IIC. In addition to the internal states of the IIC, the internal reset initializes the following:

- Bit counter (ICMR1.BC[2:0] bits)
- I²C Bus Shift Register (ICDRS)
- I²C Bus Status Registers (ICSR1 and ICSR2)
- SDAO and SCLO Output Control/Monitor (ICCR1.SDAO and ICCR1.SCLO bits)

- I²C Bus Control Register 2 (except ICCR2.BBSY bit)

For the reset conditions for each register, see [section 34.14. State of Registers When Issuing Each Condition](#).

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the IIC without initializing the port settings and the control and setting registers of the IIC. If the IIC hangs in a low-level output state, resetting the internal states cancels the low-level output state and releases the bus with the 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 34.3 IIC resets

IICRST	ICE	State	Specifications
1	0	IIC reset	Resets all registers except ICCR1.IICRST and ICCR1.ICE bits, and the internal states of the IIC
	1	Internal reset	Reset the following: • ICMR1.BC[2:0] bits • ICSR1, ICSR2, ICDRS registers • SDAO and SCLO Output Control/Monitor (ICCR1.SDAO and ICCR1.SCLO bits) • I²C Bus Control Register 2 (except ICCR2.BBSY bit) • Internal states of the IIC

ICE bit (I²C Bus Interface Enable)

The ICE bit selects the active or inactive state of the SCLn and SDAn pins. It can also be combined with the IICRST bit to initiate two types of resets. See [Table 34.3](#) for the reset descriptions.

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.

34.2.2 ICCR2 : I²C Bus Control Register 2

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	ST	Start Condition Issuance Request 0: Do not issue a start condition request 1: Issue a start condition request	R/W
2	RS	Restart Condition Issuance Request 0: Do not issue a restart condition request 1: Issue a restart condition request	R/W

Bit	Symbol	Function	R/W
3	SP	Stop Condition Issuance Request 0: Do not issue a stop condition request 1: Issue a stop condition request	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	TRS	Transmit/Receive Mode 0: Receive mode 1: Transmit mode	R/W ¹
6	MST	Master/Slave Mode 0: Slave mode 1: Master mode	R/W ¹
7	BBSY	Bus Busy Detection Flag 0: I ² C bus released (bus free state) 1: I ² C bus occupied (bus busy state)	R

Note 1. The MST and TRS bits can be written to when the ICMR1.MTWP bit is set to 1.

ST bit (Start Condition Issuance Request)

The ST bit requests transition to master mode and triggers a start condition. When this bit is set to 1, a start condition is issued when the BBSY flag is set to 0 (bus free state). For details on this function, see [section 34.10. Start, Restart, and Stop Condition Issuing Function](#).

[Setting condition]

- When 1 is written to the ST bit.

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition is issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Only set the ST bit to 1 (start condition request) when the BBSY flag is set to 0 (bus free state). Arbitration might be lost if the ST bit is set to 1 (start condition request) when the BBSY flag is 1 (bus busy state).

RS bit (Restart Condition Issuance Request)

The RS bit requests that a restart condition be issued in master mode. When this bit is set to 1 to request a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on this function, see [section 34.10. Start, Restart, and Stop Condition Issuing Function](#).

[Setting condition]

- When 1 is written to the RS bit with the BBSY flag in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition is issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Do not set the RS bit to 1 while issuing a stop condition.

Note: If 1 (restart condition request) is written to the RS bit in slave mode, the restart condition is not issued, but the RS bit remains set to 1. If the operating mode changes to master mode without the bit being cleared, a restart condition might be issued.

SP bit (Stop Condition Issuance Request)

The SP bit requests that a stop condition be issued in master mode. When this bit is set to 1, a stop condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on this function, see [section 34.10. Start, Restart, and Stop Condition Issuing Function](#).

[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the MST bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition is issued (a stop condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Writing to the SP bit is not possible while the BBSY flag is 0 (bus free state).

Note: Do not set the SP bit to 1 while a restart condition is being issued.

TRS bit (Transmit/Receive Mode)

The TRS bit indicates transmit or receive mode. The IIC is in receive mode when the TRS bit is 0 and in transmit mode when the bit is 1. The combination of this bit and the MST bit indicates the IIC operating mode.

The value of the TRS bit automatically changes to 1 for transmit mode or 0 for receive mode when a start condition is issued or detected and the R/W# bit is set. Although writing to the TRS bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not required during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1)
- When a restart condition is issued normally because of a restart condition request (when a restart condition is detected with the RS bit set to 1)
- When the R/W# bit appended to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in ICSE, 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 ICSE when the value of the received R/W# bit is 0, including when the received address is the general call address
- In slave mode, when a restart condition is detected (a start condition is detected with ICCR2.BBSY = 1 and ICCR2.MST = 0)
- When 0 is written to the TRS bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

MST bit (Master/Slave Mode)

The MST bit indicates master or slave mode. The IIC is in slave mode when the MST bit is 0 and is in master mode when the bit is 1. The combination of this bit and the TRS bit indicates the IIC operating mode.

The value of the MST bit automatically changes to 1 for master mode or 0 for slave mode when a start condition is issued or a stop condition is issued or detected. Although writing to the MST bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not required during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 0 is written to the MST bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

BBSY flag (Bus Busy Detection Flag)

The BBSY flag indicates whether the I²C bus is occupied (bus busy state) or released (bus free state). The flag is set to 1 when the SDAn line changes from high to low when the SCLn line is high, assuming that a start condition was issued. The flag is set to 0 if a start condition is not detected for the bus free time (ICBRL setting), assuming that a stop condition was issued.

[Setting condition]

- When a start condition is detected.

[Clearing conditions]

- When a start condition is not detected for the bus free time (ICBRL setting) after detecting a stop condition
- When 1 is written to the IICRST bit in ICCR1 with the ICE bit in ICCR1 set to 0 (IIC reset).

34.2.3 ICMR1 : I²C Bus Mode Register 1

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MTWP	CKS[2:0]			BCWP	BC[2:0]		

Value after reset: 0 0 0 0 1 0 0 0

Bit	Symbol	Function	R/W
2:0	BC[2:0]	Bit Counter 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits	R/W ^{*1}
3	BCWP	BC Write Protect This bit is read as 1. 0: Write enable BC[2:0] bits 1: Write protect BC[2:0] bits	W ^{*1}
6:4	CKS[2:0]	Internal Reference Clock Select Select the internal reference clock source (IICφ) for the IIC. IICφ = (PCLKB / 2 ^{CKS[2:0]}) clock	R/W
7	MTWP	MST/TRS Write Protect 0: Write protect MST and TRS bits in ICCR2 1: Write enable MST and TRS bits in ICCR2	R/W

Note 1. Rewrite the BC[2:0] bits and set the BCWP bit to 0 at the same time.

BC[2:0] bits (Bit Counter)

The BC[2:0] bits function as a counter indicating the number of bits remaining to be transferred on detection of a rising edge on the SCLn line. Although BC[2:0] are read/write bits, it is not required to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one, for an additional acknowledge bit, between transferred frames when the 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.

34.2.4 ICMR2 : I²C Bus Mode Register 2

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
0	TMOS	Timeout Detection Time Select 0: Select long mode 1: Select short mode	R/W
1	TMOL	Timeout L Count Control 0: Disable count while SCLn line is low 1: Enable count while SCLn line is low	R/W
2	TMOH	Timeout H Count Control 0: Disable count while SCLn line is high 1: Enable count while SCLn line is high	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	SDDL[2:0]	SDA Output Delay Counter 0 0 0: No output delay 0 0 1: 1 IICφ cycle (When ICMR2.DLCS = 0 (IICφ)) 1 or 2 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 0 1 0: 2 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 3 or 4 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 0 1 1: 3 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 5 or 6 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 0 0: 4 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 7 or 8 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 0 1: 5 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 9 or 10 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 1 0: 6 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 11 or 12 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 1 1: 7 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 13 or 14 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2))	R/W
7	DLCS	SDA Output Delay Clock Source Select 0: Select internal reference clock (IICφ) as the clock source for SDA output delay counter 1: Select internal reference clock divided by 2 (IICφ/2) as the clock source for SDA output delay counter*1	R/W

Note 1. The setting DLCS = 1 (IICφ/2) is only valid when SCL is low. When SCL is high, the DLCS = 1 setting becomes invalid and the clock source becomes the internal reference clock (IICφ).

TMOS bit (Timeout Detection Time Select)

The TMOS bit selects long or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE = 1). When this bit is set to 0, long mode is selected. When it is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14-bit counter. While the SCLn line is in the state that enables this counter as specified in the TMOH and TMOL bits, the counter

counts up in synchronization with the internal reference clock (IIC ϕ) as a count source. For details on this function, see [section 34.11.1. Timeout Function](#).

TMOL bit (Timeout L Count Control)

The TMOL bit enables or disables up-counting on the internal counter of the timeout function while the SCLn line is held low and the timeout function is enabled (ICFER.TMOE = 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. This setting can be used for all types of SDA output, including transmission of the acknowledge bit.

Set the SDA output delay to meet the I²C bus standard for the data enable time/acknowledge enable time,^{*1} or the SMBus standard, within [data hold time (300 ns or more + the SCL clock low-level period) - the data setup time (250 ns)]. If a value outside the standard is set, communication between devices might malfunction or falsely indicate a start or stop condition, depending on the bus state.

For details on this function, see [section 34.5. 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)

34.2.5 ICMR3 : I²C Bus Mode Register 3

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SMBS	WAIT	RDRF S	ACKW P	ACKB T	ACKB R	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	NF[1:0]	Noise Filter Stage Select 0 0: Filter out noise of up to 1 IIC ϕ cycle (single-stage filter) 0 1: Filter out noise of up to 2 IIC ϕ cycles (2-stage filter) 1 0: Filter out noise of up to 3 IIC ϕ cycles (3-stage filter) 1 1: Filter out noise of up to 4 IIC ϕ cycles (4-stage filter)	R/W
2	ACKBR	Receive Acknowledge 0: 0 received as the acknowledge bit (ACK reception) 1: 1 received as the acknowledge bit (NACK reception)	R
3	ACKBT	Transmit Acknowledge 0: Send 0 as the acknowledge bit (ACK transmission) 1: Send 1 as the acknowledge bit (NACK transmission)	R/W ^{*1}
4	ACKWP	ACKBT Write Protect 0: Write protect ACKBT bit 1: Write enable ACKBT bit	R/W
5	RDRFS	RDRF Flag Set Timing Select Low-hold is released by writing to ACKBT. 0: Set the RDRF flag on the rising edge of the 9th SCL clock cycle. The SCLn line is not held low on the falling edge of the 8th clock cycle. 1: Set the RDRF flag on the rising edge of the 8th SCL clock cycle. The SCLn line is held low on the falling edge of the 8th clock cycle.	R/W ^{*2}

Bit	Symbol	Function	R/W
6	WAIT	Low-hold is released by reading ICDRR. 0: No wait (The SCLn line is not held low during the period between the 9th clock cycle and the 1st clock cycle.) 1: Wait (The SCLn line is held low during the period between the 9th clock cycle and the 1st clock cycle.)	R/W ²
7	SMBS	SMBus/I ² C Bus Select 0: Select I ² C Bus 1: Select SMBus	R/W

Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If the application writes 1 to the ACKWP and ACKBT bits at the same time, the ACKBT bit is not set to 1.

Note 2. The WAIT and RDRFS bits are only valid in receive mode (invalid in transmit mode).

NF[1:0] bits (Noise Filter Stage Select)

The NF[1:0] bits select the number of stages in the digital noise filter. For details on this function, see [section 34.6. Digital Noise Filter Circuits](#)

Note: Set the noise range to be filtered within a range less than the SCLn line high- or low-level period. If the noise range is set to a value of [SCL clock width: high- or low-level period, whichever is shorter] - [1.5 internal reference clock (IIC_φ) cycles + analog noise filter: 120 ns (reference values)] or more, the SCL clock is regarded as noise, which might prevent the IIC from operating normally.

ACKBR bit (Receive Acknowledge)

The ACKBR bit stores the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

- When 1 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

ACKBT bit (Transmit Acknowledge)

The ACKBT bit sets the acknowledge bit to be sent in receive mode

[Setting condition]

- When 1 is written to this bit with the ACKWP bit set to 1.

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1
- When stop condition request is detected with the SP bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

ACKWP bit (ACKBT Write Protect)

The ACKWP bit controls write enabling of the ACKBT bit.

RDRFS bit (RDRF Flag Set Timing Select)

The RDRFS bit selects the RDRF flag set timing in receive mode and also selects whether to hold the 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 force a low-hold between the ninth SCL clock cycle and the first SCL clock cycle, until the receive data buffer (ICDRR) is completely read each time single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation continues without a low-hold between the ninth and the first SCL clock cycle. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCLn line is held low from the falling edge of the ninth clock cycle until the ICDRR value is read each time single-byte data is received. This enables receive operation in byte units.

Note: When the value of the WAIT bit is to be read, always read ICDRR first.

SMBS bit (SMBus/I²C Bus Select)

Setting the SMBS bit to 1 selects the SMBus and enables the HOAE bit in ICSESR.

34.2.6 ICFER : I²C Bus Function Enable Register

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x05

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SCLE	NFE	NACK E	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Function	R/W
0	TMOE	Timeout Function Enable 0: Disable 1: Enable	R/W
1	MALE	Master Arbitration-Lost Detection Enable 0: Disable the arbitration-lost detection function and disable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost 1: Enable the arbitration-lost detection function and enable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost	R/W
2	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: Disable 1: Enable	R/W
3	SALE	Slave Arbitration-Lost Detection Enable 0: Disable 1: Enable	R/W
4	NACKE	NACK Reception Transfer Suspension Enable 0: Do not suspend transfer operation during NACK reception (disable transfer suspension) 1: Suspend transfer operation during NACK reception (enable transfer suspension)	R/W
5	NFE	Digital Noise Filter Circuit Enable 0: Do not use the digital noise filter circuit 1: Use the digital noise filter circuit	R/W
6	SCLE	SCL Synchronous Circuit Enable 0: Do not use the SCL synchronous circuit 1: Use the SCL synchronous circuit	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

TMOE bit (Timeout Function Enable)

The TMOE bit enables or disables the timeout function. For details on this function, see [section 34.11.1. Timeout Function](#).

MALE bit (Master Arbitration-Lost Detection Enable)

The MALE bit specifies whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE bit (NACK Transmission Arbitration-Lost Detection Enable)

The NALE bit specifies whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode, for example when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with a different number of receive bytes.

SALE bit (Slave Arbitration-Lost Detection Enable)

The SALE bit specifies whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode, for example when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs because of noise.

NACKE bit (NACK Reception Transfer Suspension Enable)

The NACKE bit specifies whether to continue or discontinue the transfer operation when NACK is received from the slave device in transmit mode. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended. When the NACKE bit is 0, the next transfer operation continues regardless of the received acknowledge content.

For details, see [section 34.8.2. NACK Reception Transfer Suspension Function](#).

SCLE bit (SCL Synchronous Circuit Enable)

The SCLE bit specifies whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.

When the SCLE bit is set to 0 (no SCL synchronous circuit used), the IIC does not synchronize the SCL clock with the SCL input clock. With this setting, the IIC outputs the SCL clock at the transfer rate set in ICBRH and ICBRL, regardless of the SCLn line state. For this reason, if the bus load of the I²C bus line is much larger than the specification value, or if the SCL clock output overlaps in multiple masters, a short-cycle SCL clock that does not meet the specification might be output. When no SCL synchronous circuit is used, it also affects the issuing of the start, restart, and stop conditions, and the continuous output of extra SCL clock cycles.

Do not set this bit to 0 except when checking the output of the set transfer rate.

34.2.7 IC SER : I²C Bus Status Enable Register

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HOAE	—	DIDE	—	GCAE	SAR2 E	SAR1 E	SAR0 E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Function	R/W
0	SAR0E	Slave Address Register 0 Enable 0: Disable slave address in SARL0 and SARU0 1: Enable slave address in SARL0 and SARU0	R/W
1	SAR1E	Slave Address Register 1 Enable 0: Disable slave address in SARL1 and SARU1 1: Enable slave address in SARL1 and SARU1	R/W
2	SAR2E	Slave Address Register 2 Enable 0: Disable slave address in SARL2 and SARU2 1: Enable slave address in SARL2 and SARU2	R/W
3	GCAE	General Call Address Enable 0: Disable general call address detection 1: Enable general call address detection	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
5	DIDE	Device-ID Address Detection Enable 0: Disable device-ID address detection 1: Enable device-ID address detection	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	HOAE	Host Address Enable 0: Disable host address detection 1: Enable host address detection	R/W

SARyE bit (Slave Address Register y Enable) (y = 0 to 2)

The SARyE bit enables or disables the received slave address and the slave address set in SARLy and SARUy.

When this bit is set to 1, the slave address set in SARLy and SARUy is enabled and is compared with the received slave address. When this bit is set to 0, the slave address set in SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

GCAE bit (General Call Address Enable)

The GCAE bit specifies whether to ignore the general call address (0000 000b + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the IIC recognizes the received slave address as the general call address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs the data receive operation. When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

DIDE bit (Device-ID Address Detection Enable)

The DIDE bit specifies whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first frame after a start or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the IIC recognizes that the device-ID address was received. When the next R/W# bit is 0 (W), the IIC recognizes the second and the subsequent frames as slave addresses and continues the receive operation. When this bit is set to 0, the IIC ignores the received first frame even if it matches the device-ID address, and it recognizes the first frame as a normal slave address.

For details on this function, see [section 34.7.3. 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.

34.2.8 ICIER : I²C Bus Interrupt Enable Register

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x07

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMOIE	Timeout Interrupt Request Enable 0: Disable timeout interrupt (TMOI) request 1: Enable timeout interrupt (TMOI) request	R/W

Bit	Symbol	Function	R/W
1	ALIE	Arbitration-Lost Interrupt Request Enable 0: Disable arbitration-lost interrupt (ALI) request 1: Enable arbitration-lost interrupt (ALI) request	R/W
2	STIE	Start Condition Detection Interrupt Request Enable 0: Disable start condition detection interrupt (STI) request 1: Enable start condition detection interrupt (STI) request	R/W
3	SPIE	Stop Condition Detection Interrupt Request Enable 0: Disable stop condition detection interrupt (SPI) request 1: Enable stop condition detection interrupt (SPI) request	R/W
4	NAKIE	NACK Reception Interrupt Request Enable 0: Disable NACK reception interrupt (NAKI) request 1: Enable NACK reception interrupt (NAKI) request	R/W
5	RIE	Receive Data Full Interrupt Request Enable 0: Disable receive data full interrupt (IICn_RXI) request 1: Enable receive data full interrupt (IICn_RXI) request	R/W
6	TEIE	Transmit End Interrupt Request Enable 0: Disable transmit end interrupt (IICn_TEI) request 1: Enable transmit end interrupt (IICn_TEI) request	R/W
7	TIE	Transmit Data Empty Interrupt Request Enable 0: Disable transmit data empty interrupt (IICn_TXI) request 1: Enable transmit data empty interrupt (IICn_TXI) request	R/W

TMOIE bit (Timeout Interrupt Request Enable)

The TMOIE bit enables or disables timeout interrupt (TMOI) requests when the TMOF flag in ICSR2 is 1. To cancel a TMOI interrupt request, set the TMOF flag or the TMOIE bit to 0.

ALIE bit (Arbitration-Lost Interrupt Request Enable)

The ALIE bit enables or disables arbitration-lost interrupt (ALI) requests when the AL flag in ICSR2 is 1. To cancel an ALI interrupt request, set the AL flag or the ALIE bit to 0.

STIE bit (Start Condition Detection Interrupt Request Enable)

The STIE bit enables or disables start condition detection interrupt (STI) requests when the START flag in ICSR2 is 1. To cancel an STI interrupt request, set the START flag or the STIE bit to 0.

SPIE bit (Stop Condition Detection Interrupt Request Enable)

The SPIE bit enables or disables stop condition detection interrupt (SPI) requests when the STOP flag in ICSR2 is 1. To cancel an SPI interrupt request, set the STOP flag or the SPIE bit to 0.

NAKIE bit (NACK Reception Interrupt Request Enable)

The NAKIE bit enables or disables NACK reception interrupt (NAKI) requests when the NACKF flag in ICSR2 is 1. To cancel an NAKI interrupt request, set the NACKF flag or the NAKIE bit to 0.

RIE bit (Receive Data Full Interrupt Request Enable)

The RIE bit enables or disables receive data full interrupt (IICn_RXI) requests when the RDRF flag in ICSR2 is 1.

TEIE bit (Transmit End Interrupt Request Enable)

The TEIE bit enables or disables transmit end interrupt (IICn_TEI) requests when the TEND flag in ICSR2 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 ICSR2 is 1.

34.2.9 ICSR1 : I²C Bus Status Register 1

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AAS0	Slave Address 0 Detection Flag 0: Slave address 0 not detected 1: Slave address 0 detected	R/(W) ^{*1}
1	AAS1	Slave Address 1 Detection Flag 0: Slave address 1 not detected 1: Slave address 1 detected	R/(W) ^{*1}
2	AAS2	Slave Address 2 Detection Flag 0: Slave address 2 not detected 1: Slave address 2 detected	R/(W) ^{*1}
3	GCA	General Call Address Detection Flag 0: General call address not detected 1: General call address detected	R/(W) ^{*1}
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DID	Device-ID Address Detection Flag This bit is set to 1 when the first frame received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0[W]). 0: Device-ID command not detected 1: Device-ID command detected	R/(W) ^{*1}
6	—	This bit is read as 0. The write value should be 0.	R/W
7	HOA	Host Address Detection Flag This bit is set to 1 when the received slave address matches the host address (0001 000b). 0: Host address not detected 1: Host address detected	R/(W) ^{*1}

Note 1. Only 0 can be written to clear the flag.

AASy flag (Slave Address y Detection flag) (y = 0 to 2)

The AASy flag indicates whether slave address y was detected.

[Setting conditions]

For 7-bit address format (SARUy.FS = 0):

- When the received slave address matches the SVA[6:0] value in SARLy, with the SARyE bit in ICSEr set to 1 (slave address y detection enabled).
The AASy flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format (SARUy.FS = 1):

- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy), and the subsequent address matches the SARLy value, with the SARyE bit in ICSEr set to 1 (slave address y detection enabled).
The AASy flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy flag after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

For 7-bit address format (SARUy.FS = 0):

- When the received slave address does not match the SVA[6:0] value in SARLy, with the SARyE bit in ICSEr set to 1 (slave address y detection enabled).
The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format (SARUy.FS = 1):

- When the received slave address does not match a value of (11110b + SVA[1:0] in SARUy), with the SARyE bit in ICSEr set to 1 (slave address y detection enabled).
The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy), and the subsequent address does not match the SARLy value, with the SARyE bit in ICSEr set to 1 (slave address y detection enabled).
The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

GCA flag (General Call Address Detection Flag)

The GCA flag indicates whether the general call address was detected.

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSEr set to 1 (general call address detection enabled).
The GCA flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA flag after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSEr set to 1 (general call address detection enabled).
The GCA flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

DID flag (Device-ID Address Detection Flag)

The DID flag indicates whether the device-ID address was detected.

[Setting condition]

- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), with the DIDE bit in ICSEr set to 1 (device-ID address detection enabled).
The DID flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID flag after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start or restart condition is detected does not match a value of (device ID (1111 100b)), with the DIDE bit in ICSEr set to 1 (device-ID address detection enabled).
The DID flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), and the second frame does not match any slave address from 0 to 2, with the DIDE bit in ICSEr set to 1 (device-ID address detection enabled).
The DID flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

HOA flag (Host Address Detection Flag)

The HOA flag indicates whether the host address was detected.

[Setting condition]

- When the received slave address matches the host address (0001 000b), with the HOAE bit in IC SER set to 1 (host address detection enabled).

The HOA flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA flag after reading HOA = 1
- When a stop condition is detected
- When the received slave address does not match the host address (0001 000b), with the HOAE bit in IC SER set to 1 (host address detection enabled)
The HOA flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

34.2.10 ICSR2 : I²C Bus Status Register 2

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x09

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TDRE	TEND	RDRF	NACK F	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMOF	Timeout Detection Flag 0: Timeout not detected 1: Timeout detected	R/(W) ^{*1}
1	AL	Arbitration-Lost Flag 0: Arbitration not lost 1: Arbitration lost	R/(W) ^{*1}
2	START	Start Condition Detection Flag 0: Start condition not detected 1: Start condition detected	R/(W) ^{*1}
3	STOP	Stop Condition Detection Flag 0: Stop condition not detected 1: Stop condition detected	R/(W) ^{*1}
4	NACKF	NACK Detection Flag 0: NACK not detected 1: NACK detected	R/(W) ^{*1}
5	RDRF	Receive Data Full Flag 0: ICDRR contains no receive data 1: ICDRR contains receive data	R/(W) ^{*1}
6	TEND	Transmit End Flag 0: Data being transmitted 1: Data transmit complete	R/(W) ^{*1}
7	TDRE	Transmit Data Empty Flag 0: ICDRT contains transmit data 1: ICDRT contains no transmit data	R

Note 1. Only 0 can be written, to clear the flag.

TMOF flag (Timeout Detection Flag)

The TMOF flag is set to 1 when the IIC detects a timeout because the SCLn line state remains 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 (timeout function enabled) in master or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF flag after reading TMOF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

AL flag (Arbitration-Lost Flag)

The AL flag indicates that bus mastership was lost in arbitration because of a bus conflict or some other reason when a start condition was issued or an address and data was transmitted. The IIC monitors the level on the SDAn line during transmission and, if the level on the line does not match the value of the bit being output, is set the value of the AL flag to 1 to indicate that the bus is occupied by another device.

The IIC can also set the flag to indicate the detection of arbitration loss during NACK transmission in master mode or during data transmission in slave mode.

[Setting conditions]

When master arbitration-lost detection is enabled (ICFER.MALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of 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 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 apply an IIC reset or an internal reset.

Table 34.4 Relationship between arbitration-lost generation sources and arbitration-lost enable functions

ICFER			ICSR2	Error	Arbitration-lost generation source
MALE	NALE	SALE	AL		
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
			1	Transmit data mismatch	When ST in ICCR2 is set to 1 while BBSY in ICCR2 is 1
x	1	x	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master or slave receive mode
x	x	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

x: Don't care

START flag (Start Condition Detection Flag)

The START flag indicates whether a start or restart condition was detected.

[Setting condition]

- When a start or restart condition is detected.

[Clearing conditions]

- When 0 is written to the START flag after reading START = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

STOP flag (Stop Condition Detection Flag)

The STOP flag indicates whether a stop condition was detected.

[Setting condition]

- When a stop condition is detected.

[Clearing conditions]

- When 0 is written to the STOP flag after reading STOP = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

NACKF flag (NACK Detection Flag)

The NACKF flag indicates whether a NACK was detected.

[Setting condition]

- When acknowledge is not received (NACK received) from the receive device in transmit mode, with the NACKIE bit in ICFER set to 1 (transfer suspension enabled).

[Clearing conditions]

- When 0 is written to the NACKF flag after reading NACKF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1, the IIC suspends data transmission and reception. Writing to ICDRT in transmit mode or reading from ICDRR in receive mode with the NACKF flag set to 1 does not enable data transmit or receive operation. To restart data transmission or reception, set the NACKF flag to 0.

RDRF flag (Receive Data Full Flag)

The RDRF flag indicates whether the ICDRR contains receive data.

[Setting conditions]

- When receive data is transferred from ICDRS to ICDRR
The RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL clock cycle (selected in the RDRFS bit in ICMR3).
- When the received slave address matches after a start or restart condition is detected with the TRS bit in ICCR2 set to 0.

[Clearing conditions]

- When 0 is written to the RDRF flag after reading RDRF = 1
- When data is read from ICDRR
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

TEND flag (Transmit End Flag)

The TEND flag indicates completion of transmission.

[Setting condition]

- On the rising edge of the ninth SCL clock cycle while the TDRE flag is 1.

[Clearing conditions]

- When 0 is written to the TEND flag after reading TEND = 1
- When data is written to ICDRT
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

TDRE flag (Transmit Data Empty Flag)

The TDRE flag indicates whether the ICDRT contains transmit data.

[Setting conditions]

- When data is transferred from ICDRT to ICDRS and ICDRT becomes empty
- When the TRS bit in ICCR2 is set to 1
- When the received slave address matches while the TRS bit is 1.

[Clearing conditions]

- When data is written to ICDRT
- When the TRS bit in ICCR2 is set to 0
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1 while the NACKIE bit in ICFER is 1, the IIC suspends data transmission and reception. In this case, if the TDRE flag is 0 (next transmit data written), data is transferred to the ICDRS register and the ICDRT register becomes empty on the rising edge of the 9th clock cycle, but the TDRE flag does not set to 1.

34.2.11 SARLy : Slave Address Register Ly (y = 0 to 2)

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0A + 0x02 × y

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SVA[6:0]							SVA0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SVA0	10-bit Address LSB Slave address setting.	R/W
7:1	SVA[6:0]	7-bit Address/10-bit Address Lower Bits Slave address setting.	R/W

SVA0 bit (10-bit Address LSB)

When the 10-bit address format is selected (SARUy.FS = 1), the SVA0 bit functions as the LSB of a 10-bit address and is combined with the SVA[6:0] bits to form the lower 8 bits of a 10-bit address.

This bit is valid when the SARyE bit in ICSEr is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in this bit is ignored.

SVA[6:0] bits (7-bit Address/10-bit Address Lower Bits)

When the 7-bit address format is selected (SARUy.FS = 0), the SVA[6:0] bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS = 1), these bits combine with the SVA0 bit to form the lower 8 bits of a 10-bit address.

When the SARyE bit in ICSEr is 0, the setting in these bits is ignored.

34.2.12 SARUy : Slave Address Register Uy (y = 0 to 2)

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0B + 0x02 × y

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

Bit	Symbol	Function	R/W
0	FS	7-bit/10-bit Address Format Select 0: Select 7-bit address format 1: Select 10-bit address format	R/W
2:1	SVA[1:0]	10-bit Address Upper Bits Slave address setting.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

FS bit (7-bit/10-bit Address Format Select)

The FS bit selects 7- or 10-bit format for slave address y (in SARLy and SARUy).

When the SARyE bit in ICSEr is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SVA[6:0] setting in SARLy is valid, and the SVA[1:0] and SVA0 settings in SARLy are ignored.

When the SARyE bit in ICSEr is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the SVA[1:0] and SARLy settings are valid.

When the SARyE bit in ICSEr is 0 (SARLy and SARUy disabled), the SARUy.FS setting is invalid.

SVA[1:0] bits (10-bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), the SVA[1:0] bits function as the upper 2 bits of a 10-bit address.

These bits are valid when the SARyE bit in ICSEr is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in these bits is ignored.

34.2.13 ICBRL : I²C Bus Bit Rate Low-Level Register

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x10

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

Bit	Symbol	Function	R/W
4:0	BRL[4:0]	Bit Rate Low-Level Period Low-level period of SCL clock.	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W

BRL[4:0] bits (Bit Rate Low-Level Period)

The BRL[4:0] bits set the low-level period of the SCL clock. ICBRL counts the low-level period with the internal reference clock source (IICφ) specified by the CKS[2:0] bits in ICMR1. ICBRL also generates the data setup time for automatic SCL low-hold operation, see [section 34.8. Automatic Low-Hold Function for SCL](#). When the IIC is used in slave mode, the BRL[4:0] bits must be set to a value longer than the data setup time*1.

If the digital noise filter is enabled (NFE bit in ICFER is 1), set the BRL[4:0] bits to a value at least one greater than the number of stages in the noise filter. For details on the number of stages, see the description of the NF[1:0] bits in [section 34.2.5. ICMR3 : I²C Bus Mode Register 3](#).

Note 1. Data setup time (t_{SU} : DAT)

250 ns for up to 100 kbps: Standard-mode (Sm)

100 ns for up to 400 kbps: Fast-mode (Fm)

34.2.14 ICBRH : I²C Bus Bit Rate High-Level Register

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x11

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

Bit	Symbol	Function	R/W
4:0	BRH[4:0]	Bit Rate High-Level Period High-level period of SCL clock.	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W

BRH[4:0] bits (Bit Rate High-Level Period)

The BRH[4:0] bits set the high-level period of SCL clock. BRH[4:0] bits are valid in master mode. If the IIC is used only in slave mode, do not set the BRH[4:0] bits.

ICBRH counts the high-level period with the internal reference clock source (IICφ) specified in the CKS[2:0] bits in ICMR1.

If the digital noise filter is enabled (the NFE bit in ICFER is 1), set these bits to a value at least one greater than the number of stages in the noise filter. For the number of stages in the noise filter, see the description of the NF[1:0] bits in [section 34.2.5. ICMR3 : I²C Bus Mode Register 3](#).

The IIC transfer rate and the SCL clock duty are calculated using the following expressions (1) to (5):

- ICFER.SCLE = 0

$$\text{Transfer rate} = 1 / [\{(BRH + 1) + (BRL + 1)\} / IIC\phi^{*1} + tr^{*2} + tf^{*2}]$$

$$\text{Duty cycle} = [tr + \{(BRH + 1) / IIC\phi\}] / [tr + tf + \{(BRH + 1) + (BRL + 1)\} / IIC\phi]$$
- ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] = 000b (IICφ = PCLKB)

$$\text{Transfer rate} = 1 / [\{(BRH + 3) + (BRL + 3)\} / IIC\phi + tr + tf]$$

$$\text{Duty cycle} = [tr + \{(BRH + 3) / IIC\phi\}] / [tr + tf + \{(BRH + 3) + (BRL + 3)\} / IIC\phi]$$
- ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] = 000b (IICφ = PCLKB)

$$\text{Transfer rate} = 1 / [\{(BRH + 3 + nf^{*3}) + (BRL + 3 + nf)\} / IIC\phi + tr + tf]$$

$$\text{Duty cycle} = [tr + \{(BRH + 3 + nf) / IIC\phi\}] / [tr + tf + \{(BRH + 3 + nf) + (BRL + 3 + nf)\} / IIC\phi]$$
- ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] ≠ 000b

$$\text{Transfer rate} = 1 / [\{(BRH + 2) + (BRL + 2)\} / IIC\phi + tr + tf]$$

$$\text{Duty cycle} = [tr + \{(BRH + 2) / IIC\phi\}] / [tr + tf + \{(BRH + 2) + (BRL + 2)\} / IIC\phi]$$
- ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] ≠ 000b

$$\text{Transfer rate} = 1 / [\{(BRH + 2 + nf) + (BRL + 2 + nf)\} / IIC\phi + tr + tf]$$

$$\text{Duty cycle} = [tr + \{(BRH + 2 + nf) / IIC\phi\}] / [tr + tf + \{(BRH + 2 + nf) + (BRL + 2 + nf)\} / IIC\phi]$$

Note 1. IICφ = PCLKB × division ratio

Note 2. The SCLn line rise time (tr) and SCLn line fall time (tf) depend on the total bus line capacitance (Cb) and the pull-up resistor (Rp). For details, see the I²C bus standard from NXP Semiconductors.

Note 3. nf = Number of digital noise filters selected in the ICMR3.NF bit.

Table 34.5 Example of ICBRH/ICBRL Settings for Transfer Rate IIC when SCLE = 0

Transfer rate (kbps)	PCLKB (MHz)								
	8			20			32		
	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)
10	100b	23(0x17)	25(0x19)	110b	16(0x10)	13(0x0D)	110b	23(0x17)	25(0x19)
50	010b	16(0x10)	20(0x14)	011b	21(0x15)	24(0x18)	100b	16(0x10)	20(0x14)
100	001b	15(0x0F)	18(0x12)	010b	19(0x13)	23(0x17)	011b	15(0x0F)	18(0x12)
400	001b	2(0x02)	5(0x05)	001b	5(0x05)	12(0x0C)	001b	9(0x09)	20(0x14)

Note: These values are calculated using the calculation formula 1) with the following settings.

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 34.6 Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 0

Transfer rate (kbps)	PCLKB (MHz)								
	8			20			32		
	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)
10	100b	19(0x13)	27(0x1B)	110b	15(0x0F)	12(0x0C)	110b	20(0x14)	26(0x1A)
50	010b	17(0x11)	17(0x11)	011b	19(0x13)	24(0x18)	100b	16(0x10)	18(0x12)
100	001b	19(0x13)	12(0x0C)	010b	19(0x13)	21(0x15)	011b	15(0x0F)	16(0x10)
400	001b	1(0x01)	4(0x04)	001b	4(0x04)	11(0x0B)	001b	8(0x08)	19(0x13)

Note: These values are calculated using the calculation formula 4) with the following settings.

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 34.7 Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 1

Transfer rate (kbps)	PCLKB (MHz)								
	8			20			32		
	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)
10	100b	19(0x13)	23(0x17)	110b	15(0x0F)	8(0x08)	110b	19(0x13)	23(0x17)
50	010b	17(0x11)	13(0x0D)	011b	19(0x13)	20(0x14)	100b	16(0x10)	14(0x0E)
100	001b	19(0x13)	8(0x08)	010b	19(0x13)	17(0x11)	011b	15(0x0F)	12(0x0C)
400	000b	3(0x03)	7(0x07)	001b	3(0x03)	9(0x09)	001b	6(0x06)	17(0x11)

Note: These values are calculated using the calculation formula 5) with the following settings. (ICMR3.NF[1:0] = 01b).

Calculated that only the value for the 400 kbps setting at PCLKB = 8 MHz is calculated with ICMR1.CKS[2:0] = 000b.

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

34.2.15 ICDRT : I²C Bus Transmit Data Register

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x12

Bit position: 7 6 5 4 3 2 1 0

Bit field:

Value after reset: 1 1 1 1 1 1 1 1

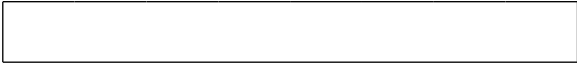
When ICDRT detects a space in the I²C Bus Shift Register (ICDRS), it transfers the transmit data that was written to ICDRT to ICDRS and starts transmitting data in transmit mode. The double-buffer structure of ICDRT and ICDRS allows continuous transmit operation if the next transmit data is written to ICDRT while the ICDRS data is being transmitted.

ICDRT can always be read and written to. Write transmit data to ICDRT once when a transmit data empty interrupt (IICn_TXI) request is generated.

34.2.16 ICDRR : I²C Bus Receive Data Register

Base address: IICn = 0x4005_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x13

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

When 1 byte of data is received, the received data is transferred from the I²C Bus Shift Register (ICDRS) to ICDRR to enable the next data to be received. The double-buffer structure of ICDRS and ICDRR allows continuous receive operation if the received data is read from ICDRR while ICDRS is receiving data. ICDRR cannot be written to. Read data from ICDRR once when a receive data full interrupt (IICn_RXI) request is generated.

If ICDRR receives the next receive data before the current data is read from ICDRR (while the RDRF flag in ICSR2 is 1), the IIC automatically holds the SCL clock low 1 cycle before the RDRF flag is set to 1 next.

34.2.17 ICDRS : I²C Bus Shift Register

Base address: n/a

Offset address: n/a

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

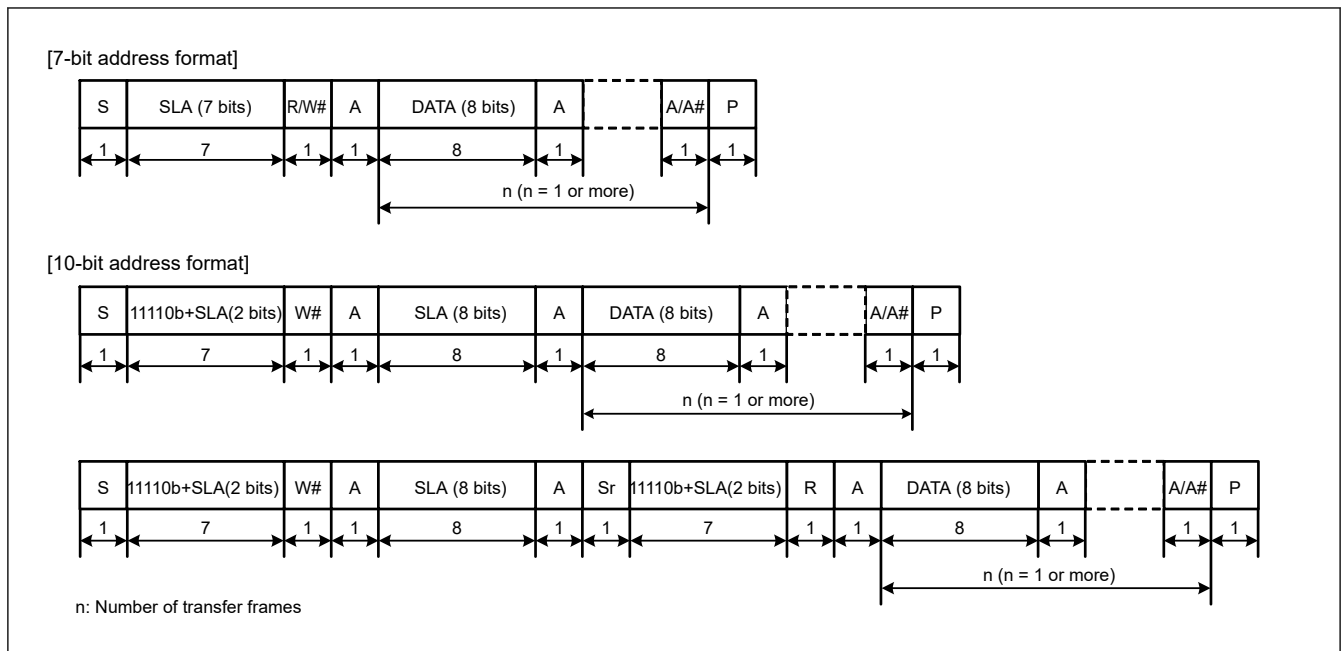
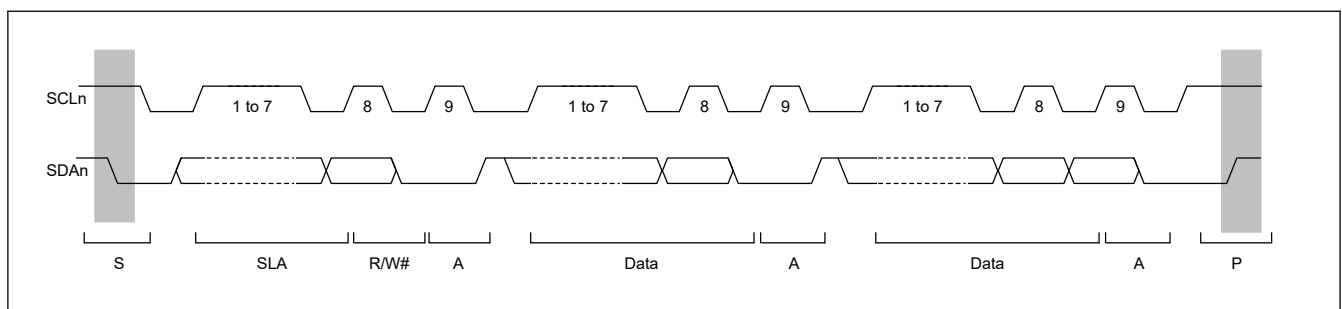
ICDRS is an 8-bit shift register for data transmit and receive. During transmission, transmit data is transferred from ICDRT to ICDRS and is transmitted from the SDAn pin. During reception, data is transferred from ICDRS to ICDRR after 1 byte of data is received. ICDRS cannot be accessed directly.

34.3 Operation

34.3.1 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 34.3](#) shows the I²C bus format, and [Figure 34.4](#) shows the I²C bus timing.

Figure 34.3 I²C bus formatFigure 34.4 I²C bus timing when the SLA setting = 7 bits

- S: Start condition. The master device drives the SDAn line low from high while the SCLn line is high.
- SLA: Slave address, by which the master device selects a slave device.
- R/W#: Indicates the direction of data transfer: from the slave device to the master device when R/W# is 1, or from the master device to the slave device when R/W# is 0.
- A: Acknowledge. The receive device drives the SDAn line low. In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.
- A#: Not Acknowledge. The receive device drives the SDAn line high.
- Sr: Restart condition. The master device drives the SDAn line low from the high level after the setup time has elapsed with the SCLn line high.
- DATA: Transmitted or received data.
- P: Stop condition. The master device drives the SDAn line high from low while the SCLn line is high.

34.3.2 Initial Settings

Before starting data transmission or reception, initialize the IIC using the procedure shown in [Figure 34.5](#).

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 34.5](#).
5. When the required register settings are complete, set the ICCR1.IICRST bit to 0 to release the IIC reset.

This procedure is not required if the IIC initialization is already complete.

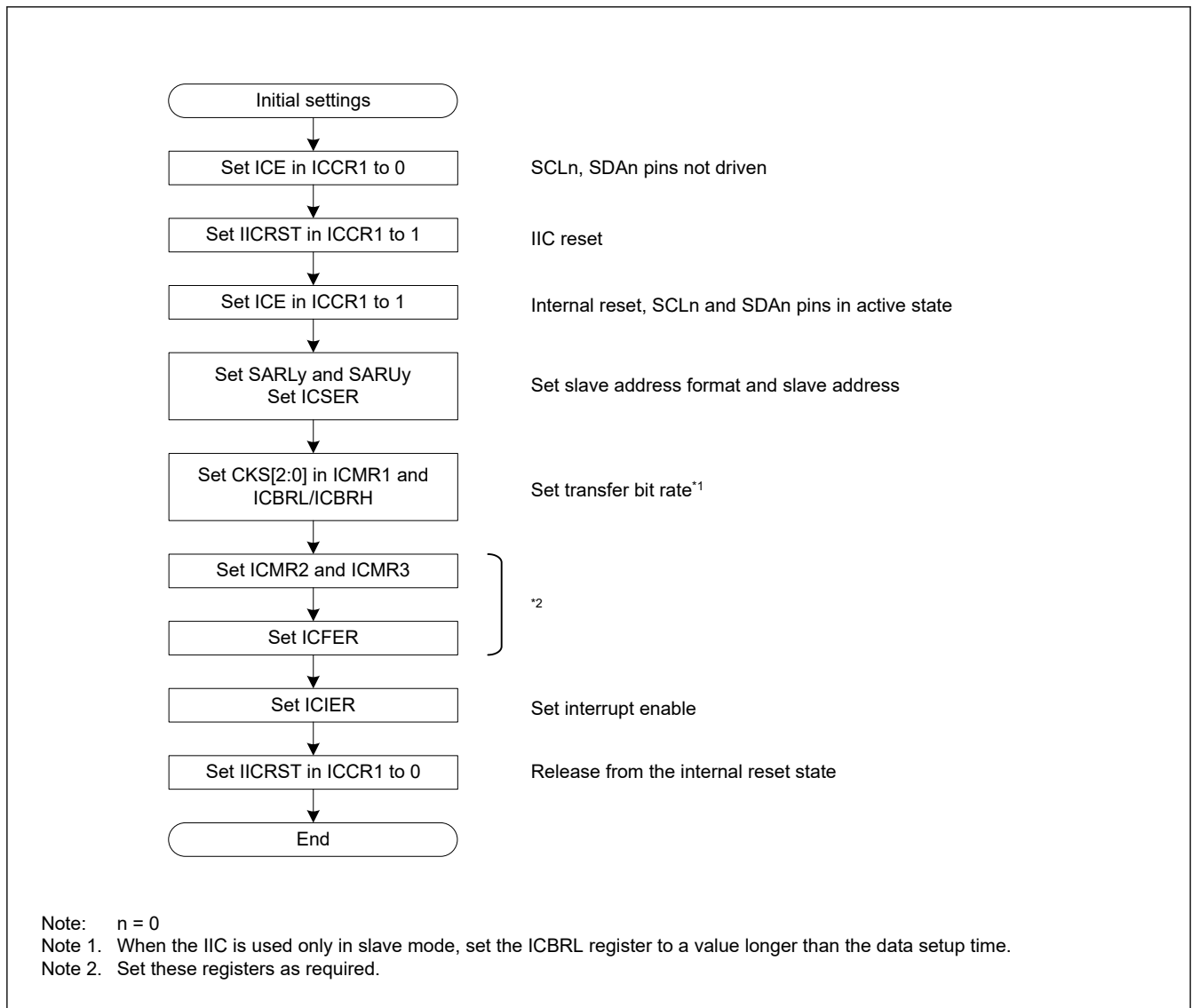


Figure 34.5 Example IIC initialization flow

34.3.3 Master Transmit Operation

In master transmit operation, the IIC outputs the SCL clock and transmitted data signals as the master device, and the slave device returns acknowledgments. [Figure 34.6](#) shows an example of master transmission, and [Figure 34.7](#) to [Figure 34.9](#) show the operation timing in master transmission.

To set up and perform master transmission:

1. Process initial settings. For details, see [section 34.3.2. Initial Settings](#).
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition request). On receiving the request, the IIC issues a start condition. At the same time, the BBSY and START flags in ICSR2 automatically set to 1 and the ST bit automatically is set to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that issuance of the start condition as requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also automatically is set to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag automatically is set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again is set to 1. After the byte containing the slave address and R/W# bit is

transmitted, the value of the TRS bit automatically updates to select master transmit or master receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the IIC continues in master transmit mode. Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.

For data transmission with an address in the 10-bit format, start by writing 11110b, the 2 higher-order bits of the slave address, and W to ICDRT as the first address transmission. For the second address transmission, write the 8 lower-order bits of the slave address to ICDRT.

4. Check that the TDRE flag in ICSR2 is 1, and then write the transmit data to the ICDRT register. The IIC automatically holds the SCLn line low until the transmit data is ready or a stop condition is issued.
5. After all bytes of transmit data are written to the ICDRT register, wait until the value in the TEND flag in ICSR2 returns to 1, and then set the SP bit in ICCR2 to 1 (stop condition requested). On receiving a stop condition request, the IIC issues the stop condition. Regarding issuing a stop condition, see [section 34.10.3. Issuing a Stop Condition](#).
6. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, it automatically sets the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 to 1.
7. Check that the ICSR2.STOP flag is 1, and then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

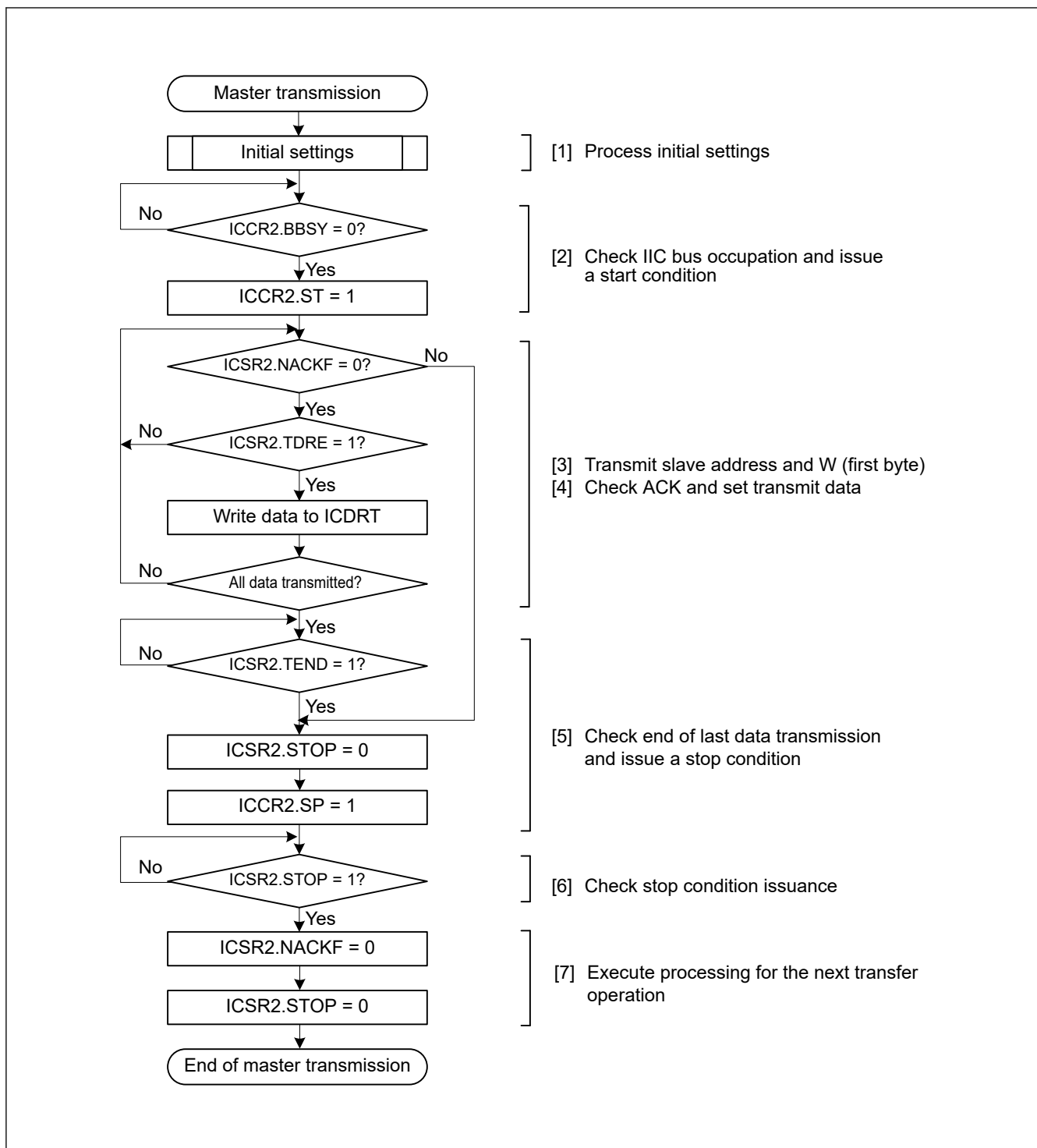


Figure 34.6 Example master transmission flow

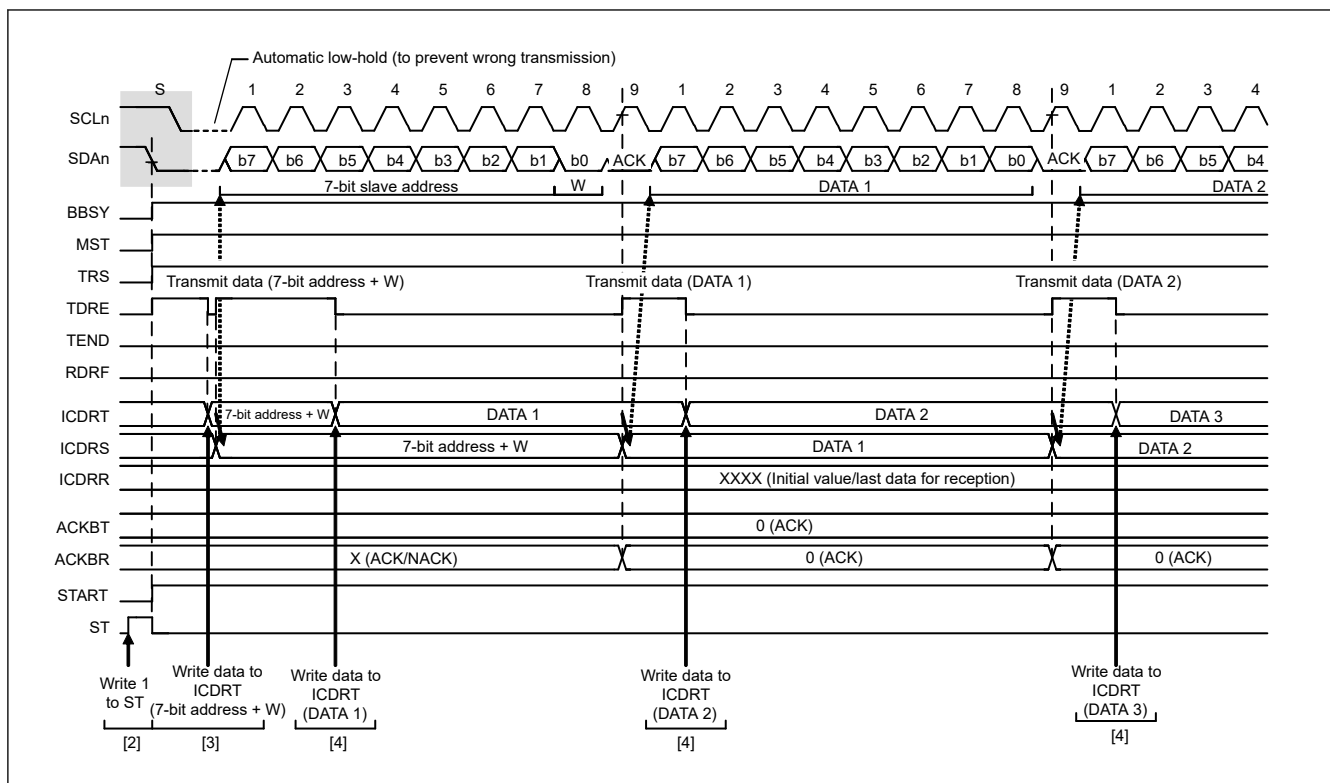


Figure 34.7 Master transmit operation timing (1) with 7-bit address format

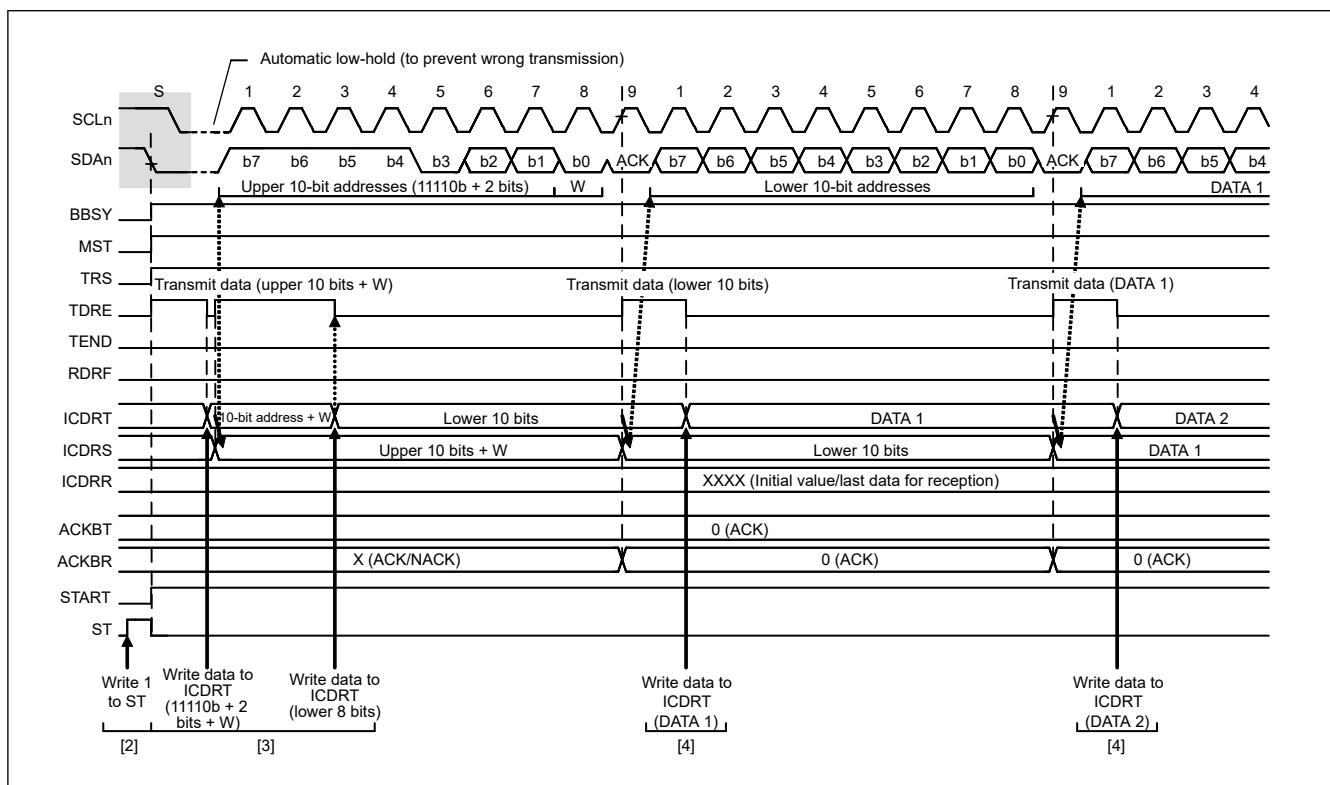


Figure 34.8 Master transmit operation timing (2) with 10-bit address format

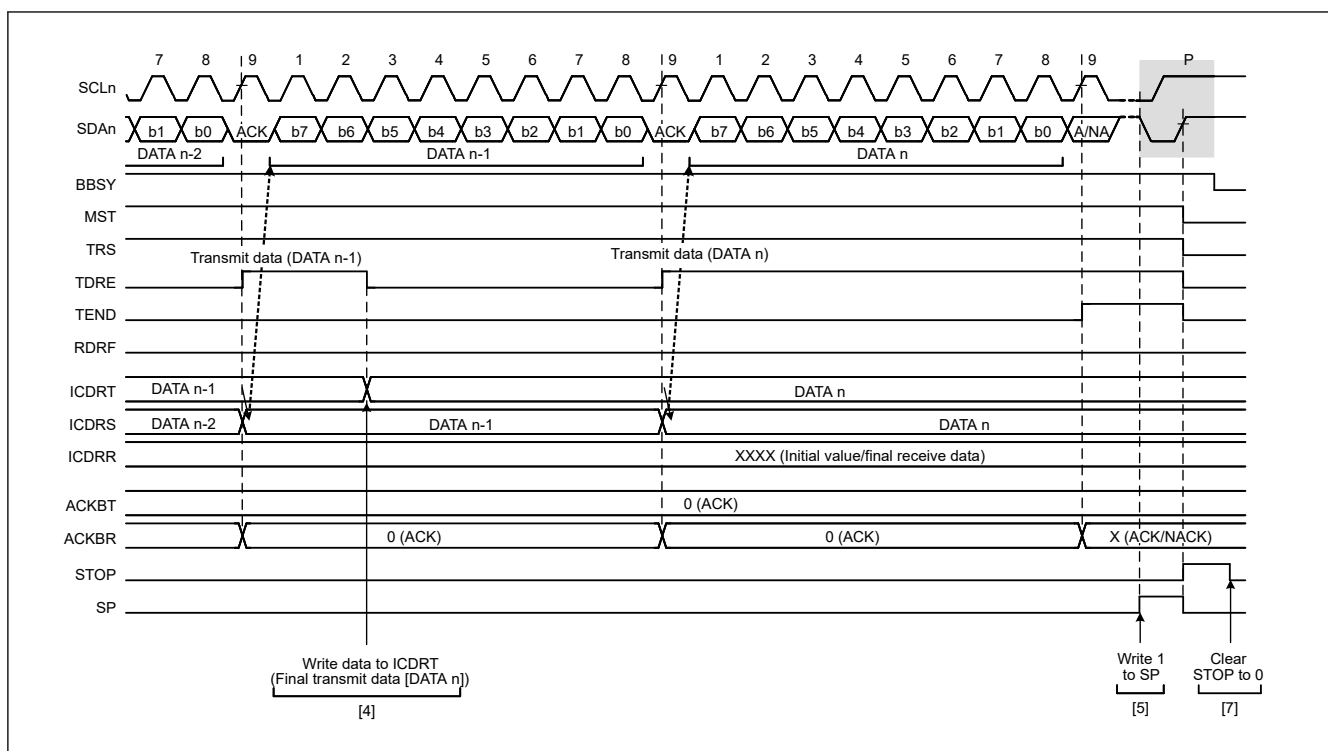


Figure 34.9 Master transmit operation timing (3)

34.3.4 Master Receive Operation

In master receive operation, the IIC as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgments. Because the IIC must start by sending a slave address to the associated slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 34.10 and Figure 34.11 show examples of master reception (7-bit address format), and Figure 34.12 to Figure 34.14 show the operation timing in master reception.

To set up and perform master reception:

1. Process initial settings. For details, see [section 34.3.2. Initial Settings](#).
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition request). On receiving the request, the IIC issues a start condition. When the IIC detects the start condition, the BBSY and START flags in ICSR2 automatically set to 1, and the ST bit automatically is set to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA line match while the ST bit is 1, the IIC recognizes that issuance of the start condition as requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also automatically is set to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag automatically is set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again is set to 1. When the byte containing the slave address and R/W# bit is transmitted, the value of the ICCR2.TRS bit automatically updates to select transmit or receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit is 1, the TRS bit is set to 0 on the rising edge of the ninth cycle of the SCL clock, placing the IIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag automatically is set to 1. Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition. For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a restart condition. After that, transmitting 11110b, the two higher-order bits of the slave address, and the R bit places the IIC in master receive mode.
4. Dummy read ICDRR after confirming that the RDRF flag in ICSR2 is 1. This makes the IIC start output of the SCL clock and start data reception.

5. After 1 byte of data is received, the RDRF flag in ICSR2 is set to 1 on the rising edge of the 8th or 9th cycle of the SCL clock, as selected in the RDRFS bit in ICMR3. Reading ICDRR at this time produces the received data, and the RDRF flag is automatically set to 0 at the same time. Additionally, the value of the acknowledgment field received during the ninth cycle of the SCL clock is returned as the value set in the ICMR3.ACKBT bit. If the next byte to be received is the second-to-last byte, set the ICMR3.WAIT bit to 1 for wait insertion before reading ICDRR, containing the second-to-last byte. In addition to enabling NACK output, even when interrupts or other operations result in delays in setting the ICMR3.ACKBT bit to 1 (NACK) in step (6), this fixes the SCLn line to the low level on the rising edge of the ninth clock cycle in reception of the last byte, which enables the issuing of a stop condition.
6. When the ICMR3.RDRFS bit is 0, and the slave device must be notified that it is to end transfer for data reception after transfer of the next and final byte, set the ICMR3.ACKBT bit to 1 (NACK).
7. After reading the second-to-last byte from the ICDRR register, if the value of the ICSR2.RDRF flag is 1, write 1 to the SP bit in ICCR2 (stop condition requested), and then read the last byte from ICDRR. When ICDRR is read, the IIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is complete or the SCLn line is released from the low-hold state.
8. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, detection of the stop condition sets the ICSR2.STOP flag to 1.
9. Check that the ICSR2.STOP flag is 1, then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

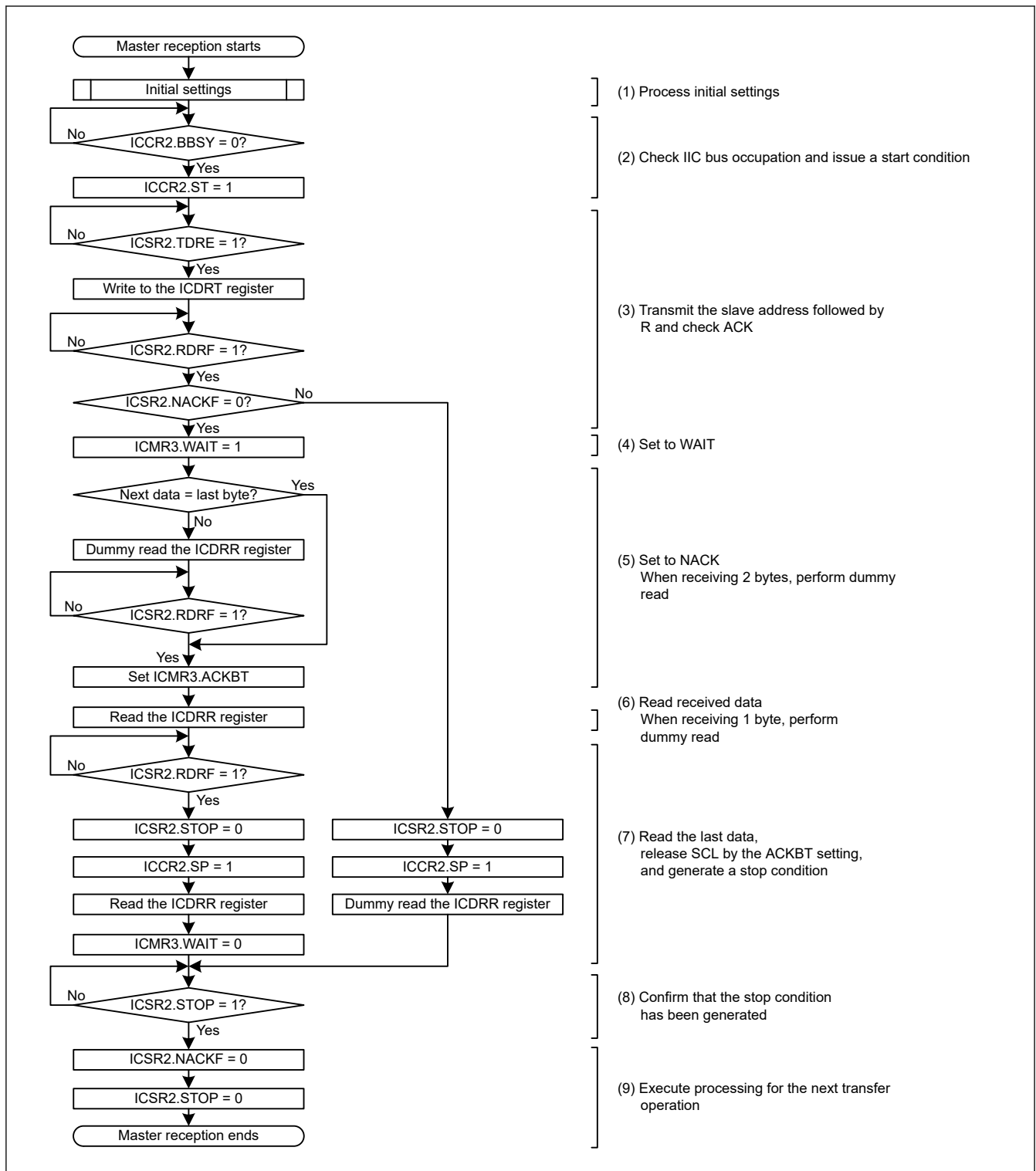


Figure 34.10 Example master reception flow with 7-bit address format of 1 byte or 2 bytes

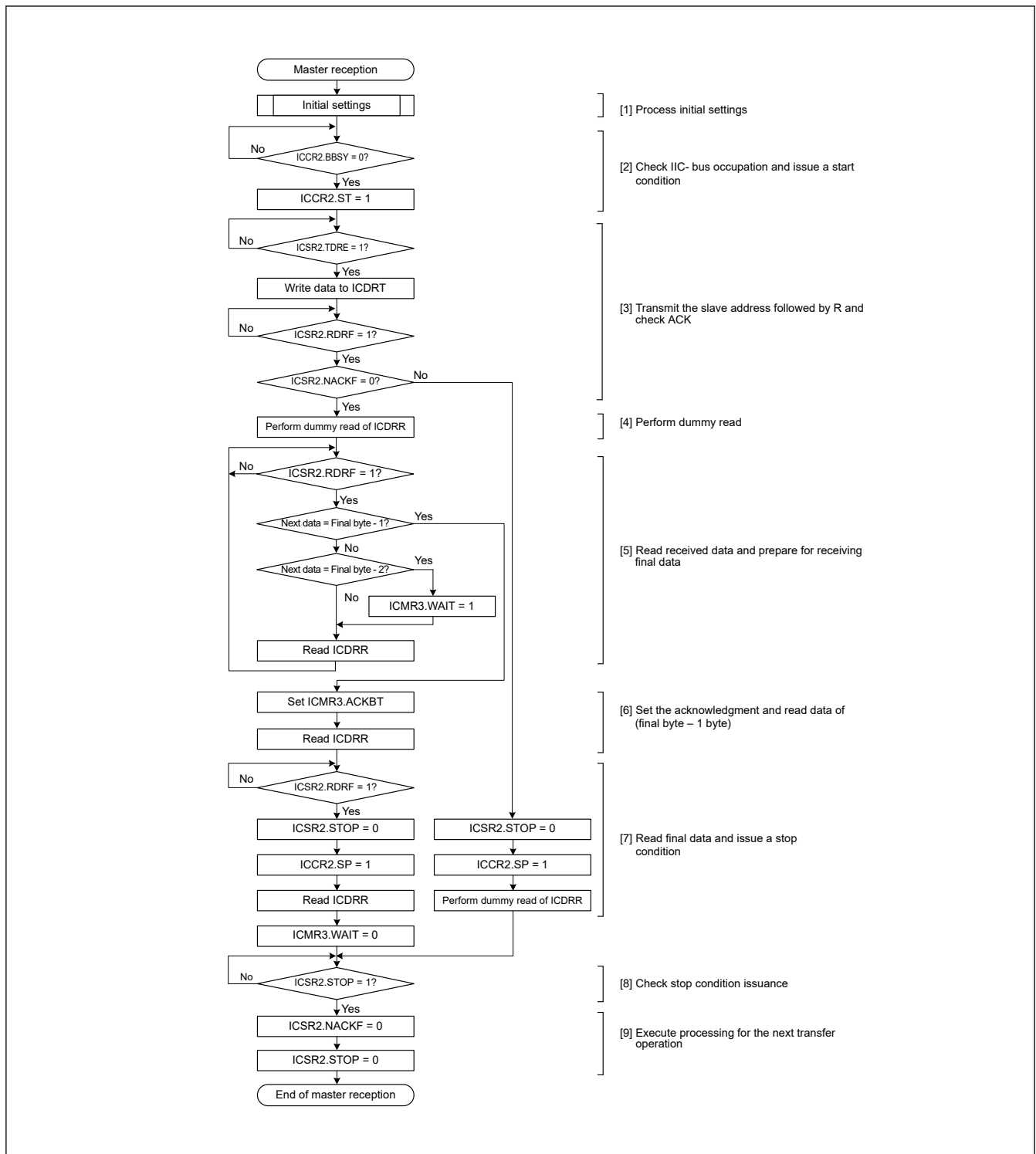


Figure 34.11 Example master reception flow with 7-bit address format of 3 or more bytes

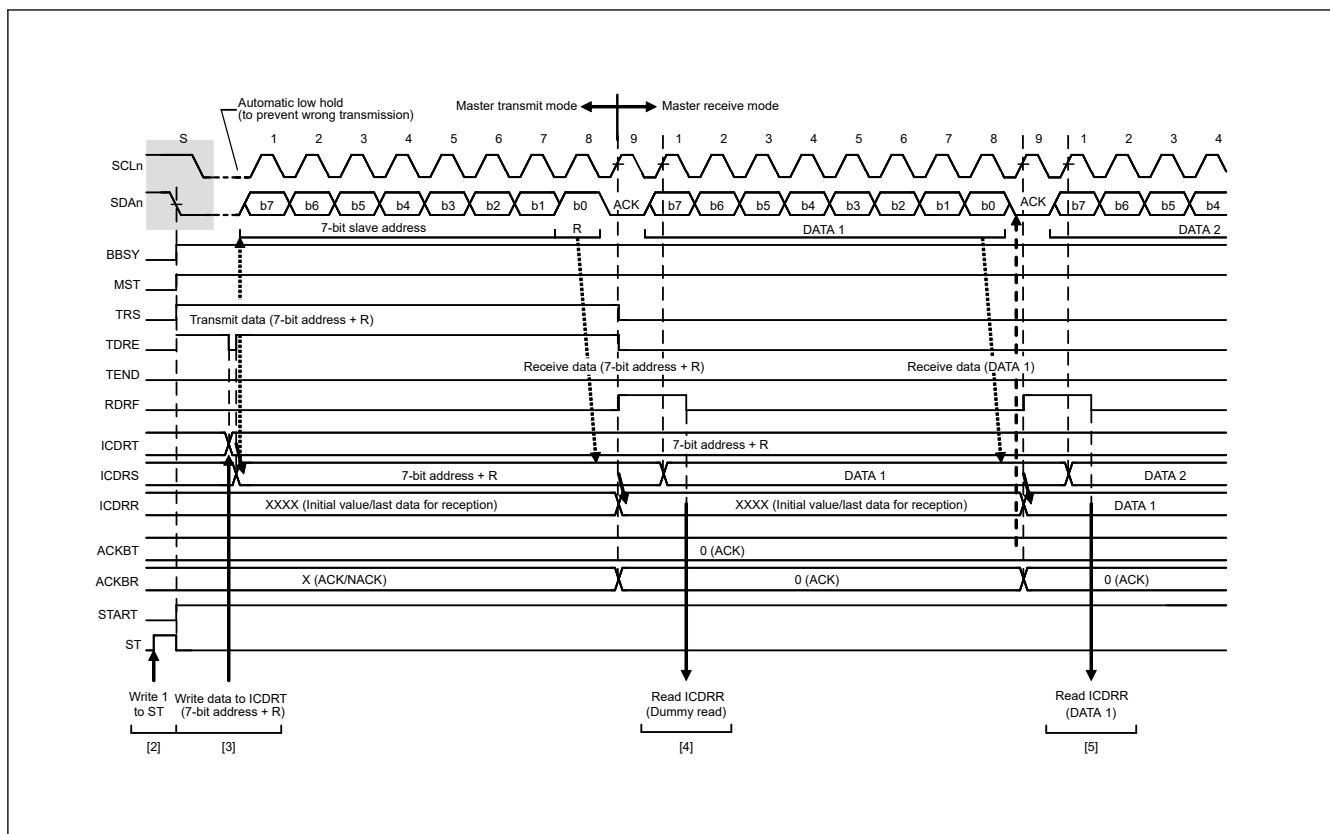


Figure 34.12 Master receive operation timing (1) with 7-bit address format when RDRFS = 0

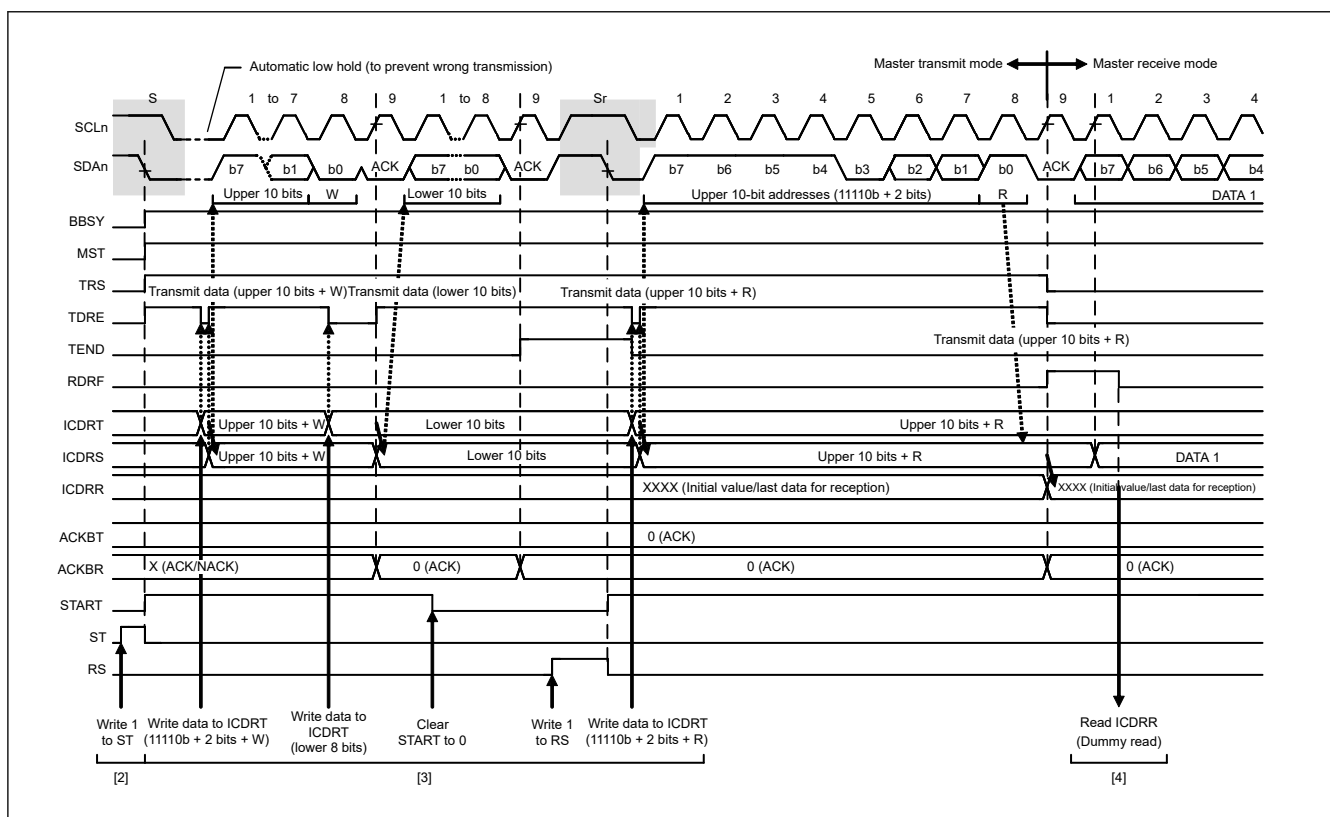


Figure 34.13 Master receive operation timing (2) with 10-bit address format when RDRFS = 0



In slave transmit operation, the master device outputs the SCL clock, the IIC transmits data as a slave device, and the master device returns acknowledgments.

To set up and perform slave transmission:

1. Initialize the IIC using the procedure in [section 34.3.2. Initial Settings](#).
After initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 1 on the rising edge of the ninth cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of the SCL clock. If the value of the R/W# bit is 1, the IIC automatically places itself in slave transmit mode by setting both the ICCR2.TRS bit and the ICSR2.TDRE flag to 1.
3. Check that the ICSR2.TDRE flag is 1, then write the transmit data to the ICDRT register. If the IIC receives no acknowledge from the master device (receives an NACK signal) while the ICFER.NACKE bit is 1, the IIC suspends transfer of the next data.
4. Wait until the ICSR2.TEND flag is set to 1 while the ICSR2.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDRT register. When the ICSR2.NACKF flag or the TEND flag is 1, the IIC drives the SCLn line low on the ninth falling edge of the SCL clock.
5. When the ICSR2.NACKF or ICSR2.TEND flag is 1, dummy read ICDRR to complete the processing. This releases the SCLn line.
6. On detecting the stop condition, the IIC automatically sets the ICSR1.HOA, GCA, and AASy flags (y = 0 to 2), the ICSR2.TDRE and TEND flags, and the ICCR2.TRS bit to 0, and enters slave receive mode.
7. Check that the ICSR2.STOP flag is 1, then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

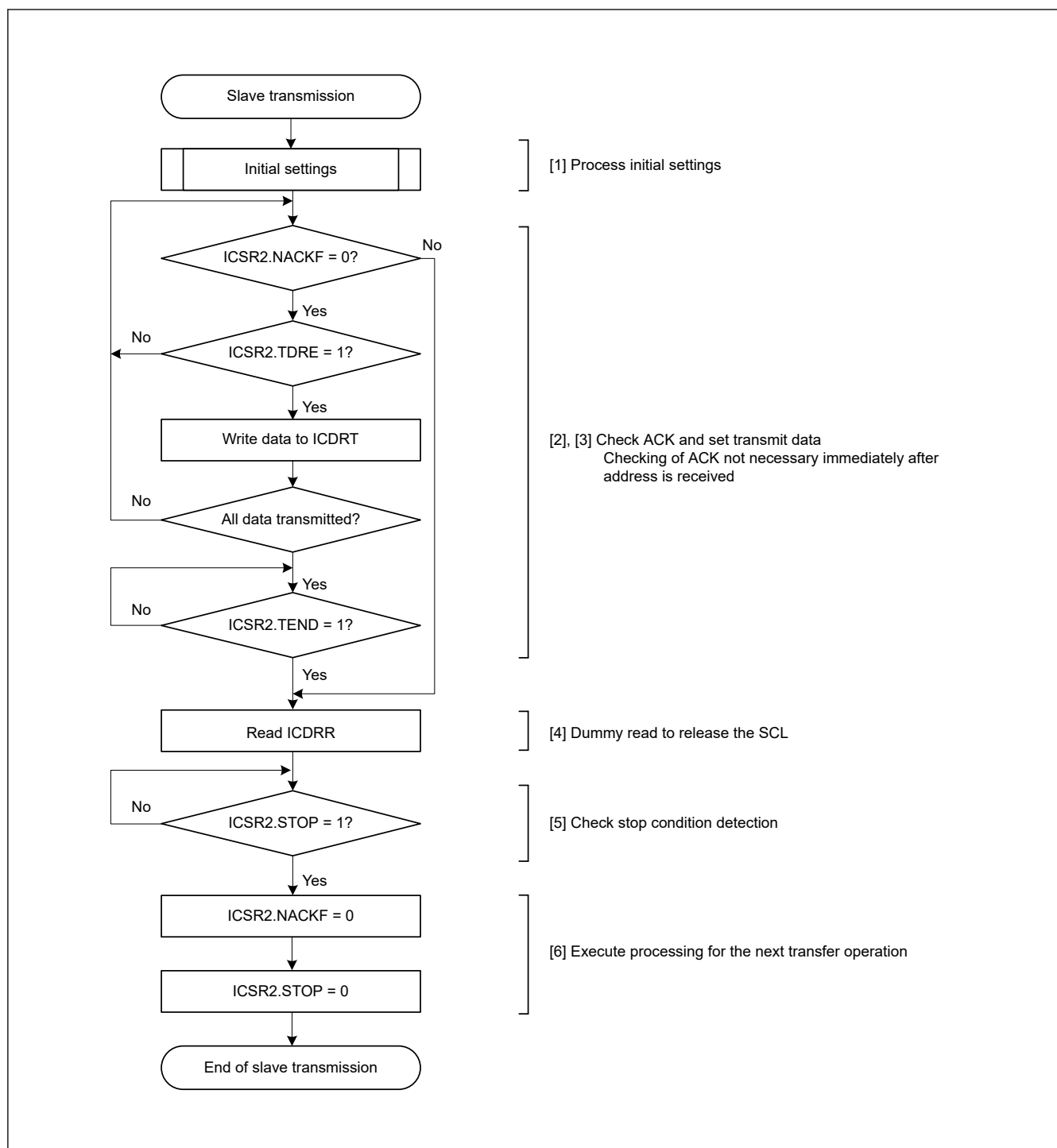


Figure 34.15 Example slave transmission flow

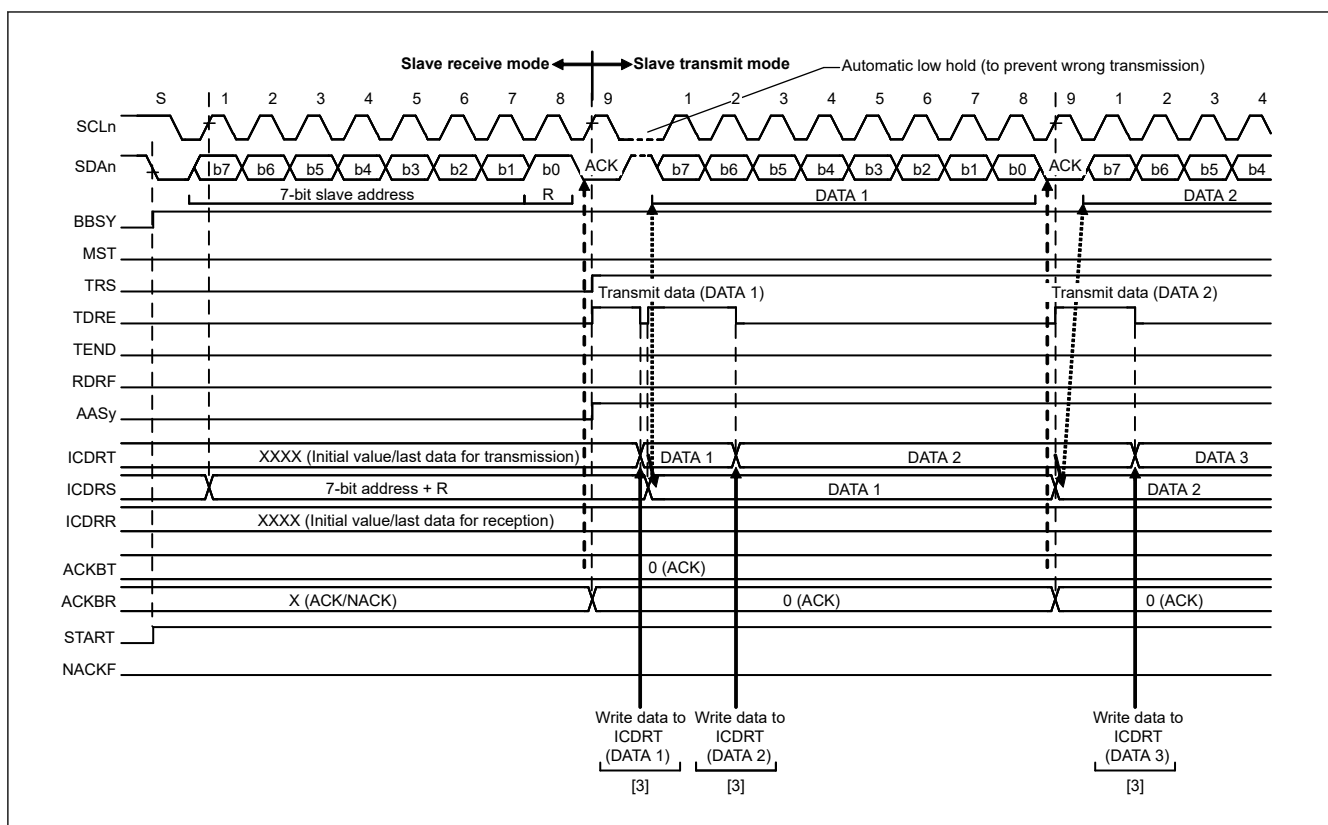


Figure 34.16 Slave transmit operation timing (1) with 7-bit address format

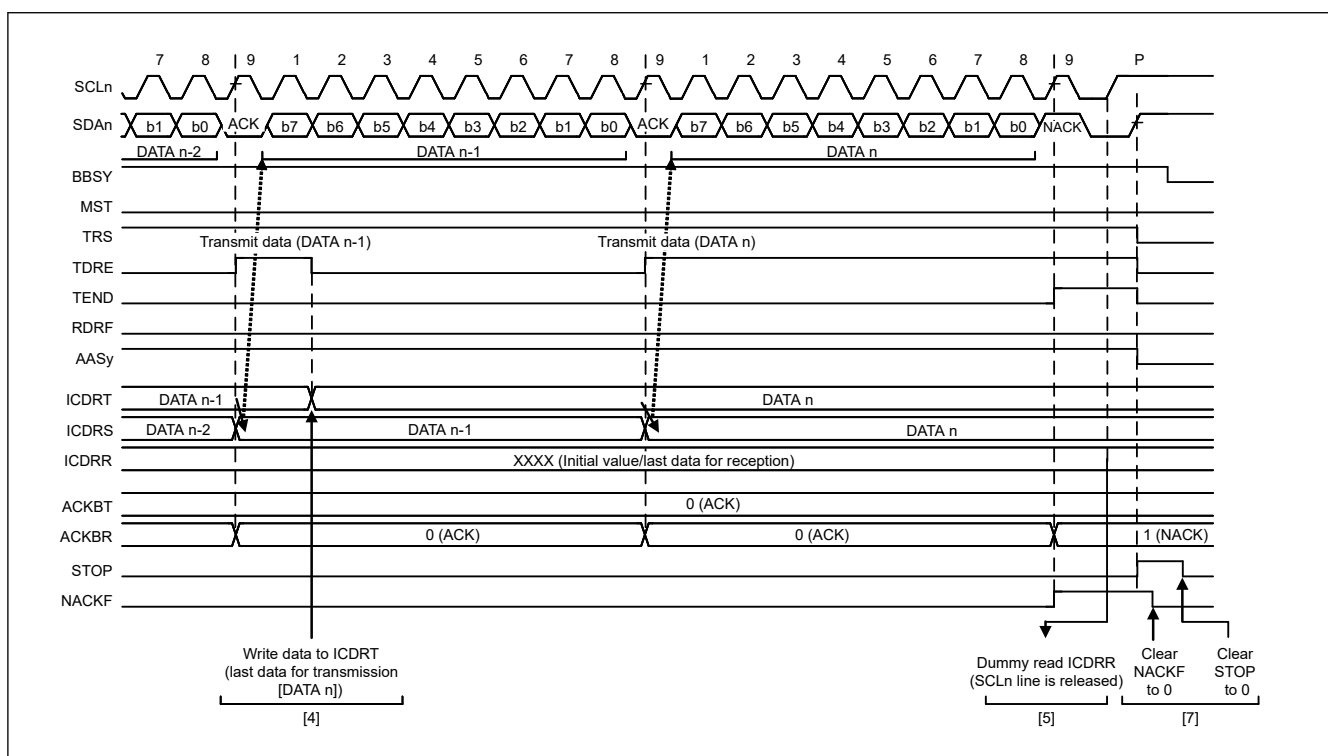


Figure 34.17 Slave transmit operation timing (2)

34.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the IIC returns acknowledgments as a slave device.

Figure 34.18 shows an example of slave reception, and Figure 34.19 and Figure 34.20 show the operation timing in slave reception.

To set up and perform slave reception:

1. Initialize the IIC using the procedure in [section 34.3.2. Initial Settings](#).
After initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 1 on the rising edge of the ninth cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of the SCL clock. If the value of the R/W# bit is 0, the IIC continues to place itself in slave receive mode and sets the RDRF flag in ICSR2 to 1.
3. Check that the ICSR2.STOP flag is 0 and the ICSR2.RDRF flag is 1, then dummy read ICDRR. The dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected.
4. When ICDRR is read, the IIC automatically sets the ICSR2.RDRF flag to 0. If reading of ICDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the IIC holds the SCLn line low until 1 SCL cycle before the point where RDRF must be set. In this case, reading ICDRR releases the SCLn line from being held low.
When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read ICDRR until all the data is completely received.
5. On detecting the stop condition, the IIC automatically clears the ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 0.
6. Check that the ICSR2.STOP flag is 1, then set the ICSR2.STOP flag to 0 for the next transfer operation.

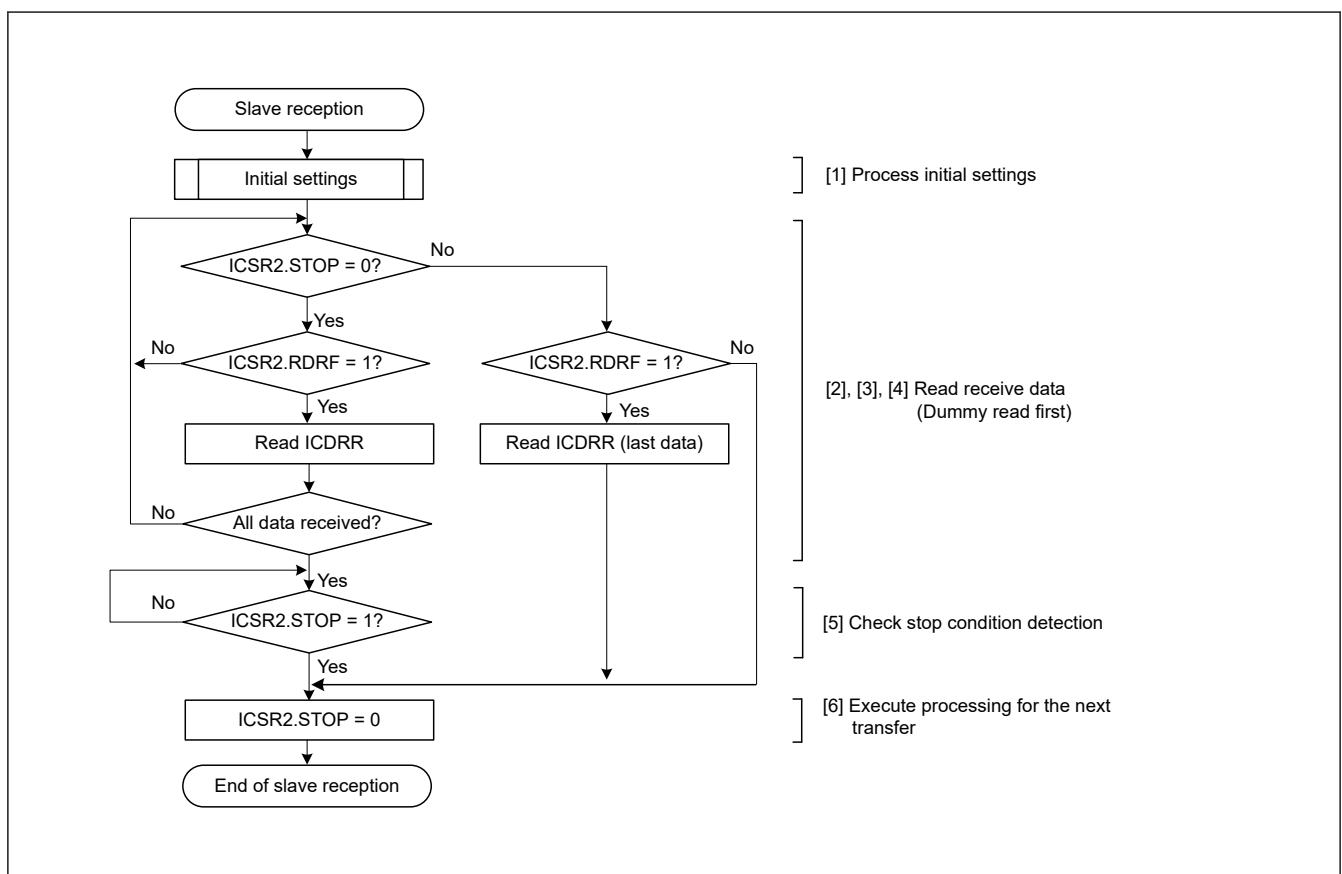


Figure 34.18 Example slave reception flow

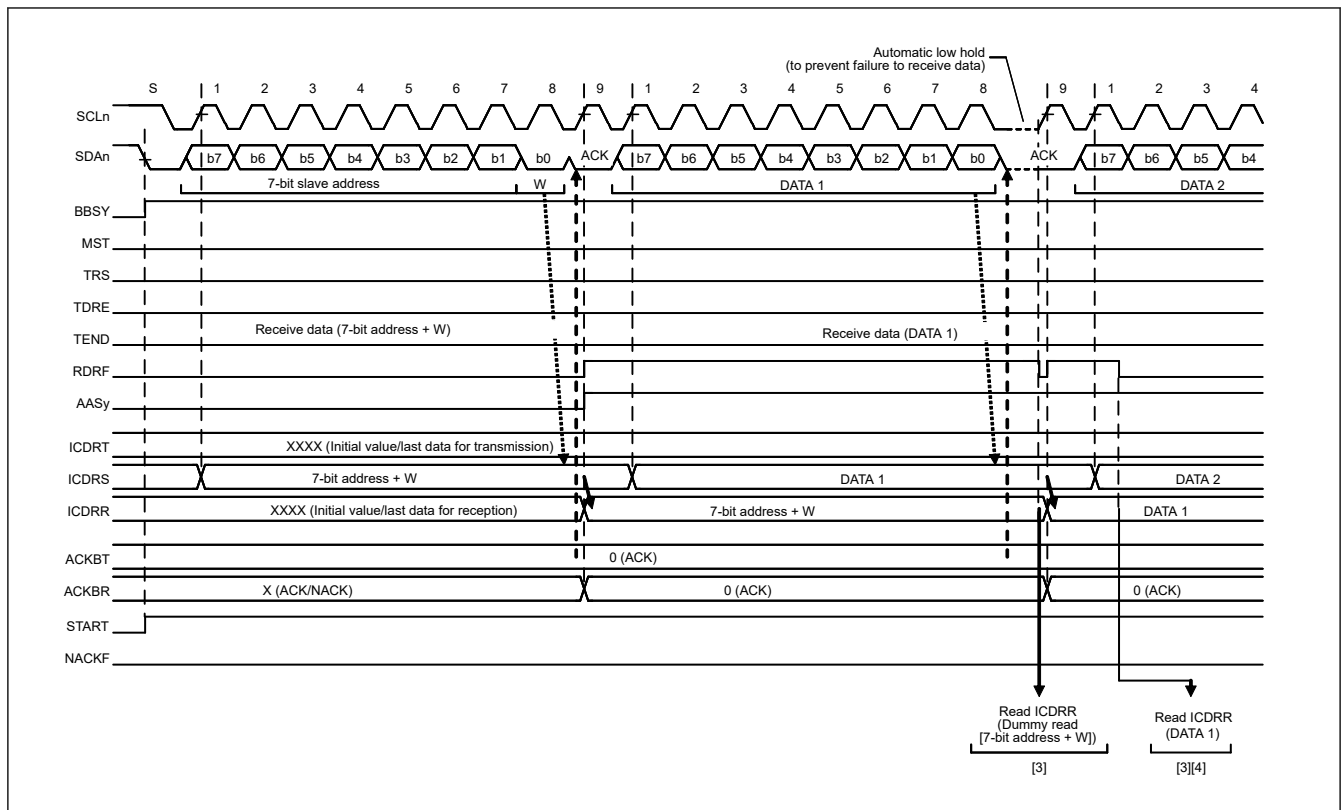


Figure 34.19 Slave receive operation timing (1) with 7-bit address format when RDRFS = 0

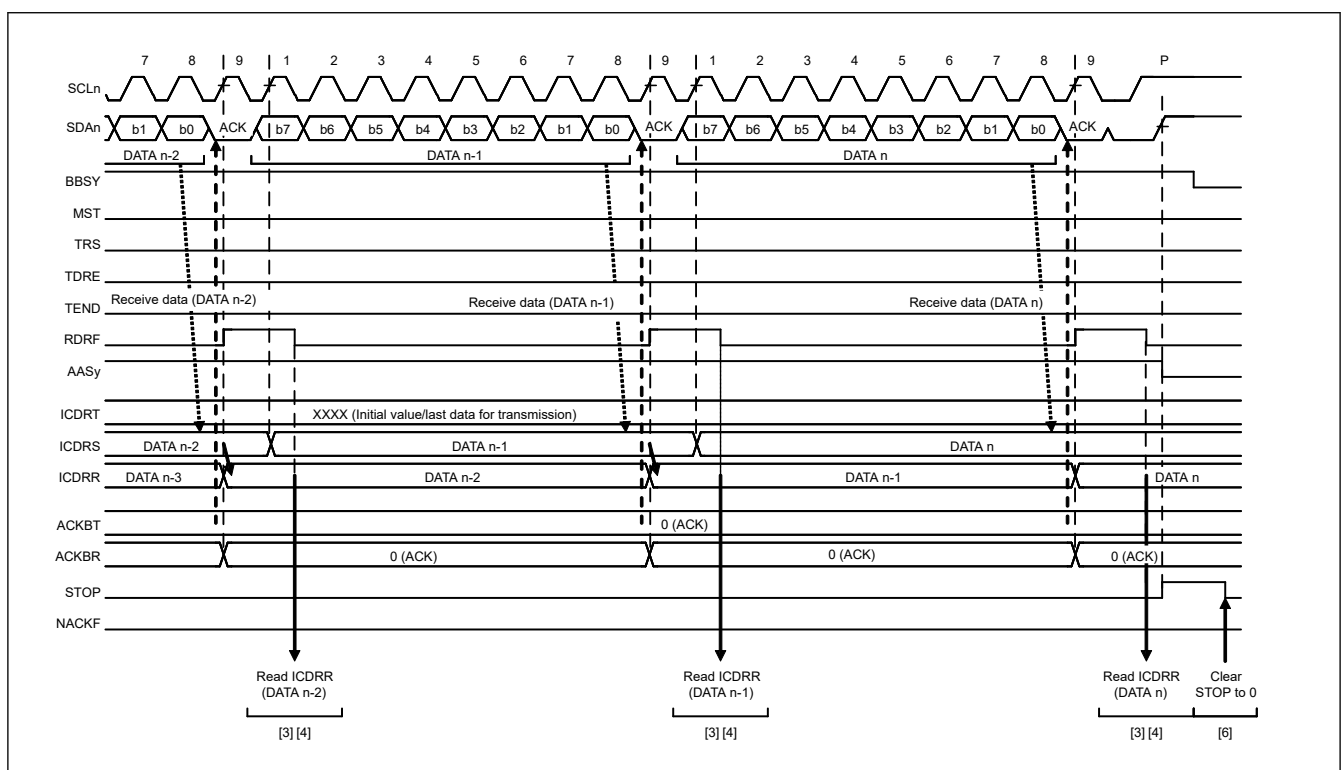


Figure 34.20 Slave receive operation timing (2) when RDRFS = 0

34.4 SCL Synchronization Circuit

For generation of the SCL clock, the IIC starts counting the value for the high-level period specified in ICBRH when it detects a rising edge on the SCLn line, and it drives the SCLn line low when it completes counting. When the IIC detects the falling edge of the SCLn line, it starts counting the value for the low-level period specified in ICBRL, and then it stops

driving the SCLn line, releasing the line, when it completes counting. The IIC repeats this process to generate the SCL clock.

If multiple master devices are connected to the I²C bus, a collision of SCL signals might arise because of contention with another master device. In such cases, the master devices must synchronize their SCL signals. Because this synchronization of SCL signals must be bit-by-bit, the IIC is equipped with an SCL synchronization circuit to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCLn line while in master mode.

When the IIC detects a rising edge on the SCLn line and starts counting the high-level period specified in ICBRH.BRH[4:0], and the level on the SCLn line falls because an SCL signal is being generated by another master device, the IIC performs the following:

1. Stops counting when it detects the falling edge.
2. Drives the level on the SCLn line low.
3. Starts counting the low-level period specified in ICBRL.BRL[4:0].

When the IIC finishes counting the low-level period, it stops driving the SCLn line low to release the line. If the low-level period of the SCL clock signal from the other master device is longer than the low-level period set in the IIC, the low-level period of the SCL signal is extended. When the low-level period for the other master device ends, the SCL signal rises because the SCLn line is released.

When the IIC finishes outputting the low-level period of the SCL clock, the SCLn line is released and the SCL clock rises. That is, when SCL signals from more than one master are contending, the high-level period of the SCL signal is synchronized with that of the clock with the narrower period, and the low-level period of the SCL signal is synchronized with that of the clock with the broader period. However, such synchronization of the SCL signal is only enabled when the SCLE bit in ICFER is set to 1.

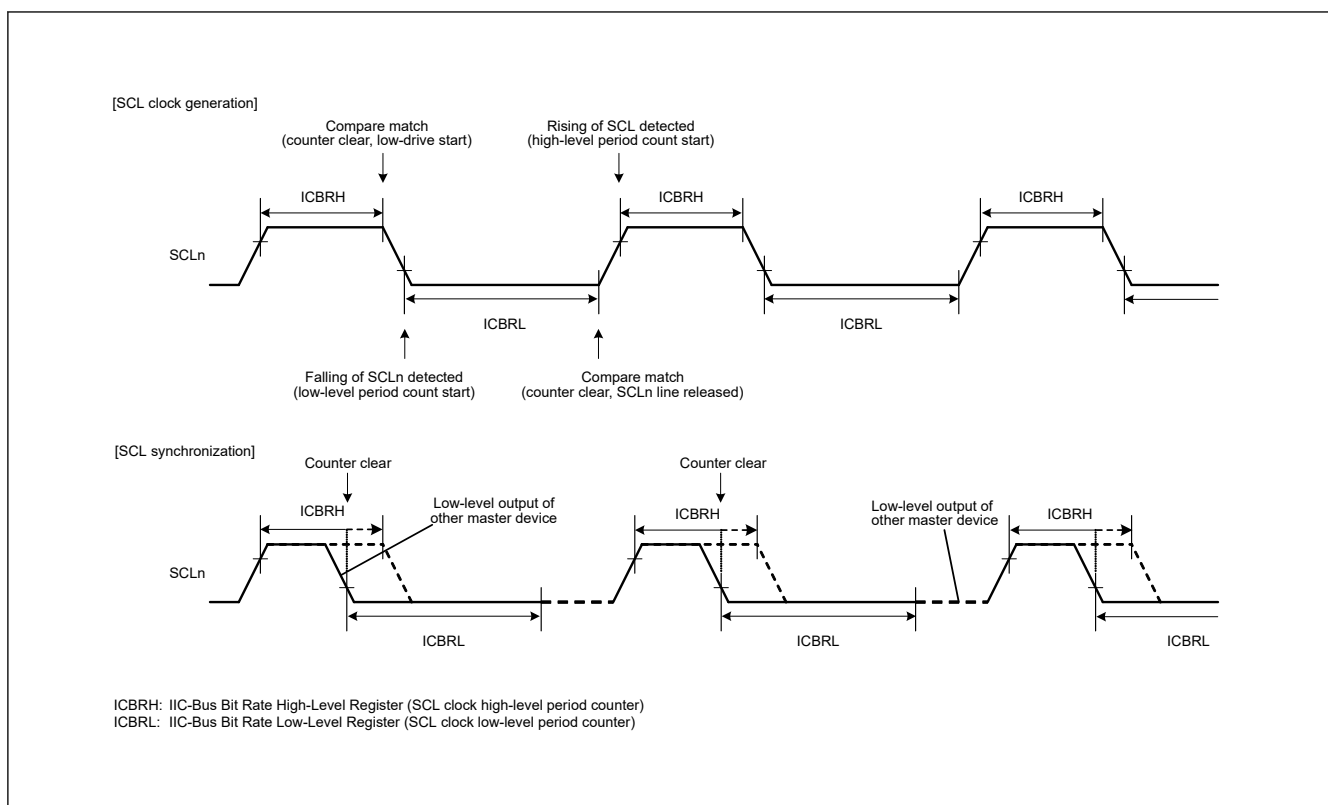


Figure 34.21 Generation and synchronization of SCL signal from IIC

34.5 SDA Output Delay Function

The IIC module incorporates a function for delaying output on the SDA line. The delay can be applied to all output on the SDA line, including issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals.

With this function, SDA output is delayed from the detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval during which the SCL clock is low. This approach helps prevent erroneous operation of communications devices, with the aim of satisfying the 300-ns minimum data-hold time requirement of the SMBus specification. The output delay function is enabled by setting the SDDL[2:0] bits in ICMR2 to a value other than 000b, and disabled by setting the same bits to 000b.

When the SDA output delay function is enabled, for example, the DLCS bit in ICMR2 selects the clock source for the SDA output delay counter, either as the internal base clock (IIC ϕ) for the IIC module or as the internal base clock divided by 2 (IIC ϕ /2). The counter counts the number of cycles set in the SDDL[2:0] bits in ICMR2. When the delay cycles count is reached, the IIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.

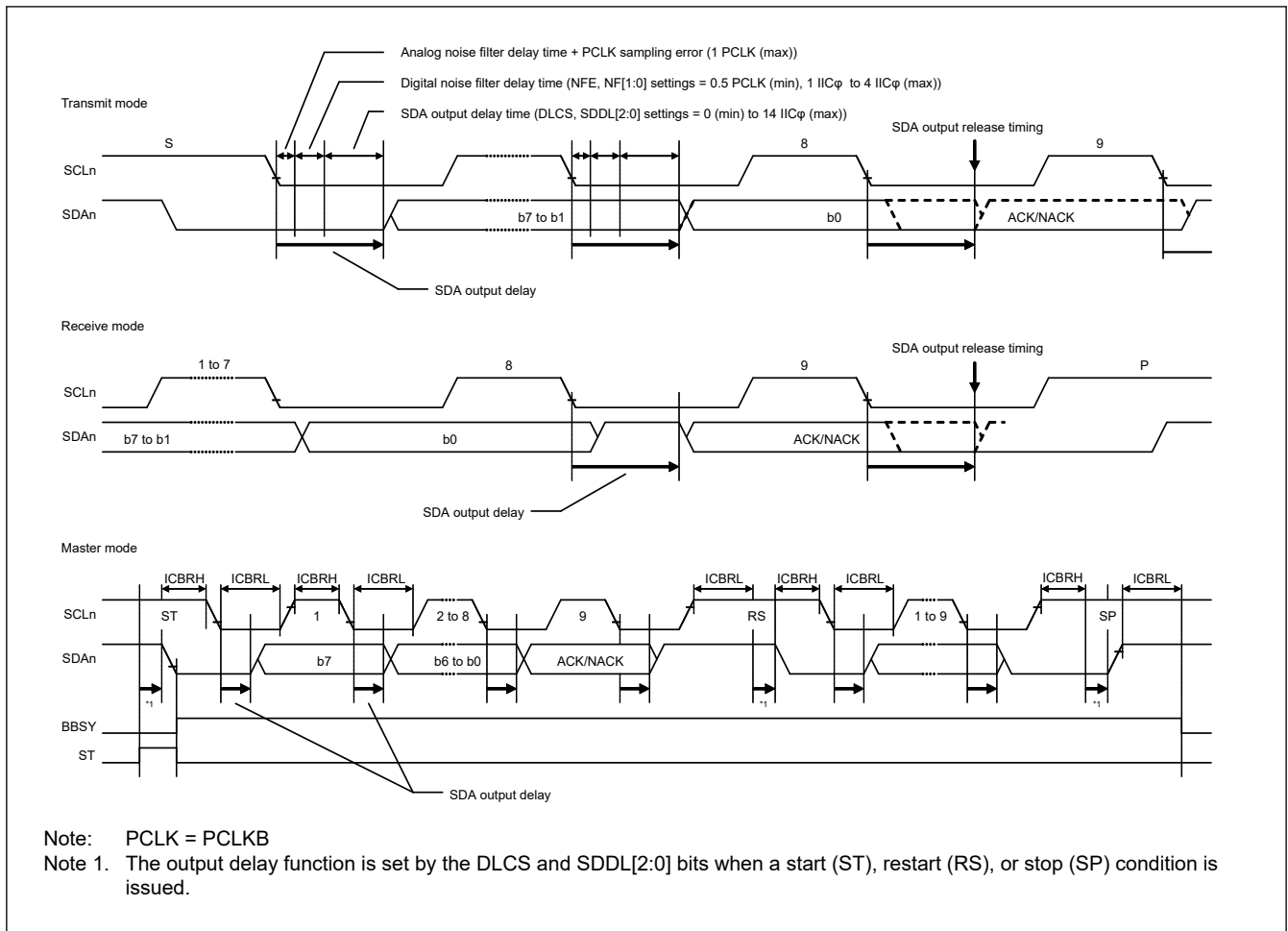


Figure 34.22 SDA output delay function

34.6 Digital Noise Filter Circuits

The internal circuitry sees the states of the SCLn and SDA pins through analog and digital noise-filter circuits. [Figure 34.23](#) 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 IIC ϕ cycles.

The input signal to the SCLn pin (or SDA pin) is sampled on falling edges of the IIC ϕ signal. When the input signal level matches the output level of the number of valid flip-flop circuit stages as selected in the NF[1:0] bits in ICMR3, the signal level is seen in the subsequent stage. If the signal levels do not match, the previous value is saved.

If the ratio between the frequency of the internal operating clock (PCLKB) and the transfer rate is small, for example, if data transfer is 400 kbps with PCLKB = 4 MHz, the characteristics of the digital noise filter might lead to the elimination of

required signals as noise. In such cases, it is possible to disable the digital noise-filter circuit by setting the ICFER.NFE bit to 0, and use only the analog noise filter circuit.

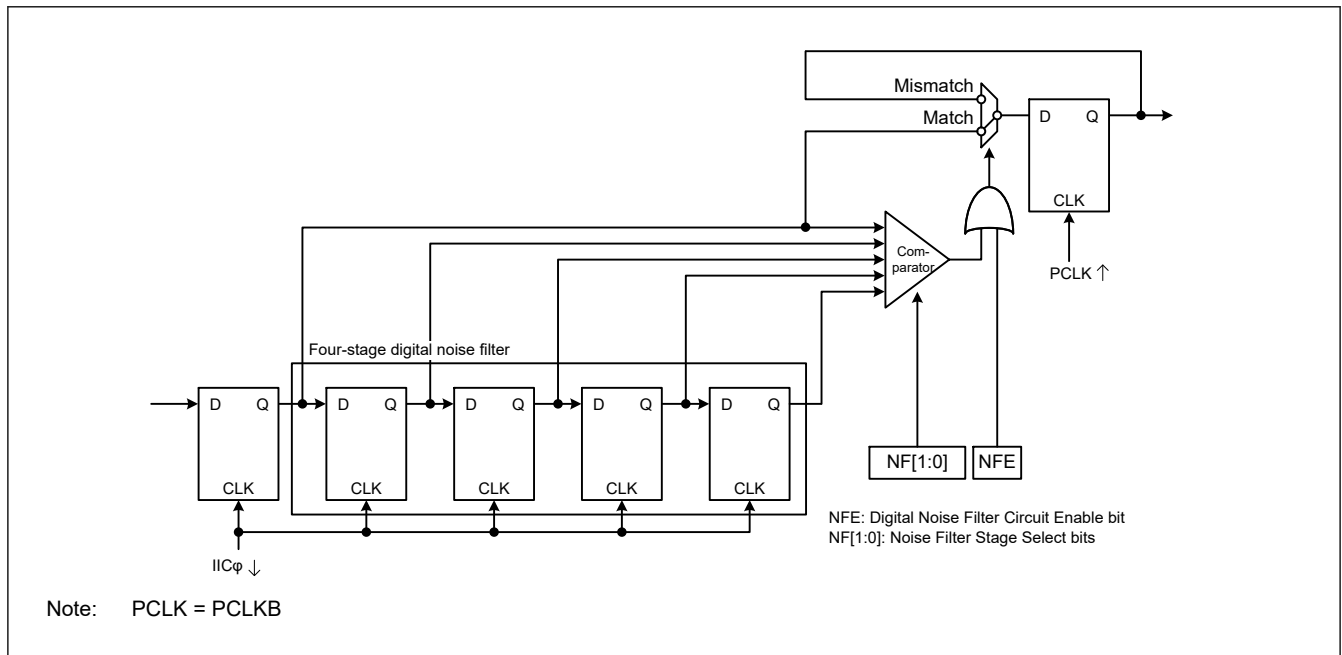


Figure 34.23 Digital noise filter circuit block diagram

34.7 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.

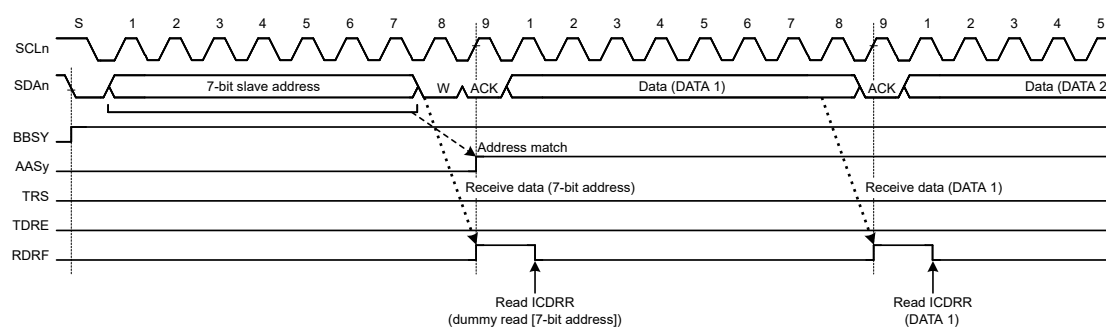
34.7.1 Slave-Address Match Detection

The IIC can set three unique slave addresses and has a slave address detection function for each unique slave address. When the SARyE bit ($y = 0$ to 2) in ICSER is set to 1, the slave addresses set in SARUy and SARLy ($y = 0$ to 2) can be detected.

When the IIC detects a match of the set slave address, the associated AASy flag ($y = 0$ to 2) in ICSR1 is set to 1 on the rising edge of the ninth SCL clock cycle, and the RDRF flag in ICSR2 or the TDRE flag in ICSR2 is set to 1 by the subsequent R/W# bit. This causes a receive data full interrupt (IICn_RXI) or transmit data empty interrupt (IICn_TXI) to be generated. The AASy flag identifies which slave address is specified.

Figure 34.24 to Figure 34.26 show the AASy flag set timing in three cases.

[7-bit address format: slave reception]



[7-bit address format: slave transmission]

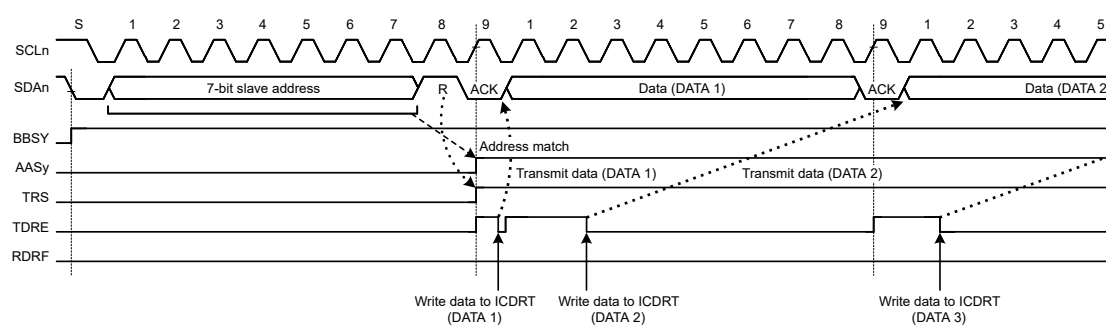
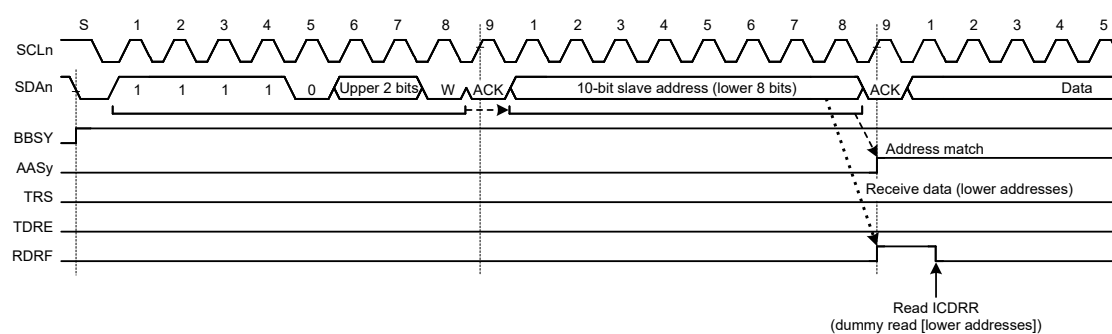


Figure 34.24 AASy flag set timing with 7-bit address format

[10-bit address format: slave reception]



[10-bit address format: slave transmission]

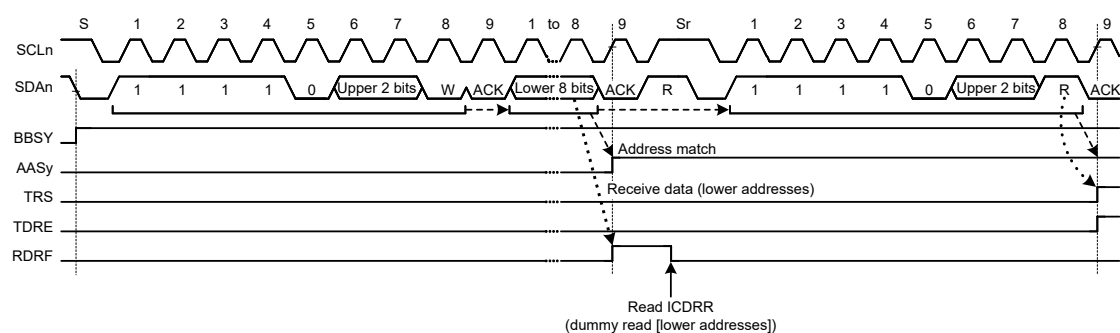


Figure 34.25 AASy flag set timing with 10-bit address format

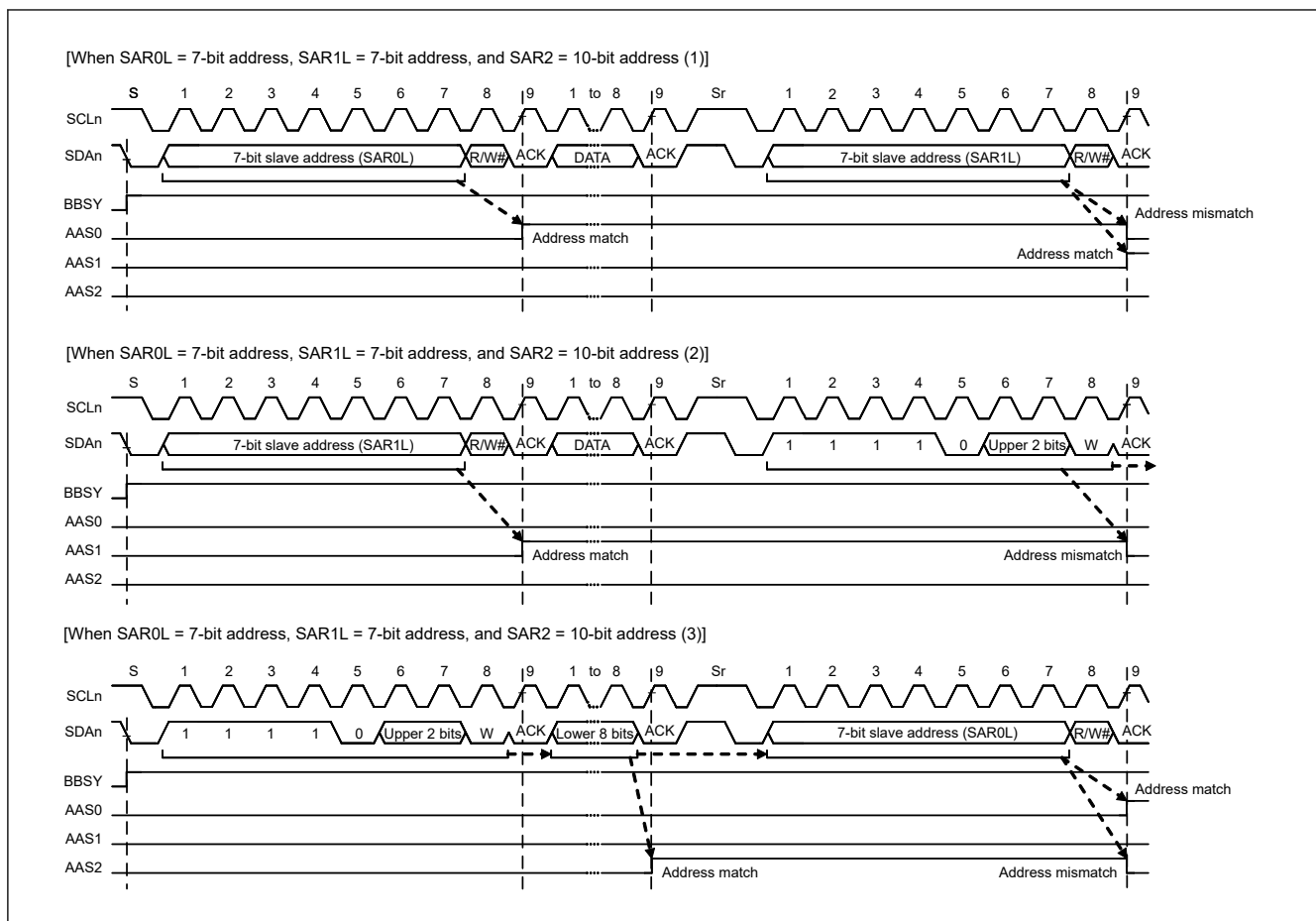


Figure 34.26 AASy flag set and clear timing with mixed 7-bit and 10-bit address formats

34.7.2 Detection of General Call Address

The IIC provides detection of the general call address (0000 000b + 0 [W]). This is enabled by setting the GCAE bit in IC SER to 1.

If the address received after a start or restart condition is issued is 0000 000b + 1[R] (start byte), the IIC recognizes this as the address of a slave device with an all-zero address, but not as the general call address.

When the IIC detects the general call address, both the GCA flag in ICSR1 and the RDRF flag in ICSR2 set to 1 on the rising edge of the ninth cycle of the SCL clock. This leads to the generation of a receive data full interrupt (IICn_RXI). The value of the GCA flag can be checked to confirm that the general call address was transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

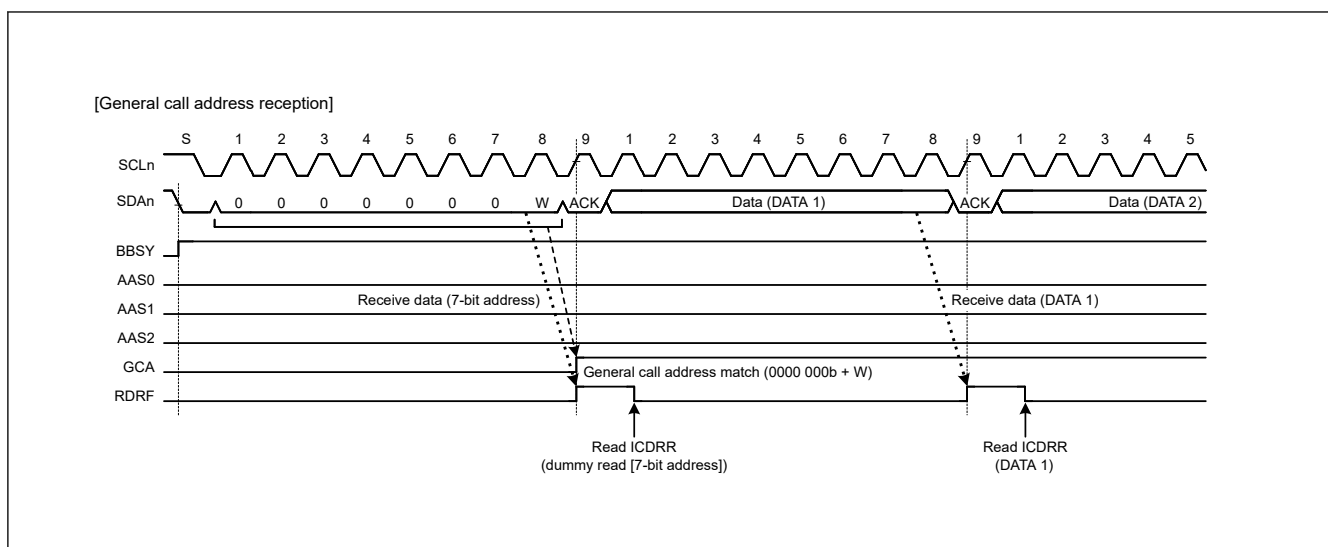


Figure 34.27 Timing of GCA flag setting during reception of general call address

34.7.3 Device-ID Address Detection

The IIC module provides detection of device-ID address compliant with the I²C bus specification (revision 03). When the IIC receives 1111 100b as the first byte after a start or restart condition is issued with the DIDE bit in ICSR 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 eighth SCL clock cycle when the subsequent R/W# bit is 0, then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the IIC sets the associated AASy flag (y = 0 to 2) in ICSR1 to 1.

When the first byte received after the issue of a start or restart condition matches the device ID address (1111 100b) again and the subsequent R/W# bit is 1, the IIC does not compare the second and subsequent bytes and sets the ICSR2.TDRE flag to 1.

In the device ID address detection function, the IIC sets the DID flag to 0 if a match with the IIC slave address is not obtained or a match with the device ID address is not obtained after a match with the IIC slave address and a restart condition is not detected. If the first byte after detection of a start or restart condition matches the device ID address (1111 100b), and the R/W# bit is 0, the IIC sets the DID flag to 1 and compares the second and subsequent bytes with the slave address of the IIC. If the R/W# bit is 1, the DID flag holds the previous value and the IIC does not compare the second and subsequent bytes. Therefore, the reception of a device ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Additionally, prepare the device ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal transmit data. For details on the information that must be included in device ID fields, contact NXP Semiconductors.

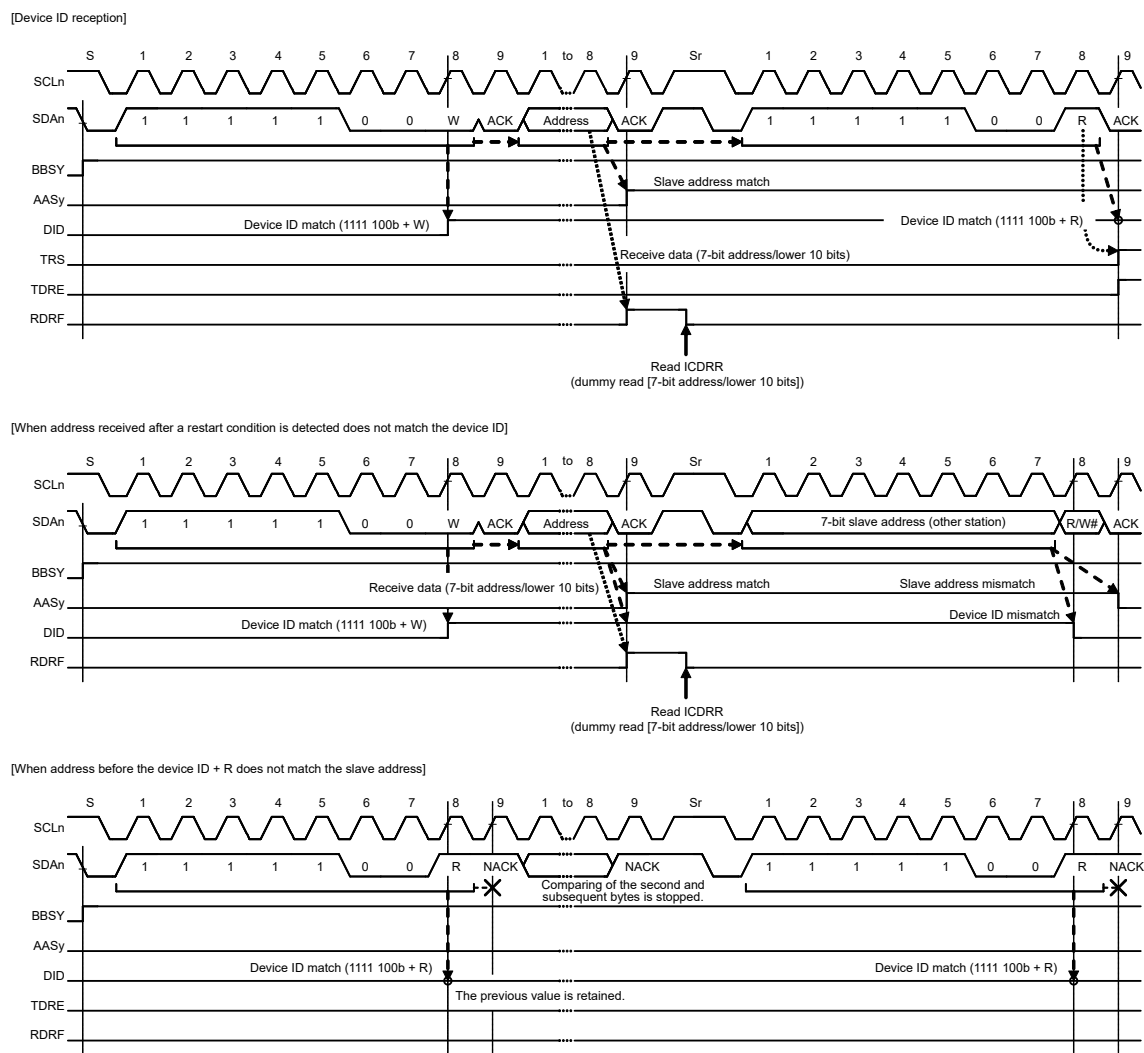


Figure 34.28 AASy and DID flag set and clear timing during reception of device ID

34.7.4 Host Address Detection

The IIC provides host address detection when operating in SMBus. When the HOAE bit in ICSR is set to 1 while the SMBS bit in ICMR3 is 1, the IIC can detect the host address (0001 000b) in slave receive mode (MST and TRS bits = 00b in ICCR2).

When the IIC detects the host address, the HOA flag in ICSR1 is set to 1 on the rising edge of the 9th SCL clock cycle, and at the same time, the RDRF flag in ICSR2 is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (IICn_RXI) to be generated. The HOA flag indicates that the host address was sent from another device.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit = 1), the IIC can also detect the host address. After the host address is detected, the IIC operates in the same manner as in normal slave operation.

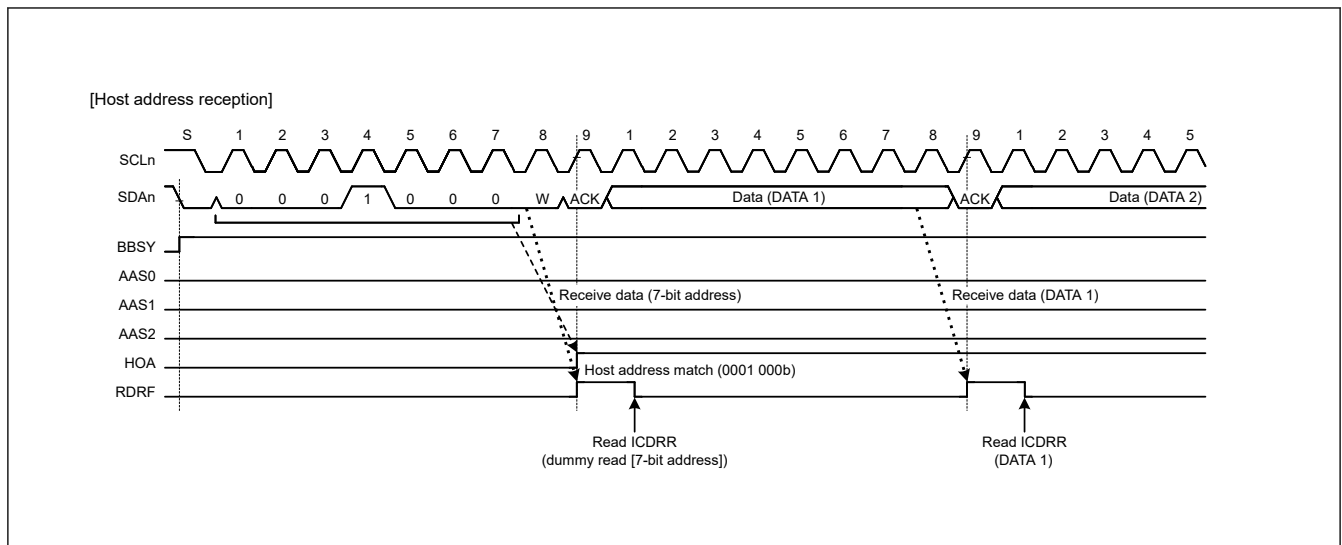


Figure 34.29 HOA flag set timing during reception of host address

34.8 Automatic Low-Hold Function for SCL

34.8.1 Function to Prevent Wrong Transmission of Transmit Data

If the I²C Bus Shift Register (ICDRS) is empty and data has not been written to the I²C Bus Transmit Data Register (ICDRT) with the IIC in transmission mode (TRS bit = 1 in ICCR2), the SCLn line is automatically held low over the subsequent intervals. This low-hold period is extended until the transmit data is written, which prevents the unintended transmission of erroneous data.

Master transmit mode:

- Low-level interval after a start or restart condition is issued
- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

Slave transmit mode:

- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

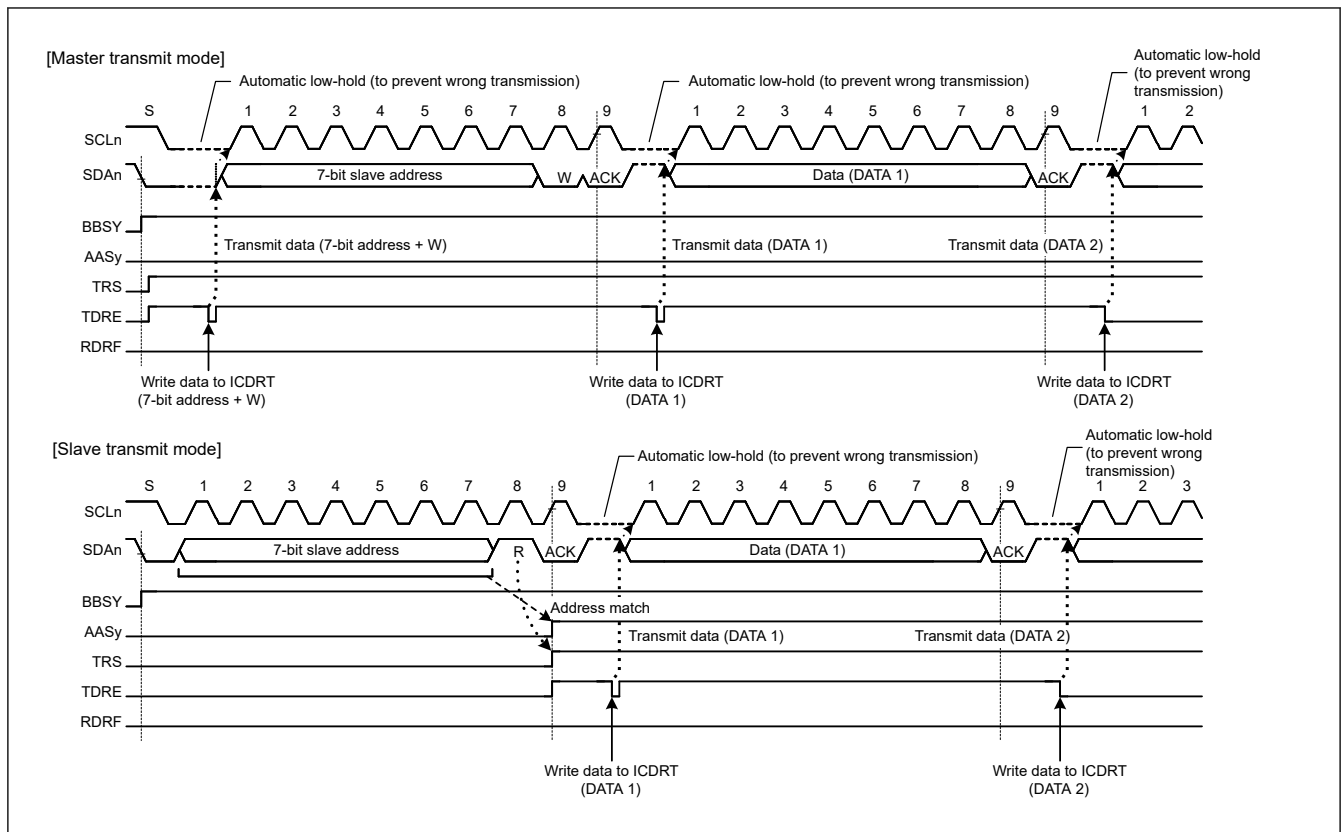


Figure 34.30 Automatic low-hold operation in transmit mode

34.8.2 NACK Reception Transfer Suspension Function

This function suspends transfer operation when NACK is received in transmit mode (TRS bit = 1 in ICCR2). This function is enabled when the NACKE bit in ICFER is set to 1. If the next transmit data is already written (TDRE flag = 0 in ICSR2) when NACK is received, the next data transmission on the falling edge of the 9th SCL clock cycle is automatically suspended. This prevents the SDAn line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (NACKF flag = 1 in ICSR2), transmit and receive operations are discontinued. To restore transmit or receive operation, you must 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.

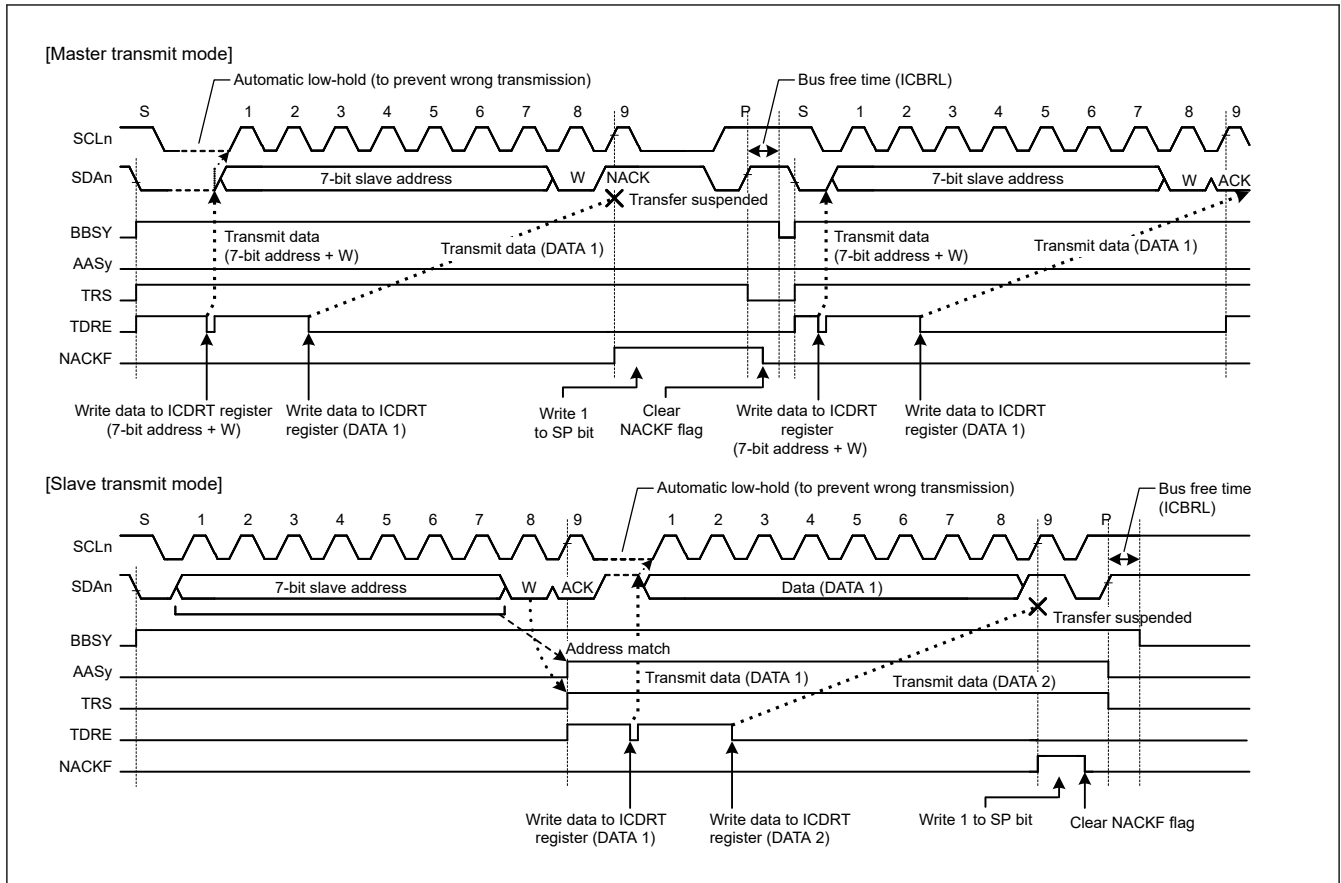


Figure 34.31 Suspension of data transfer when NACK is received, when NACK = 1

34.8.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRT) read is delayed for a period of one transfer frame or more with receive data full (RDRF flag = 1 in ICSR2) in receive mode (TRS = 0 in ICCR2), the IIC holds the SCLn line low automatically immediately before the next data is received to prevent a failure to receive data.

This function is enabled even if the read processing of the final receive data is delayed and, in the meantime, the IIC slave address is designated after a stop condition is issued. This function does not interfere with other communication because the IIC does not hold the SCLn line low when a mismatch with its own slave address occurs after a stop condition is issued.

Periods in which the SCLn line is held low can be selected with a combination of the WAIT and RDRFS bits in ICMR3.

(1) 1-byte receive operation and automatic low-hold function using the WAIT bit

When the WAIT bit in ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the WAIT bit function. Additionally, when the ICMR3.RDRFS bit is 0, the IIC automatically sends the ACKBT bit value in ICMR3 for the acknowledge bit in the period from the falling edge of the 8th SCL clock cycle to the falling edge of the 9th SCL clock cycle, and automatically holds the SCLn line low on the falling edge of the 9th SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from ICDRT, which enables byte-wise receive operation.

The WAIT bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

(2) 1-byte receive operation (ACK/NACK transmission control) and automatic low-hold function using the RDRFS bit

When the RDRFS bit in ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the RDRF flag in ICSR2 is set to 1 (receive data full) on the rising edge of the eighth SCL clock cycle, and the SCLn line is automatically held low on the falling edge of the eighth SCL clock cycle. This low-hold is released by writing a value to the ACKBT bit in ICMR3, but cannot be released by reading data from ICDRT, which enables receive operation through the ACK or NACK transmission control based on the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

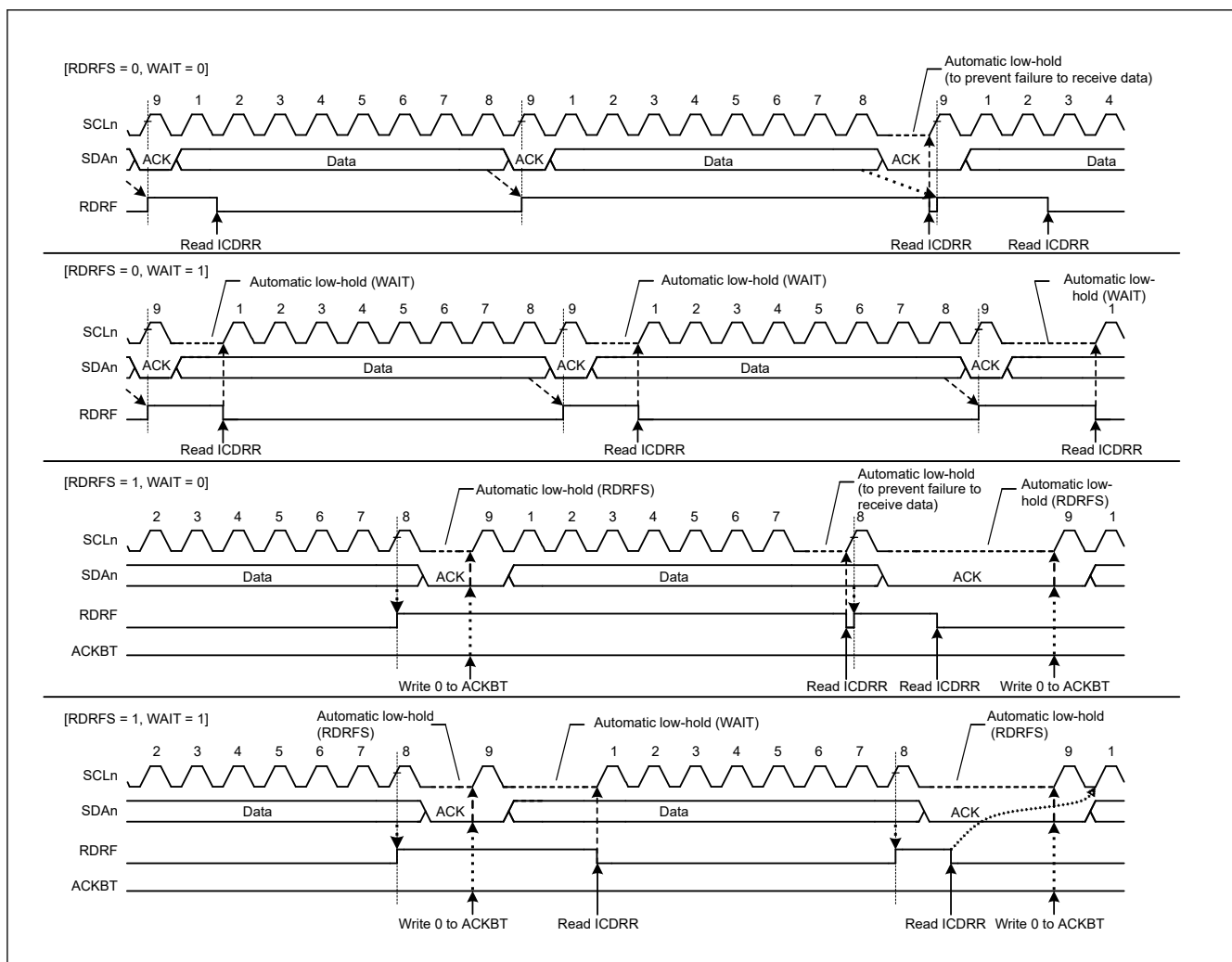


Figure 34.32 Automatic low-hold operation in receive mode using the RDRFS and WAIT bits

34.9 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.

34.9.1 Master Arbitration-Lost Detection (MALE Bit)

The IIC drives the SDAn line low to issue a start condition. However, if the SDAn line was already driven low by another master device issuing a start condition, the IIC regards its own start condition as an error and considers this a loss in arbitration. Priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the ST bit in ICCR2 to 1 while the bus is busy (BBSY flag = 1 in ICCR2), the IIC regards this as a double-issuing-of-start-condition error and considers itself to have lost in arbitration. This prevents a failure of transfer resulting from a start condition being issued while transfer is in progress.

When a start condition is issued successfully, if the transmit data including the address bits (internal SDA output level) and the level on the SDAn line do not match (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state) and a low level is detected on the SDAn line, the IIC loses in arbitration.

After a loss in arbitration of mastership, the IIC immediately enters slave receive mode. If a slave address, including the general call address, matches its own address at this time, the IIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the MALE bit in ICFER is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Mismatching of the internal level for output on SDA and the level on the SDA_n line after a start condition was issued by setting the ST bit in ICCR2 to 1 while the BBSY flag in ICCR2 is set to 0 (erroneous issuing of a start condition)
- Setting of the ST bit in ICCR2 to 1 (start condition double-issue error) while the BBSY flag is 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDA_n line in master transmit mode (MST and TRS bits = 11b in ICCR2).

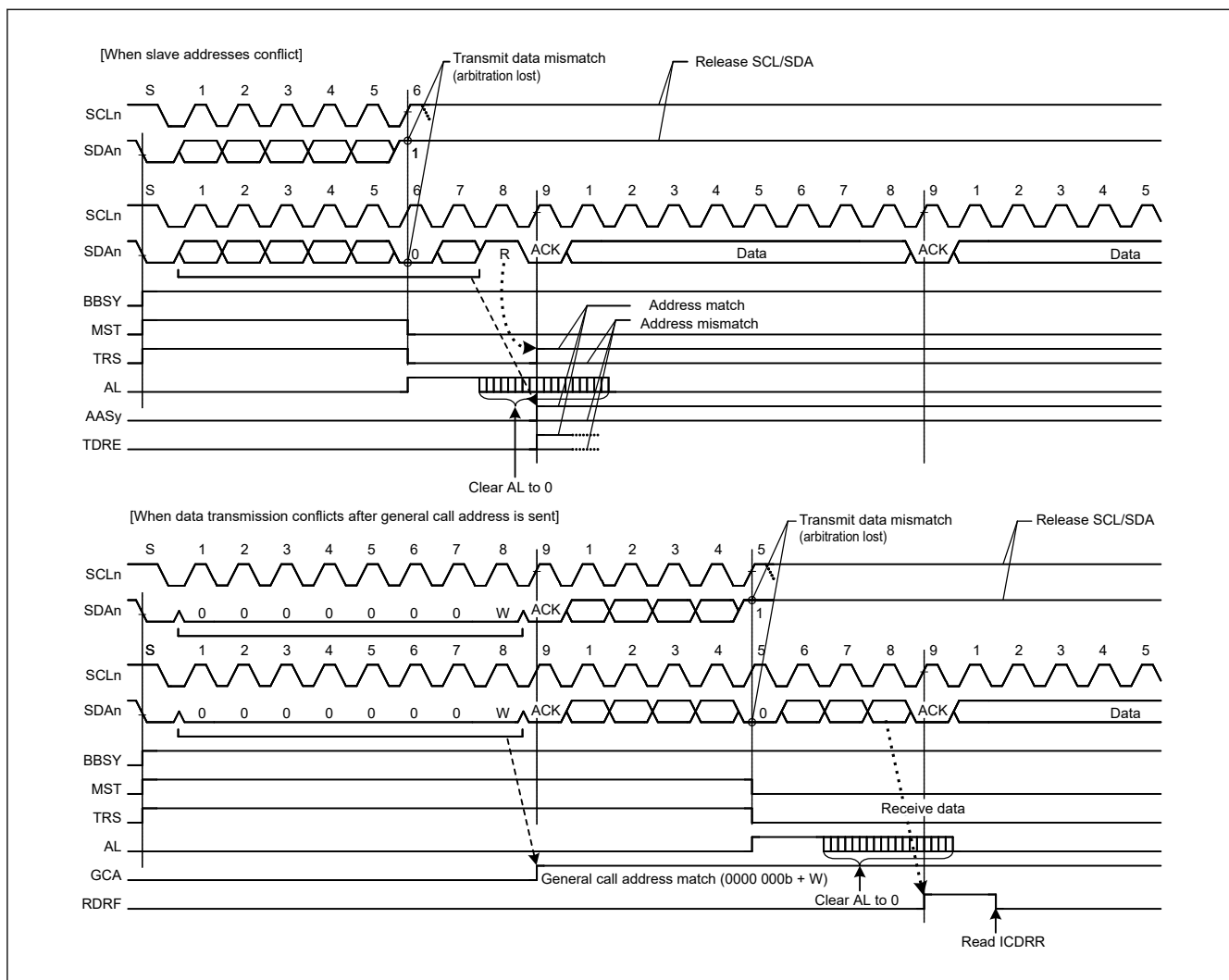


Figure 34.33 Examples of master arbitration-lost detection when MALE = 1

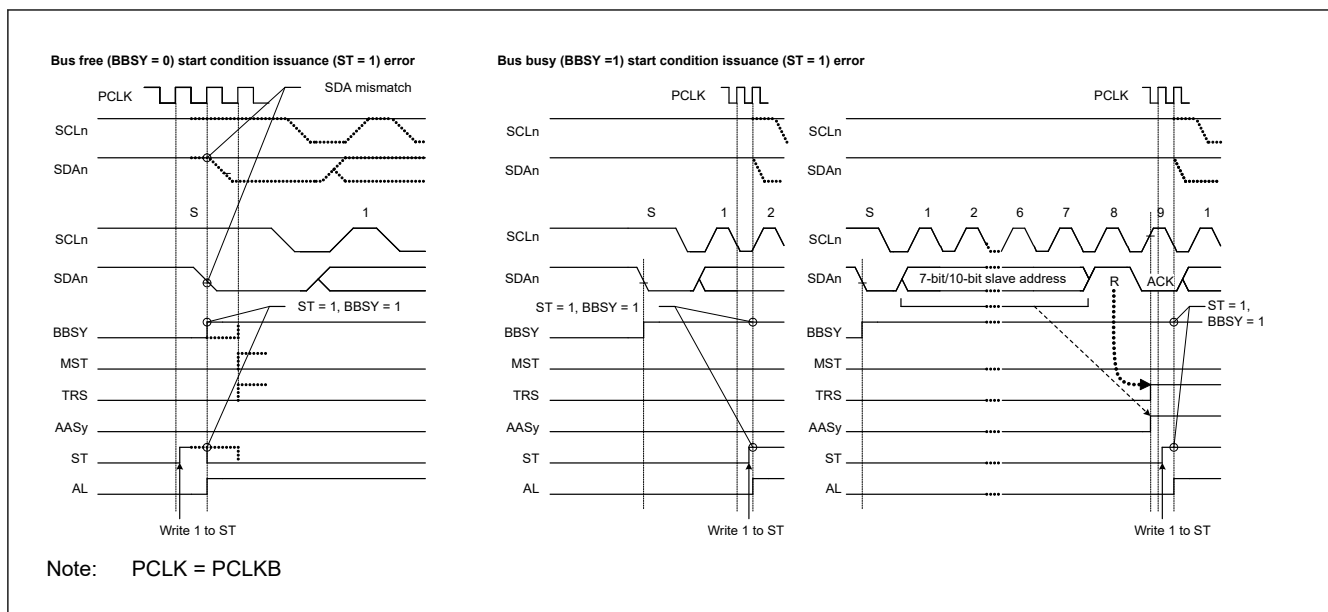


Figure 34.34 Arbitration-lost when start condition is issued when MALE = 1

34.9.2 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 SDA_n line (high output as the internal SDA output, meaning the SDA_n pin is in the high-impedance state) and the low level is detected on the SDA_n line during transmission of NACK in receive mode. Arbitration is lost because of a conflict between NACK and ACK transmissions when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such a conflict occurs when multiple master devices send or receive the same information through a single slave device. Figure 34.35 shows an example of arbitration-lost detection during transmission of NACK.

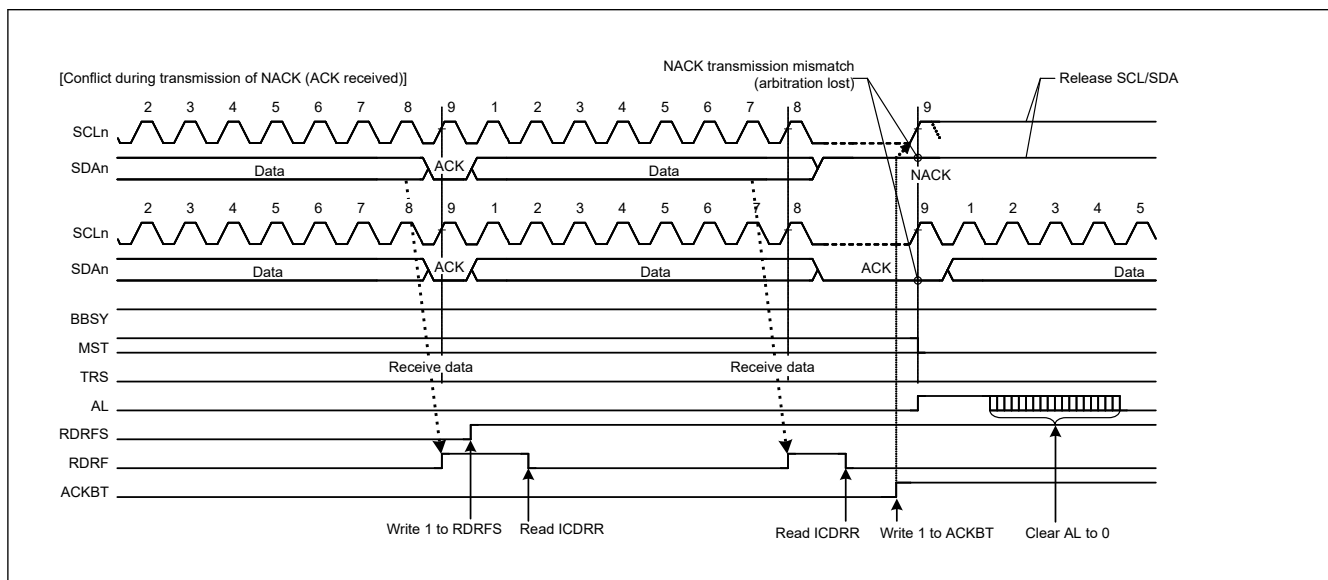


Figure 34.35 Example of arbitration-lost detection during transmission of NACK when NALE = 1

The following explains arbitration-lost detection using an example in which two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives 2 bytes of data from the slave device, and master B receives 4 bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in either master A or master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Master A sends NACK when it has received 2 final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received the required 4 bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like

this occurs, master A cannot detect the ACK transmitted by master B and issues a stop condition. The stop condition issue conflicts with the SCL clock output of master B, which disrupts communication.

When the IIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost. If arbitration is lost during transmission of NACK, the IIC immediately cancels the slave match condition and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing, such as 0xFF transmission processing, which is required if the UDID (Unique Device Identifier) of the assigned address does not match in the Get UDID general processing after the Assign Address command.

The IIC detects arbitration-lost during transmission of NACK when the following condition is met with the NALE bit in ICFER set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the SDA_n line (ACK is received) during transmission of NACK (ACKBT bit = 1 in ICMR3).

34.9.3 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 SDA_n line do not match (high output as the internal SDA output, meaning the SDA_n pin is in the high-impedance state), and the low level is detected on the SDA_n line in slave transmit mode. This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

When the IIC loses slave arbitration, the IIC is immediately released from the slave-matched state and enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminates subsequent redundant processing for the transmission of 0xFF.

The IIC detects slave arbitration-lost when the following condition is met with the SALE bit in ICFER set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When transmit data excluding acknowledge (internal SDA output level) does not match the SDA_n line in slave transmit mode (MST and TRS bits = 01b in ICCR2).

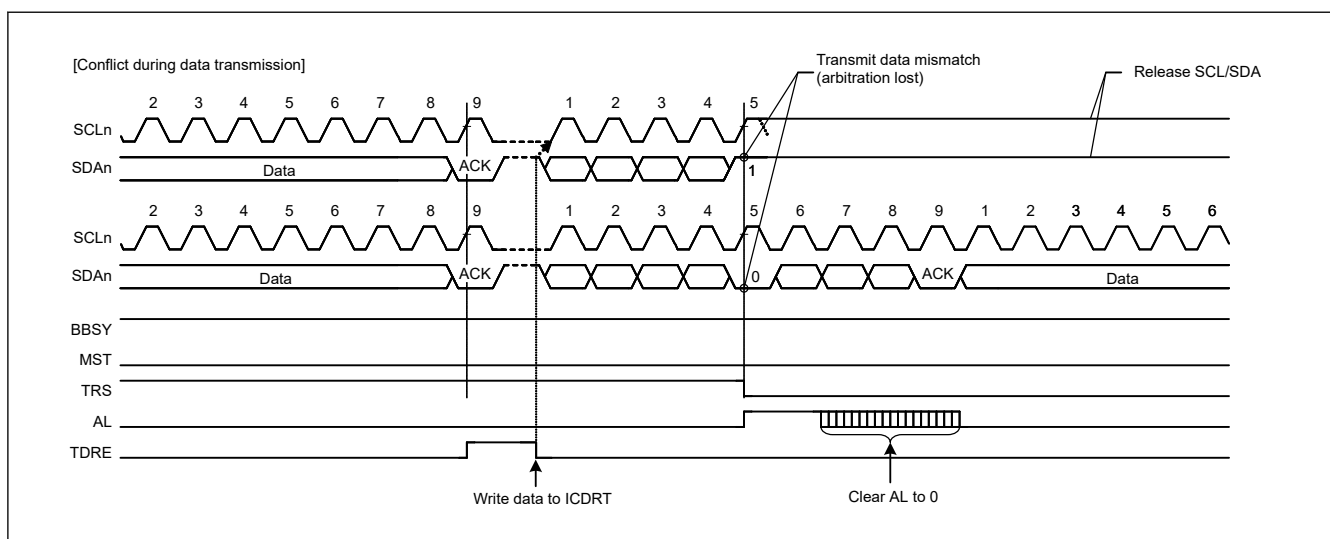


Figure 34.36 Example of slave arbitration-lost detection when SALE = 1

34.10 Start, Restart, and Stop Condition Issuing Function

34.10.1 Issuing a Start Condition

The IIC issues a start condition when the ST bit in ICCR2 is set to 1. When the ST bit is set to 1, a start condition request is made, and the IIC issues a start condition when the BBSY flag in ICCR2 is 0 (bus free state). When a start condition is issued normally, the IIC automatically shifts to the master transmit mode.

To issue a start condition:

1. Drive the SDA_n line low (high level to low level).
2. Ensure that the time set in ICBRH and the start condition hold time elapse.
3. Drive the SCL_n line low (high level to low level).
4. Detect low level on the SCL_n line and ensure the low-level period of the SCL_n line set in ICBRL elapses.

34.10.2 Issuing a Restart Condition

The IIC issues a restart condition when the RS bit in ICCR2 is set to 1. When the RS bit is set to 1, a restart condition request is made, and the IIC issues a restart condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a restart condition:

1. Release the SDA_n line.
2. Ensure the low-level period of the SCL_n line set in ICBRL elapses.
3. Release the SCL_n line (low level to high level).
4. Detect a high level on the SCL_n line and ensure the time set in ICBRL and the restart condition setup time elapse.
5. Drive the SDA_n line low (high level to low level).
6. Ensure the time set in ICBRH and the restart condition hold time elapse.
7. Drive the SCL_n line low (high level to low level).
8. Detect a low level on the SCL_n line and ensure the low-level period of the SCL_n line set in ICBRL elapses.

Note: When issuing restart condition requests, write the slave address to ICDRT after confirming that ICCR2.RS = 0. Data written while ICCR2.RS = 1 is not forwarded because of the retransmission condition before the occurrence.

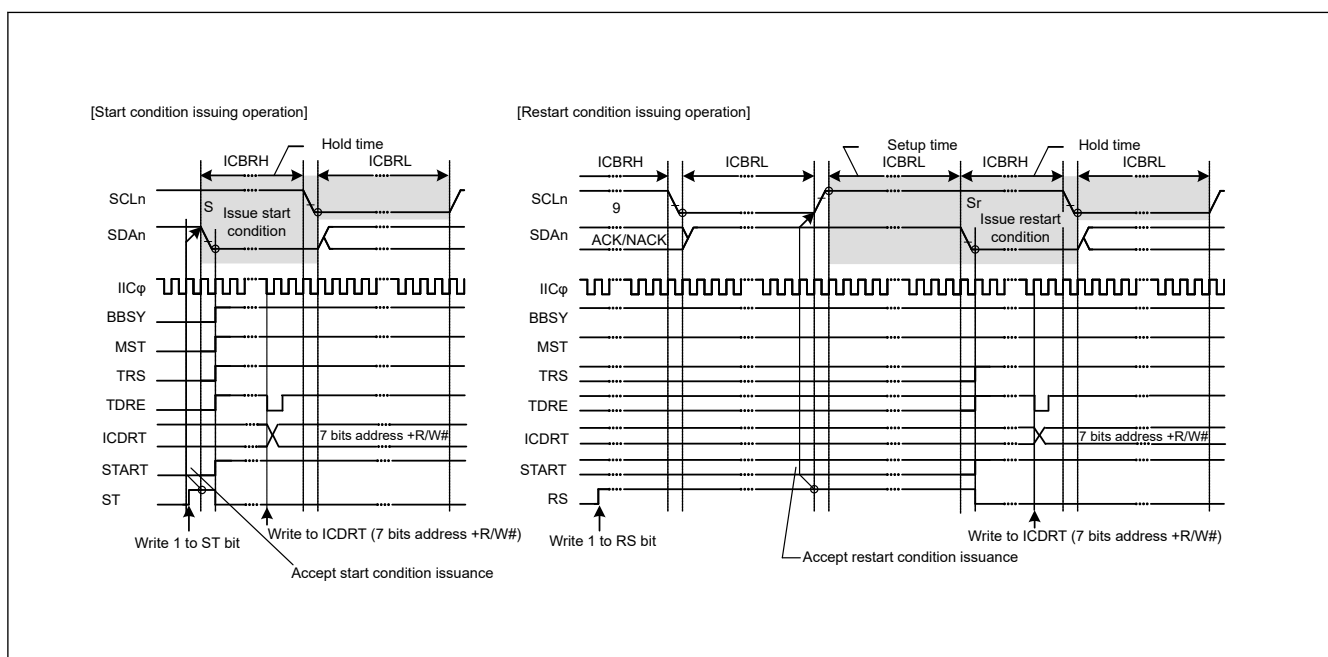


Figure 34.37 Start and restart condition issue timing using the ST and RS bits

Figure 34.38 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 34.3.2. Initial Settings](#).
2. Read the BBSY flag in IICR2 to check that the bus is open, then set the ST bit in ICCR2 to 1 (start condition issuance request). On receiving the request, the IIC issues a start condition. At the same time, the BBSY and the START flags in ICSR2 are automatically set to 1 and the ST bit is automatically set to 0. If the start condition is detected and the internal levels for the SDA output state and the levels on the SDA_n line match while the ST bit is 1, the IIC recognizes that a start condition is successfully issued as requested by the ST bit has been successfully completed. The MST and TRS bits in ICCR2 are automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 when the TRS bit is set to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. After the data for transmission is written to ICDRT, the TDRE flag is automatically set to 0, data is transferred from ICDRT to ICDRS, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode according to the value of the transmitted R/W# bit. If the value of the R/W# bit is 0, the IIC continues in master transmit mode. If the ICSR2.NACKF flag is 1 at this time, indicating that no slave device recognized the address or that there was an error in communications, write 1 to ICCR2.SP bit to issue a stop condition. To transmit data with an address in the 10-bit format, start by writing 1111 0b, the 2 upper bits of the slave address, and W to ICDRT as the first address transmission. Then, as the second address transmission, write the 8 lower bits of the slave address to ICDRT.
4. After confirming that the TDRE flag in ICSR2 is 1, write data for transmission to the ICDRT register. The IIC automatically holds the SCL_n line low until data for transmission is ready, a restart condition is issued or a stop condition is issued.
5. After all bytes of data for transmission are written to the ICDRT register, wait until the value of the TEND flag in ICSR2 returns to 1. Then after checking that the START flag in ICSR2 is 1, set the START flag in ICSR2 to 0.
6. Set the RS bit in ICCR2 to 1 (restart condition issue request). On receiving the request, the IIC issues a restart condition.
7. After checking that the START flag in ICSR2 is 1, write the value for transmission (the slave address and the R/W# bit) to ICDRT.

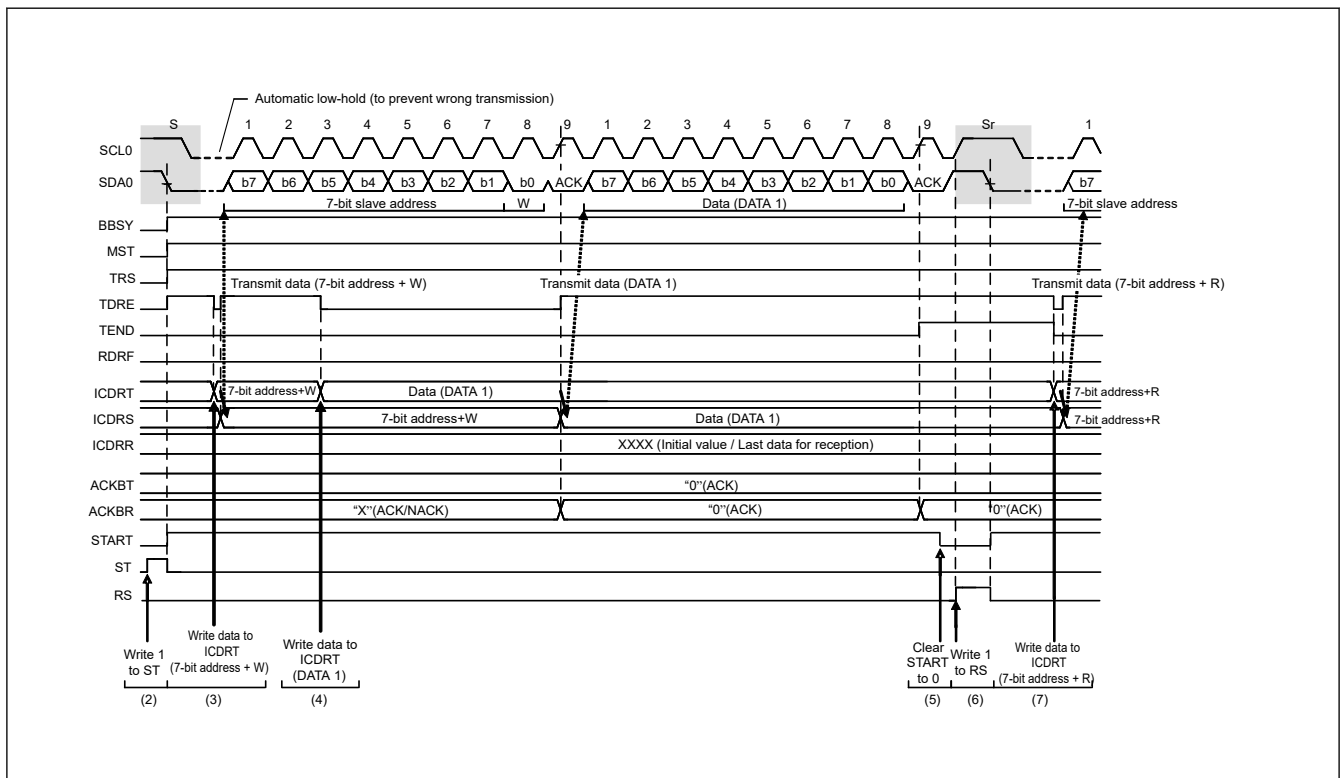


Figure 34.38 Restart condition issue timing after master transmission.

34.10.3 Issuing a Stop Condition

The IIC issues a stop condition when the SP bit in ICCR2 is set to 1. When the SP bit is set to 1, a stop condition request is made, and the IIC issues a stop condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a stop condition:

1. Drive the SDA_n line low (high level to low level).
2. Ensure the low-level period of the SCL_n line set in ICBRL elapses.
3. Release the SCL_n line (low level to high level).
4. Detect a high level on the SCL_n line and ensure the time set in ICBRH and the stop condition setup time elapse.
5. Release the SDA_n line (low level to high level).
6. Ensure the time set in ICBRL and the bus free time elapse.
7. Clear the BBSY flag to 0 to release the bus mastership.

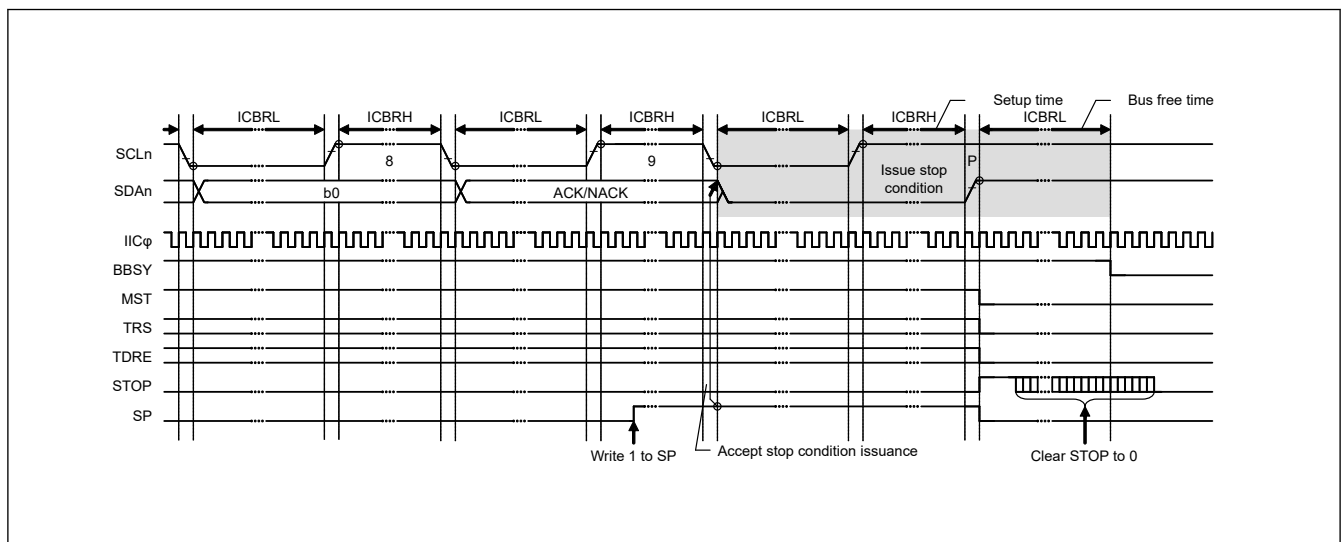


Figure 34.39 Stop condition issue timing using the SP bit

34.11 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 SCL_n line or SDA_n line.

To manage bus hanging, the IIC has a timeout function to detect hanging by monitoring the SCL_n line, and a function for outputting an extra SCL clock cycle to release the bus from:

- A timeout function to detect hanging by monitoring the SCL_n line
- The IIC reset function
- An internal reset function.

By checking the SCLO, SDAO, SCLI, and SDAI bits in ICCR1, it is possible to see whether the IIC or its communicating partner is placing the low level on the SCL_n or SDA_n line.

34.11.1 Timeout Function

The timeout function can detect when the SCL_n line is stuck longer than the predetermined time. The IIC can detect an abnormal bus state by monitoring that the SCL_n line is stuck low or high for a predetermined time.

The timeout function monitors the SCL_n line state and counts the low- or high-level period using the internal counter. The timeout function resets the internal counter each time the SCL_n line changes (rises or falls), but continues to count unless the SCL_n line changes. If the internal counter overflows because no SCL_n line changes, the IIC can detect the timeout and report the bus hung state.

This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state when the SCLn line is stuck low or high during the following conditions:

- The bus is busy (ICCR2.BBSY flag is 1) in master mode (ICCR2.MST bit is 1)
- The IIC slave address is detected (ICSR1 register is not 0x00) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0)
- The bus is open (ICCR2.BBSY flag is 0) while a start condition is requested (ICCR2.ST bit is 1).

The internal counter of the timeout function uses the internal reference clock (IIC ϕ) set in the CKS[2:0] bits in ICMR1 as a count source. It functions as a 16-bit counter when long mode is selected (TMOS bit = 0 in ICMR2) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCLn line level (low, high, or both levels) during which this counter is activated can be selected in the TMOH and TMOL bits in ICMR2. If both TMOL and TMOH bits are set to 0, the internal counter is disabled.

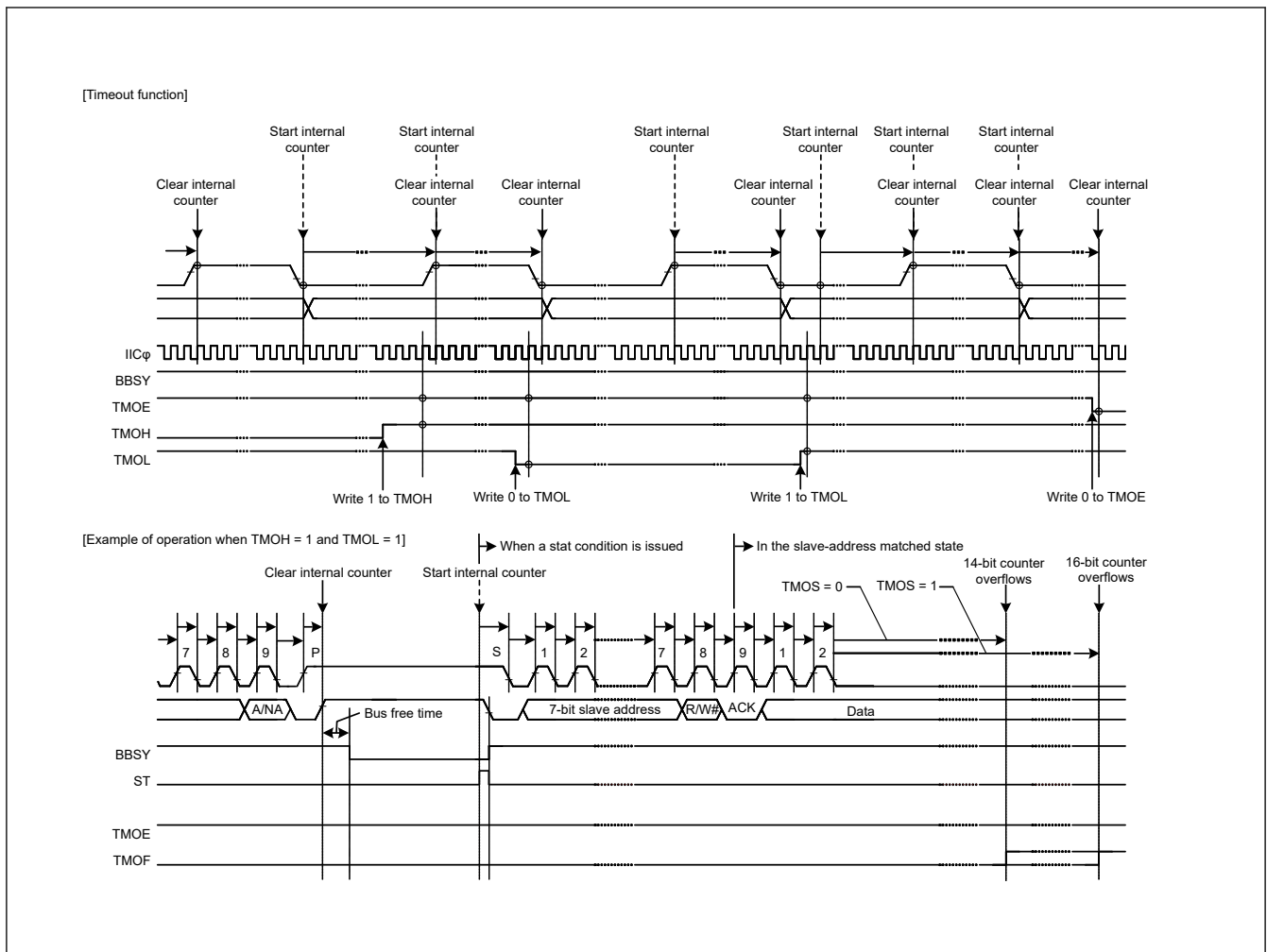


Figure 34.40 Timeout function using the TMOE, TMOS, TMOH, and TMOL bits

34.11.2 Extra SCL Clock Cycle Output Function

In master mode, this function outputs extra SCL clock cycles to release the SDAn line of the slave device from being held low because the master is out of synchronization with the slave device. This function is mainly used in master mode to release the SDAn line of the slave device from being fixed low by including extra cycles of SCL output from the IIC. It uses single cycles of the SCL clock for a bus error where the IIC cannot issue a stop condition because the slave device is holding the SDAn line at the low level. Do not use this function in normal situations. Using it when communications are proceeding correctly leads to malfunctions.

When the CLO bit in ICCR1 is set to 1 in master mode, a single cycle of the SCL clock at the transfer rate specified in the CKS[2:0] bits in ICMR1, and in the ICBRH and ICBRL registers, is output as an extra clock cycle. After output of this

single cycle of the SCL clock, the CLO bit is automatically set to 0. At this time, if ICCR2.BBSY = 1, the SCL pin continues to output low, and when ICCR2.BBSY = 0, the SCL pin goes high. After confirming that the CLO bit is 0 by software, write 1 to the CLO bit to output the additional clock continuously.

When the IIC module is in master mode and the slave device is holding the SDAn line low because synchronization with the slave device is lost because of effects like noise, the output of a stop condition is not possible. This function can be used to output extra cycles of SCL one by one to make the slave device release the SDAn line from being held low, and so recover the bus from an unusable state. Release of the SDAn line by the slave device can be monitored by reading the SDAI bit in ICCR1. After confirming the release of the SDAn line by the slave device, complete communications by reissuing the stop condition.

[Output conditions for using the CLO bit in ICCR1]

- When the bus is open (BBSY flag in ICCR2 = 0) or in master mode (MST bit = 1 and BBSY flag = 1 in ICCR2)
- When the communication device does not hold the SCLn line low.

Figure 34.41 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

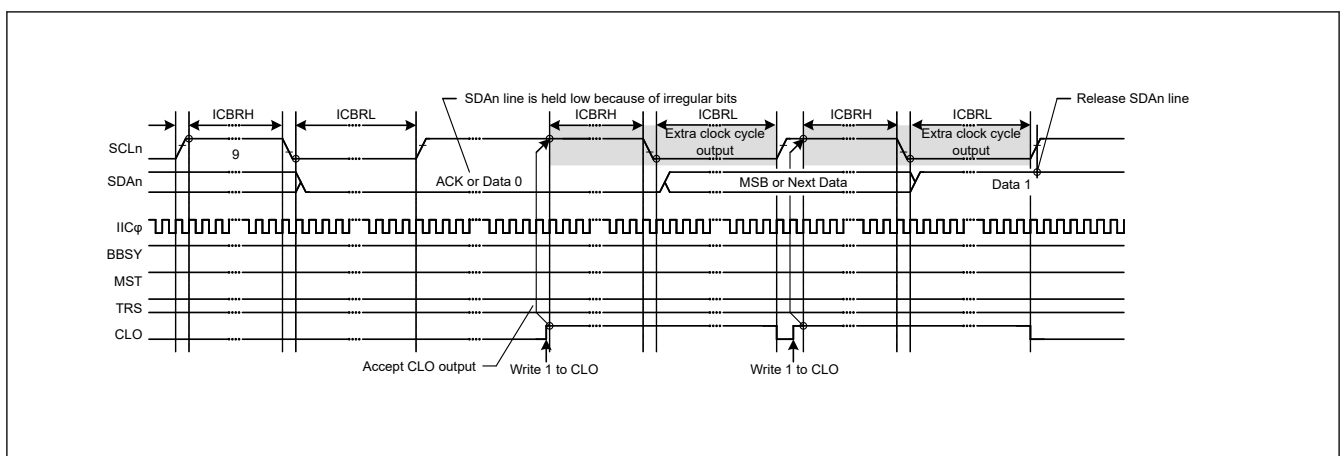


Figure 34.41 Extra SCL clock cycle output function using the CLO bit

34.11.3 IIC Reset and Internal Reset

The IIC module incorporates a function for resetting itself. It uses two types of resets:

- An IIC reset, which initializes all registers, including the BBSY flag in ICCR2.
- An internal reset, which releases the IIC from the slave-address matched state and initializes the internal counter while saving other settings.

After issuing a reset, always set the IICRST bit in ICCR1 to 0. Both types of resets are valid for release from bus-hung states, because both restore the output state of the SCLn and SDAn pins to the high-impedance state.

Issuing a reset during slave operation might lead to a loss of synchronization between the master device clock and the slave device clock, so avoid this when possible. In addition, monitoring of the bus state, such as for the presence of a start condition, is not possible during an IIC reset (ICE and IICRST bits = 01b in ICCR1).

For a detailed description of the IIC and internal resets, see [section 34.14. State of Registers When Issuing Each Condition](#).

34.12 SMBus Operation

The IIC supports data communication conforming to the SMBus Specification (version 2.0). To perform SMBus communication, set the SMBS bit in ICMR3 to 1. To use the transfer rate within a range of 10 to 100 kbps of the SMBus standard, set the CKS[2:0] bits in ICMR1, the ICBRH, and ICBRL registers. In addition, specify the values in the DLCS bit in ICMR2 and the SDDL[2:0] bits in ICMR2 to meet the data hold time specification of 300 ns or more. When the IIC is used only as a slave device, the transfer rate setting is not required, but ICBRL must be set to a value longer than the data setup time (250 ns).

For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (SARL0, SARL1, and SARL2), and set the associated FS bit (7- or 10-bit address format select) in SARUy (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the SALE bit in ICFER to 1 to enable the slave arbitration-lost detection function.

34.12.1 SMBus Timeout Measurement

(1) Measuring slave device timeout

The following period (timeout interval: $T_{\text{LOW:SEXT}}$) 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] $T_{\text{LOW:SEXT}}$: 25 ms (maximum) of the SMBus standard.

If the time measured with the GPT exceeds the clock low-level detection timeout T_{TIMEOUT} : 25 ms (minimum) of the SMBus standard, the slave device must release the bus by writing 1 to the IICRST bit in ICCR1 to issue an internal reset of the IIC. When an internal reset is issued, the IIC stops driving the bus for the SCLn and SDAn pins, making them output high-impedance, which releases the bus.

(2) Measuring master device timeout

The following periods (timeout interval: $T_{\text{LOW:MEXT}}$) 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) $T_{\text{LOW:MEXT}}$: 10 ms (maximum) of the SMBus standard, and the total of all $T_{\text{LOW:MEXT}}$ values from start condition to stop condition must be within $T_{\text{LOW:SEXT}}$: 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) $T_{\text{LOW:MEXT}}$: 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).

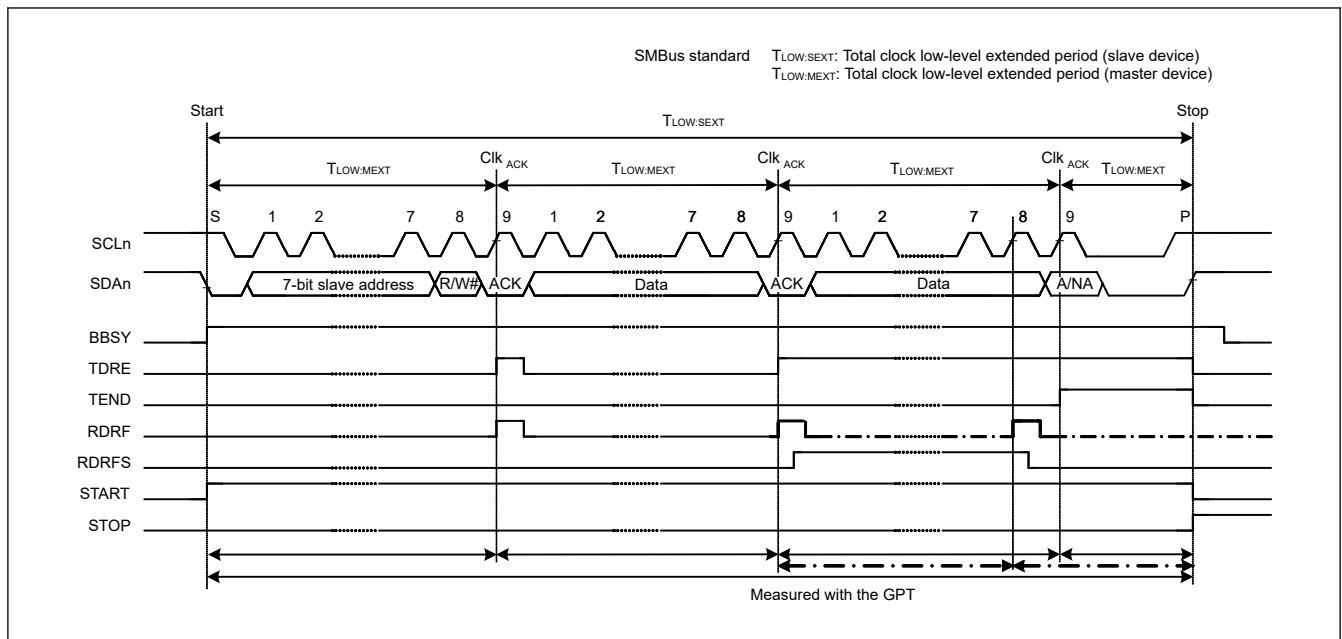


Figure 34.42 SMBus timeout measurement

34.12.2 Packet Error Code (PEC)

The MCU provides a CRC calculator that enables transmission of a Packet Error Code (PEC) or allows checking of the received data in SMBus data communication for the IIC. For the CRC-generating polynomials of the CRC calculator, see [section 37, 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.

34.12.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communicating over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address, or to request its own slave address from the SMBus host.

For a product using the MCU to operate as an SMBus host or ARP master, the host address (0001 000b) sent from the slave device must be detected as a slave address, and so the IIC provides a function for detecting the host address. To detect the host address as a slave address, set the SMBS bit in ICMR3 and the HOAE bit in ICSER to 1. Operation after the host address is detected is the same as normal slave operation.

34.13 Interrupt Sources

The IIC issues five types of interrupt requests:

- Transfer error or event generation (arbitration-lost, NACK detection, timeout detection, start condition detection, and stop condition detection)
- Receive data full
- Transmit data empty
- Transmit end

[Table 34.8](#) lists details about the interrupt requests. The receive data full and transmit data empty interrupts can activate data transfer by the DTC or DMAC.

Table 34.8 Interrupt sources

Symbol	Interrupt source	Interrupt flag	DTC or DMAC activation	Interrupt condition
IICn_EEI* ⁴	Transfer error or event occurrence	AL	Not possible	AL = 1, ALIE = 1
		NACKF		NACKF = 1, NAKIE = 1
		TMOF		TMOF = 1, TMOIE = 1
		START		START = 1, STIE = 1
		STOP		STOP = 1, SPIE = 1
IICn_RXI* ² * ⁴	Receive data full	RDRF	Possible	RDRF = 1, RIE = 1
IICn_TXI* ¹ * ⁴	Transmit data empty	RDRF	Possible	TDRE = 1, TIE = 1
IICn_TEI* ³ * ⁴	Transmit end	TEND	Not possible	TEND = 1, TEIE = 1

Note: There is a delay between the execution of a write instruction for a peripheral module by the CPU and the actual writing to the module. When an interrupt flag is cleared or masked, read the relevant flag again to check whether clearing or masking is complete, then return from interrupt handling. Not doing so creates the possibility of repeated processing of the same interrupt.

Note 1. Because IICn_TXI is edge-detected, it does not require clearing. Additionally, the TDRE flag in ICSR2 (condition for IICn_TXI) is automatically set to 0 when transmit data is written to the ICDRT register or a stop condition is detected (STOP flag = 1 in ICSR2).

Note 2. Because IICn_RXI is edge-detected, it does not require clearing. Additionally, the RDRF flag in ICSR2 (condition for IICn_RXI) is automatically set to 0 when data is read from ICDRR.

Note 3. When using the IICn_TEI interrupt, clear the TEND flag in ICSR2 in the IICn_TEI interrupt handling. The TEND flag in ICSR2 automatically is set to 0 when transmit data is written to the ICDRT register or a stop condition is detected (STOP flag = 1 in ICSR2).

Note 4. Channel number (n = 0, 1).

Clear or mask each flag during interrupt handling.

34.13.1 Buffer Operation for IICn_TXI and IICn_RXI Interrupts

If the conditions for generating an IICn_TXI or IICn_RXI interrupt are satisfied while the associated IR flag is 1, the interrupt request is not output for the ICU but is saved internally. One request per source can be saved internally.

An interrupt request that is saved in the ICU is output when the ICU.IELSRn.IR flag becomes 0. Internally saved interrupt requests are automatically cleared under normal conditions. They can also be cleared by writing 0 to the interrupt enable bit within the associated peripheral module.

34.14 State of Registers When Issuing Each Condition

The IIC has two dedicated resets, IIC reset and Internal reset. Table 34.9 lists the registers states when issuing each condition.

Table 34.9 Register states when issuing each condition (1 of 2)

Registers		Reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
ICCR1	ICE, IICRST	In reset	Saved	Saved	Saved	Saved
	SCLO, SDAO		In reset	In reset		
	Others			Saved		
ICCR2	BBSY	In reset	In reset	Saved	Set	Saved
	ST			In reset	In reset	Saved
	TRS, MST				Set	In reset
	Others				In reset	In reset
ICMR1	BC[2:0]	In reset	In reset	In reset	In reset	Saved
	Others			Saved	Saved	
ICMR2		In reset	In reset	Saved	Saved	Saved

Table 34.9 Register states when issuing each condition (2 of 2)

Registers		Reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
ICMR3	ACKBAT	In reset	In reset	Saved	Saved	In reset
	Others					Saved
ICFER		In reset	In reset	Saved	Saved	Saved
ICSER		In reset	In reset	Saved	Saved	Saved
ICIER		In reset	In reset	Saved	Saved	Saved
ICSR1		In reset	In reset	In reset	Saved	In reset
ICSR2	TDRE, TEND	In reset	In reset	In reset	Saved	In reset
	START				Set	
	STOP				Saved	Set
	Others				Saved	Saved
SARL0, SARL1, SARL2 SARU0, SARU1, SARU2		In reset	In reset	Saved	Saved	Saved
ICBRH, ICBRL		In reset	In reset	Saved	Saved	Saved
ICDRT		In reset	In reset	Saved	Saved	Saved
ICDRR		In reset	In reset	Saved	Saved	Saved
ICDRS		In reset	In reset	In reset	Saved	Saved
Timeout function		In reset	In reset	In reset	Operating	Operating
Bus free time measurement		In reset	In reset	Operating	Operating	Operating

34.15 Event Link Output

The IIC0 module handles the event output for the Event Link Controller (ELC) for the following sources:

(1) Transfer error event

When a transfer error event occurs, the associated event signal can be output to another module by the ELC.

(2) Receive data full

When a receive data register becomes full, the associated event signal can be output to another module by the ELC.

(3) Transmit data empty

When a transmit data register becomes empty, the associated event signal can be output to another module by the ELC.

(4) Transmit end

On completion of the transfer, the associated event signal can be output to another module by the ELC.

34.15.1 Interrupt Handling and Event Linking

Each of the IIC interrupt types (see [Table 34.8](#)) 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 34.8](#).

34.16 Usage Notes

34.16.1 Settings for the Module-Stop Function

The Module Stop Control Register B (MSTPCRB) can enable or disable IIC operation. The IIC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

34.16.2 Notes on Starting Transfer

If the IR flag associated with the IIC interrupt is 1 when transfer is started (ICCR1.ICE bit = 1), follow the procedure in this section to clear the interrupts before enabling operations. Starting transfer with the IR flag set to 1 while the ICCR1.ICE bit is 1 leads to an interrupt request being internally saved after transfer starts, and this can lead to unexpected behavior of the IR flag.

To clear interrupts before starting transfer operation:

1. Confirm that the ICCR1.ICE bit is 0.
2. Set the relevant interrupt enable bits, such as ICIER.TIE to 0.
3. Read the relevant interrupt enable bits, such as ICIER.TIE, and confirm that the value is 0.
4. Set the IR flag to 0.

35. Serial Peripheral Interface (SPI)

35.1 Overview

The Serial Peripheral Interface (SPI) provides high-speed full-duplex synchronous serial communications with multiple processors and peripheral devices. [Table 35.1](#) lists the SPI specifications, [Figure 35.1](#) shows a block diagram, and [Table 35.2](#) lists the I/O pins.

Table 35.1 SPI specifications (1 of 2)

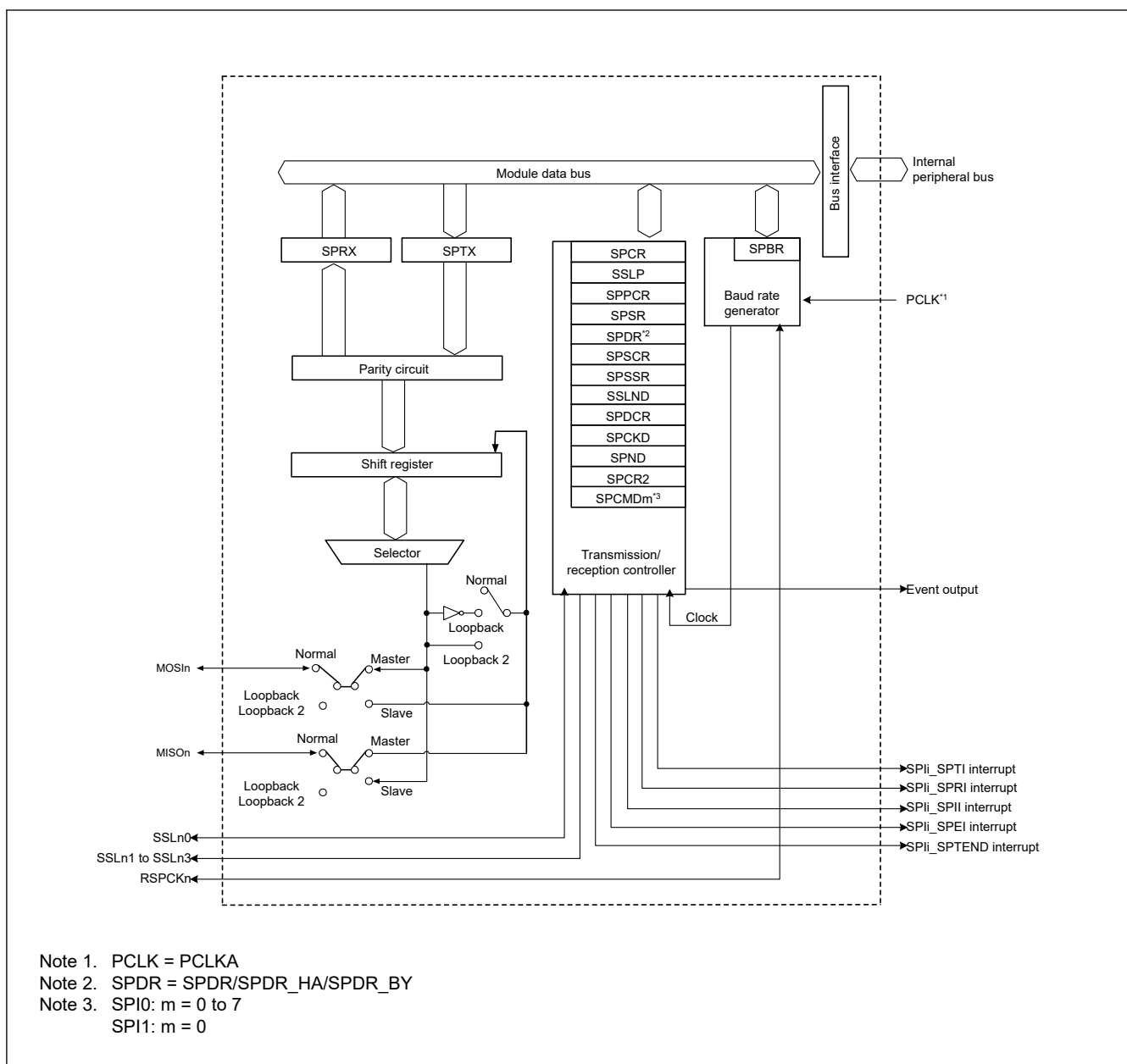
Parameter	Specifications
Number of channels	Two channels
SPI transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (SPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method) • Transmit-only operation available • Communication mode selectable to full-duplex or transmit-only • RSPCK polarity switching • RSPCK phase switching
Data format	• MSB-first or LSB-first selectable • Transfer bit length selectable to 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits • [SPI0] 128-bit transmit and receive buffers • [SPI0] Up to four frames transferable in one round of transmission or reception (each frame consisting of up to 32 bits) • [SPI1] 32-bit transmit and receive buffers • [SPI1] Up to one frames transferable in one round of transmission or reception (each frame consisting of up to 32 bits)
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing PCLKA (the division ratio ranges from divided by 2 to divided by 4096) • In slave mode, the minimum PCLKA clock divided by 4 can be input as RSPCK (PCLKA divided by 4 is the maximum RSPCK frequency) Width at high level: 2 PCLKA cycles; width at low level: 2 PCLKA cycles
Buffer configuration	• Double buffer configuration for the transmit and receive buffers • 128 bits*2 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 (SSLn: SSLn0 to SSLn3) (n = A, B) for each channel • In single-master mode, SSLn0 to SSLn3 pins are output • In multi-master mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins either for output or unused • In slave mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins unused • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity
Control in master transfer	• Transfers of up to eight commands each can be executed sequentially in looped execution (only SPI0) • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, MSB- or LSB-first, burst (only SPI0), 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

Table 35.1 SPI specifications (2 of 2)

Parameter	Specifications
Interrupt sources	Interrupt sources: • Receive buffer full interrupt • Transmit buffer empty interrupt • SPI error interrupt (mode fault error, overrun error, parity error) • SPI idle interrupt (SPI idle) • Transmission-complete interrupt
Event link function (only SPI0)	The following events can be output to the Event Link Controller (ELC): • Receive buffer full signal • Transmit buffer empty signal • Mode fault, underrun, overrun, or parity error signal • SPI idle signal • Transmission-complete signal
Other functions	• Switching between CMOS output and open-drain output • SPI initialization function • Loopback mode
Module-stop function	Module-stop state can be set to reduce power consumption.

Note 1. In master reception and when the RSPCK auto-stop function is enabled, an overrun error does not occur because the transfer clock is stopped on overrun error detection.

Note 2. SPI1 is 32-bit.

**Figure 35.1 SPI block diagram**

The SPI automatically switches the I/O direction of the SSLn0 pin. SSLn0 is set as an output when the SPI is a single master, and as an input when the SPI is a multi-master or a slave. The RSPCKn, MOSIn, and MISO_n 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 35.3.2. Controlling the SPI Pins](#).

Table 35.2 SPI I/O pins (1 of 2)

Channel	Pin name	I/O	Description
SPI0	RSPCKA	I/O	Clock input/output pin
	MOSIA	I/O	Master transmit data input/output
	MISOA	I/O	Slave transmit data input/output
	SSLA0	I/O	Slave selection input/output
	SSLA1 to SSLA3	Output	Slave selection output
SPI1	RSPCKB	I/O	Clock input/output pin
	MOSIB	I/O	Master transmit data input/output

Table 35.2 SPI I/O pins (2 of 2)

Channel	Pin name	I/O	Description
	MISOB	I/O	Slave transmit data input/output
	SSLB0	I/O	Slave selection input/output
	SSLB1 to SSLB3	Output	Slave selection output

Note: Pin names are indicated as "...A" or "...An" for SPI0, and "...B" or "...Bn" for SPI1 (n = 0, 1, 2, or 3).

35.2 Register Descriptions

35.2.1 SPCR : SPI Control Register

Base address: SPI0 = 0x4007_2000
SPI1 = 0x4007_2100

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	SPMS	SPI Mode Select 0: Select SPI operation (4-wire method) 1: Select clock synchronous operation (3-wire method)	R/W
1	TXMD	Communications Operating Mode Select 0: Select full-duplex synchronous serial communications 1: Select serial communications with transmit-only	R/W
2	MODFEN	Mode Fault Error Detection Enable 0: Disable detection of mode fault errors 1: Enable detection of mode fault errors	R/W
3	MSTR	SPI Master/Slave Mode Select 0: Select slave mode 1: Select master mode	R/W
4	SPEIE	SPI Error Interrupt Enable 0: Disable SPI error interrupt requests 1: Enable SPI error interrupt requests	R/W
5	SPTIE	Transmit Buffer Empty Interrupt Enable 0: Disable transmit buffer empty interrupt requests 1: Enable transmit buffer empty interrupt requests	R/W
6	SPE	SPI Function Enable 0: Disable SPI function 1: Enable SPI function	R/W
7	SPRIE	SPI Receive Buffer Full Interrupt Enable 0: Disable SPI receive buffer full interrupt requests 1: Enable SPI receive buffer full interrupt requests	R/W

SPMS bit (SPI Mode Select)

The SPMS bit selects SPI operation (4-wire method) or clock synchronous operation (3-wire method).

The SSLn0 to SSLn3 pins are not used in clock synchronous operation. The RSPCKn, MOSIn, and MISO pins handle communications. For clock synchronous operation in master mode (MSTR = 1), the SPCMDm.CPHA bit can be set to either 0 or 1. For clock synchronous operation in slave mode (MSTR = 0), always set the CPHA bit to 1. Do not perform operations if the CPHA bit is set to 0 for clock synchronous operation in slave mode (MSTR = 0).

TXMD bit (Communications Operating Mode Select)

The TXMD bit selects full-duplex synchronous serial communications or transmit-only operations. When this bit is set to 1, the SPI only performs transmit operations and not receive operations (see [section 35.3.6. Data Transfer Modes](#)), and receive buffer full interrupt requests cannot be used.

MODFEN bit (Mode Fault Error Detection Enable)

The MODFEN bit enables or disables the detection of mode fault errors (see [section 35.3.8. Error Detection](#)). In addition, the SPI determines the I/O direction of the SSLn0 to SSLn3 pins based on combination of the MODFEN and MSTR bits (see [section 35.3.2. 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, MISO_n, 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 35.3.8. 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 35.3.8. Error Detection](#).

Setting the SPE bit to 0 disables the SPI function and initializes a part of the module function. For details, see [section 35.3.9. Initializing the SPI](#). In addition, a transmit buffer empty interrupt request is generated when the SPE bit is changed from 0 to 1 or from 1 to 0.

SPRIE bit (SPI Receive Buffer Full Interrupt Enable)

The SPRIE bit enables or disables the generation of an SPI receive buffer full interrupt request when the SPI detects a receive buffer full write after completion of a serial transfer.

35.2.2 SSLP : SPI Slave Select Polarity Register

Base address: SPI0 = 0x4007_2000
SPI1 = 0x4007_2100

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SSL0P	SSLn0 Signal Polarity Setting 0: Set SSLn0 signal to active-low 1: Set SSLn0 signal to active-high	R/W
1	SSL1P	SSLn1 Signal Polarity Setting 0: Set SSLn1 signal to active-low 1: Set SSLn1 signal to active-high	R/W
2	SSL2P	SSLn2 Signal Polarity Setting 0: Set SSLn2 signal to active-low 1: Set SSLn2 signal to active-high	R/W
3	SSL3P	SSLn3 Signal Polarity Setting 0: Set SSLn3 signal to active-low 1: Set SSLn3 signal to active-high	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

35.2.3 SPPCR : SPI Pin Control Register

Base address: SPI0 = 0x4007_2000
SPI1 = 0x4007_2100

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPLP	SPI Loopback 0: Normal mode 1: Loopback mode (receive data = inverted transmit data)	R/W
1	SPLP2	SPI Loopback 2 0: Normal mode 1: Loopback mode (receive data = transmit data)	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	MOIFV	MOSI Idle Fixed Value 0: Set level output on MOSIn pin during MOSI idling to low 1: Set level output on MOSIn pin during MOSI idling to high	R/W
5	MOIFE	MOSI Idle Value Fixing Enable 0: Set MOSI output value to equal final data from previous transfer 1: Set MOSI output value to equal value set in the MOIFV bit	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

SPLP bit (SPI Loopback)

The SPLP bit selects the mode of the SPI pins. When this bit is set to 1, the SPI shuts off the path between the MISO_{in} pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSI_{in} pin and the shift register if the SPCR.MSTR bit is 0. The SPI then inverts the value of the input path for the shift register and connects it to the output path (loopback mode). For more information, see [section 35.3.12. Loopback Mode](#).

SPLP2 bit (SPI Loopback 2)

The SPLP2 bit selects the mode of the SPI pins. When this bit is set to 1, the SPI shuts off the path between the MISO_{in} pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSI_{in} pin and the shift register if the SPCR.MSTR bit is 0. The SPI then connects the value of the input path for the shift register to the output path (loopback mode) without inverting the value. For more information, see [section 35.3.12. Loopback Mode](#).

MOIFV bit (MOSI Idle Fixed Value)

The MOIFV bit determines the MOSI pin output value during the SSL negation period (including the SSL retention period during a burst transfer (only SPI0)) when the MOIFE bit is 1 in master mode.

MOIFE bit (MOSI Idle Value Fixing Enable)

The MOIFE bit fixes the MOSI output value when the SPI is in master mode and in an SSL negation period (including the SSL retention period during a burst transfer (only SPI0)). When the MOIFE bit is 0, the SPI outputs the last data from the previous serial transfer during the SSL negation period to the MOSI pin. When the MOIFE bit is 1, the SPI outputs the fixed value set in the MOIFV bit to the MOSI pin.

35.2.4 SPSR : SPI Status Register

Base address: SPI0 = 0x4007_2000
SPI1 = 0x4007_2100

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SPRF	—	SPTE F	UDRF	PERF	MODF	IDLNF	OVRF

Value after reset: 0 0 1 0 0 0 0 0

Bit	Symbol	Function	R/W
0	OVRF	Overrun Error Flag 0: No overrun error occurred 1: Overrun error occurred	R/W ^{*1}
1	IDLNF	SPI Idle Flag 0: SPI is in the idle state 1: SPI is in the transfer state	R
2	MODF	Mode Fault Error Flag 0: No mode fault or underrun error occurred 1: Mode fault error or underrun error occurred	R/W ^{*1}
3	PERF	Parity Error Flag 0: No parity error occurred 1: Parity error occurred	R/W ^{*1}
4	UDRF	Underrun Error Flag The UDRF bit is valid when MODF flag is 1. 0: Mode fault error occurred (MODF = 1) 1: Underrun error occurred (MODF = 1)	R/W ^{*1 *2}
5	SPTEF	SPI Transmit Buffer Empty Flag 0: Data is in the transmit buffer 1: No data is in the transmit buffer	R/W ^{*3}
6	—	This bit is read as 0. The write value should be 0.	R/W
7	SPRF	SPI Receive Buffer Full Flag 0: No valid data is in SPDR/SPDR_HA 1: Valid data is in SPDR/SPDR_HA	R/W ^{*3}

Note 1. Only 0 can be written to clear the flag after reading 1.

Note 2. Clear the UDRF flag at the same time as the MODF flag.

Note 3. The write value should be 1.

OVRF flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error. In master mode (SPCR.MSTR bit = 1) and when the RSPCK clock auto-stop function is enabled (SPCR1.SCKASE bit = 1), overrun errors do not occur. This flag does not set to 1. For details, see [section 35.3.8.1. Overrun errors](#).

[Setting condition]

- When the next serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer is full.

[Clearing condition]

- When 0 is written to the OVRF flag after the OVRF flag is confirmed to be 1 by a read of SPSR.

IDLNF flag (SPI Idle Flag)

The IDLNF flag indicates the transfer status of the SPI.

[Setting conditions]

Master mode

- When none of the conditions in the master mode [Clearing condition] is met.

Slave mode

- When the SPE bit in SPCR is 1, enabling the SPI function.

[Clearing conditions]

Master mode

When condition 1 or all other conditions are satisfied.

Condition 1: The SPE bit in SPCR is 0, indicating that the SPI is initialized.

Condition 2: The transmit buffer (SPTX) is empty, indicating that data for the next transfer is not set.

Condition 3: The SPI internal sequencer is in the idle state, indicating that operation up to next-access delay is complete.

Condition 4: The SPCP[2:0] bits in SPSSR are 000 (at the beginning of sequence control) (This is for only SPI0)

Slave mode

- When condition 1 is satisfied.

MODF flag (Mode Fault Error Flag)

The MODF flag indicates the occurrence of a mode fault error or an underrun error. The UDRF flag indicates which error occurred.

[Setting conditions]

Multi-master mode

- When the input level of the SSLni pin changes to the active level while the SPCR.MSTR bit is 1 (master mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), triggering a mode fault error.

Slave mode

- When condition 1 or 2 is satisfied.

Condition 1: The SSLni pin is negated before the RSPCK cycle required for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), triggering a mode fault error.

Condition 2: The serial transfer begins with the SPCR.MSTR bit is set to 0 (slave mode), the SPCR.SPE bit is set to 1, and the transmission data not prepared, triggering an underrun error.

The active level of the SSLni signal is determined by the SSLP.SSLiP bit (SSLi signal polarity setting).

[Clearing condition]

- When SPSR is read while this flag is 1, and then 0 is written to this flag.

PERF flag (Parity Error Flag)

The PERF flag indicates the occurrence of a parity error.

[Setting condition]

- When a serial transfer ends while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1, triggering a parity error.

[Clearing condition]

- When SPSR is read while this flag is 1, and then 0 is written to this flag.

UDRF flag (Underrun Error Flag)

The UDRF flag indicates the occurrence of an underrun error.

[Setting condition]

- When the serial transfer begins with the SPCR.MSTR bit is set to 0 (slave mode), the SPCR.SPE bit is set to 1, and the transmission data not prepared, triggering an underrun error.

[Clearing condition]

- When SPSR is read while this flag is 1, and then 0 is written to this flag.

SPTEF flag (SPI Transmit Buffer Empty Flag)

The SPTEF flag indicates the status of the transmit buffer for the SPI Data Register (SPDR/SPDR_HA).

[Setting conditions]

- When condition 1. or 2. is satisfied.
 1. The SPCR.SPE bit is 0, indicating that the SPI is initialized.
 2. Transmit data (the frame size specified by the SPDCR.SPFC[1:0]) is transferred from the transmit buffer to the shift register. (SPFC[1:0] is only SPI0)

[Clearing condition]

- [SPI0] When data is written to SPDR/SPDR_HA/SPDR_BY equals the number of frames set in the SPFC[1:0] bits in the SPI Data Control Register (SPDCR).
- [SPI1] When data is written to SPDR/SPDR_HA/SPDR_BY.

Data can only be written to SPDR/SPDR_HA/SPDR_BY when the SPTEF flag is 1. If data is written to the transmit buffer of SPDR/SPDR_HA when the SPTEF flag is 0, data in the transmit buffer is not updated.

SPRF flag (SPI Receive Buffer Full Flag)

The SPRF flag indicates the status of the receive buffer for the SPI Data Register (SPDR/SPDR HA).

[Setting condition]

- [SPI0] Received data with the frame size specified by the SPDCR.SPFC[1:0] bits have been transferred to the SPDR from the shift register while the SPRF flag is 0. And satisfy the following. However, the SPRF flag does not change from 0 to 1 while the OVRF flag = 1.
 - The SPCR.TXMD bit is 0 (transmit-receive master mode, transmit-receive slave mode)

[SPI1] Received data have been transferred to the SPDR from the shift register while the SPRF flag is 0. And satisfy the following. However, the SPRF flag does not change from 0 to 1 while the OVRF flag = 1.

- The SPCR.TXMD bit is 0 (transmit-receive master mode, transmit-receive slave mode)

[Clearing condition]

- When received data is read from the SPDR/SPDR HA.

35.2.5 SPDR/SPDR HA/SPDR BY : SPI Data Register

Base address: SPI0 = 0x4007_2000
SPI1 = 0x4007_2100

Offset address: 0x04

Bit position: 31 0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	SPI Data	R/W

SPDR/SPDR_HA/SPDR_BY is the interface with the buffers that hold data for transmission and reception by the SPI. When accessing this register in words (the SPDCR.SPLW bit is 1), access SPDR. When accessing it in halfwords (the SPLW bit is 0), access SPDR_HA. When accessing it in byte (the SPDCR.SPBYP bit is 1), access SPDR_BY.

The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to SPDR/SPDR_HA. Figure 35.2 shows the configuration of the SPDR/SPDR_HA register.

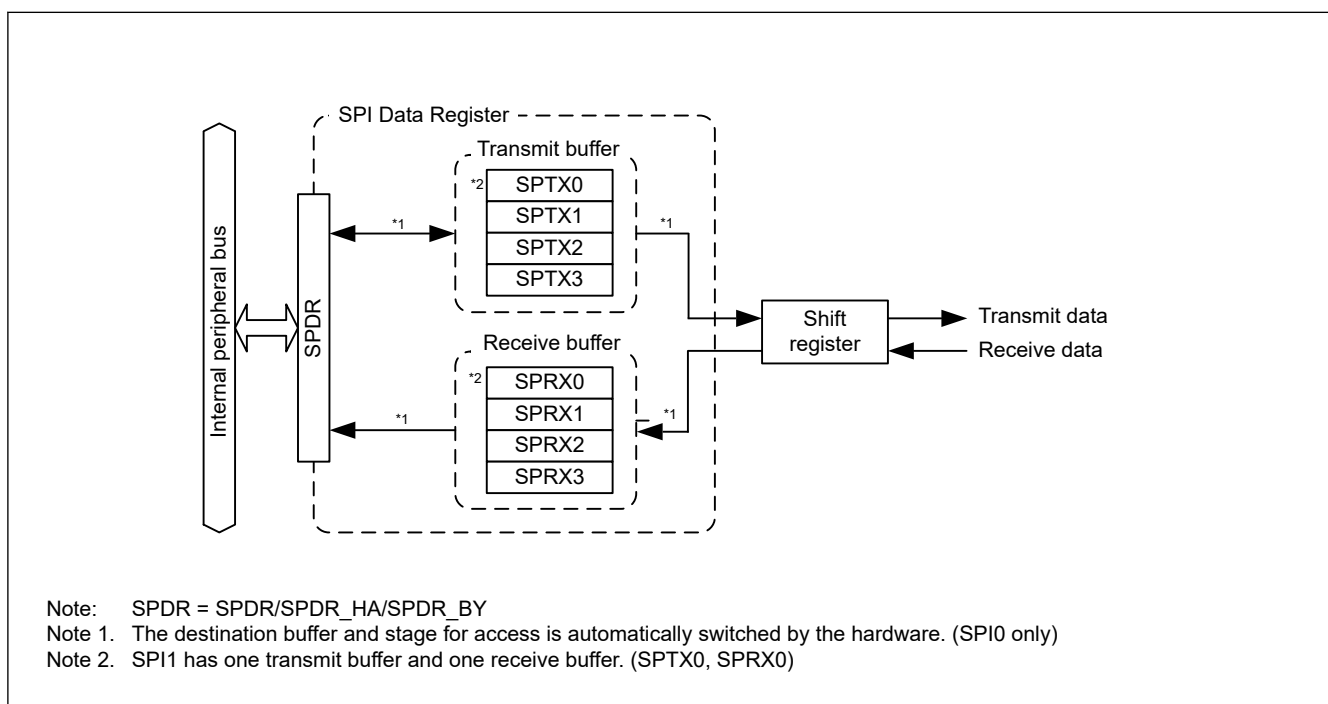


Figure 35.2 Configuration of SPDR/SPDR_HA/SPDR_BY

The transmit and receive buffers each have four stages^{*1}. The eight stages^{*1} of the buffer are all mapped to the single address of SPDR/SPDR_HA/SPDR_BY.

Data written to SPDR/SPDR_HA/SPDR_BY is written to a transmit-buffer stage (SPTX_n) (n = 0 to 3), and then transmitted from the buffer. The receive buffer holds received data on completion of reception. The receive buffer is not updated if an overrun is generated.

Additionally, if the data length is not 32 bits, bits not referred to in SPTX_n (n = 0 to 3) are stored in the associated bits in SPRX_n (n = 0 to 3). For example, if the data length is 9 bits, the received data is stored in the SPRX_n[8:0] bits, and the SPTX_n[31:9] bits are stored in the SPRX_n[31:9] bits.

Note 1. For SPI1, the buffers each have one stage, for a total of two stages of the buffer.

(1) Bus interface

SPDR/SPDR_HA/SPDR_BY is an interface with 32-bit wide transmit and receive buffers, each of which has four stages^{*1}, for a total of 32 bytes^{*1}. The 32 bytes^{*1} are mapped to the 4-byte address space for SPDR/SPDR_HA/SPDR_BY. Additionally, the unit of access for SPDR/SPDR_HA/SPDR_BY is selected by the SPI Word Access/Halfword Access Specification bit in the SPI Data Control Register (SPDCR.SPLW). SPDR can also be accessed with the access size specified by the SPI Byte Access bit in the SPI Data Control Register (SPDCR.SPBYP).

Flush the transmission data at the LSB end of the register, and store the received data at the LSB end.

The following sections describe the operations involved in writing to and reading from SPDR/SPDR_HA/SPDR_BY.

Note 1. For SPI1, the buffers each have one stage, for a total of 4 bytes.

Writing

Data written to SPDR/SPDR_HA/SPDR_BY is written to a transmit buffer (SPTXn). This is not affected by the value of the SPDCR.SPRDTD bit, unlike when reading from SPDR/SPDR_HA/SPDR_BY. The transmit buffer includes a transmit buffer write pointer^{*1} that is automatically updated to reference the next stage each time data is written to SPDR/SPDR_HA/SPDR_BY.

Figure 35.3 shows the configuration of the bus interface with the transmit buffer when writing to SPDR/SPDR_HA/SPDR_BY.

Note 1. SPI0 only

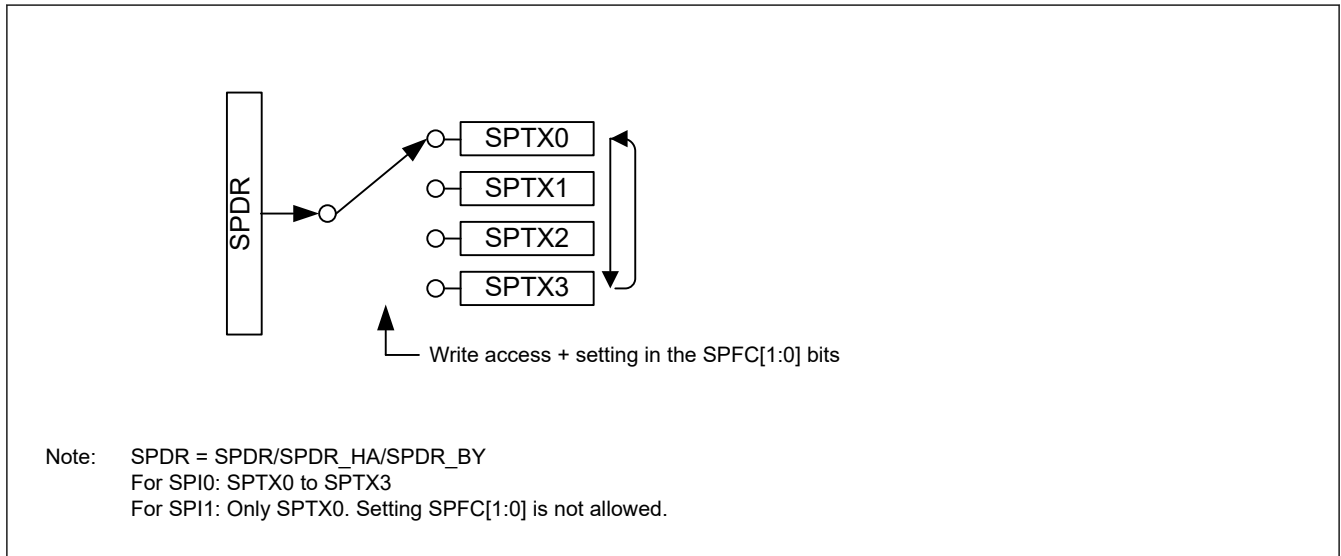


Figure 35.3 Configuration of SPDR/SPDR_HA/SPDR_BY for write access

The sequence for switching the transmit buffer write pointer differs with the setting of the number of frames specification bits in the SPI Data Control Register (SPDCR.SPFC[1:0] (only SPI0)). The relationship of the SPFC[1:0] setting and the sequence of pointer switching from SPTX0 to SPTX3 is as follows:

- When SPFC[1:0] = 00b: SPTX0 → SPTX0 → SPTX0 → ...
- When SPFC[1:0] = 01b: SPTX0 → SPTX1 → SPTX0 → SPTX1 → ...
- When SPFC[1:0] = 10b: SPTX0 → SPTX1 → SPTX2 → SPTX0 → SPTX1 → ...
- When SPFC[1:0] = 11b: SPTX0 → SPTX1 → SPTX2 → SPTX3 → SPTX0 → SPTX1 → ...

When 1 is written to the SPI Function Enable bit in the SPI Control Register (SPCR.SPE) while the bit is 0, SPTX0 is the destination for the next write.

When writing to the transmit buffer (SPTXn) after generating the transmit buffer empty interrupt (when SPSR.SPTEF is 1), write the number of frames set in SPFC[1:0] (only SPI0) in the SPI Data Control Register (SPDCR). Even when the specified number of frames is written to the transmit buffer (SPTXn), the value of the buffer is not updated after completion of the writing and before generation of the next transmit buffer empty interrupt (when SPTEF is 0).

Reading

SPDR/SPDR_HA/SPDR_BY can be accessed to read the value of a receive buffer (SPRXn) or a transmit buffer (SPTXn). The setting in the SPI Receive/Transmit Data Select bit in the SPI Data Control Register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer. The sequence of reading the SPDR/SPDR_HA/SPDR_BY register is controlled by the independent receive buffer and transmit buffer read pointers.

Figure 35.4 shows the configuration of the bus interface with the receive and transmit buffers for reading from SPDR/SPDR_HA/SPDR_BY.

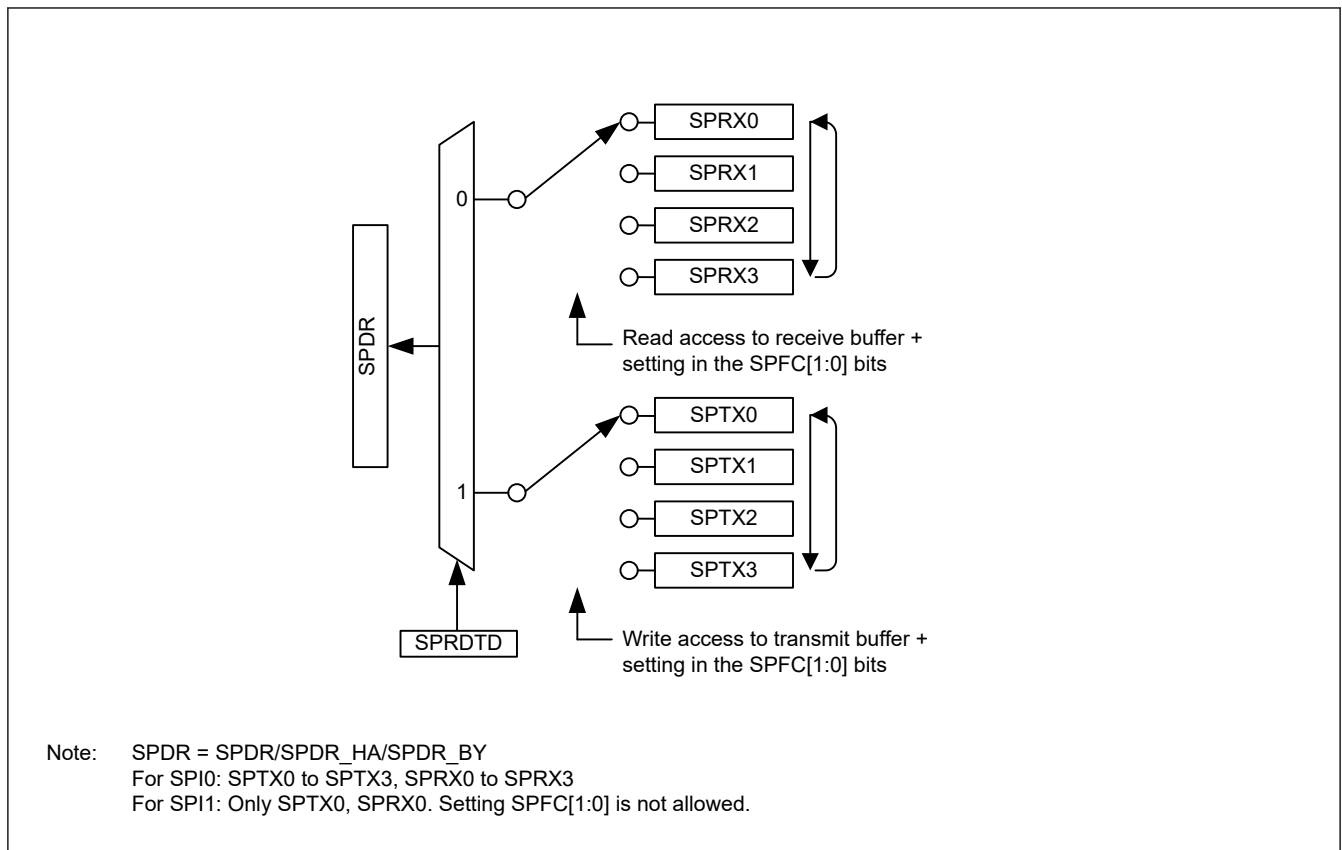


Figure 35.4 Configuration of SPDR/SPDR HA/SPDR_BY for read access

Reading the receive buffer switches the receive buffer read pointer to the next buffer automatically. The switching sequence for the receive buffer read pointer is the same as that for the transmit buffer write pointer. However, when 1 is written to the SPI Function Enable bit in the SPI Control Register (SPCR.SPE) while the value of the bit is 1, SPRX0 is referenced by the buffer read pointer for the next read.

The transmit buffer read pointer is updated when writing to SPDR/SPDR_HA/SPDR_BY, but not updated when reading from the transmit buffer. When reading from the transmit buffer, the value most recently written to SPDR/SPDR_HA/SPDR_BY is read.

After a transmit buffer empty interrupt is generated, reading from the transmit buffer returns all 0s after the completion of writing the number of frames of data specified in the SPDCR.SPFC[1:0] (only SPI0) bits, until the next buffer empty interrupt is generated (when SPTEF is 0).

Note: [In case of SPI1] After generation of the transmit buffer empty interrupt, 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).

35.2.6 SPSCR : SPI Sequence Control Register

Base address: SPI0 = 0x4007 2000

Offset address: 0x08

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

Bit	Symbol	Function	R/W
2:0	SPSLN[2:0]	SPI Sequence Length Specification The sequence length that is set in these bits determines the order in which the SPCMD0 to SPCMD7 registers are referenced. The setting defines the relationship between the sequence length and the SPCMD0 to SPCMD7 registers referenced by the SPI. In slave mode, the SPI references SPCMD0. 0 0 0: Sequence Length is 1 (Referenced SPCMDn, n = 0→0→...) 0 0 1: Sequence Length is 2 (Referenced SPCMDn, n = 0→1→0→...) 0 1 0: Sequence Length is 3 (Referenced SPCMDn, n = 0→1→2→0→...) 0 1 1: Sequence Length is 4 (Referenced SPCMDn, n = 0→1→2→3→0→...) 1 0 0: Sequence Length is 5 (Referenced SPCMDn, n = 0→1→2→3→4→0→...) 1 0 1: Sequence Length is 6 (Referenced SPCMDn, n = 0→1→2→3→4→5→0→...) 1 1 0: Sequence Length is 7 (Referenced SPCMDn, n = 0→1→2→3→4→5→6→0→...) 1 1 1: Sequence Length is 8 (Referenced SPCMDn, n = 0→1→2→3→4→5→6→7→0→...)	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: All bits are reserved in SPI1

SPSCR specifies the sequence length when the SPI operates in master mode. Before changing the SPSLN[2:0] bits while both the SPCR.MSTR and SPCR.SPE bits are 1, check that the SPSR.IDLNF flag is 0.

SPSLN[2:0] bits (SPI Sequence Length Specification)

The SPSLN[2:0] bits specify the sequence length when the SPI in master mode performs sequential operations. The SPI in master mode changes the SPCMD0 to SPCMD7 registers to be referenced, and the order in which they are referenced is based on this sequence length setting. In slave mode, SPCMD0 is referenced.

35.2.7 SPSSR : SPI Sequence Status Register

Base address: SPI0 = 0x4007_2000

Offset address: 0x09

Bit position: 7 6 5 4 3 2 1 0

Bit field:	—	SPECM[2:0]	—	SPCP[2:0]

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	SPCP[2:0]	SPI Command Pointer 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
3	—	This bit is read as 0.	R
6:4	SPECM[2:0]	SPI Error Command 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
7	—	This bit is read as 0.	R

SPSSR indicates the sequence control status when the SPI operates in master mode. Any writes to SPSSR are ignored.

SPCP[2:0] bits (SPI Command Pointer)

The SPCP[2:0] bits indicate the SPCMDm register that is referenced to by the pointer during sequence control by the SPI. For the SPI sequence control, see [section 35.3.10.1. Master mode operation](#).

SPECM[2:0] bits (SPI Error Command)

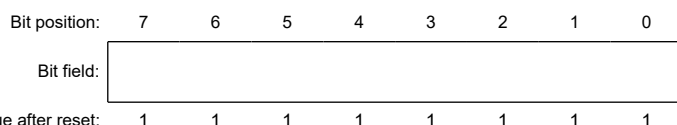
The SPECM[2:0] bits indicate the SPCMDm register that is specified in the SPCP[2:0] bits when an error is detected during sequence control by the SPI. The SPI updates the SPECM[2:0] bits only when an error is detected. If both the SPSR.OVRF and SPSR.MODF flags are 0 and there is no error, the values of the SPECM[2:0] bits have no meaning.

For the SPI error detection function, see [section 35.3.8. Error Detection](#). For the SPI sequence control, see [section 35.3.10.1. Master mode operation](#).

35.2.8 SPBR : SPI Bit Rate Register

Base address: SPI0 = 0x4007_2000
SPI1 = 0x4007_2100

Offset address: 0x0A



Bit	Symbol	Function	R/W
7:0	n/a	Bit rate	R/W

SPBR sets the bit rate in master mode.

When the SPI is in slave mode, the bit rate depends on the bit rate of the input clock, regardless of the settings in SPBR and the SPCMDm.BRDV[1:0] bits (bit rate division setting). Use bit rates that satisfy the electrical characteristics of the device.

The bit rate is determined by combinations of the SPBR and SPCMDm.BRDV[1:0] settings in the SPI Command Register. The equation for calculating the bit rate is given as follows:

$$\text{Bit rate} = \frac{f(\text{PCLK})}{2 \times (n + 1) \times 2^N}$$

(PCLK = PCLKA)

In the equation, n denotes an SPBR setting (0, 1, 2, ..., 255), and N denotes a BRDV[1:0] setting (0, 1, 2, 3).

[Table 35.3](#) lists examples of the relationship between the SPBR settings, the BRDV[1:0] settings, and bit rates.

Table 35.3 Relationship between SPBR settings, BRDV[1:0] settings, and bit rates

SPBR (n)	SPCMDm. BRDV[1:0] bits (N)	Division ratio	Bit rate				
			PCLKA = 32 MHz	PCLKA = 36 MHz	PCLKA = 40 MHz	PCLKA = 50 MHz	PCLKA = 60 MHz
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps	25.0 Mbps	30.0 Mbps
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps	12.5 Mbps	15.0 Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps	8.33 Mbps	10.0 Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps	6.25 Mbps	7.50 Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps	5.00 Mbps	6.00 Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps	4.16 Mbps	5.00 Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps	2.08 Mbps	2.50 Mbps
5	2	48	667 kbps	750 kbps	833 kbps	1.04 Mbps	1.25 Mbps
5	3	96	333 kbps	375 kbps	417 kbps	521 kbps	625 kbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps	12.2 kbps	14.6 kbps

35.2.9 SPDCR : SPI Data Control Register

Base address: SPI0 = 0x4007_2000
SPI1 = 0x4007_2100

Offset address: 0x0B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SPBY T	SPLW	SPRD TD	—	—	SPFC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SPFC[1:0]	Number of Frames Specification (only SPI0) 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	SPRDTD	SPI Receive/Transmit Data Select 0: Read SPDR/SPDR_HA values from receive buffer 1: Read SPDR/SPDR_HA values from transmit buffer, but only if the transmit buffer is empty	R/W
5	SPLW	SPI Word Access/Halfword Access Specification 0: Set SPDR_HA to valid for halfword access 1: Set SPDR to valid for word access	R/W
6	SPBYT	SPI Byte Access Specification 0: SPDR/SPDR_HA is accessed in halfword or word (SPLW is valid) 1: SPDR_BY is accessed in byte (SPLW is invalid)	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

[For SPI0]

The SPI Data Control Register (SPDCR) is used to read the number of frames that can be stored in the SPDR register, read the SPDR register, and to set the access width for the SPDR register to word access, halfword access, or byte access. Up to four frames can be transmitted or received in one round of transmission or reception. The amount of data in each transfer is controlled by the combination of the SPCMDm.SP[B3:0] bits, the SPCR.SP[SLN2:0] bits, and the SPFC[1:0] bits.

When changing the SPFC[1:0] bits while the SPCR.SPE bit is 1, check that the SPSR.IDLNF flag is 0.

[For SPI1]

The SPI data control register (SPDCR) is used to read the SPDR register, and to set the access width for the SPDR register to word access, halfword access, or byte access.

SPFC[1:0] bits (Number of Frames Specification)

The SPFC[1:0] bits specify the number of frames that can be stored in SPDR/SPDR_HA per transfer activation. Up to four frames can be transmitted or received in one round of transmission or reception.

When the number of transmission data frames specified in the SPFC[1:0] bits is written to the SPDR/SPDR_HA register, SPI clears the SPSR.SPTEF flag to 0 and begins transmitting. After that, when the number of transmission data frames specified in the SPFC[1:0] bits is transmitted to the shift register, the SPI generates the transmit buffer empty interrupt (SPSR.SPTEF sets to 1).

When the number of data frames specified in the SPFC[1:0] bits is received, the SPI generates the receive buffer full interrupt (SPSR.SPRF sets to 1).

Table 35.4 Settable combinations of the SP[SLN2:0] and SPFC[1:0] bits (1 of 2)

Setting	SP[SLN2:0]	SPFC[1:0]	Number of frames in a single sequence	Number of frames at which transmission or receive buffer is filled
1-1	000b	00b	1	1

Table 35.4 Settable combinations of the SPSLN[2:0] and SPFC[1:0] bits (2 of 2)

Setting	SPSLN[2:0]	SPFC[1:0]	Number of frames in a single sequence	Number of frames at which transmission or receive buffer is filled
1-2	000b	01b	2	2
1-3	000b	10b	3	3
1-4	000b	11b	4	4
2-1	001b	01b	2	2
2-2	001b	11b	4	4
3	010b	10b	3	3
4	011b	11b	4	4
5	100b	00b	5	1
6	101b	00b	6	1
7	110b	00b	7	1
8	111b	00b	8	1

SPRDTD bit (SPI Receive/Transmit Data Select)

The SPRDTD bit selects whether the SPDR/SPDR_HA reads values from the receive buffer or from the transmit buffer. If reading is from the transmit buffer, the last value written to SPDR/SPDR_HA register is read. Read the transmit buffer after an SPI transmit buffer empty interrupt is generated until data of frames specified by SPFC[1:0] (only SPI0) has been written (while the SPSR.SPTEF flag is 1).

For details, see [section 35.2.5. SPDR/SPDR_HA/SPDR_BY : SPI Data Register](#).

SPLW bit (SPI Word Access/Halfword Access Specification)

The SPLW bit specifies the access width for SPDR. Access to SPDR_HA in halfwords is valid when the SPLW bit is 0 and access to SPDR in words is valid when the SPLW bit is 1. Also, when this bit is 0, set the SPI data length setting bits, SPCMDm.SPB[3:0], from 8 to 16 bits. Do not perform any operations when a data length of 20, 24, or 32 bits is specified.

SPBYT bit (SPI Byte Access Specification)

The SPBYT bit is used to set the data width of access to the SPI Data Register (SPDR). When SPBYT = 0, use word or half word access to SPDR/SPDR_HA. When SPBYT = 1 (in that case, SPLW is invalid), use byte access to SPDR_BY.

When SPBYT = 1, set the SPI data length bits (SPB[3:0]) in the SPI Command Register m (SPCMDm) to 8 bits. If SPB[3:0] are set to 9 to 16, 20, 24, or 32 bit, subsequent operation is not guaranteed.

35.2.10 SPCKD : SPI Clock Delay Register

Base address: SPI0 = 0x4007_2000
SPI1 = 0x4007_2100

Offset address: 0x0C

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

Bit	Symbol	Function	R/W
2:0	SCKDL[2:0]	RSPCK Delay Setting 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

SPCKD specifies the RSPCK delay, the period from the beginning of SSLni signal assertion to RSPCK oscillation, when the SPCMDm.SCKDEN bit is 1.

SCKDL[2:0] bits (RSPCK Delay Setting)

The SCKDL[2:0] bits specify an RSPCK delay value when the SPCMDm.SCKDEN bit is 1. When using the SPI in slave mode, set the SCKDL[2:0] bits to 000b.

35.2.11 SSLND : SPI Slave Select Negation Delay Register

Base address: SPI0 = 0x4007_2000
SPI1 = 0x4007_2100

Offset address: 0x0D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SLNDL[2:0]		

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	SLNDL[2:0]	SSL Negation Delay Setting 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

SSLND specifies the SSL negation delay, the period from the transmission of a final RSPCK edge to the negation of the SSLni signal during a serial transfer by the SPI in master mode. If the contents of SSLND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, do not perform subsequent operations.

SLNDL[2:0] bits (SSL Negation Delay Setting)

The SLNDL[2:0] bits specify an SSL negation delay value when the SLNDEN bit in SPCMDn is 1 and the SPI is in master mode. When using the SPI in slave mode, set the SLNDL[2:0] bits to 000b.

35.2.12 SPND : SPI Next-Access Delay Register

Base address: SPI0 = 0x4007_2000
SPI1 = 0x4007_2100

Offset address: 0x0E

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

Bit	Symbol	Function	R/W
2:0	SPNDL[2:0]	SPI Next-Access Delay Setting 0 0 0: 1 RSPCK + 2 PCLKA 0 0 1: 2 RSPCK + 2 PCLKA 0 1 0: 3 RSPCK + 2 PCLKA 0 1 1: 4 RSPCK + 2 PCLKA 1 0 0: 5 RSPCK + 2 PCLKA 1 0 1: 6 RSPCK + 2 PCLKA 1 1 0: 7 RSPCK + 2 PCLKA 1 1 1: 8 RSPCK + 2 PCLKA	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

SPND specifies the next-access delay, the non-active period of the SSLni signal after termination of a serial transfer, when the SPCMDm.SPNDEN bit is 1.

SPNDL[2:0] bits (SPI Next-Access Delay Setting)

The SPNDL[2:0] bits specify a next-access delay when the SPCMDm.SPNDEN bit is 1. When using the SPI in slave mode, set the SPNDL[2:0] bits to 000b.

35.2.13 SPCR2 : SPI Control Register 2

Base address: SPI0 = 0x4007_2000
SPI1 = 0x4007_2100

Offset address: 0x0F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SCKA SE	PTE	SPIIE	SPOE	SPPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPPE	Parity Enable 0: Do not add parity bit to transmit data and do not check parity bit of receive data 1: When SPCR.TXMD = 0: Add parity bit to transmit data and check parity bit of receive data When SPCR.TXMD = 1: Add parity bit to transmit data but do not check parity bit of receive data	R/W
1	SPOE	Parity Mode 0: Select even parity for transmission and reception 1: Select odd parity for transmission and reception	R/W
2	SPIIE	SPI Idle Interrupt Enable 0: Disable idle interrupt requests 1: Enable idle interrupt requests	R/W
3	PTE	Parity Self-Testing 0: Disable self-diagnosis function of the parity circuit 1: Enable self-diagnosis function of the parity circuit	R/W

Bit	Symbol	Function	R/W
4	SCKASE	RSPCK Auto-Stop Function Enable 0: Disable RSPCK auto-stop function 1: Enable RSPCK auto-stop function	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

SPPE bit (Parity Enable)

The SPPE bit enables or disables the parity function.

When the SPCR.TXMD bit is 0 and this bit is 1, the parity bit is added to transmit data and parity checking is performed for receive data.

When the SPCR.TXMD bit is 1 and this bit is 1, the parity bit is added to transmit data but parity checking is not performed for receive data.

SPOE bit (Parity Mode)

The SPOE bit specifies odd or even parity.

When even parity is set, parity bit addition is performed so that the total number of bits whose value is 1 in the transmit or receive character plus the parity bit is even. Similarly, when odd parity is set, parity bit addition is performed so that the total number of bits whose value is 1 in the transmit or receive character plus the parity bit is odd.

The SPOE bit is only valid when the SPPE bit is 1.

SPIIE bit (SPI Idle Interrupt Enable)

The SPIIE bit enables or disables the generation of SPI idle interrupt requests when an idle state is detected in the SPI and the SPSR.IDLNF flag clears is set to 0.

PTE bit (Parity Self-Testing)

The PTE bit enables self-diagnosis of the parity circuit to check whether the parity function is operating correctly.

SCKASE bit (RSPCK Auto-Stop Function Enable)

The SCKASE bit enables or disables the RSPCK auto-stop function. When this function is enabled, the RSPCK clock is stopped before an overrun error occurs, when data is received in master mode. For details, see [section 35.3.8.1. Overrun errors](#).

35.2.14 SPCMDm : SPI Command Register m (m = 0 to 7)

Base address: SPI0 = 0x4007_2000
SPI1 = 0x4007_2100

Offset address: 0x10 + 0x02 × m

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SCKD EN	SLND EN	SPND EN	LSBF	SPB[3:0]			SSLK P	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA		
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1

Bit	Symbol	Function	R/W
0	CPHA	RSPCK Phase Setting 0: Select data sampling on leading edge, data change on trailing edge 1: Select data change on leading edge, data sampling on trailing edge	R/W
1	CPOL	RSPCK Polarity Setting 0: Set RSPCK low during idle 1: Set RSPCK high during idle	R/W

Bit	Symbol	Function	R/W
3:2	BRDV[1:0]	Bit Rate Division Setting 0 0: Base bit rate 0 1: Base bit rate divided by 2 1 0: Base bit rate divided by 4 1 1: Base bit rate divided by 8	R/W
6:4	SSLA[2:0]	SSL Signal Assertion Setting 0 0 0: SSL0 0 0 1: SSL1 0 1 0: SSL2 0 1 1: SSL3 Others: Setting prohibited	R/W
7	SSLKP	SSL Signal Level Keeping (only SPI0) 0: Negate all SSL signals on completion of transfer 1: Keep SSL signal level from the end of transfer until the beginning of the next access	R/W
11:8	SPB[3:0]	SPI Data Length Setting 0x0: 20 bits 0x1: 24 bits 0x2: 32 bits 0x3: 32 bits 0x8: 9 bits 0x9: 10 bits 0xA: 11 bits 0xB: 12 bits 0xC: 13 bits 0xD: 14 bits 0xE: 15 bits 0xF: 16 bits Others: 8 bits	R/W
12	LSBF	SPI LSB First 0: MSB-first 1: LSB-first	R/W
13	SPNDEN	SPI Next-Access Delay Enable 0: Select next-access delay of 1 RSPCK + 2 PCLKA 1: Select next-access delay equal to the setting in the SPI Next-Access Delay Register (SPND)	R/W
14	SLNDEN	SSL Negation Delay Setting Enable 0: Select SSL negation delay of 1 RSPCK 1: Select SSL negation delay equal to the setting in the SPI Slave Select Negation Delay Register (SSLND)	R/W
15	SCKDEN	RSPCK Delay Setting Enable 0: Select RSPCK delay of 1 RSPCK 1: Select RSPCK delay equal to the setting in the SPI Clock Delay Register (SPCKD)	R/W

[SPI0]

The SPCMDm registers specify the transfer format for the SPI in master mode. Each channel has eight SPCMDm (m = 0 to 7). Some of the bits in the SPCMD0 registers are used to set the transfer mode for the SPI in slave mode. The SPI in master mode sequentially references the SPCMDm registers based on the settings in the SPSCR.SPSSLN[2:0] bits and executes the serial transfer that is set in the referenced SPCMDm registers.

Set the SPCMDm registers while the transmit buffer is empty (SPSR.SPTEF is 1 and data for the next transfer is not set) and before the setting of the data to be transmitted when that SPCMDm registers is referenced.

The SPCMDm registers referenced by the SPI in master mode can be checked by means of the SPSSR.SPCP[2:0] bits.

[SPI1]

The SPCMDm register sets the transfer format for the SPI in master mode.

Set this register while the transmit buffer is empty (SPSR.SPTEF is 1 and data for the next transfer is not set), and before the setting of data to be transmitted when this register is referenced.

CPHA bit (RSPCK Phase Setting)

The CPHA bit selects the RSPCK phase of the SPI in master or slave mode. Data communications between SPI modules require the same RSPCK phase setting between the modules.

CPOL bit (RSPCK Polarity Setting)

The CPOL bit selects the RSPCK polarity of the SPI in master or slave mode. Data communications between SPI modules require the same RSPCK polarity setting between the modules.

BRDV[1:0] bits (Bit Rate Division Setting)

The BRDV[1:0] bits determine the bit rate in combination with the settings in the SPBR register. (see [section 35.2.8. SPBR : SPI Bit Rate Register](#)). The SPBR settings determine the base bit rate. The BRDV[1:0] setting selects the bit rate obtained by dividing the base bit rate by 1, 2, 4, or 8. Different BRDV[1:0] bit settings can be specified in the SPCMD0 register. This enables execution of serial transfers at different bit rates for each command.

SSLA[2:0] bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSLn_i signal assertion when the SPI performs serial transfers in master mode. When an SSLn_i signal is asserted, its polarity is determined by the value set in the associated SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSLn₀ pin acts as input).

When using the SPI in slave mode, set the SSLA[2:0] bits to 000b.

SSLKP bit (SSL Signal Level Keeping)

When the SPI in master mode performs a serial transfer, the SSLKP bit specifies whether the SSLn_i signal level for the current command is to be kept or negated between the SSL negation associated with the current command and the SSL assertion associated with the next command. Setting the SSLKP bit to 1 enables a burst transfer. For details, see [section 35.3.10.1. Master mode operation](#). When using the SPI in slave mode, set the SSLKP bit to 0.

SPB[3:0] bits (SPI Data Length Setting)

The SPB[3:0] bits specify the transfer data length for the SPI in master or slave mode.

LSBF bit (SPI LSB First)

The LSBF bit specifies the data format of the SPI in master or slave mode to MSB-first or LSB-first.

SPNDEN bit (SPI Next-Access Delay Enable)

The SPNDEN bit specifies the next-access delay, the period from the time the SPI in master mode terminates a serial transfer and sets the SSLn_i signal inactive until the SPI enables the SSLn_i signal assertion for the next access. If the SPNDEN bit is 0, the SPI sets the next-access delay to 1 RSPCK + 2 PCLKA. 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 SSLn_i signal to inactive. If the SLNDEN bit is 0, the SPI sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the SPI negates the SSL signal at the SSL negation delay according to the SSLND setting.

When using the SPI in slave mode, set the SLNDEN bit to 0.

SCKDEN bit (RSPCK Delay Setting Enable)

The SCKDEN bit specifies the SPI clock delay, the period from the point when the SPI in master mode asserts the SSLn_i signal until the RSPCK starts oscillation. If the SCKDEN bit is 0, the SPI sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, the SPI starts the oscillation of RSPCK at an RSPCK delay according to the SPCKD setting.

When using the SPI in slave mode, set the SCKDEN bit to 0.

35.3 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.

35.3.1 Overview of SPI Operation

The SPI is capable of synchronous serial transfers in the following modes:

- Slave mode (SPI operation)
- Single master mode (SPI operation)
- Multi-master mode (SPI operation)
- Slave mode (clock synchronous operation)
- Master mode (clock synchronous operation)

The SPI mode can be selected by using the MSTR, MODFEN, and SPMS bits in SPCR. [Table 35.5](#) lists the relationship between SPI modes and SPCR settings, and a description of each mode.

Table 35.5 Relationship between SPCR settings and SPI modes (1 of 2)

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
RSPCKn pins	Input	Output	Output/Hi-Z	Input	Output
MOSIn pin	Input	Output	Output/Hi-Z	Input	Output
MISO pin	Output/Hi-Z	Input	Input	Output	Input
SSLn0 pins	Input	Output	Input	Hi-Z ^{*1}	Hi-Z ^{*1}
SSLn1 to SSLn3 pins	Hi-Z ^{*1}	Output	Output/Hi-Z	Hi-Z ^{*1}	Hi-Z ^{*1}
SSL polarity change function	Supported	Supported	Supported	—	—
Max transfer rate	PCLKA/4	PCLKA/2	PCLKA/2	PCLKA/4	PCLKA/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two				
Clock phase	Two	Two	Two	One (CPHA = 1)	Two
First transfer bit	MSB/LSB				
Transfer data length	8 to 16, 20, 24, 32 bits				
Burst transfer ^{*5}	Possible (CPHA = 1)	Possible (CPHA = 0, 1)	Possible (CPHA = 0, 1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer trigger	SSL input active or RSPCK oscillation	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)	RSPCK oscillation	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)
Sequence control ^{*5}	Not supported	Supported	Supported	Not supported	Not supported

Table 35.5 Relationship between SPCR settings and SPI modes (2 of 2)

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
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* ^{3,2}				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported
Underrun error detection	Supported	Not supported	Not supported	Supported	Not supported

Note 1. This function is not supported in this mode.

Note 2. When the SPCR.TXMD bit is 1, detection of receiver buffer full, overrun error, and parity error are not performed.

Note 3. When the SPCR2.SPPE bit is 0, parity error detection is not performed.

Note 4. When the SPCR2.SCKASE bit is 1, overrun error detection does not proceed.

Note 5. Only SPI0

35.3.2 Controlling the SPI Pins

Based on the settings of the MSTR, MODFEN, and SPMS bits in SPCR and the PmnPFS.NCODR bit for I/O Ports, the SPI can switch pin states. Table 35.6 lists the relationship between pin states and bit settings. Setting the PmnPFS.NCODR bit for an I/O port to 0 selects the CMOS output. Setting it to 1 selects the open-drain output. The I/O port settings must follow this relationship.

Table 35.6 Relationship between pin states and bit settings (1 of 2)

Mode	Pin	Pin state* ²	
		PmnPFS.NCODR bit for I/O ports = 0	PmnPFS.NCODR bit for I/O ports = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3	CMOS output	Open-drain output
	MOSIn	CMOS output	Open-drain output
	MISO _n	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCKn* ³	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSLn0	Input	Input
	SSLn1 to SSLn3* ³	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSIn* ³	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISO _n	Input	Input
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCKn	Input	Input
	SSLn0	Input	Input
	SSLn1 to SSLn3* ⁵	Hi-Z* ¹	Hi-Z* ¹
	MOSIn	Input	Input
	MISO _n * ⁴	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3* ⁵	Hi-Z* ¹	Hi-Z* ¹
	MOSIn	CMOS output	Open-drain output
	MISO _n	Input	Input

Table 35.6 Relationship between pin states and bit settings (2 of 2)

Mode	Pin	Pin state ^{*2}	
		PmnPFS.NCODR bit for I/O ports = 0	PmnPFS.NCODR bit for I/O ports = 1
Slave mode (clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCKn	Input	Input
	SSLn0 to SSLn3 ^{*5}	Hi-Z ^{*1}	Hi-Z ^{*1}
	MOSIn	Input	Input
	MISO _n	CMOS output	Open-drain output

Note 1. This function is not supported in this mode.

Note 2. SPI settings are not reflected in multiplexed pins for which the SPI function is not selected.

Note 3. When SSLn0 is at the active level, the pin state is Hi-Z. Whether or not the input signal is at the active level determines the setting of the SSLP.SSL0P bit.

Note 4. When SSLn0 is at the non-active level or the SPCR.SPE bit is 0, the pin state is Hi-Z. Whether or not the input signal is at the active level determines the setting of the SSLP.SSL0P bit.

Note 5. These pins are available for use as I/O port pins.

The SPI in single-master mode (SPI operation) or multi-master mode (SPI operation) determines the MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer (only SPI0)) based on the MOIFE and MOIFV bit settings in SPPCR, as listed in [Table 35.7](#).

Table 35.7 MOSI signal value determination during SSL negation

MOIFE bit	MOIFV bit	MOSIn signal value during SSL negation
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

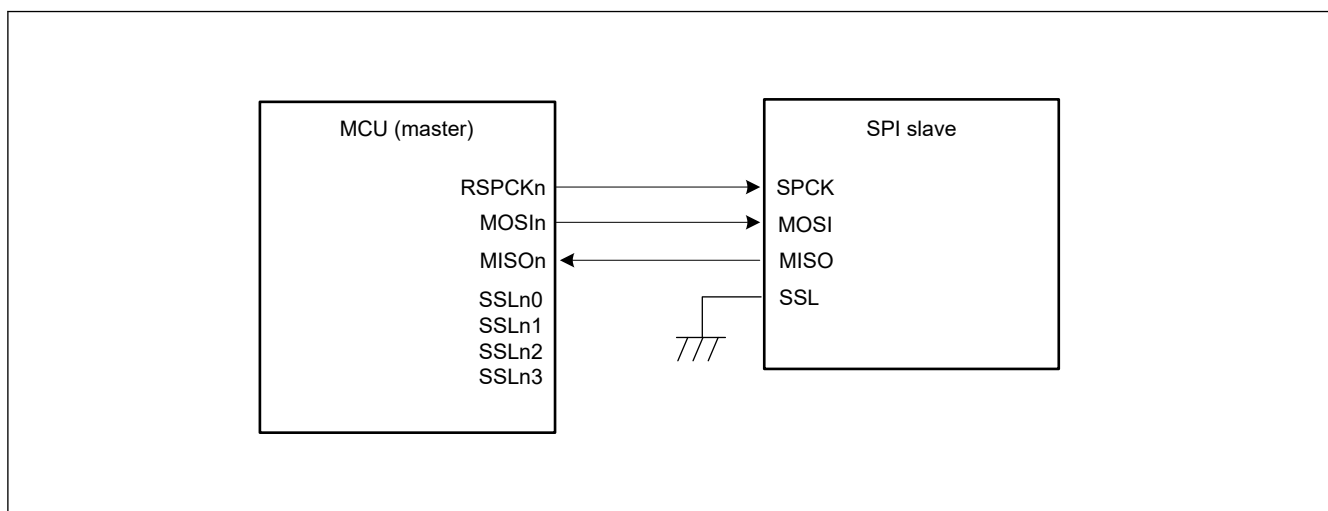
35.3.3 SPI System Configuration Examples

35.3.3.1 Single-master/single-slave with the MCU as a master

[Figure 35.5](#) shows a single-master/single-slave SPI system configuration example where the MCU is used as a master. In the single-master/single-slave configuration, the 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 is maintained in the selected state.^{*1}

Note 1. In the transfer format configured when the SPCMDm.CPHA bit is 0, the SSL signal for some slave devices cannot be fixed to an active level. In this case, always connect the SSLn_i output of the MCU to the SSL input of the slave device.

The MCU (master) drives the RSPCKn and MOSIn signals. The SPI slave drives the MISO signal.

**Figure 35.5 Single-master/single-slave configuration example with the MCU as a master**

35.3.3.2 Single-master/single-slave with the MCU as a slave

Figure 35.6 shows a single-master/single-slave SPI system configuration example where the MCU is used as a slave. When the MCU operates as a slave, the SSLn0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI signals. The MCU (slave) drives the MISO signal.^{*1}

Note 1. When SSLn0 is at a non-active level, the pin state is Hi-Z.

In the single-slave configuration when the SPCMDm.CPHA bit is set to 1, the SSLn0 input of the MCU (slave) is fixed to the low level and the MCU (slave) is maintained in the selected state. This enables serial transfer execution (Figure 35.7).

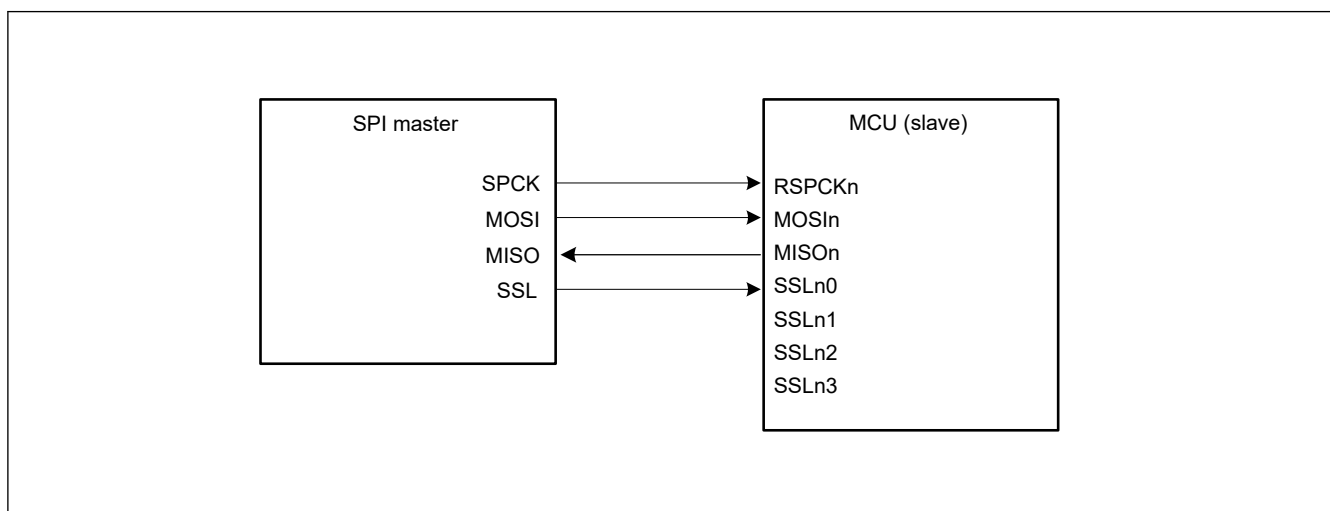


Figure 35.6 Single-master/single-slave configuration example with the MCU as a slave and CPHA = 0

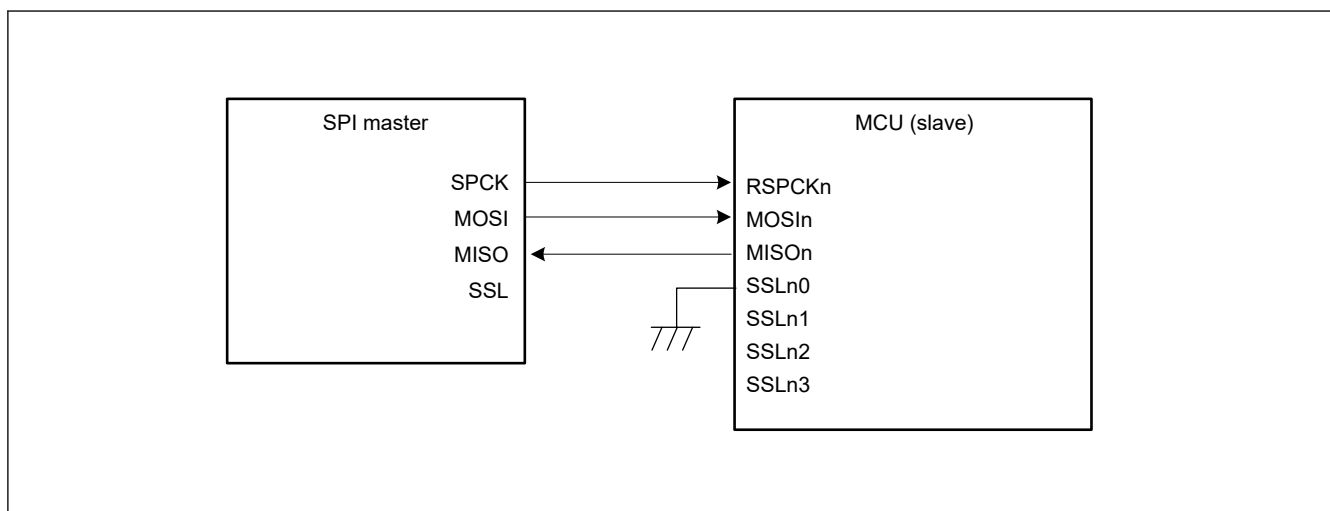


Figure 35.7 Single-master/single-slave configuration example with the MCU as a slave and CPHA = 1

35.3.3.3 Single-master/multi-slave with the MCU as a master

Figure 35.8 shows a single-master/multi-slave SPI system configuration example where the MCU is used as a master. In this example, the SPI system includes the MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKn and MOSIn outputs of the MCU (master) are connected to the RSPCK and MOSI inputs of SPI slaves 0 to 3. The MISO outputs of SPI slaves 0 to 3 are all connected to the MISO input of the MCU (master). The SSLn0 to SSLn3 outputs of the MCU (master) are connected to the SSL inputs of SPI slaves 0 to 3, respectively.

The MCU (master) drives the RSPCKn, MOSIn, and SSLn0 to SSLn3 signals. Of the SPI slaves 0 to 3, the slave that receives low-level input into the SSL input drives the MISO signal.

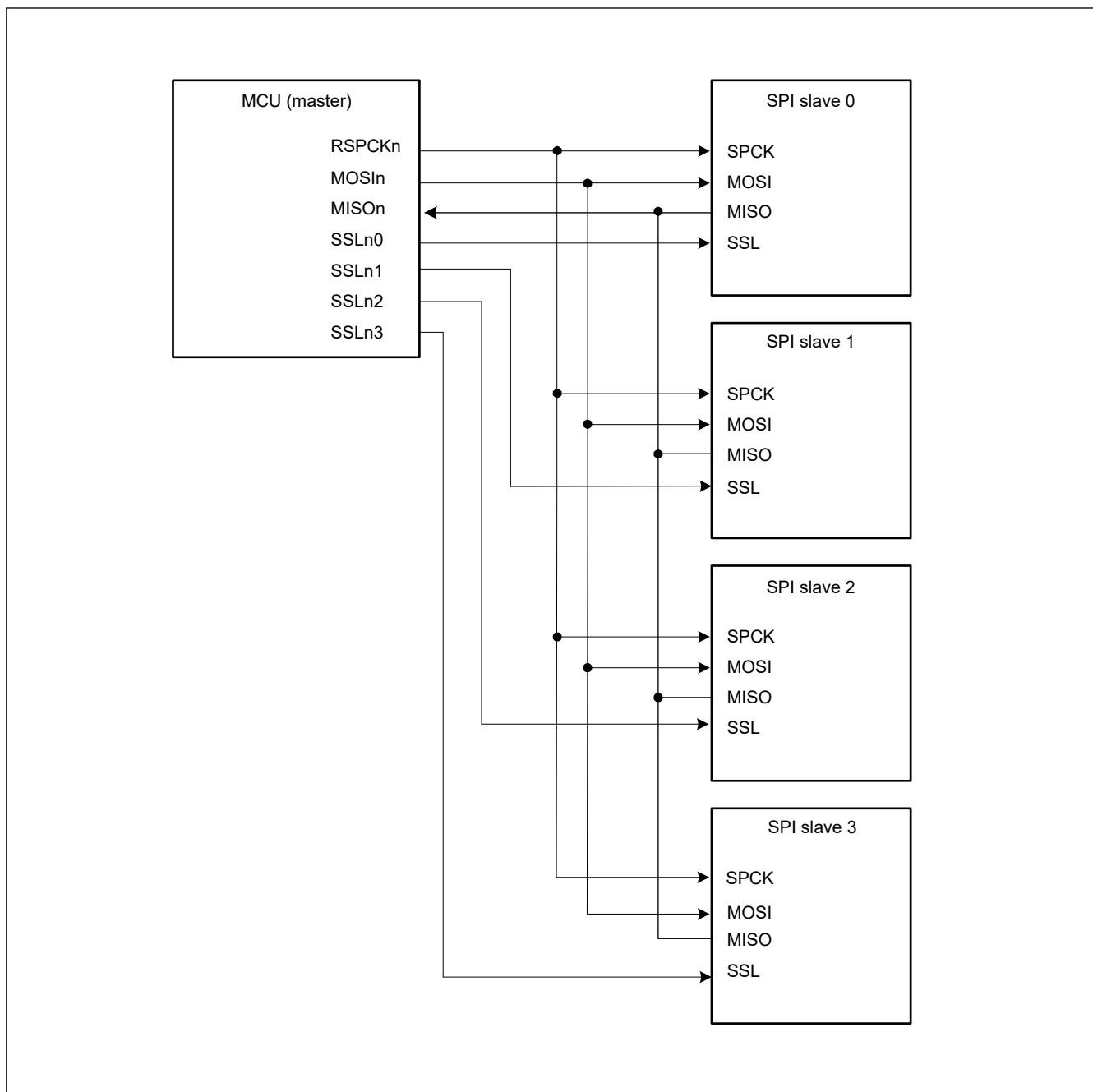


Figure 35.8 Single-master/multi-slave configuration example with the MCU as a master

35.3.3.4 Single-master/multi-slave with the MCU as a slave

Figure 35.9 shows a single-master/multi-slave SPI system configuration example where the MCU is used as a slave. In this example, the SPI system includes an SPI master and two MCUs (slaves X and Y).

The SPCK and MOSI outputs of the SPI master are connected to the RSPCKn and MOSIn inputs of the MCUs (slaves X and Y). The MISO outputs of the MCUs (slaves X and Y) are all connected to the MISO input of the SPI master. The SSLX and SSLY outputs of the SPI master are connected to the SSLn0 inputs of the MCUs (slaves X and Y, respectively).

The SPI master drives the SPCK, MOSI, SSLX, and SSLY signals. Of the MCUs (slaves X and Y), the slave that receives low-level input into the SSLn0 input drives the MISO output signal.

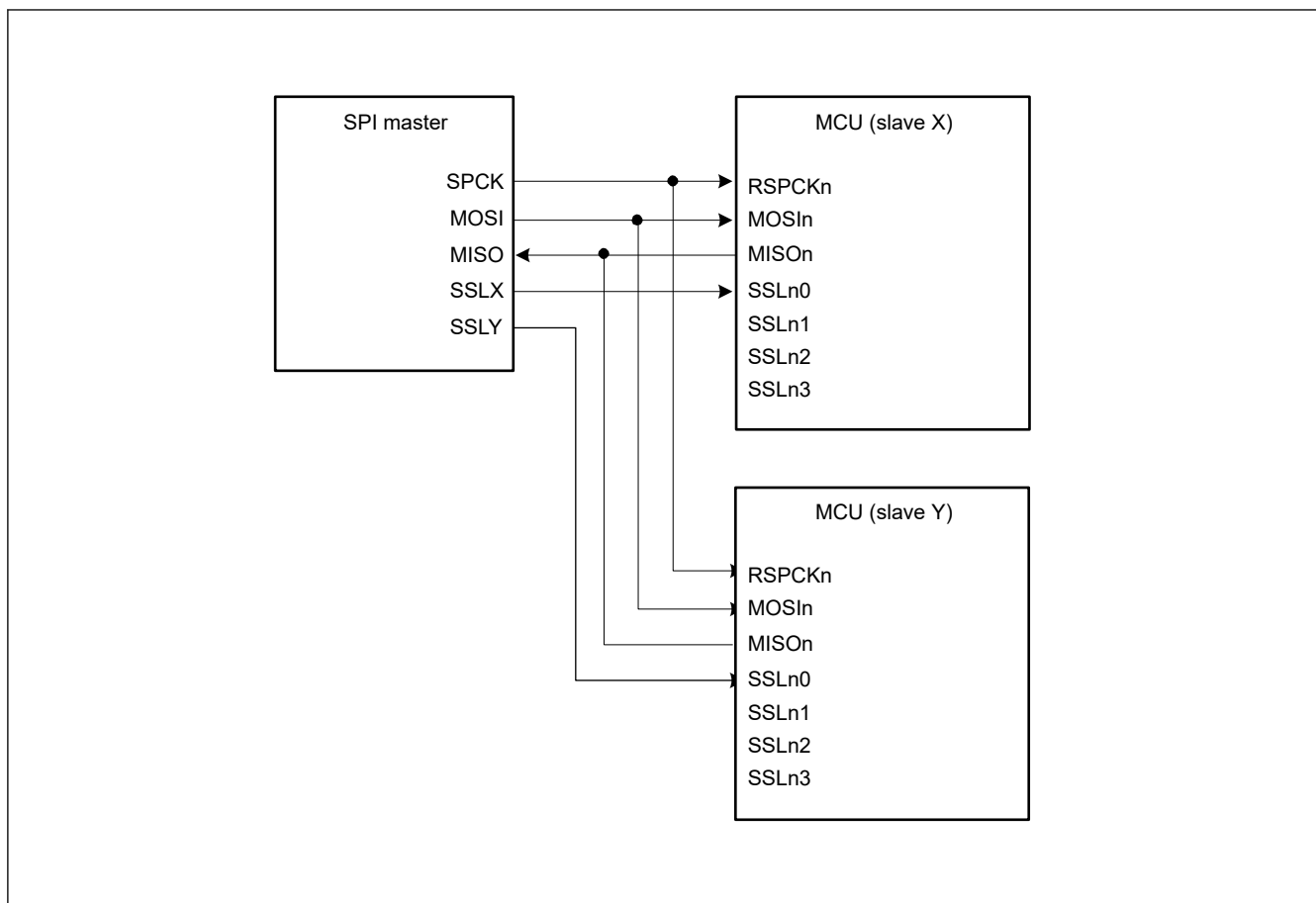


Figure 35.9 Single-master/multi-slave configuration example with the MCU as a slave

35.3.3.5 Multi-master/multi-slave with the MCU as a master

Figure 35.10 shows a multi-master/multi-slave SPI system configuration example where the MCU is used as a master. In this example, the SPI system includes two MCUs (masters X and Y) and two SPI slaves (SPI slaves 1 and 2).

The RSPCKn and MOSIn outputs of the MCUs (masters X and 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 MISOOn inputs of the MCUs (masters X and 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 (masters X and Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, because the system can be comprised solely of SSLn0 input, and SSLn1 and SSLn2 outputs for slave connections, the SSLn3 output of the MCU is not required.

The MCU drives the RSPCKn, MOSIn, SSLn1, and SSLn2 signals 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.

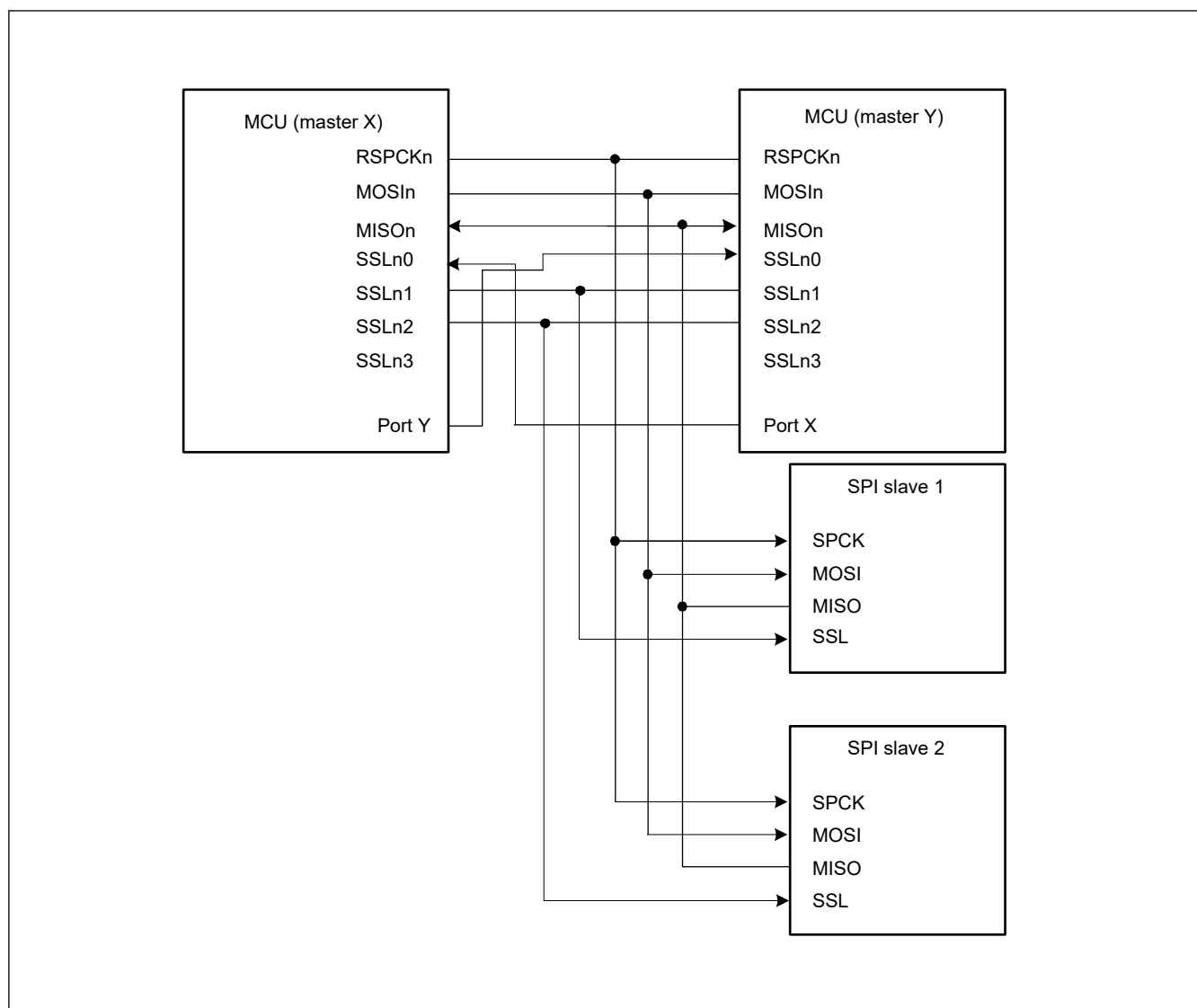


Figure 35.10 Multi-master/multi-slave configuration example with the MCU as a master

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

Figure 35.11 shows a master and slave in clock synchronous mode configuration example where the MCU is used as 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.

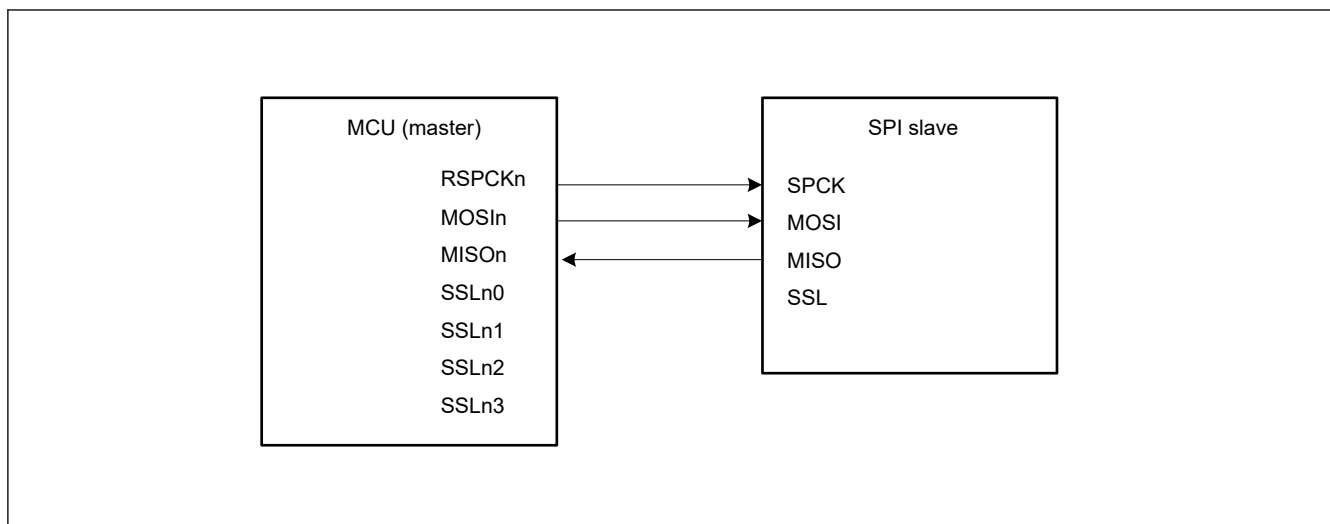


Figure 35.11 Clock synchronous master/slave configuration example with the MCU as a master

35.3.3.7 Master and slave in clock synchronous mode with the MCU as a slave

Figure 35.12 shows a master and slave in clock synchronous mode configuration example where the MCU is used as a slave. When the MCU operates as a slave (clock synchronous operation), the MCU (slave) drives the MISOOn signal and the SPI master drives the SPCK and MOSI signals. In addition, SSLn0 to SSLn3 of the MCU (slave) are not used.

The MCU (slave) can only execute serial transfers in the single-slave configuration when the SPCMDm.CPHA bit is set to 1.

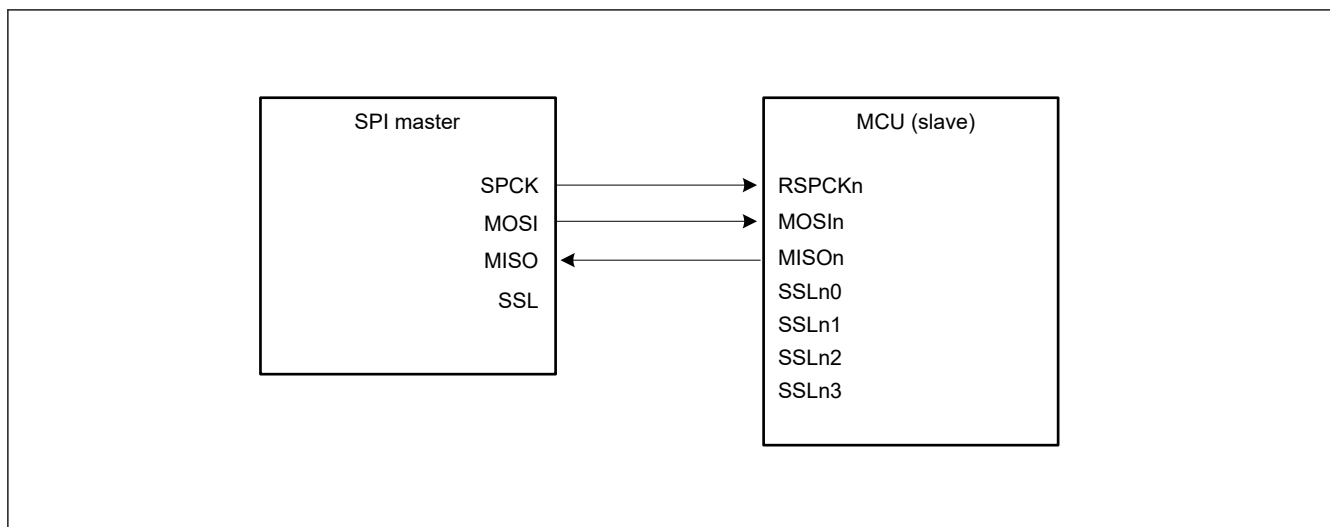


Figure 35.12 Clock synchronous master/slave configuration example with the MCU as a slave and CPHA = 1

35.3.4 Data Formats

The data format of the SPI depends on the settings in SPI Command Register m (SPCMDm)^{*1} and the parity enable bit in SPI Control Register 2 (SPCR2.SPPE). Regardless of whether the MSB or LSB is first, the SPI treats the range from the LSB bit in the SPI Data Register (SPDR/SPDR_HA) to the bit associated with the selected data length, as transfer data.

Note 1. For SPI0, m = 0 to 7; for SPI1, m = 0.

This section shows the format of one frame of data before or after transfer.

Data format with parity disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the SPI data length setting in SPI Command Register m (SPCMDm.SPB[3:0]).

Data format with parity enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the SPI data length setting in SPI Command Register m (SPCMDm.SPB[3:0]). In this case, however, the last bit is a parity bit.

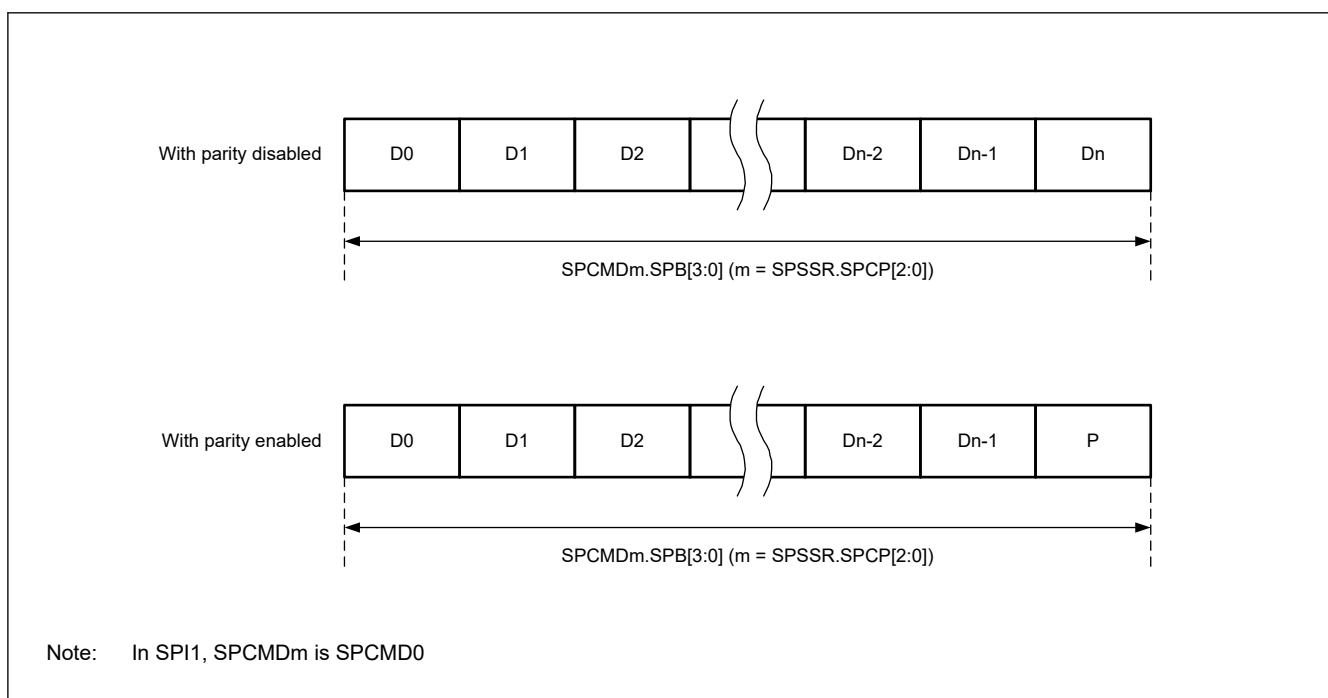


Figure 35.13 Data format with parity disabled and enabled

35.3.4.1 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 35.14 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, a SPI data length of 32 bits, and MSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission is shifted out from the shift register from T31 to T30, and continuing to T00.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R31 to R00 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.

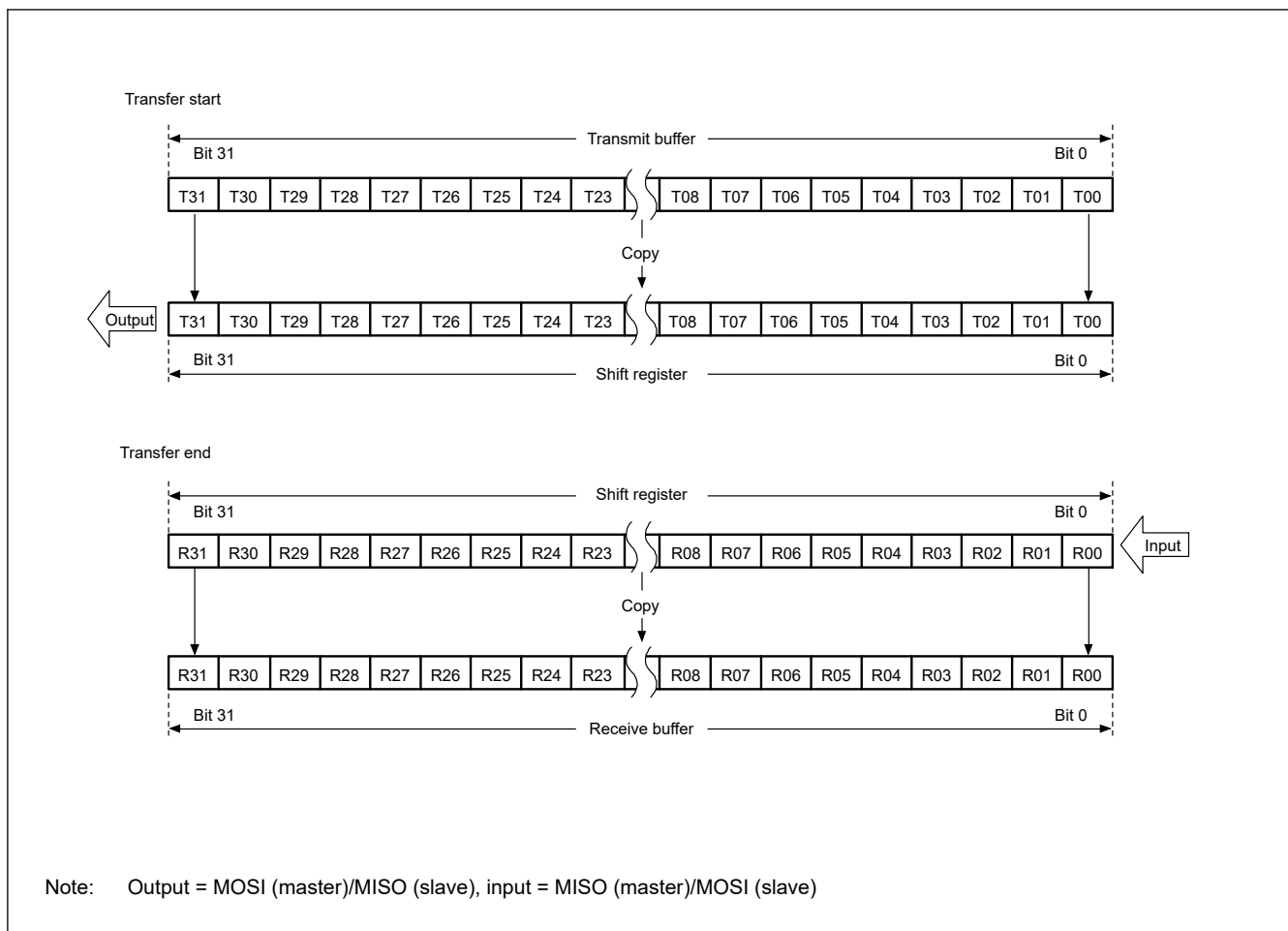


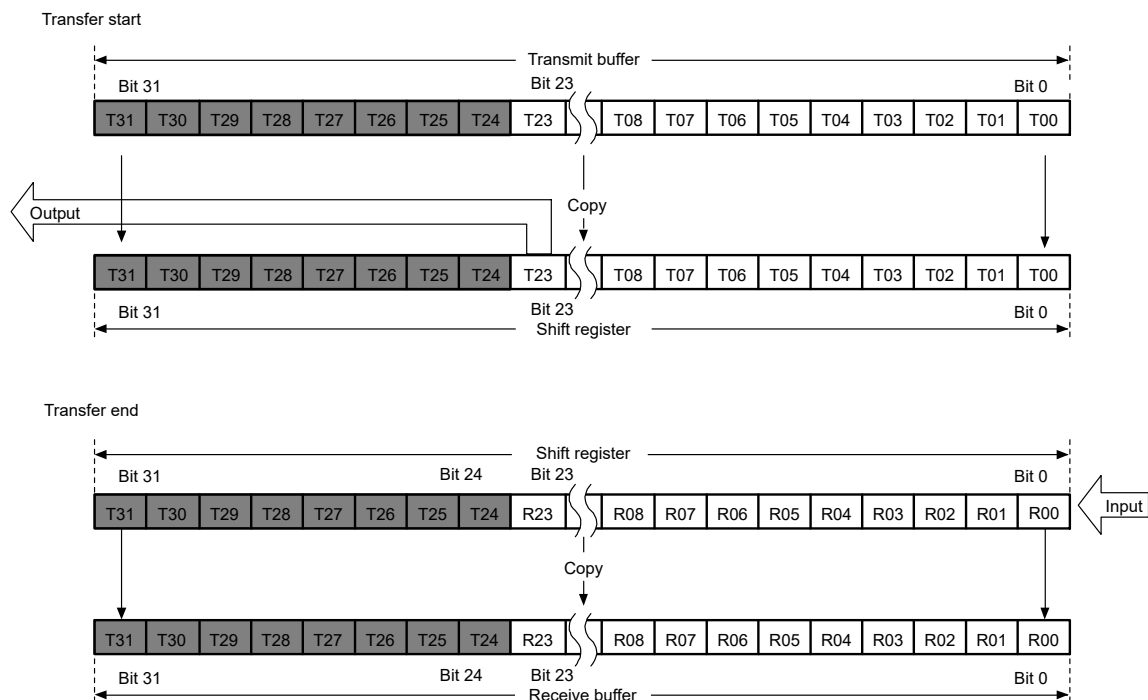
Figure 35.14 MSB-first transfer with 32-bit data and parity disabled

(2) MSB-first transfer with 24-bit data

Figure 35.15 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 the R23 to R00 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. The upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 during transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 35.15 MSB-first transfer with 24-bit data and parity disabled

(3) LSB-first transfer with 32-bit data

Figure 35.16 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 in order from T00 to T01, and continuing to T31.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R00 to R31 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.

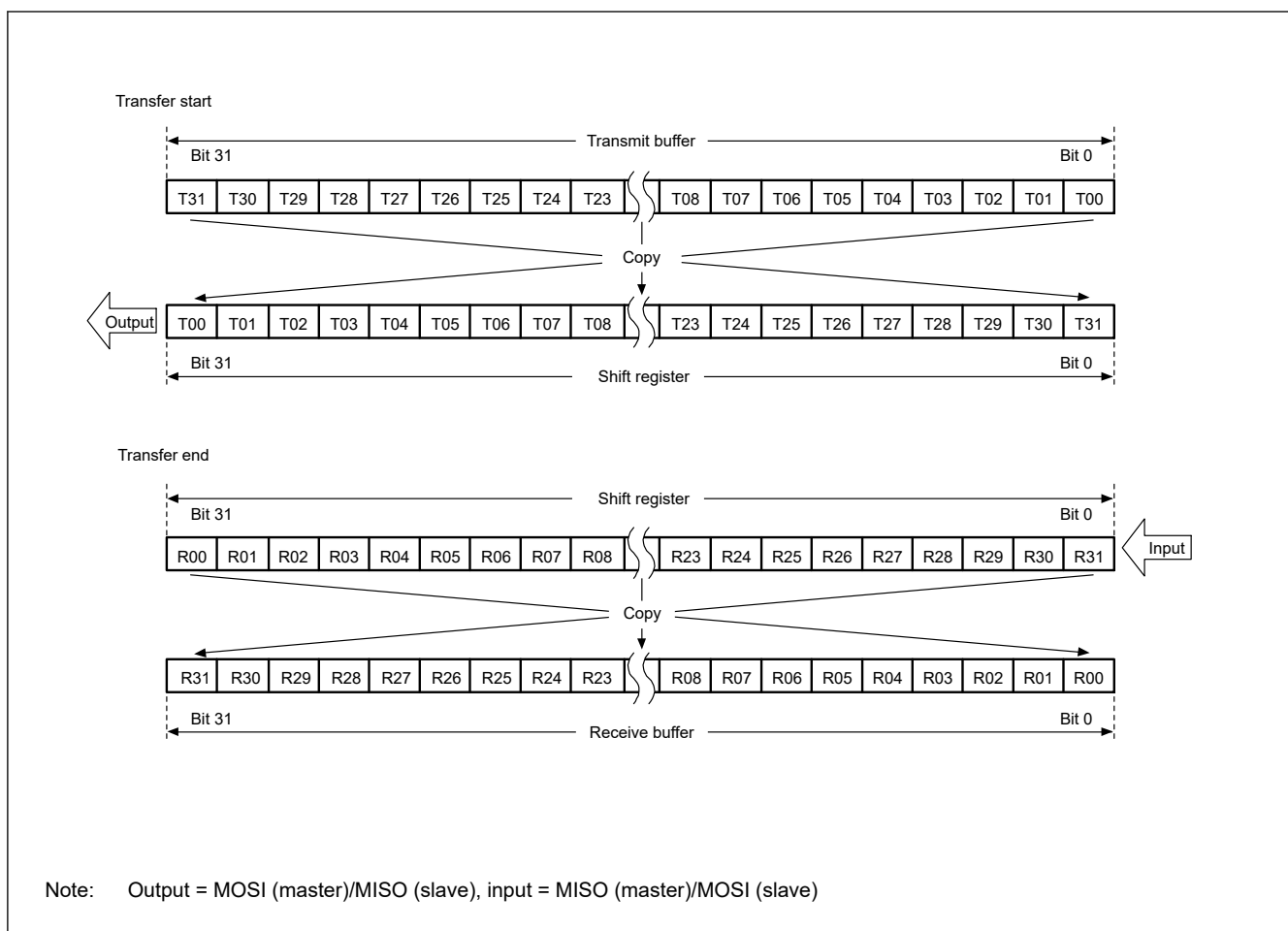
**Figure 35.16 LSB-first transfer with 32-bit data and parity disabled****(4) LSB-first transfer with 24-bit data**

Figure 35.17 shows the operation of the SPI Data Register (SPDR) and the shift register in transfers with parity disabled, an SPI data length of 24 bits for an example that is not 32, and LSB-first selected.

In transmission, the lower 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit-by-bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission is shifted out from the shift register from T00 to T01, and continuing to T23.

In reception, received data is shifted in bit-by-bit through bit[8] of the shift register. When the R00 to R23 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.

The upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to T31 to T24 during transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.

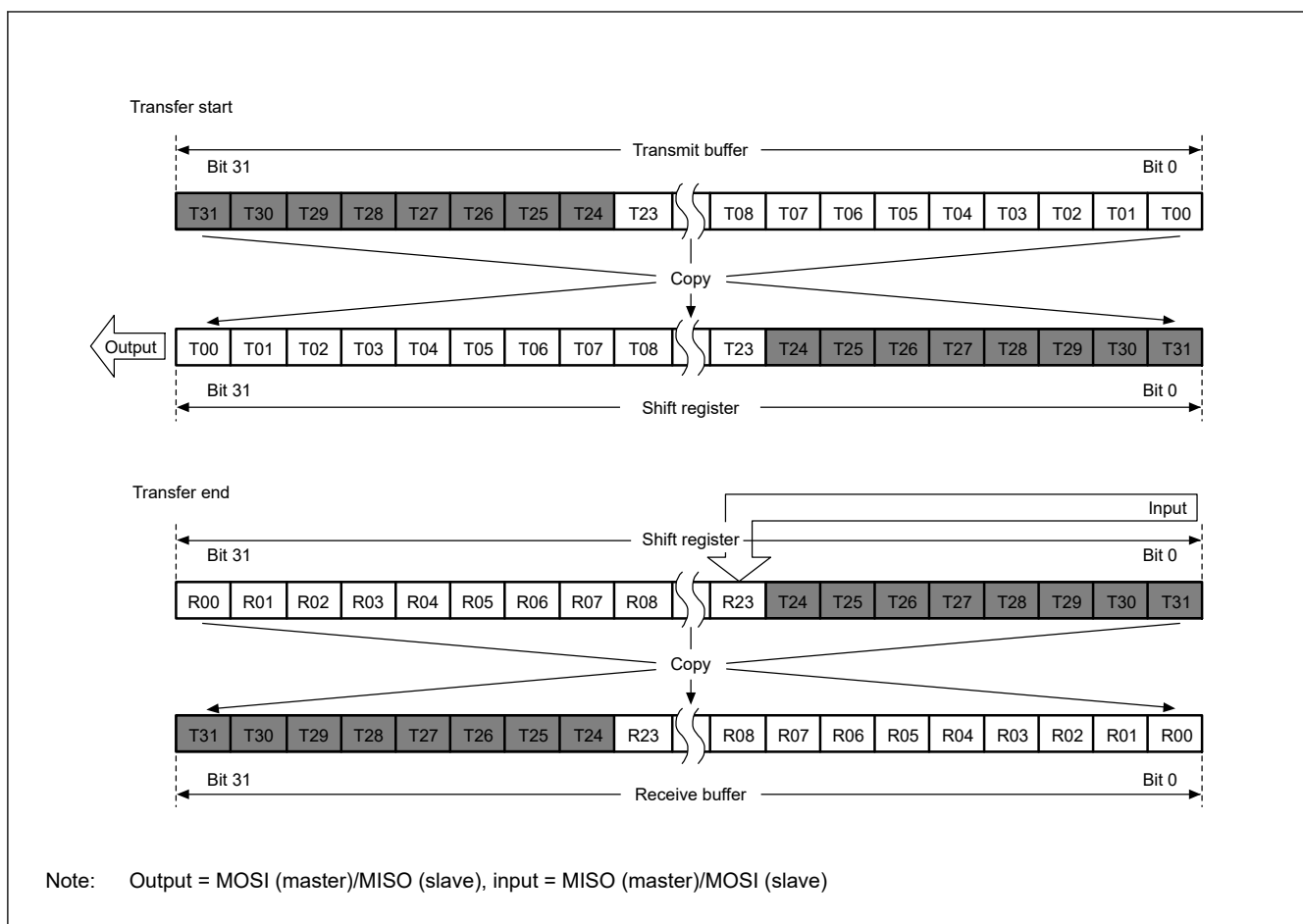


Figure 35.17 LSB-first transfer with 24-bit data and parity disabled

35.3.4.2 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 35.18 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 value is copied to the shift register. Data is transmitted in the order T31, T30, ..., T01, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R31 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R31 to P is checked for parity.

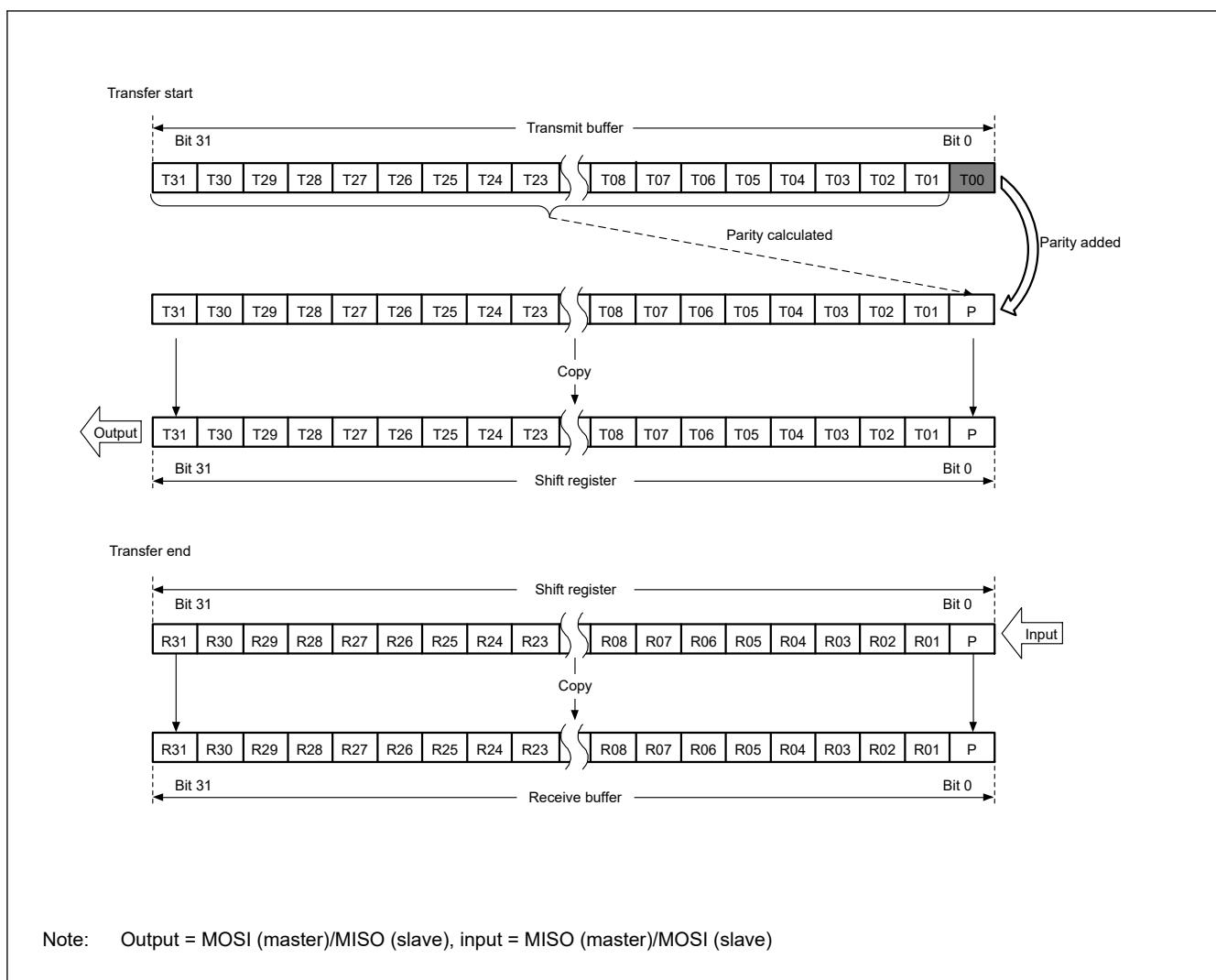


Figure 35.18 MSB-first transfer with 32-bit data and parity enabled

(2) MSB-first transfer with 24-bit data

Figure 35.19 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, a SPI data length of 24 bits, and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole value is copied to the shift register. Data is transmitted in the order T23, T22, ..., T01, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R23 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P is checked for parity. The upper 8 bits of the transmit buffer is stored in the upper 8 bits of the receive buffer. Writing 0 to T31 to T24 during transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.

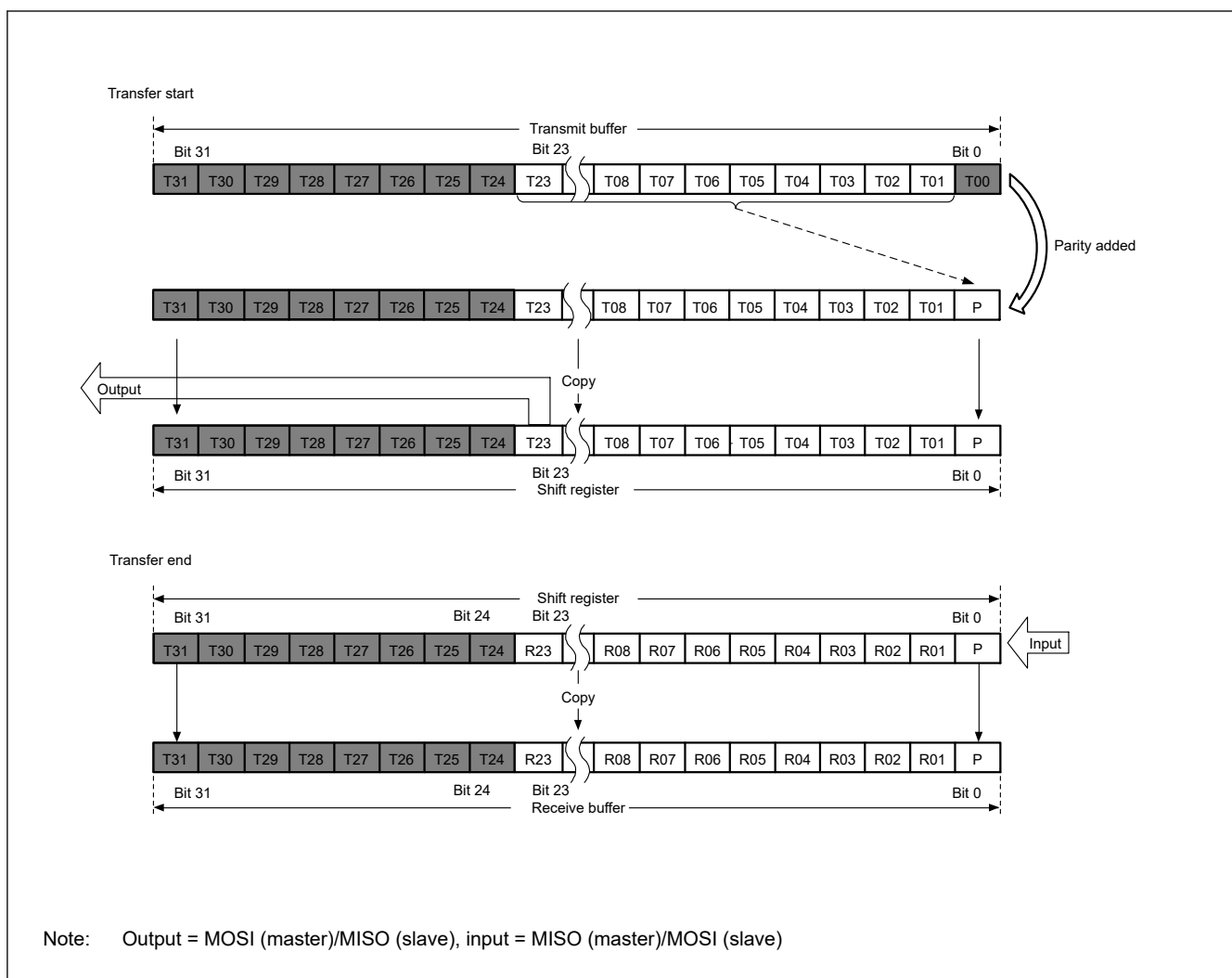


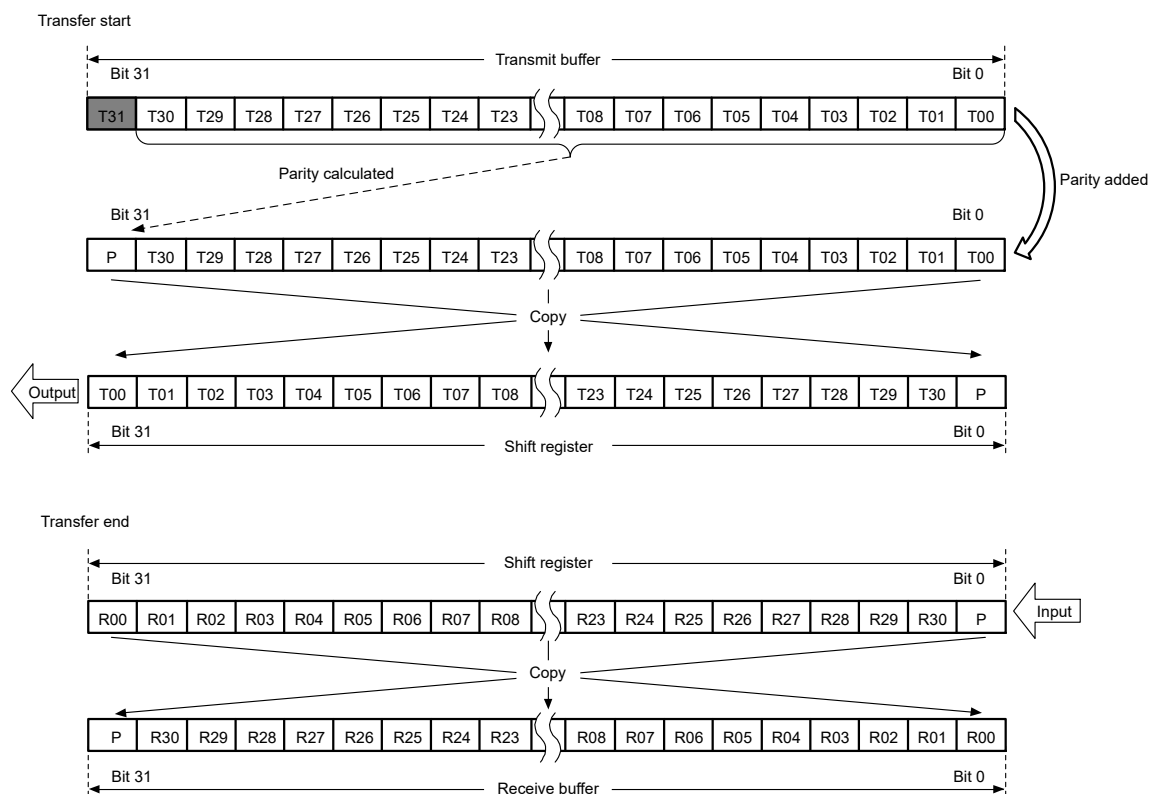
Figure 35.19 MSB-first transfer with 24-bit data and parity enabled

(3) LSB-first transfer with 32-bit data

Figure 35.20 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 the R00 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 35.20 LSB-first transfer with 32-bit data and parity enabled

(4) LSB-first transfer with 24-bit data

Figure 35.21 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, a SPI data length of 24 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole value is copied to the shift register. Data is transmitted in the order T00, T01, ..., T22, and P.

In reception, received data is shifted in bit-by-bit through bit[8] of the shift register. When the R00 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P is checked for parity. The upper 8 bits of the transmit buffer is stored in the upper 8 bits of the receive buffer. Writing 0 to T31 to T24 during transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.

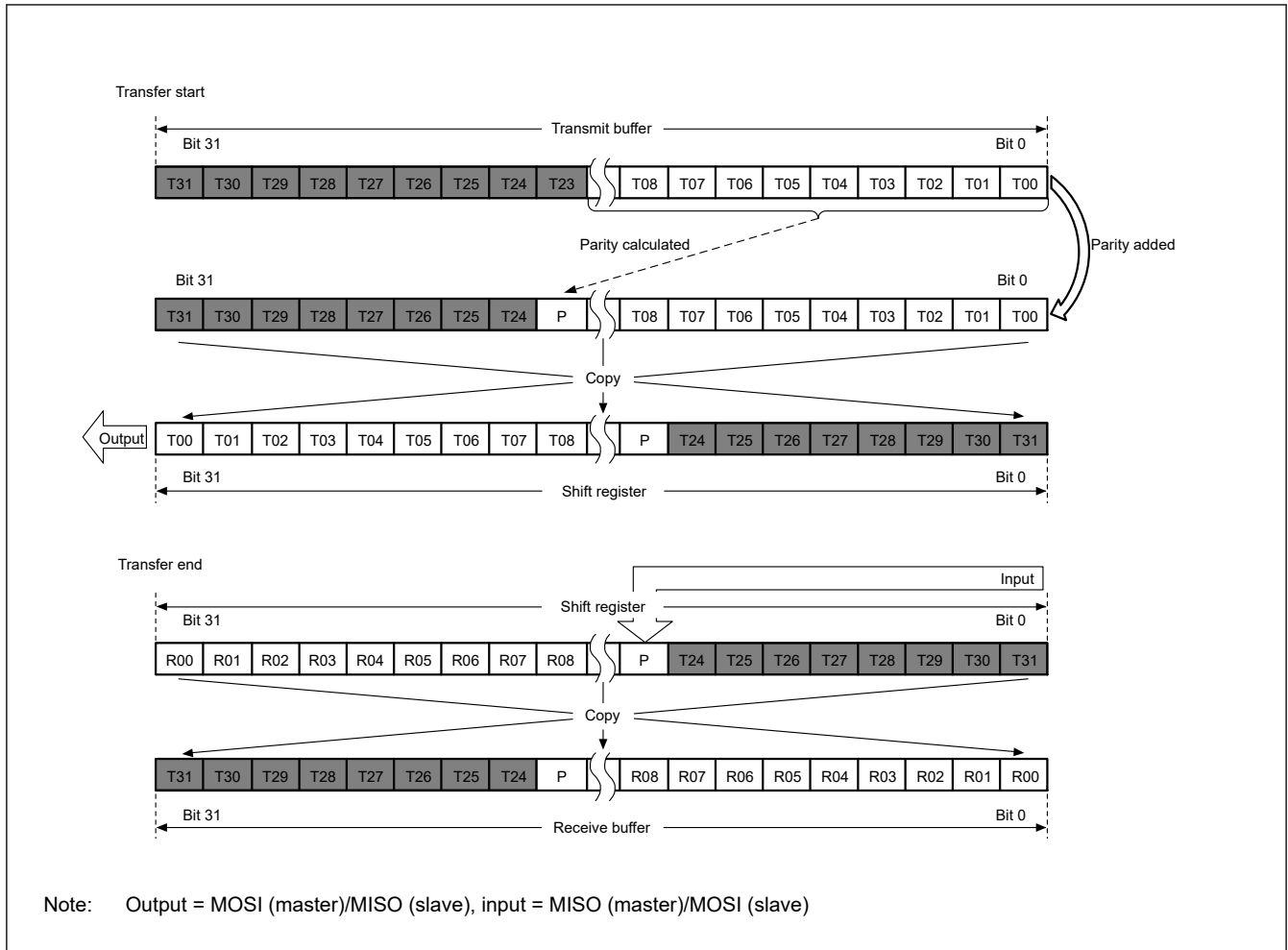


Figure 35.21 LSB-first transfer with 24-bit data and parity enabled

35.3.5 Transfer Formats

35.3.5.1 When CPHA = 0

Figure 35.22 shows an example transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Do not perform clock synchronous operation (SPCR.SPMS = 1) when the SPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 35.22, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.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 35.3.2. Controlling the SPI Pins](#).

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIn and MISO signals begins at an SSLni signal assertion. The first RSPCKn signal change that occurs after the SSLni signal assertion becomes the first transfer data fetch. After this, data is sampled every 1 RSPCK cycle. The change timing for MOSIn and MISO signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCK signal operation timing as it only affects the signal polarity.

t1 denotes the RSPCK delay, the period from an SSLni signal assertion to RSPCKn oscillation. t2 denotes the SSL negation delay, the period from the termination of RSPCKn oscillation to an SSLni signal negation. t3 denotes the next-access delay, the period in which SSLni signal assertion is suppressed for the next transfer after the end of serial transfer. t1, t2, and t3 are controlled by a master device running on the SPI system. For a description of t1, t2, and t3 when the SPI is in master mode, see [section 35.3.10.1. Master mode operation](#).

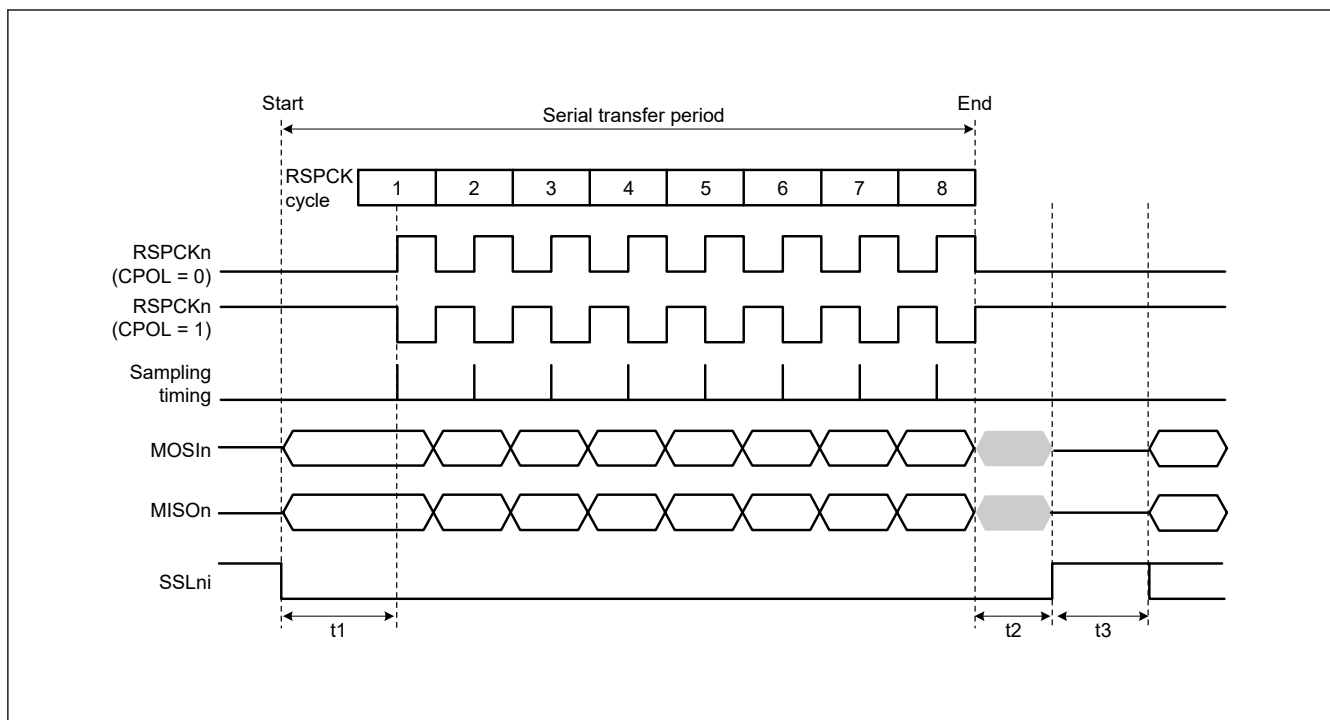


Figure 35.22 SPI transfer format when CPHA = 0

35.3.5.2 When CPHA = 1

[Figure 35.23](#) shows an example transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLni signals are not used, and only the three signals RSPCKn, MOSIn, and MISOOn handle communications. In [Figure 35.23](#), RSPCK (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0 and RSPCK (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI mode (master or slave mode). For details, see [section 35.3.2. Controlling the SPI Pins](#).

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOOn signal begins at an SSLni signal assertion. The output of valid data to the MOSIn and MISOOn signals begins at the first RSPCKn signal change that occurs after the SSLni signal assertion. After this, data is updated every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKn signal operation timing. It only affects the signal polarity.

t1, t2, and t3 are the same as those when CPHA = 0. For a description of t1, t2, and t3 when the SPI of the MCU is in master mode, see [section 35.3.10.1. Master mode operation](#).

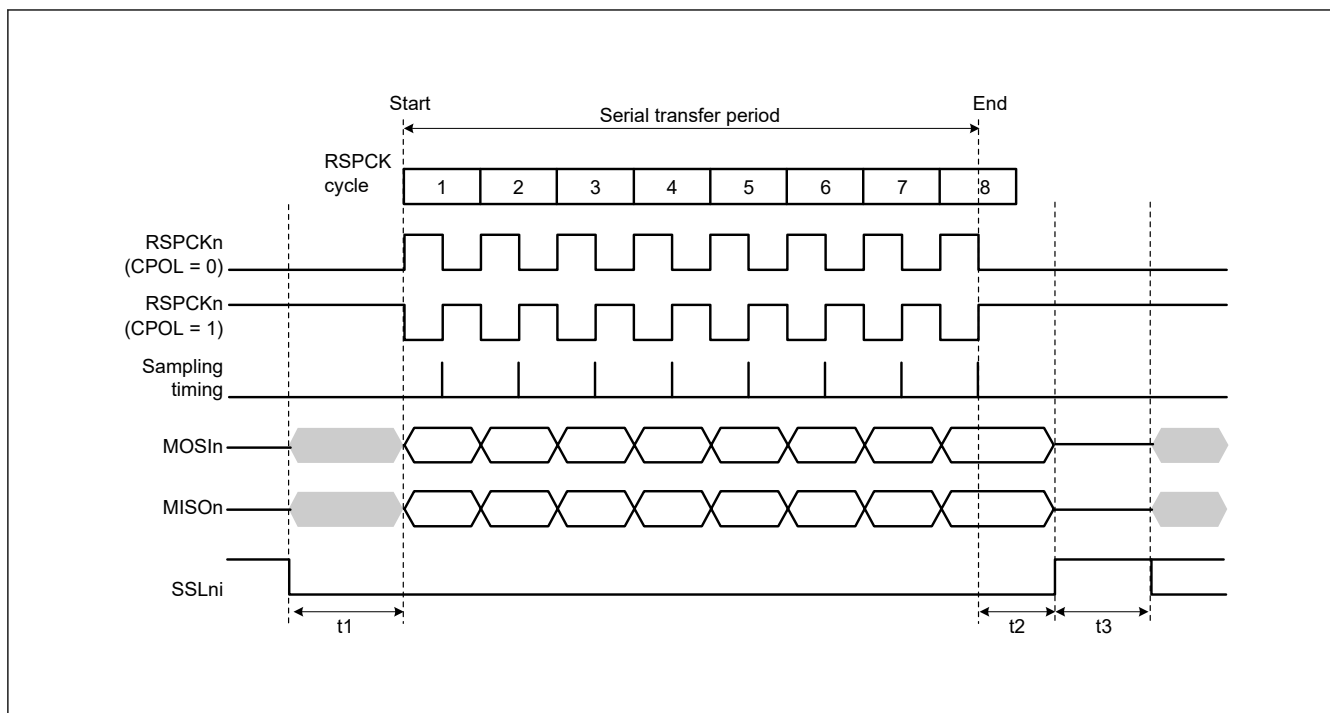


Figure 35.23 SPI transfer format when CPHA = 1

35.3.6 Data Transfer Modes

Full-duplex synchronous serial communications or transmit operations can only be selected in the Communications Operating Mode Select bit (SPCR.TXMD). The register accesses shown in [Figure 35.24](#) and [Figure 35.25](#) indicate the condition of access to the SPDR/SPDR_HA register, where W denotes a write cycle.

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

[Figure 35.24](#) shows an example of operation when the Communications Operating Mode Select bit (SPCR.TXMD) is set to 0. In this example, the SPI performs an 8-bit serial transfer when SPDCR.SPFC[1:0] (only SPI0) bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

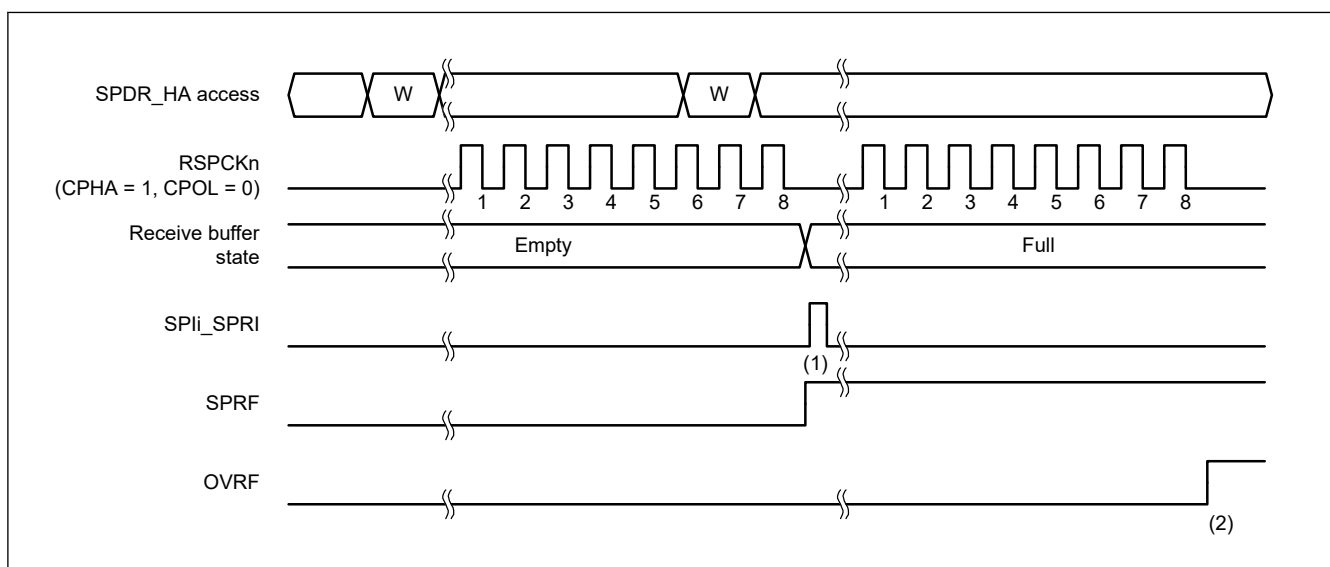


Figure 35.24 Operation example when SPCR.TXMD = 0

The operation of the flags at timings (1) and (2) in [Figure 35.24](#) is as follows:

1. When a serial transfer ends with the receive buffer of SPDR_HA empty, the SPI generates a receive buffer full interrupt request (SPIi_SPRI), the SPI sets the SPSR.SPRF flag to 1, and copies the received data in the shift register to the receive buffer.
2. When a serial transfer ends with the receive buffer of SPDR_HA holding data that was received in the previous serial transfer, the SPI sets the SPSR.OVRF flag to 1, and discards the received data in the shift register. For details about the operation of the SPSR.OVRF flag, see [section 35.3.8.1. Overrun errors](#).

35.3.6.2 Transmit-Only Serial Communications (SPCR.TXMD = 1)

[Figure 35.25](#) shows an example of operation when the Communications Operating Mode Select bit (SPCR.TXMD) is set to 1. In this example, the SPI performs an 8-bit serial transfer when SPDCR.SPFC[1:0] (only SPI0) bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

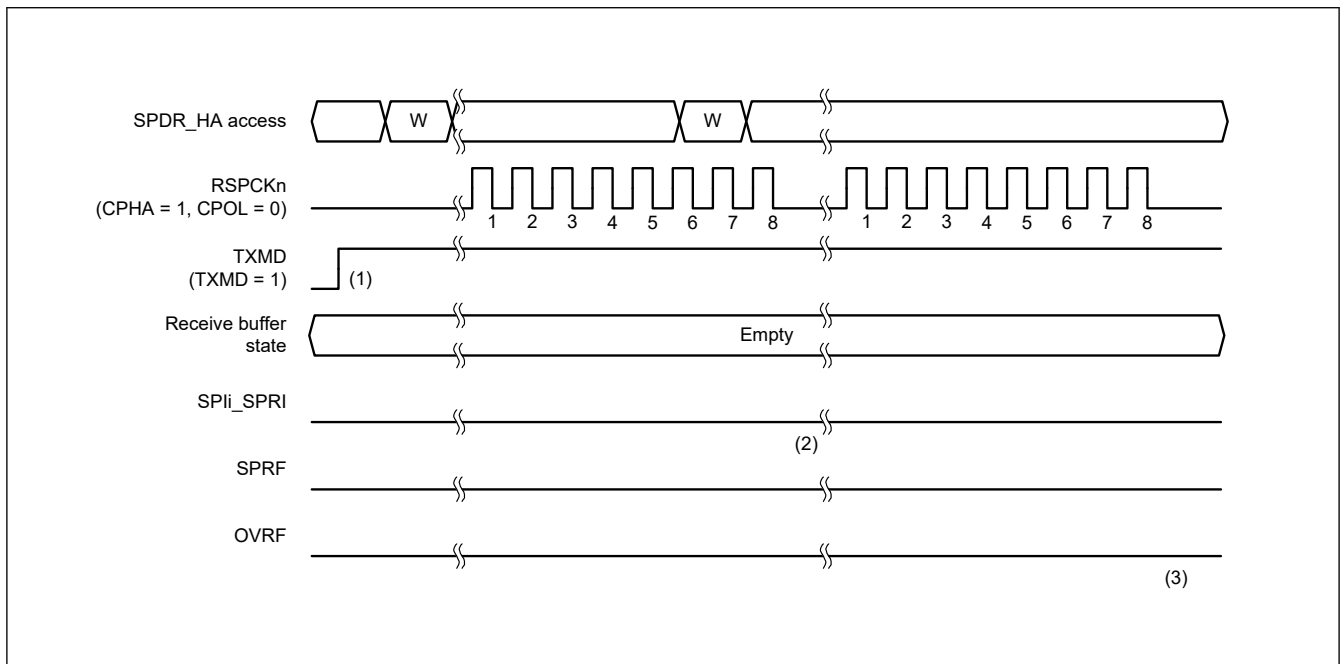


Figure 35.25 Operation example when SPCR.TXMD = 1

The operation of the flags at timings (1) to (3) in [Figure 35.25](#) is as follows:

1. Make sure there is no data left in the receive buffer (the SPSR.SPRF flag is 0) and the SPSR.OVRF flag is 0 before entering transmit-only mode (SPCR.TXMD = 1).
2. When a serial transfer ends with the receive buffer of SPDR_HA empty, if the transmit-only mode is selected (SPCR.TXMD = 1), the SPSR.SPRF flag retains the value of 0, and the SPI does not copy the data in the shift register to the receive buffer.
3. Because the receive buffer of SPDR_HA does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

In transmit-only mode (SPCR.TXMD = 1), the SPI transmits data but does not receive data. Therefore, the SPSR.SPRF and SPSR.OVRF flags remain 0 at timings (1) to (3).

35.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts

[Figure 35.26](#) and [Figure 35.27](#) show examples of operation of the transmit buffer empty interrupt (SPIi_SPTI) and the receive buffer full interrupt (SPIi_SPRI). The SPDR_HA register accesses shown in these figures indicate the conditions of access to the register, where W denotes a write cycle and R a read cycle. In [Figure 35.26](#), the SPI performs an 8-bit serial transfer when SPCR.TXMD bit is 0, the SPDCR.SPFC[1:0] bits (only SPI0) are 00b, the SPCMDm.CPHA bit is 0, and the SPCMDm.CPOL bit is 0. In [Figure 35.27](#), the SPI performs an 8-bit serial transfer when SPCR.TXMD bit is 0, the

SPDCR.SPFC[1:0] bits (only SPI0) are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

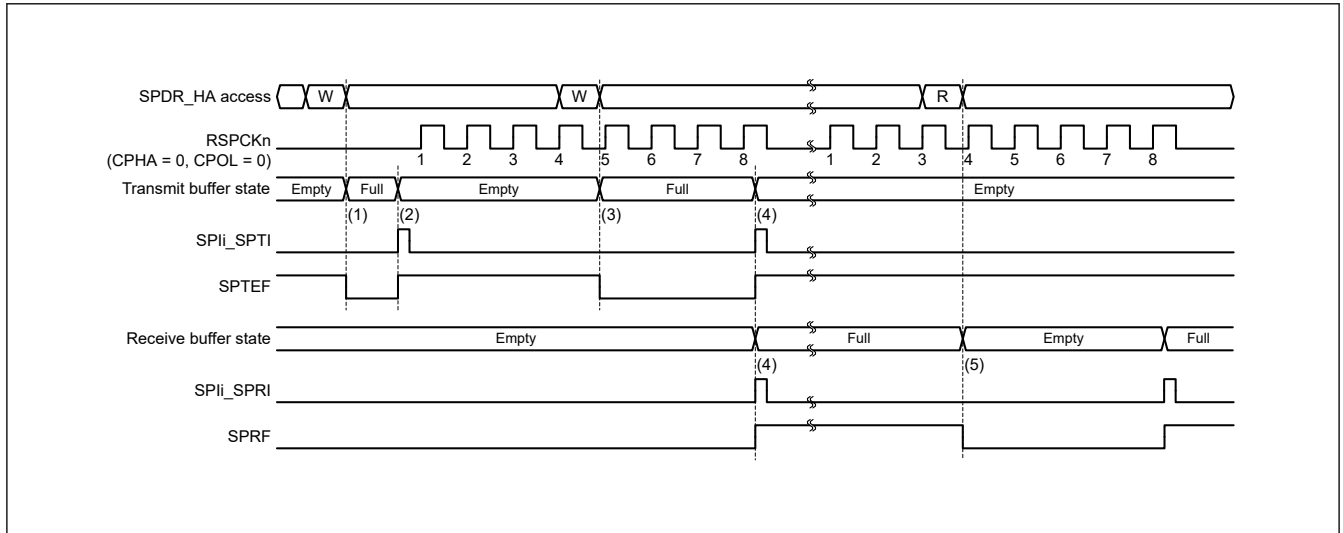


Figure 35.26 Operation example of the SPli_SPTI and SPli_SPRI interrupts when CPHA = 0 and CPOL = 0 in master mode

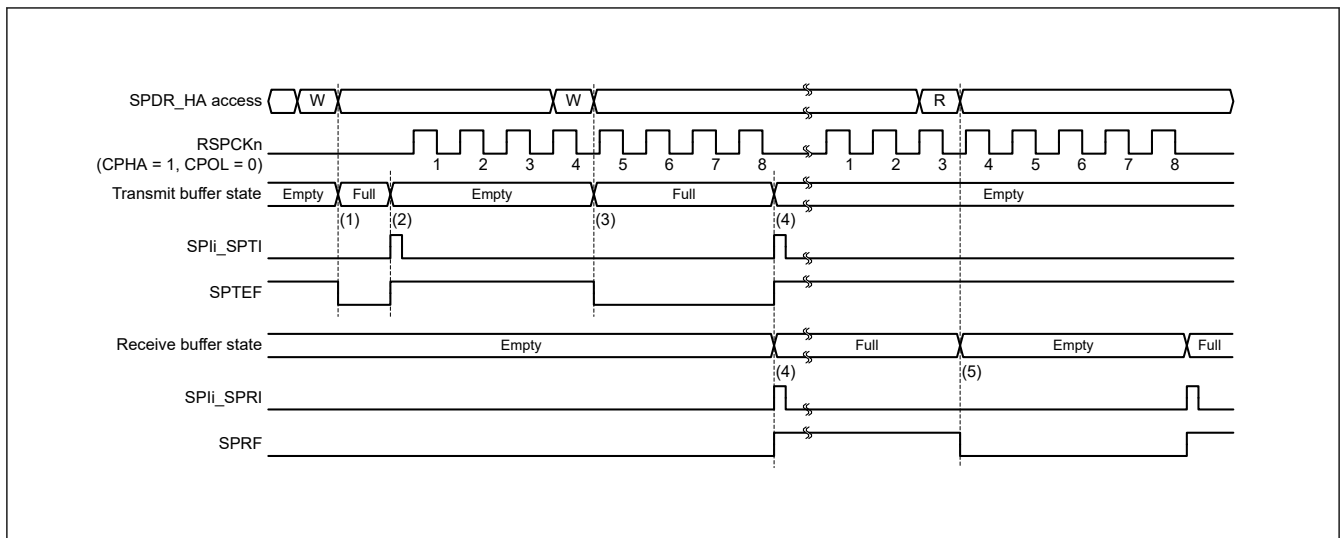


Figure 35.27 Operation example of the SPli_SPTI and SPli_SPRI interrupts when CPHA = 1 and CPOL = 0 in master mode

The operation of the SPI at timings (1) to (5) in [Figure 35.26](#) and [Figure 35.27](#) is as follows:

1. When transmit data is written to SPDR_HA with the transmit buffer of SPDR_HA is empty and data for the next transfer is not set, the SPI writes data to the transmit buffer and clears the SPSR.SPTEF flag to 0.
2. If the shift register is empty, the SPI copies the data in the transmit buffer to the shift register, generates a transmit buffer empty interrupt request (SPli_SPTI), and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the SPI mode. For details, see [section 35.3. Operation](#), and [section 35.3.11. Clock Synchronous Operation](#).
3. When transmit data is written to SPDR_HA either by the transmit buffer empty interrupt routine, or by the processing of the transmit buffer empty using the SPTEF flag, the SPI writes data to the transmit buffer and clears the SPTEF flag to 0. Because the data being transferred serially is stored in the shift register, the SPI does not copy the data in the transmit buffer to the shift register.
4. When the serial transfer ends with the receive buffer of SPDR_HA empty, the SPI copies the receive data in the shift register to the receive buffer, generates a receive buffer full interrupt request (SPli_SPRI), and sets the SPRF flag to 1. Because the shift register becomes empty on completion of the serial transfer, if the transmit buffer is full before the serial transfer ended, the SPI sets the SPTEF flag to 1 and copies data in the transmit buffer to the shift register. Even

when received data is not copied from the shift register to the receive buffer in an overrun error status, on completion of the serial transfer, the SPI determines that the shift register is empty, so data transfer from the transmit buffer to the shift register is enabled.

5. When SPDR_HA is read either by the receive buffer full interrupt routine or processing of the receive buffer full interrupt using the SPRF flag, the receive data can be read.

If SPDR_HA is written to when the transmit buffer holds data that is not yet transmitted (the SPTEF flag is 0), the SPI does not update data in the transmit buffer. When writing to SPDR_HA, always use either a transmit buffer empty interrupt request or processing of the transmit buffer empty interrupt using the SPTEF flag. To use a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1. If the SPI function is disabled (the SPCR.SPE bit is 0), set the SPTIE bit to 0.

When serial transfer ends and the receive buffer is full (the SPRF flag is 1), the SPI does not copy data from the shift register to the receive buffer, and it detects an overrun error (see [section 35.3.8. Error Detection](#)). To prevent a receive data overrun error, read the received data using a receive buffer full interrupt request before the next serial transfer ends. To use an SPI receive buffer full interrupt, set the SPCR.SPRIE bit to 1.

Transmission and reception interrupts or the associated IELSRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers.

Similarly, the SPTEF and SPRF flags can be used to confirm the states of the transmit and receive buffers. See [section 16, Interrupt Controller Unit \(ICU\)](#) for the interrupt vector numbers.

35.3.8 Error Detection

In normal SPI serial transfers, data written to the transmit buffer of SPDR/SPDR_HA is transmitted, and received data can be read from the receive buffer of SPDR/SPDR_HA. If access is made to SPDR/SPDR_HA, an abnormal transfer might occur, depending on the status of the transmit or receive buffer or the status of the SPI at the beginning or end of serial transfer.

If an abnormal transfer occurs, the SPI detects the event as an underrun error, overrun error, parity error, or mode fault error. [Table 35.8](#) lists the relationship between non-normal transfer operations and the SPI error detection function.

Table 35.8 Relationship between non-normal transfer operations and SPI error detection (1 of 2)

Operation	Occurrence condition	SPI operation	Error detection
1	SPDR/SPDR_HA is written when the transmit buffer is full.	- The contents of the transmit buffer are kept - Write data is missing	None
2	SPDR/SPDR_HA is read when the receive buffer is empty.	The contents of the receive buffer and previously received data are output.	None
3	Serial transfer is started in slave mode when the SPI is not able to transmit data.	- Serial transfer is suspended - Transmit or receive data is missing - Driving of the MISO_n output signal is stopped - SPI function is disabled	Underrun error
4	Serial transfer terminates when the receive buffer is full.	- Keeps the contents of the receive buffer - Missing receive data	Overrun error
5	An incorrect parity bit is received during full-duplex synchronous serial communication with the parity function enabled in following mode: - Transmit-receive master mode - Transmit-receive slave mode	The parity error flag is asserted	Parity error
6	The SSL _n 0 input signal is asserted when the serial transfer is idle in multi-master mode.	- Driving of the RSPCK_n, MOSI_n, SSL_n1 to SSL_n3 output signals is stopped - SPI function is disabled	Mode fault error
7	The SSL _n 0 input signal is asserted during serial transfer in multi-master mode.	- Serial transfer is suspended - Transmit or receive data is missing - Driving of the RSPCK_n, MOSI_n, SSL_n1 to SSL_n3 output signals is stopped - SPI function is disabled	Mode fault error

Table 35.8 Relationship between non-normal transfer operations and SPI error detection (2 of 2)

Operation	Occurrence condition	SPI operation	Error detection
8	The SSLn0 input signal is negated during serial transfer in slave mode.	- Serial transfer is suspended - Transmit or receive data is missing - Driving of the MISO_n output signal is stopped - SPI function is disabled	Mode fault error

In operation 1 described in [Table 35.8](#), the SPI does not detect an error. To prevent data omission during writes to SPDR/SPDR_HA, the writes to SPDR/SPDR_HA must be executed using a transmit buffer empty interrupt request (when the SPSR.SPTEF flag is 1).

Similarly, the SPI does not detect an error in operation 2. To prevent extraneous data from being read, SPDR/SPDR_HA read must be executed with an SPI receive buffer full interrupt request (when the SPSR.SPRF flag is 1).

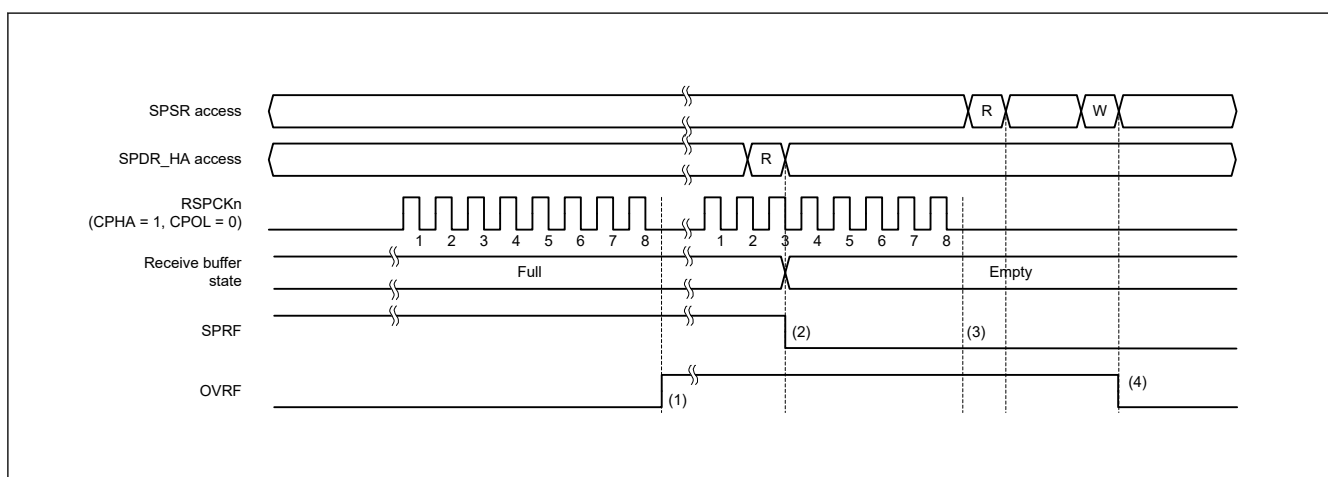
For information on the other errors, see the following sections:

- Underrun error, indicated in operation 3, see [section 35.3.8.4. Underrun errors](#)
- Overrun error, indicated in operation 4, see [section 35.3.8.1. Overrun errors](#)
- Parity error, indicated in operation 5, see [section 35.3.8.2. Parity errors](#)
- Mode fault error, indicated in operations 6 to 8, see [section 35.3.8.3. Mode fault errors](#)
- For the transmit and receive interrupts, see [section 35.3.7. Transmit Buffer Empty and Receive Buffer Full Interrupts](#).

35.3.8.1 Overrun errors

If a serial transfer ends when the receive buffer of SPDR/SPDR_HA is full, the SPI detects an overrun error and sets the SPSR.OVRF flag to 1. When the OVRF flag is 1, the SPI does not copy data from the shift register to the receive buffer, so the data prior to the error occurrence is retained in the receive buffer. To set the OVRF flag to 0, write 0 to the OVRF flag after the CPU reads SPSR with the OVRF flag set to 1.

[Figure 35.28](#) shows an example of operation of the OVRF and SPRF flags. The SPSR and SPDR_HA accesses shown in [Figure 35.28](#) indicate the condition of accesses to the SPSR and SPDR_HA register, where W denotes a write cycle, and R a read cycle. In this example, the SPI performs an 8-bit serial transfer when SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given for RSPCK_n in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

**Figure 35.28 Operation example of the OVRF and SPRF flags**

The operation of the flags at timings (1) to (4) in [Figure 35.28](#) 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 the data in the shift register to the receive buffer. Even when the SPPE bit is 1, parity errors are not detected.
2. When SPDR/SPDR_HA is read, the SPI outputs the data in the receive buffer. The SPRF flag is then set to 0. The receive buffer becoming empty does not set the OVRF flag to 0.

3. If the serial transfer ends with the OVRF flag set to 1 (overrun error occurred), the SPI does not copy data in the shift register to the receive buffer (the SPRF flag does not set to 1). A receive buffer full interrupt is not generated. Even when the SPPE bit is 1, parity errors are not detected. In an overrun error state when the SPI does not copy the received data from the shift register to the receive buffer, on termination of the serial transfer, the SPI determines that the shift register is empty. This enables data transfer from the transmit buffer to the shift register.
4. If 0 is written to the OVRF flag after SPSR is read when the OVRF flag is 1, the OVRF flag clears is set to 0.

The occurrence of an overrun can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. When executing a serial transfer, you must ensure that overrun errors are detected early, for example by reading SPSR immediately after SPDR/SPDR_HA/SPDR_BY is read.

If an overrun error occurs and the OVRF flag sets to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

When the RSPCK auto-stop function is enabled (SPCR2.SCKASE = 1) in master mode, an overrun error does not occur. [Figure 35.29](#) and [Figure 35.30](#) show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

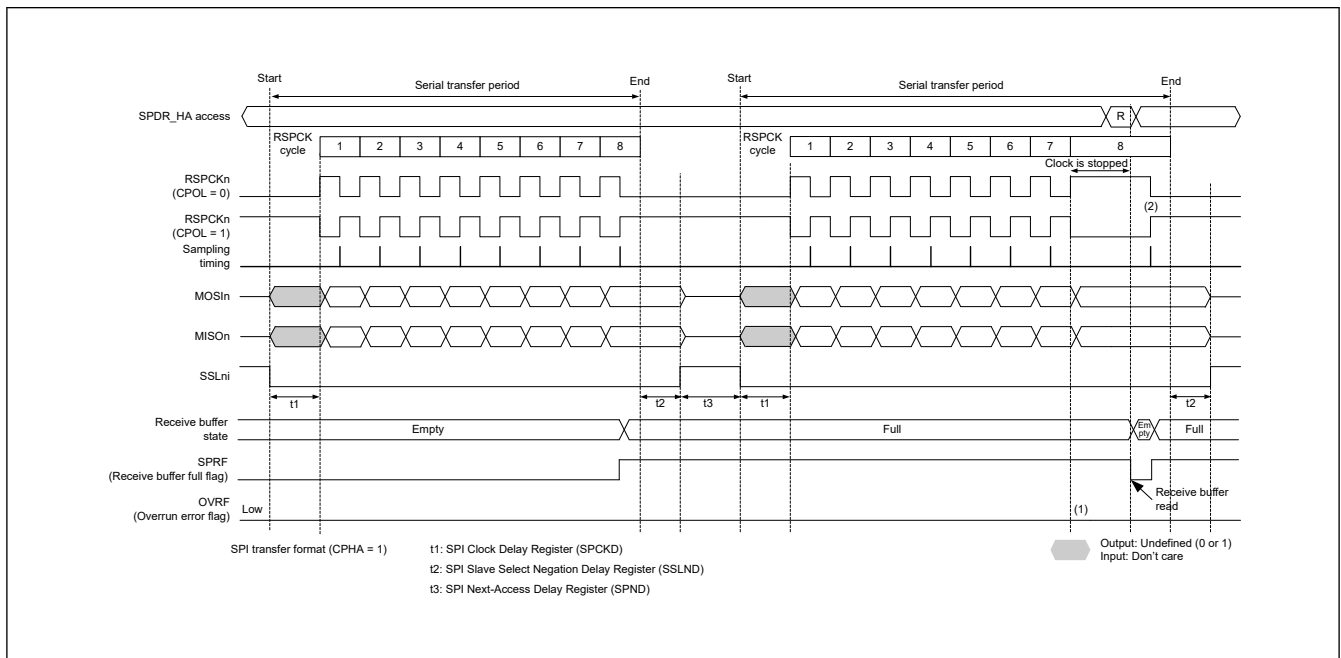


Figure 35.29 Clock stop waveform when serial transfer continues while receive buffer is full in master mode (CPHA = 1)

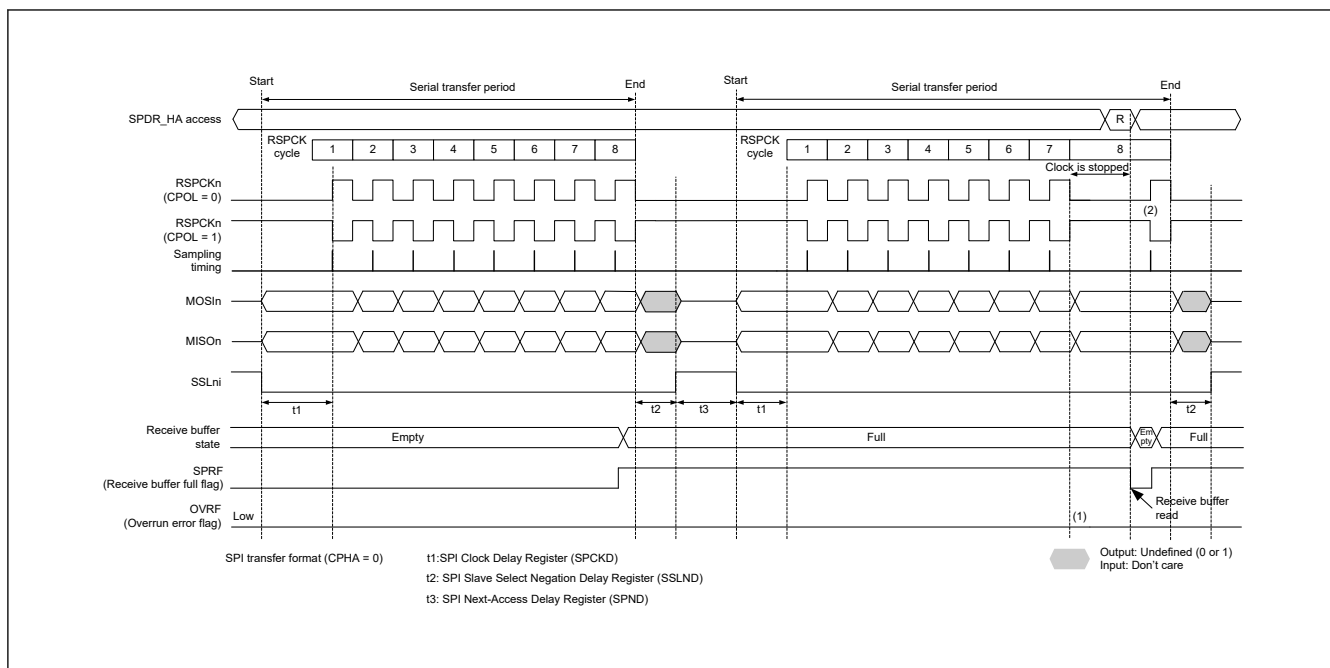


Figure 35.30 Clock stop waveform when serial transfer continues while receive buffer is full in master mode (CPHA = 0)

The operation of the flags at timings (1) and (2) in [Figure 35.29](#) and [Figure 35.30](#) 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 the SPSR.SPRF flag is set to 0).

35.3.8.2 Parity errors

When full-duplex synchronous serial communications is performed with the SPCR.TXMD bit set to 0 and the SPCR2.SPPE bit set to 1, when serial transfer ends, the SPI checks whether there are parity errors. On detecting a parity error in the received data, the SPI sets the SPSR.PERF flag to 1. Because the SPI does not copy data in the shift register to the receive buffer when the SPSR.OVRF flag is set to 1, parity error detection is not performed for the received data. To set the PERF flag to 0, write 0 to the PERF flag after the SPSR register is read with the PERF flag set to 1.

[Figure 35.31](#) shows an example of operation of the OVRF and PERF flags. The SPSR access shown in [Figure 35.31](#) indicates the condition of access to the register, where W denotes a write cycle, and R a read cycle. In this example, full-duplex serial communication is performed while the SPCR2.SPPE bit is 1. The SPI performs an 8-bit serial transfer when SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

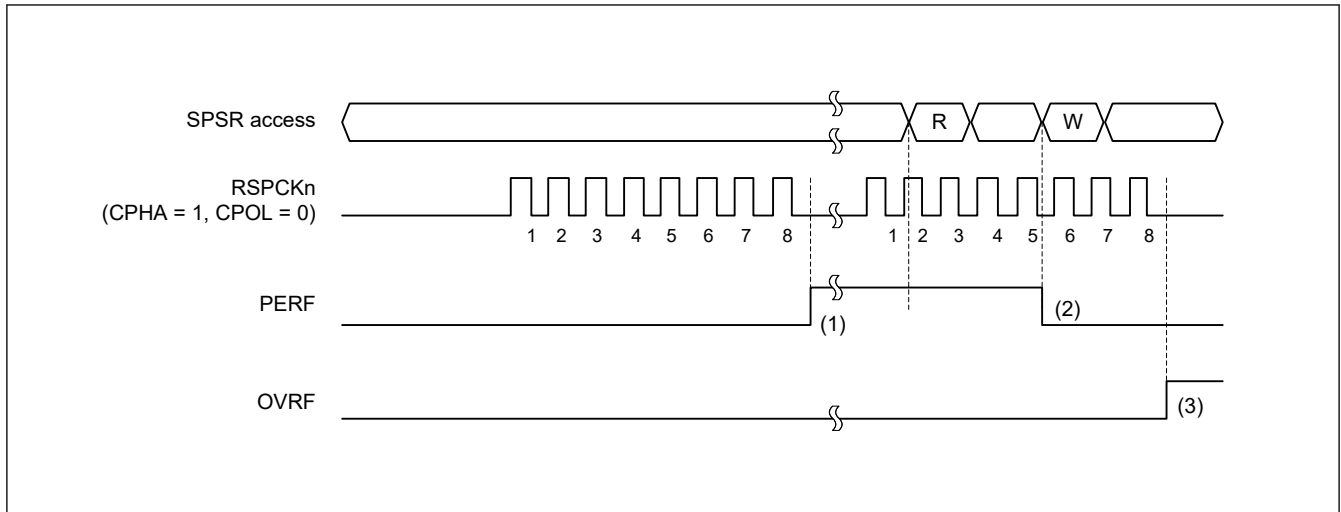


Figure 35.31 Operation example of the PERF flag

The operation of the flags at timings (1) to (3) in [Figure 35.31](#) is as follows:

1. If a serial transfer terminates with the SPI not detecting an overrun error, the SPI copies the data in the shift register to the receive buffer. The SPI checks the received data at this time and sets the PERF flag to 1 if a parity error is detected.
2. If 0 is written to the PERF flag after the SPSR register is read when the PERF flag is 1, the PERF flag is set to 0.
3. When the SPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The SPI does not perform parity error detection at this time.

Parity errors can be checked for by either reading the SPSR register or using an SPI error interrupt and reading the SPSR register. When executing a serial transfer, such checks are required to ensure early detection of parity errors. When the SPI is used in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits (Only SPI0).

35.3.8.3 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. On detecting the mode fault error, the SPI copies the value of the pointer to SPCMDm to the SPDCR.SPFC[1:0] bits (only SPI0). The active level of the SSLn0 signal is determined by the SSLP.SSL0P bit.

When the MSTR bit is 0, the SPI operates in slave mode. The SPI detects a mode fault error if the MODFEN bit of the SPI in slave mode is 1, and the SPMS bit is 0, and if the SSLn0 input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

On detecting a mode fault error, the SPI stops the driving of the output signals and clears the SPCR.SPE bit to 0 (see [section 35.3.9. Initializing the SPI](#)). For multi-master configuration, detection of a mode fault error is used to stop the driving of output signals and the SPI function, which allows the master to be released.

The occurrence of a mode fault error can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. Detecting mode-fault errors without using the SPI error interrupt requires polling of SPSR. When using the SPI in master mode, the value of the pointer to the SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits (Only SPI0).

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.

35.3.8.4 Underrun errors

While the SPI is operating in slave mode (SPCR.MSTR bit = 0), if serial transfer is started before transmit data output is ready with the SPCR.SPE bit set to 1 (SPI function enabled), the SPI detects an underrun error and sets the SPSR.MODF and SPSR.UDRF flags to 1.

On detecting an underrun error, the SPI stops the driving of output signals and clears the SPCR.SPE bit to 0 (see [section 35.3.9. Initializing the SPI](#)).

The occurrence of underrun errors can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. Detecting underrun errors without using the SPI error interrupt requires polling of SPSR.

When the MODF flag is 1, writing 1 to the SPE bit is ignored by the SPI. To enable the SPI function after the detection of an underrun error, the MODF flag must be set to 0.

35.3.9 Initializing the SPI

If 0 is written to the SPCR.SPE bit or if the SPI sets the SPE bit to 0 because it detected a mode fault error or an underrun error, the SPI disables the SPI function and initializes some of the module functions. When a system reset is generated, the SPI initializes all of the module functions. This section describes initialization by clearing of the SPCR.SPE bit and by a system reset.

35.3.9.1 Initialization by clearing of the SPCR.SPE bit

When the SPCR.SPE bit is set to 0, the SPI initializes by:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the SPI
- Initializing the transmit buffer of the SPI (the SPSR.STEF flag sets to 1)

Initialization by clearing of the SPE bit does not initialize the control bits of the SPI. For this reason, the SPI can be started in the same transfer mode in use prior to initialization when the SPE bit is set to 1 again.

The SPSR.SPRF, SPSR.OVRF, SPSR.MODE, SPSR.PERF, and SPSR.UDRF flags are not initialized, and the value of the SPI Sequence Status Register (SPSSR) is not initialized (only SPI0). Therefore, even after the SPI is initialized, data from the receive buffer can be read to check the error status during an SPI transfer.

The transmit buffer is initialized to an empty state (the SPSR.SPTEF flag sets to 1). Therefore, if the SPCR.SPTIE bit is set to 1 after SPI initialization, a transmit buffer empty interrupt is generated. To disable any transmit buffer empty interrupts when the SPI is initialized, write 0 to the SPTIE bit simultaneously while writing 0 to the SPE bit.

35.3.9.2 Initialization by system reset

A system reset completely initializes the SPI by initializing all SPI control bits, status bits, and data registers, in addition to meeting the requirements described in [section 35.3.9.1. Initialization by clearing of the SPCR.SPE bit](#).

35.3.10 SPI Operation

35.3.10.1 Master mode operation

The only difference between single- and multi-master mode operation is the use of mode fault error detection (see [section 35.3.8. Error Detection](#)). In single-master mode, the SPI does not detect mode fault errors whereas in multi-master mode, it does. This section explains operations that are common to both modes.

(1) Starting a serial transfer

The SPI updates the data in the transmit buffer (SPTX) when data is written to the SPI Data Register (SPDR/SPDR_HA) with the SPI transmit buffer empty, data for the next transfer is not set, and the SPSR.STEF flag is 0. When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits (only SPI0) are written to the SPDR/SPDR_HA/SPDR_BY, the SPI copies data from the transmit buffer to the shift register and starts serial transfer. On copying transmit data to the shift register, the SPI changes the status of the shift register to full. On termination of the serial transfer, it changes the status of the shift register to empty. The status of the shift register cannot be referenced.

The polarity of the SSLn output pins depends on the SSLP register settings. For details on the SPI transfer format, see [section 35.3.5. Transfer Formats](#).

(2) Terminating a serial transfer

Regardless of the SPCMDm.CPHA bit setting, the SPI terminates the serial transfer after transmitting an RSPCKn edge associated with the final sampling timing. If free space is available in the receive buffer (SPRX) (the SPSR.SPRF flag is 0), on termination of the serial transfer, the SPI copies data from the shift register to the receive buffer of the SPDR/SPDR_HA register.

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMDm.SPB[3:0] bit settings. The polarity of the SSLn output pin depends on the SSLP register settings. For details on the SPI transfer format, see [section 35.3.5. Transfer Formats](#).

(3) Sequence control (only SPI0)

The transfer format used in master mode is determined by the SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers.

The SPSCR register determines the sequence configuration for serial transfers that are executed by the SPI in master mode. The following items are set in the SPCMDm register:

- SSLn pin output signal value
- MSB- or LSB-first
- Data length
- Some of the bit rate settings
- RSPCK polarity and phase
- Whether SPCKD is to be referenced
- Whether SSLND is to be referenced
- Whether SPND is to be referenced

SPBR holds some of the bit rate settings, including SPCKD (SPI clock delay), SSLND (SSL negation delay), and SPND (next-access delay).

Based on the sequence length assigned in SPSCR, the SPI makes up a sequence comprised of a part or all of the SPCMDm register. The SPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the SPI function is enabled, the SPI loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The SPI increments the pointer each time the next-access delay period for a data transfer ends. On completion of the serial transfer that corresponds to the final command in the sequence, the SPI sets the pointer to SPCMD0, and in this way the sequence is executed repeatedly.

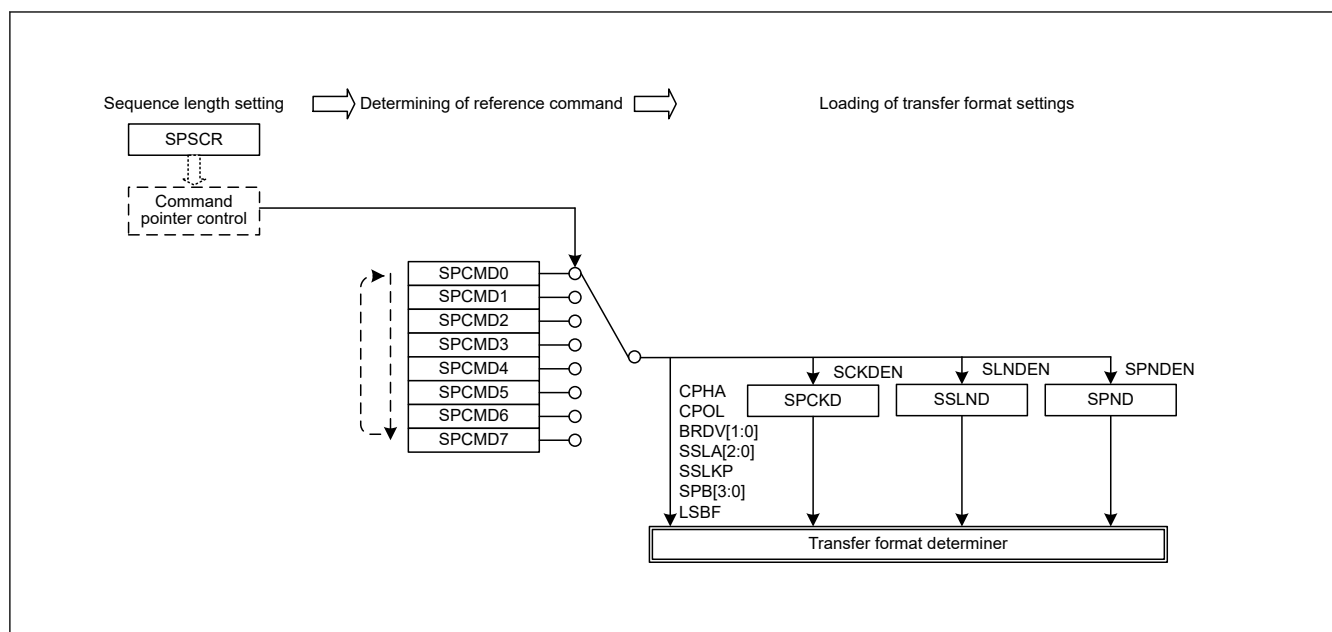


Figure 35.32 Procedure for determining the form of a serial transfer in master mode

In this section, a frame is the combination of the data in SPDR/SPDR_HA and the settings in SPCMDm.

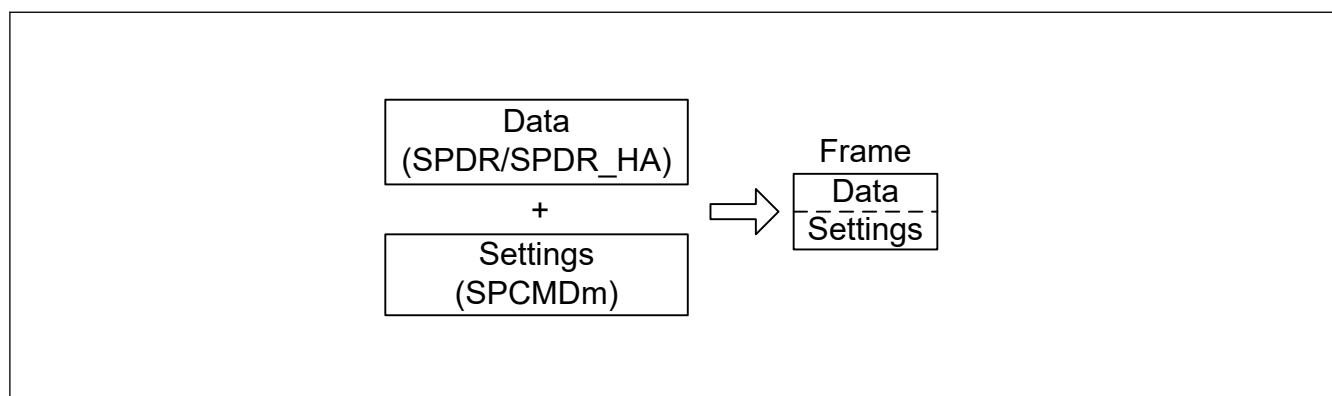
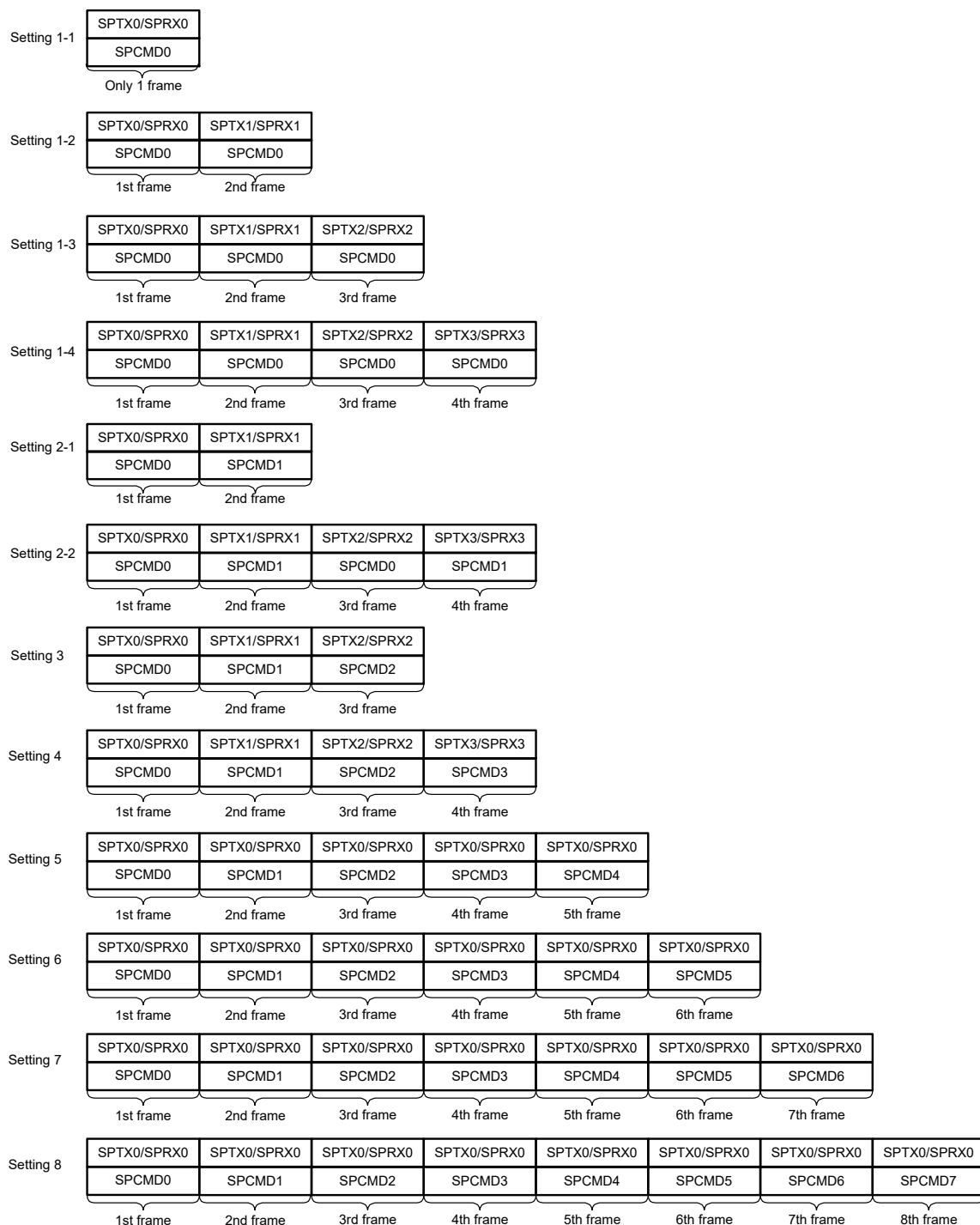


Figure 35.33 Conceptual diagram of frames

Figure 35.34 shows the correspondence between the commands and the transmit and receive buffers in the sequence of operations specified by the settings in Table 35.4.



Note: For SPI1, only 1 frame

Figure 35.34 Correspondence between SPI Command Register and transmit and receive buffers in sequence operations

(4) Burst transfers (only SPI0)

If the SPCMDm.SSLKP bit that the SPI references during the current serial transfer is 1, the SPI maintains the SSLni signal level during the serial transfer until the beginning of the SSLni signal assertion for the next serial transfer. If the SSLni

signal level for the next serial transfer is the same as the SSLni signal level for the current serial transfer, the SPI can execute continuous serial transfers while keeping the SSLni signal assertion status (burst transfer).

Figure 35.35 shows an example of an SSLni signal operation for a burst transfer that is implemented using the SPCMD0 and SPCMD1 register settings. This section describes SPI operations (1) to (7) shown in Figure 35.35.

Note: The polarity of the SSLni output signal depends on the SSLP register settings.

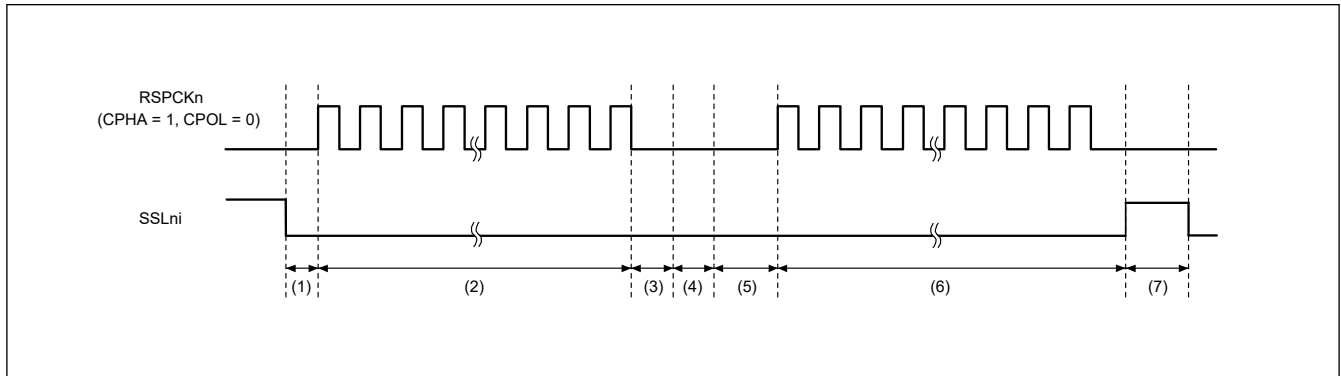


Figure 35.35 Example of burst transfer operation using the SSLKP bit

The SPI operation at times (1) to (7) in the figure is as follows:

1. Based on the SPCMD0 settings, the SPI asserts the SSLni signal and inserts RSPCK delays.
2. The SPI executes serial transfers in accordance with the SPCMD0 settings.
3. The SPI inserts an SSL negation delay.
4. Because the SPCMD0.SSLKP bit is 1, the SPI keeps the SSLni signal value specified in SPCMD0. This period is sustained at a minimum for a period equal to the next-access delay in SPCMD0. If the shift register is empty after the passage of the minimum period, this period is sustained until the transmit data is stored in the shift register for the next transfer.
5. Based on the SPCMD1 settings, the SPI asserts the SSLni signal and inserts RSPCK delays.
6. The SPI executes serial transfers in accordance with the SPCMD1 settings.
7. Because the SPCMD1.SSLKP bit is 0, the SPI negates the SSLni signal. In addition, a next-access delay is inserted in accordance with SPCMD1.

If the SSLni signal output settings in the SPCMDm register where 1 is assigned to the SSLKP bit are different from the SSLni signal output settings in the SPCMDm register to be used in the next transfer, the SPI switches the SSLni signal status to SSLni signal assertion as shown in (5) in Figure 35.35. This corresponds to the command for the next transfer.

Note: If such an SSLni signal switching occurs, the slaves that drive the MISO_n signal compete, and collision of signal levels might occur.

The SPI in master mode references the SSLni signal operation within the module when the SSLKP bit is not used. When the SPCMDm.CPHA bit is 0, the SPI can accurately start serial transfers by using the SSLni signal assertion for the next transfer that is detected internally.

(5) RSPCK delay (t₁)

The RSPCK delay value of the SPI in master mode depends on the SPCMDm.SCKDEN bit setting and the SPCKD.SCKDL[2:0] bits setting. The SPI determines the SPCMDm register to be referenced during a serial transfer by pointer control (only SPI0), and determines an RSPCK delay using the SPCMDm.SCKDEN bit and SPCKD.SCKDL[2:0] bits, as listed in Table 35.9. For a definition of RSPCK delay, see section 35.3.5. Transfer Formats.

Table 35.9 Relationship between the SPCMDm.SCKDEN bit, SPCKD.SCKDL[2:0] bits, and RSPCK delay (1 of 2)

SPCMDm.SCKDEN bit	SPCKD.SCKDL[2:0] bits	RSPCK delay
0	000b to 111b	1 RSPCK

Table 35.9 Relationship between the SPCMDm.SCKDEN bit, SPCKD.SCKDL[2:0] bits, and RSPCK delay (2 of 2)

SPCMDm.SCKDEN bit	SPCKD.SCKDL[2:0] bits	RSPCK delay
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(6) SSL negation delay (t2)

The SSL negation delay value of the SPI in master mode depends on the SPCMDm.SLNDEN bit setting and the SSLND.SLNDL[2:0] bits setting. The SPI determines the SPCMDm register to be referenced by pointer control during a serial transfer, and determines an SSL negation delay using the SPCMDm.SLNDEN bit and SSLND.SLNDL[2:0] bits, as listed in [Table 35.10](#). For a definition of SSL negation delay, see [section 35.3.5. Transfer Formats](#).

Table 35.10 Relationship between the SPCMDm.SLNDEN bit, SSLND.SLNDL[2:0] bits, and SSL negation delay

SPCMDm.SLNDEN bit	SSLND.SLNDL[2:0] bits	SSL negation delay
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(7) Next-access delay (t3)

The next-access delay value of the SPI in master mode depends on the SPCMDm.SPNDEN bit setting and the SPND.SPNDL[2:0] bits setting. The SPI determines the SPCMDm register to be referenced during serial transfer by pointer control (only SPI0), and then determines a next-access delay during serial transfer using the SPCMDm.SPNDEN bit and SPND.SPNDL[2:0] bits, as listed in [Table 35.11](#). For a definition of next-access delay, see [section 35.3.5. Transfer Formats](#).

Table 35.11 Relationship between the SPCMDm.SPNDEN bit, SPND.SPNDL[2:0] bits, and next-access delay (1 of 2)

SPCMDm.SPNDEN bit	SPND.SPNDL[2:0] bits	Next-access delay
0	000b to 111b	1 RSPCK + 2 PCLKA

Table 35.11 Relationship between the SPCMDm.SPNDEN bit, SPND.SPNDL[2:0] bits, and next-access delay (2 of 2)

SPCMDm.SPNDEN bit	SPND.SPNDL[2:0] bits	Next-access delay
1	000b	1 RSPCK + 2 PCLKA
	001b	2 RSPCK + 2 PCLKA
	010b	3 RSPCK + 2 PCLKA
	011b	4 RSPCK + 2 PCLKA
	100b	5 RSPCK + 2 PCLKA
	101b	6 RSPCK + 2 PCLKA
	110b	7 RSPCK + 2 PCLKA
	111b	8 RSPCK + 2 PCLKA

(8) Initialization flow

[Figure 35.36](#) shows an example of SPI initialization flow when the SPI is in master mode. For information on how to set up the Interrupt Controller Unit (ICU), DMAC and I/O ports, see the descriptions given in the individual blocks.

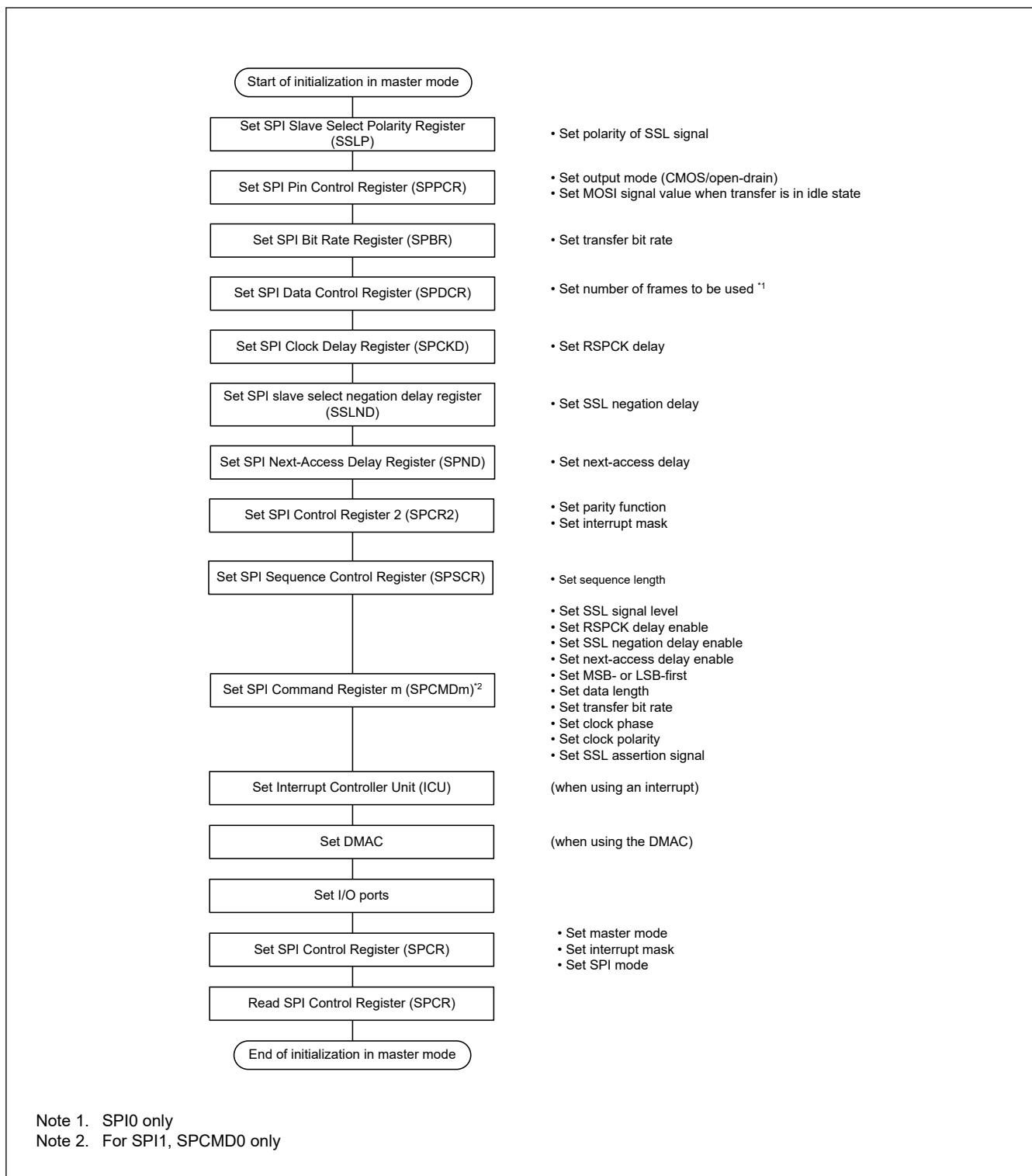


Figure 35.36 Example of initialization flow in master mode for SPI operation

(9) Software processing flow

Figure 35.37 to Figure 35.39 show examples of the software processing flow.

Transmit processing flow

When transmitting data, with the SPI_i_SPII interrupt enabled, the CPU is notified of the completion of data transmission after the last data write for transmission.

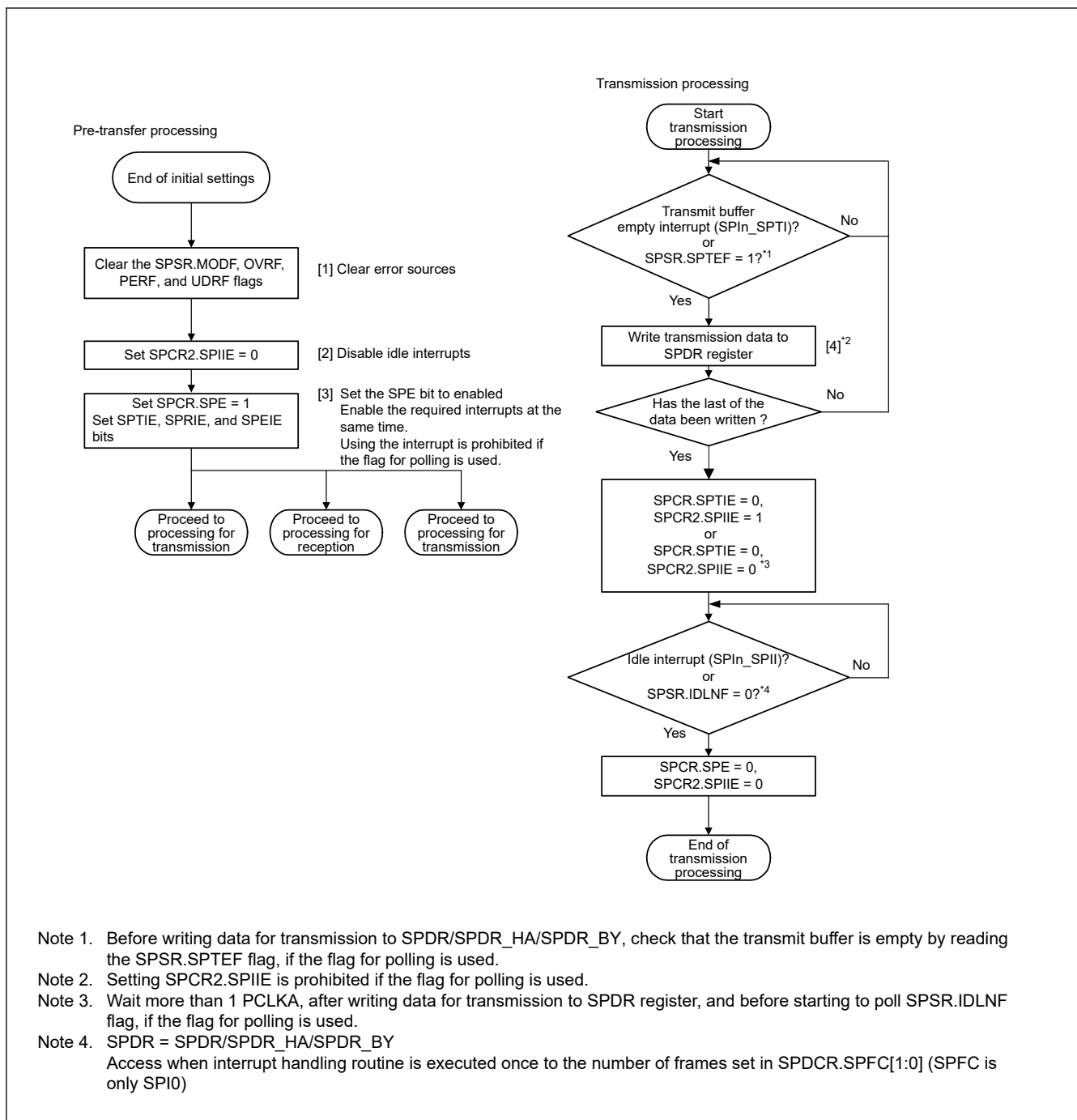


Figure 35.37 Transmission flow in master mode

Receive processing flow

The SPI cannot handle receive-only operation. Even when there is no data to transmit, it is necessary to transmit dummy data.

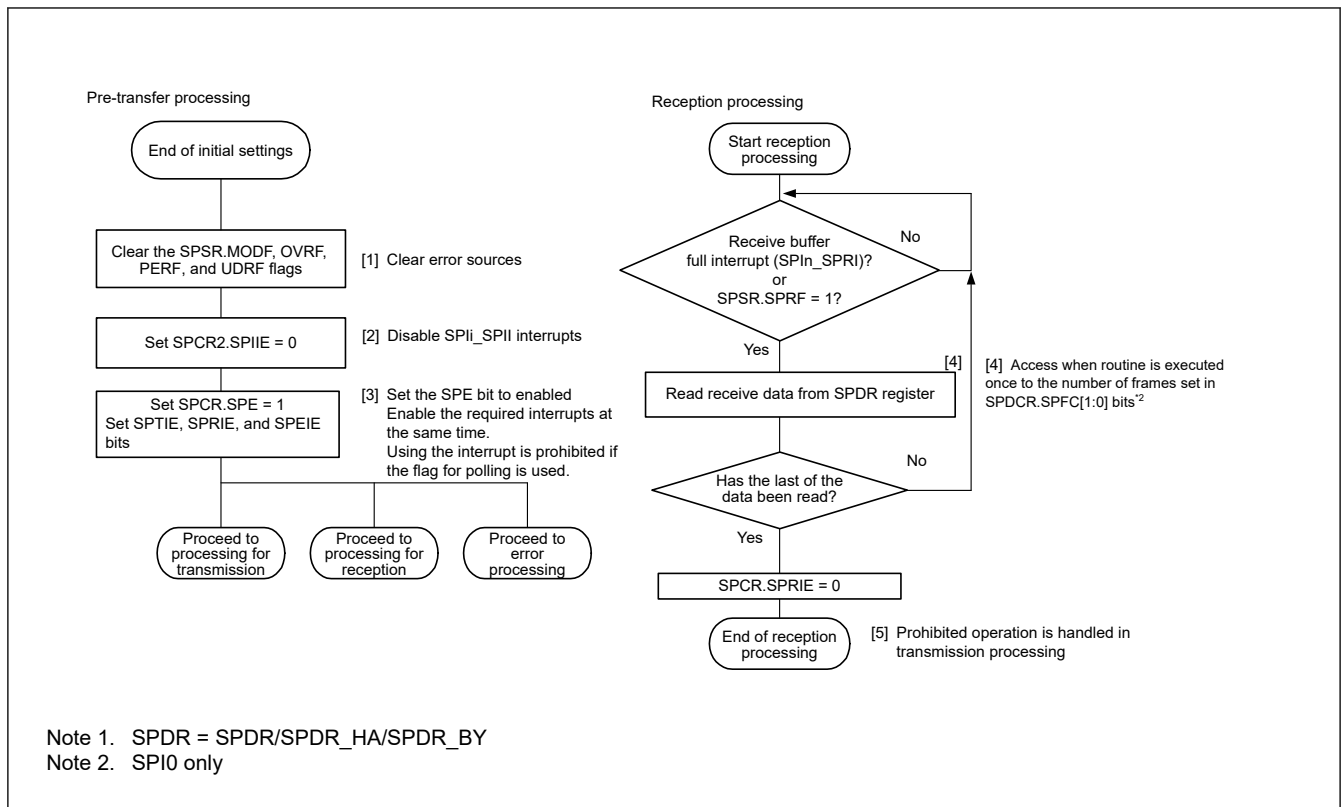


Figure 35.38 Reception flow in master mode

Error processing flow

The SPI detects the following errors:

- Mode fault error
- Underrun error
- Overrun error
- Parity error

When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, the SPCR.SPE bit is not cleared and operations for transmission and reception continue. Therefore, Renesas recommends clearing the SPCR.SPE bit to stop operations for errors other than mode fault errors. Not doing so leads to updating of the SPSR.SPECM[2:0] bits.

When an error is detected using an interrupt, clear the ICU.IELSRn.IR flag in the error processing routine. If this is not done, the ICU.IELSRn.IR flag might continue to indicate the SPIi_SPTI or SPIi_SPRI interrupt request. If the SPIi_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

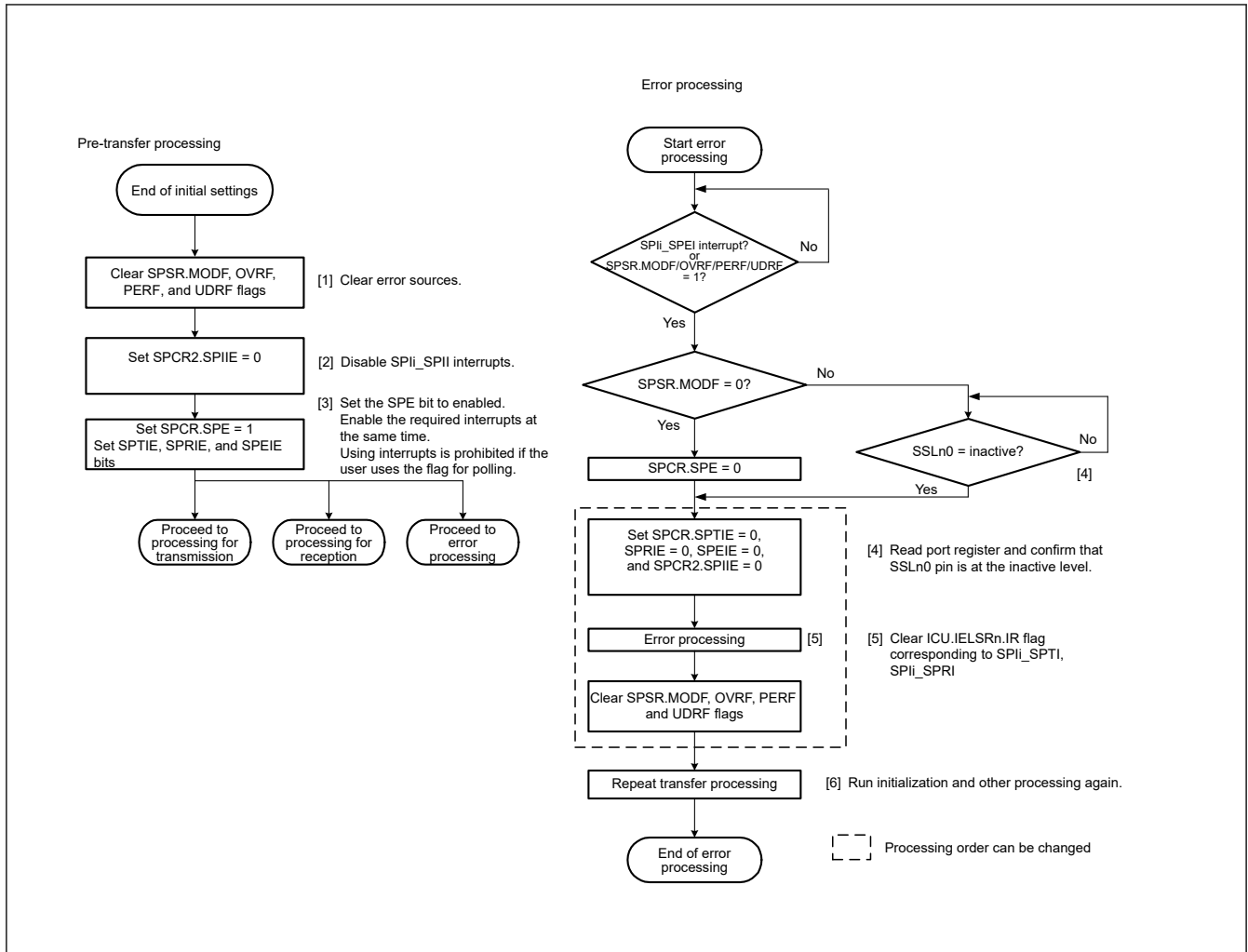


Figure 35.39 Error processing flow in master mode

35.3.10.2 Slave mode operation

(1) Starting a serial transfer

When the SPCMD0.CPHA bit is 0, if the SPI detects an SSLn0 input signal assertion, it must drive valid data to the MISON output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLn0 input signal triggers the start of a serial transfer.

When the CPHA bit is 1, if the SPI detects the first RSPCKn edge in an SSLn0 signal asserted condition, it must drive valid data to the MISON output signal. For this reason, when the CPHA bit is 1, the first RSPCKn edge in an SSLn0 signal asserted condition triggers the start of a serial transfer.

Regardless of the CPHA bit setting, the SPI drives the MISON output signal on SSLn0 signal assertion. The data that is output by the SPI is either valid or invalid, depending on the CPHA bit setting.

For details on the SPI transfer format, see [section 35.3.5. Transfer Formats](#). The polarity of the SSLn0 input signal depends on the SSLP.SSL0P setting.

(2) Terminating a serial transfer

Regardless of the SPCMD0.CPHA bit setting, the SPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPSR.SPRF flag is 0), on termination of serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR/SPDR_HA register. On termination of a serial transfer, the SPI changes the status of the shift register to empty, regardless of the receive buffer state. A mode fault error occurs if the SPI detects an SSLn0 input signal negation from the beginning of serial transfer to the end of serial transfer (see [section 35.3.8. Error Detection](#)).

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length is determined by the SPCMD0.SPB[3:0] bits setting. The polarity of the SSLn0 input signal is determined by the SSLP.SSL0P bit setting. For details on the SPI transfer format, see [section 35.3.5. Transfer Formats](#).

(3) Notes on single-slave operations

If the SPCMD0.CPHA bit is 0, the SPI starts serial transfers when it detects the assertion edge for an SSLn0 input signal. In the configuration shown in [Figure 35.7](#), if the SPI is used in single-slave mode, the SSLn0 signal is fixed at an active state. Therefore, when the CPHA bit is set to 0, the SPI cannot correctly start a serial transfer. For the SPI to correctly execute transmit and receive operations in slave mode when the SSLn0 input signal is fixed at an active state, the CPHA bit must be set to 1. Do not fix the SSLn0 input signal if there is a requirement for setting the CPHA bit to 0.

(4) Burst transfer (only SPI0)

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLn0 input signal. When the CPHA bit is 1, the serial transfer period is the period from the first RSPCKn edge to the sampling timing for the reception of the final bit in an SSLn0 signal active state. Even when the SSLn0 input signal remains at the active level, the SPI can accommodate burst transfers, because it can detect the start of an access.

When the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

(5) Initialization flow

[Figure 35.40](#) shows an example of initialization flow for SPI operation when the SPI is in slave mode. For a description of how to set up the ICU, DTC, and I/O ports, see the descriptions given in the individual blocks.

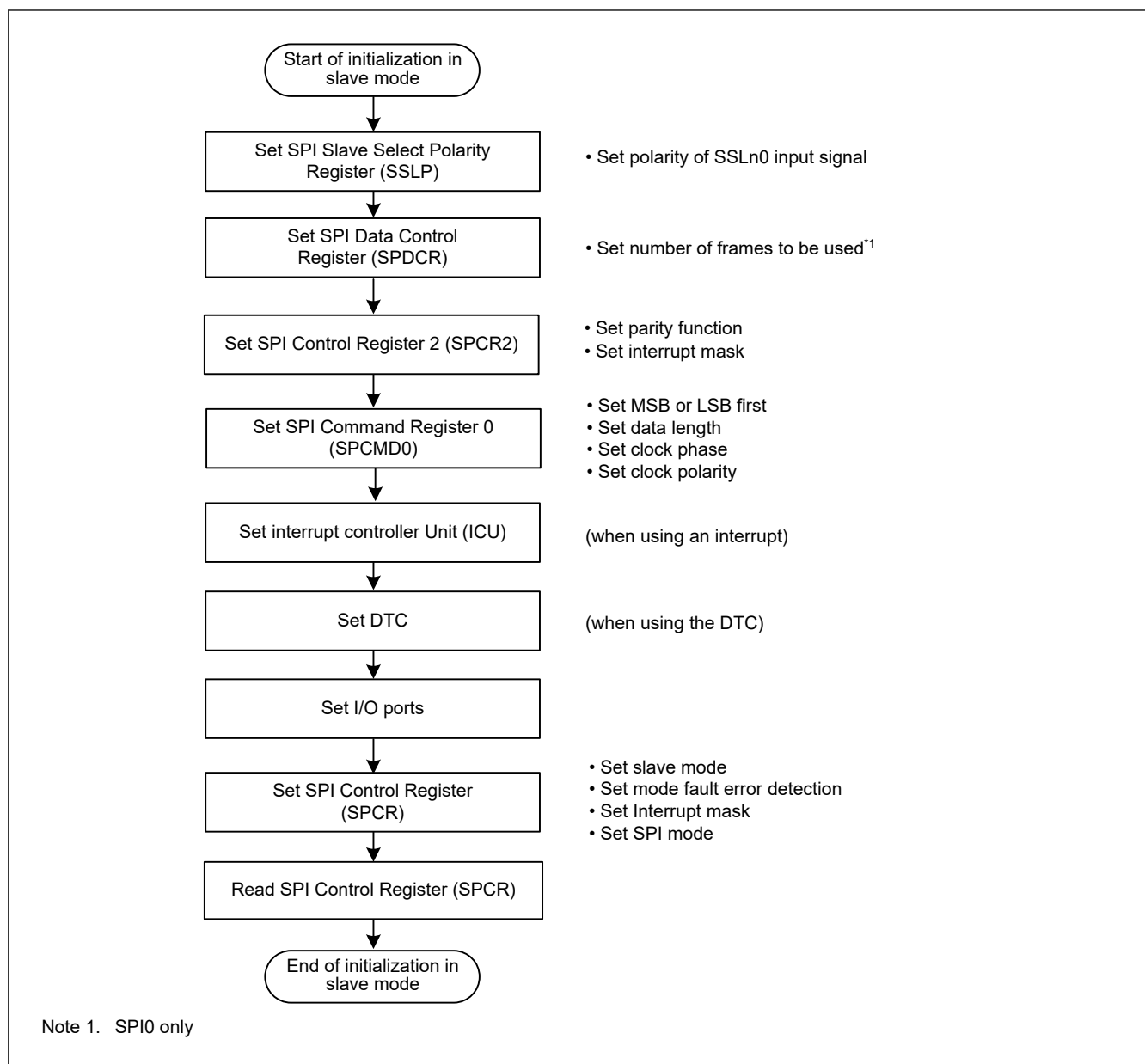
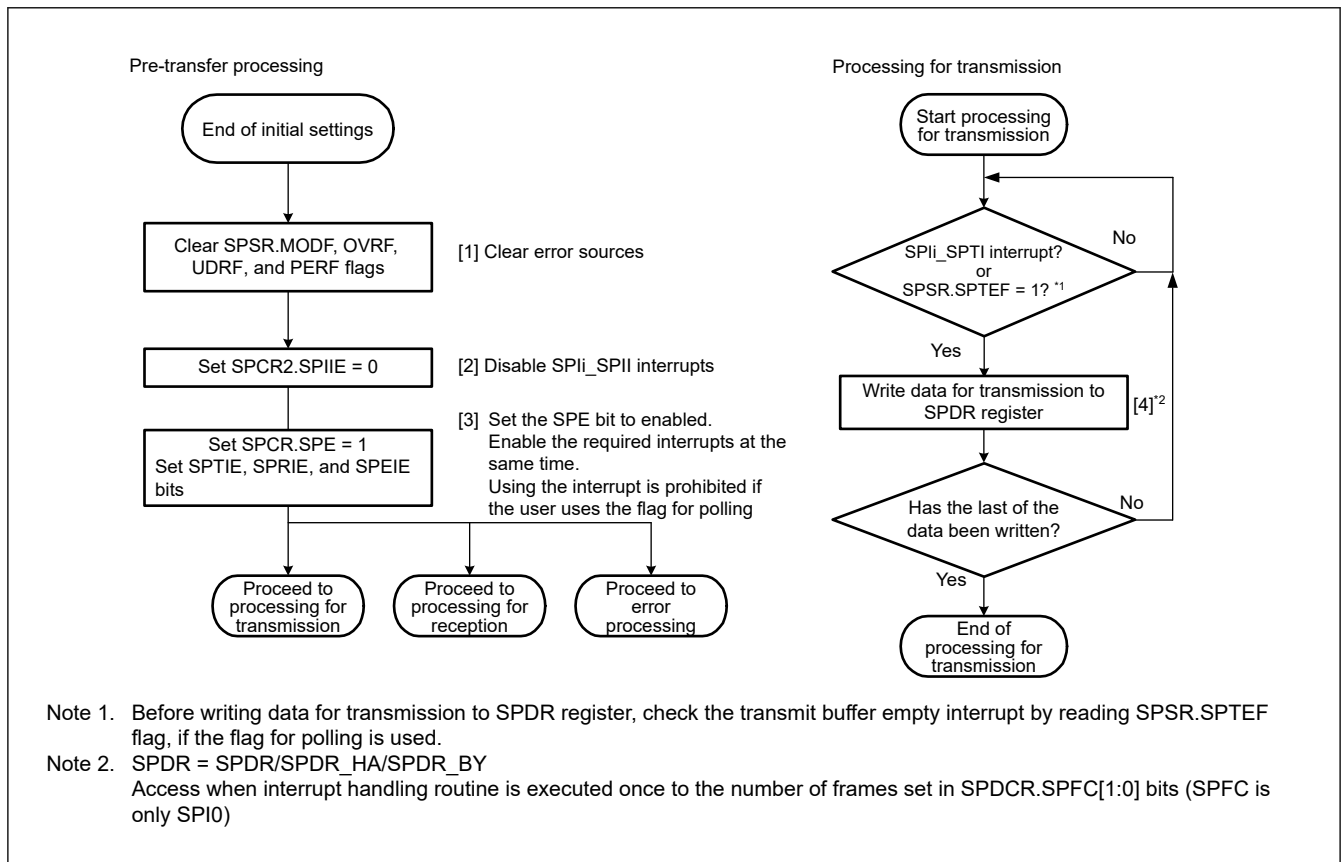


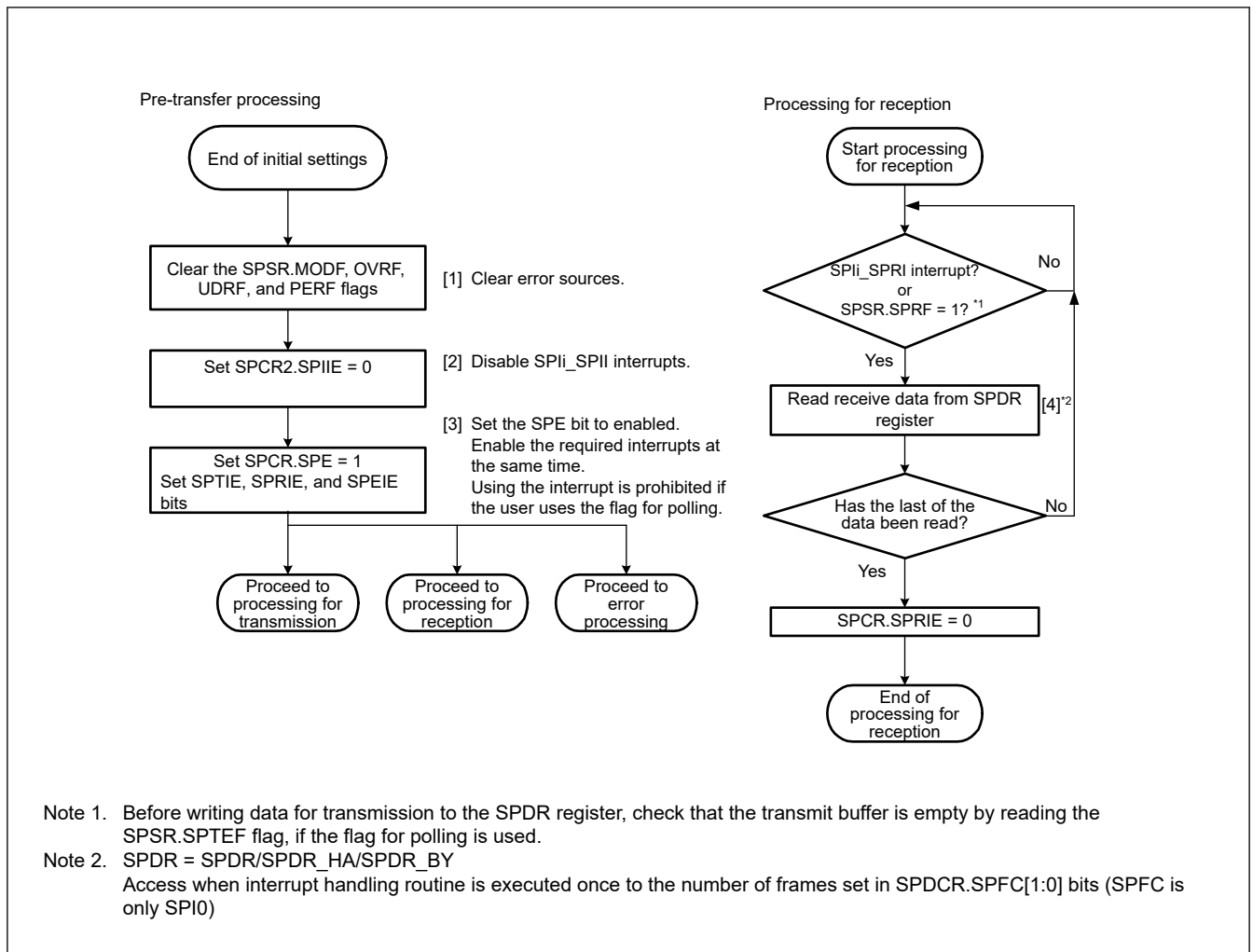
Figure 35.40 Example initialization flow in slave mode for SPI operation

(6) Software processing flow

Figure 35.41 to Figure 35.43 show examples of the flow of software processing.

Transmit processing flow**Figure 35.41 Transmission flow in slave mode****Receive processing flow**

The SPI does not handle receive-only operation, so processing for transmission is required.

**Figure 35.42 Reception flow in slave mode****Error processing flow**

In slave mode operation, even when a mode fault error is generated, the SPSR.MODF flag can be cleared regardless of the state of the SSLn0 pin.

When an error is detected by using an interrupt, clear the ICU.IELSRn.IR flag in the error processing routine. If this is not done, the ICU.IELSRn.IR flag might continue to indicate the SPIi_SPTI or SPIi_SPRI interrupt request. If the SPIi_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

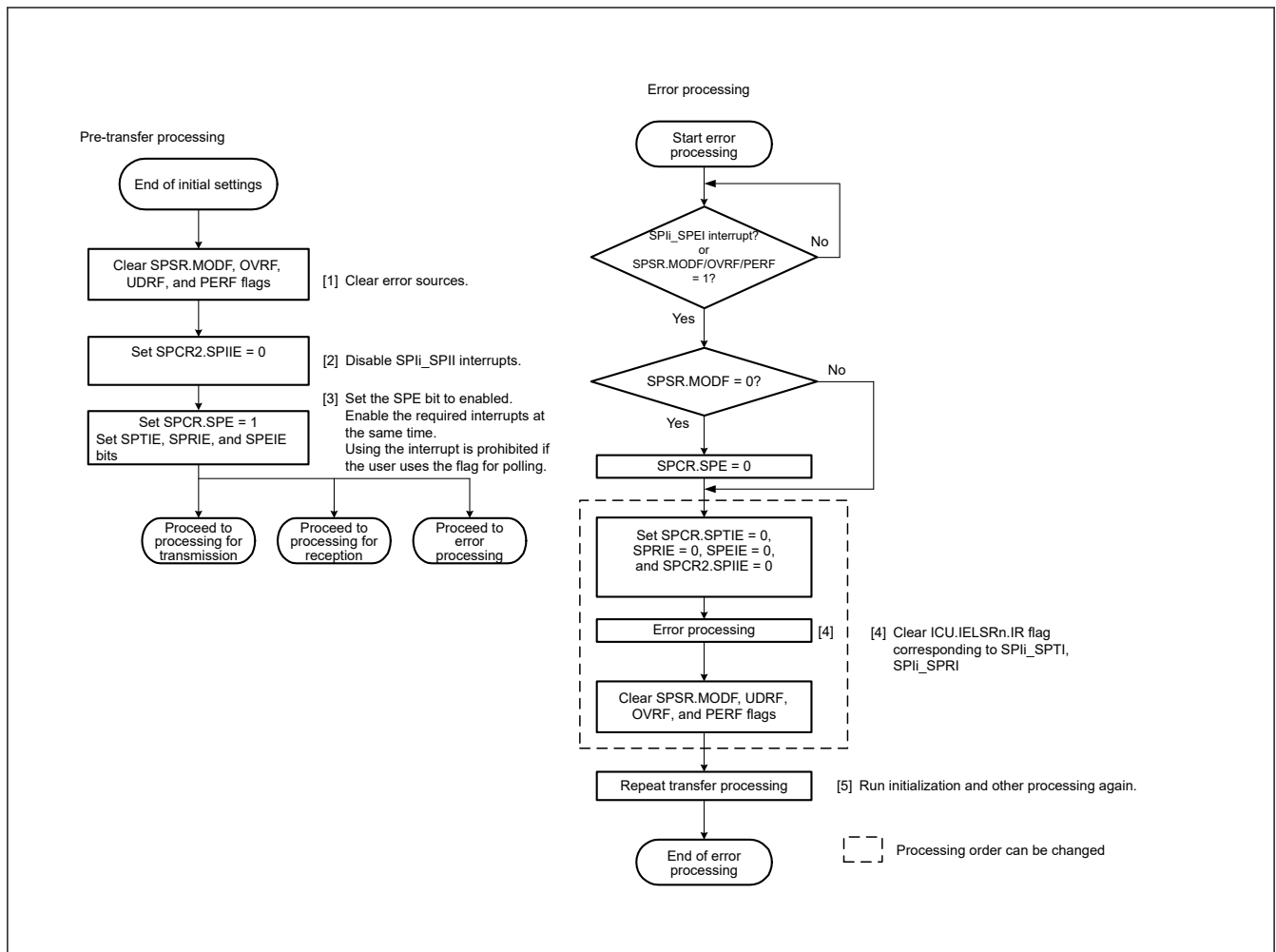


Figure 35.43 Error processing flow for slave mode

35.3.11 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the SPI. In clock synchronous operation, the SSLn pin is not used, and the RSPCKn, MOSIn, and MISO pins handle communications. All SSLn pins are available as I/O port pins.

Although clock synchronous operation does not require the use of the SSLni pin, operation of the module is the same as in SPI operation. In both master mode and slave mode operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected, because the SSLni pin is not used.

Additionally, do not perform operation if clock synchronous operation is enabled when the SPCMDm.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

35.3.11.1 Master mode operation

(1) Starting serial transfer

The SPI updates the data in the transmit buffer (SPTX) of SPDR/SPDR_HA when data is written to the SPDR/SPDR_HA register with the transmit buffer empty, the data for the next transfer not set and the SPSR.SPTEF flag is 1. When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits (only SPI0) are written to the SPDR/SPDR_HA, the SPI copies data from the transmit buffer to the shift register and starts serial transmission. On copying transmit data to the shift register, the SPI changes the status of the shift register to full, and on termination of serial transfer, it changes the status of the shift register to empty. The status of the shift register cannot be referenced.

Transfer in clock synchronous operation is conducted without the SSLn0 output signal. For details on the SPI transfer format, see [section 35.3.5. Transfer Formats](#).

(2) Terminating serial transfer

The SPI terminates the serial transfer after transmitting an RSPCKn edge corresponding to the sampling timing. If free space is available in the receive buffer (the SPSR.SPRF flag is 0), on termination of serial transfer, the SPI copies data from the shift register to the receive buffer of the SPI Data Register (SPDR/SPDR_HA).

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMDm.SPB[3:0] bits setting. Transfer in clock synchronous operation is conducted without the SSLn0 output signal. For details on the SPI transfer format, see [section 35.3.5. Transfer Formats](#).

(3) Sequence control

The transfer format used in master mode is determined by the SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers. Although the SSLni signals are not output in clock synchronous operation, these settings are valid.

The SPSCR register determines the sequence configuration for serial transfers that are executed by the SPI in master mode. The following parameters are specified in the SPCMDm register:

- SSLni output signal value
- MSB or LSB first
- Data length
- Some of the bit rate settings
- RSPCKn polarity and phase
- Whether SPCKD is to be referenced
- Whether SSLND is to be referenced
- Whether SPND is to be referenced

SPBR holds some of the bit rate settings such as SPCKD, an SPI clock delay value, SSLND, an SSL negation delay, and SPND, a next-access delay value.

Based on the sequence length that is assigned to SPSCR, the SPI makes up a sequence comprised of a part or all of SPCMDm register. The SPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the SPI function is enabled, the SPI loads the pointer to the commands in SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer. The SPI increments the pointer each time the next-access delay period for a data transfer ends. On completion of the serial transfer that corresponds to the final command comprising the sequence, the SPI sets the pointer to the SPCMD0 register, and in this manner the sequence is executed repeatedly.

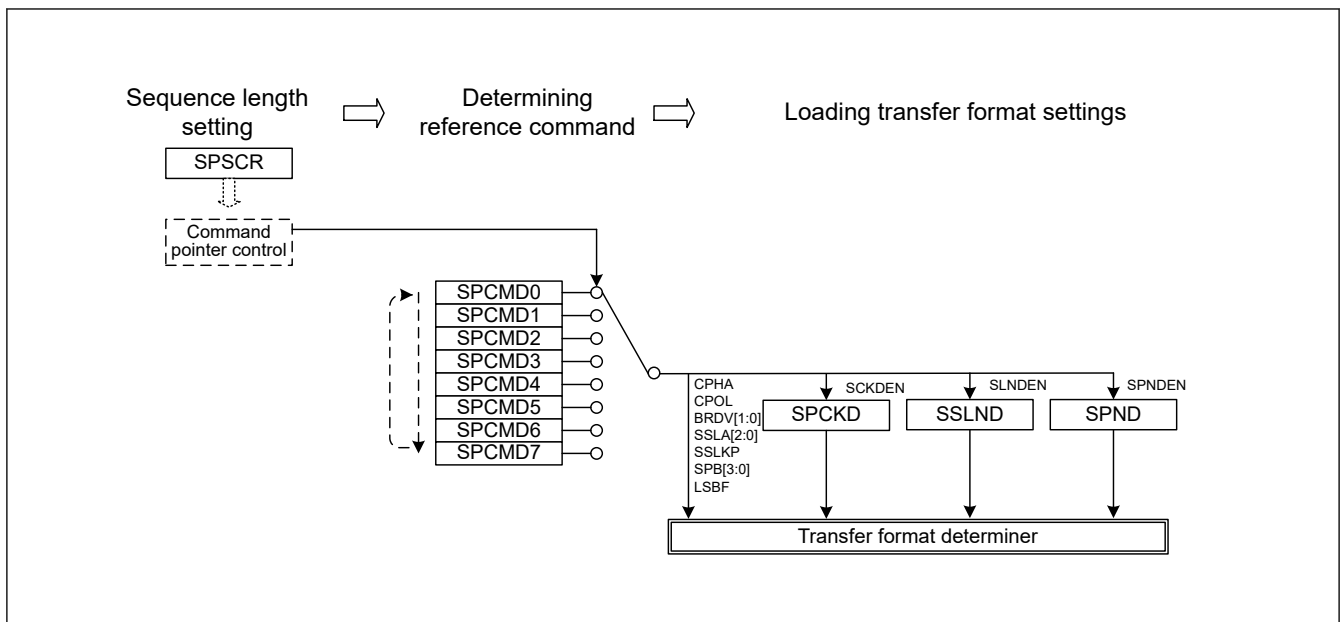


Figure 35.44 Procedure for determining the form of serial transmission in master mode

In this section, a frame is the combination of the data (SPDR/SPDR_HA) and the settings (SPCMDm).

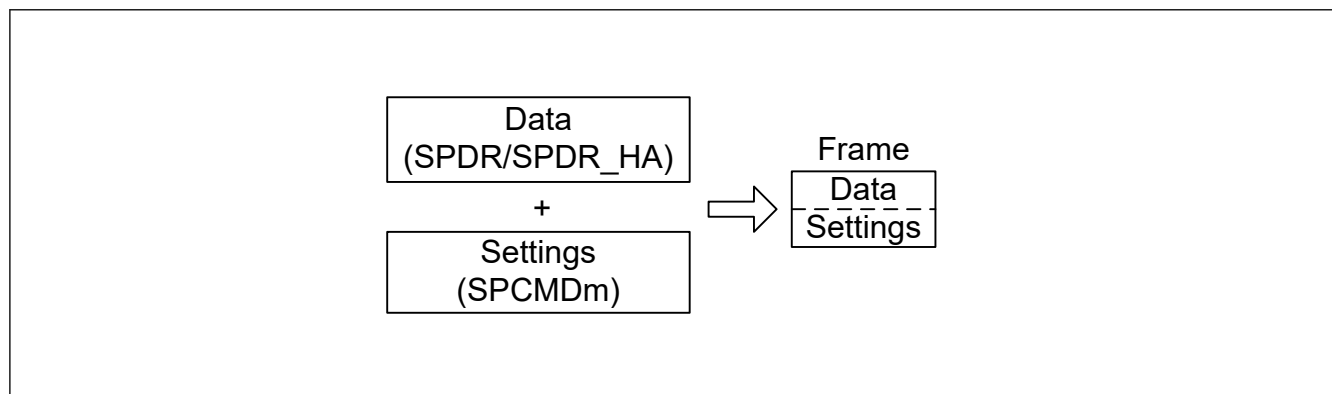


Figure 35.45 Conceptual diagram of frames

[Figure 35.46](#) shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in [Table 35.4](#).

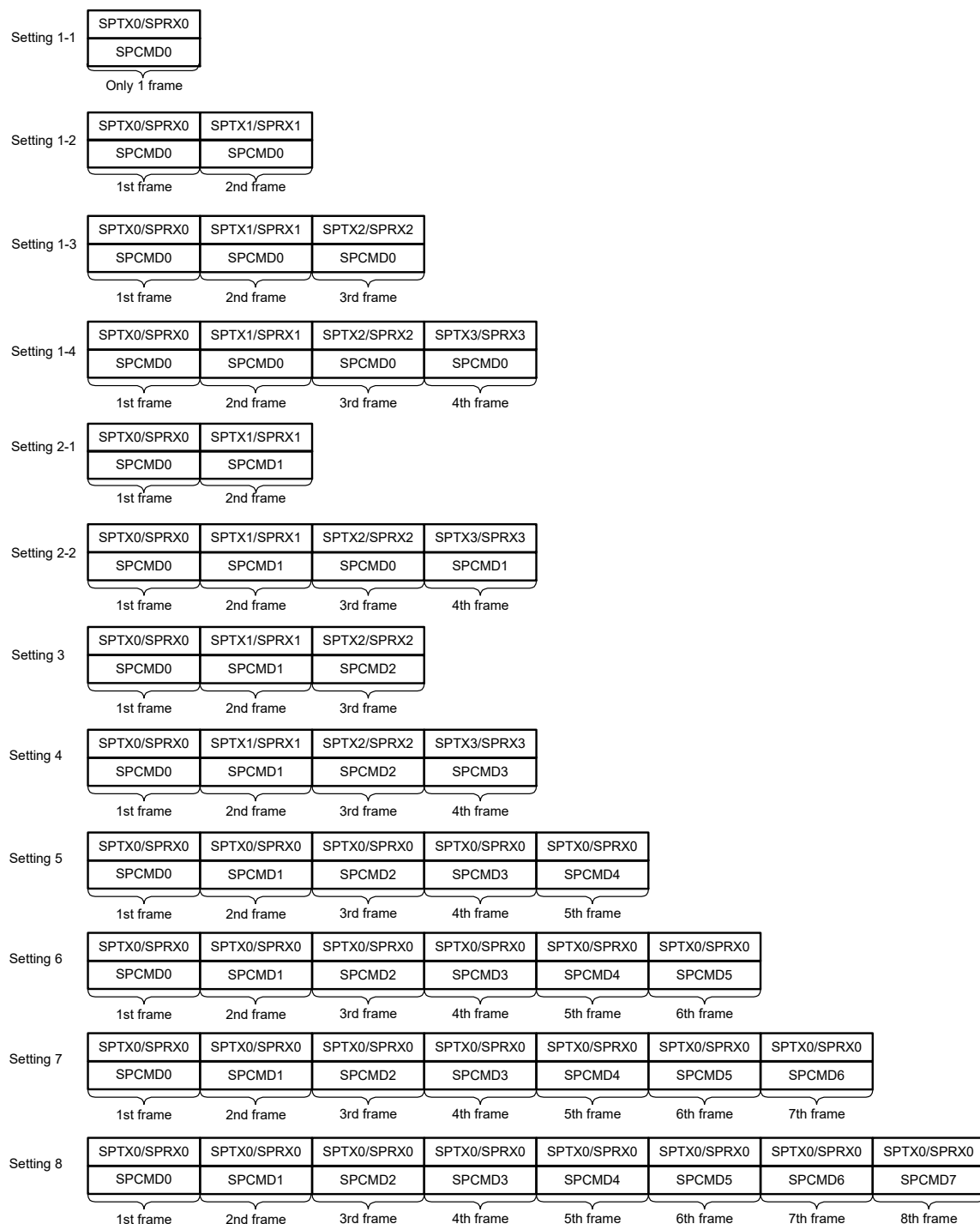


Figure 35.46 Correspondence between SPI Command Register and transmit and receive buffers in sequence operations

(4) Initialization flow

Figure 35.47 shows an example of initialization flow for clock synchronous operation when the SPI is used in master mode. For information on how to set up the ICU, DMAC or DTC, and I/O ports, see the descriptions given in the individual blocks.

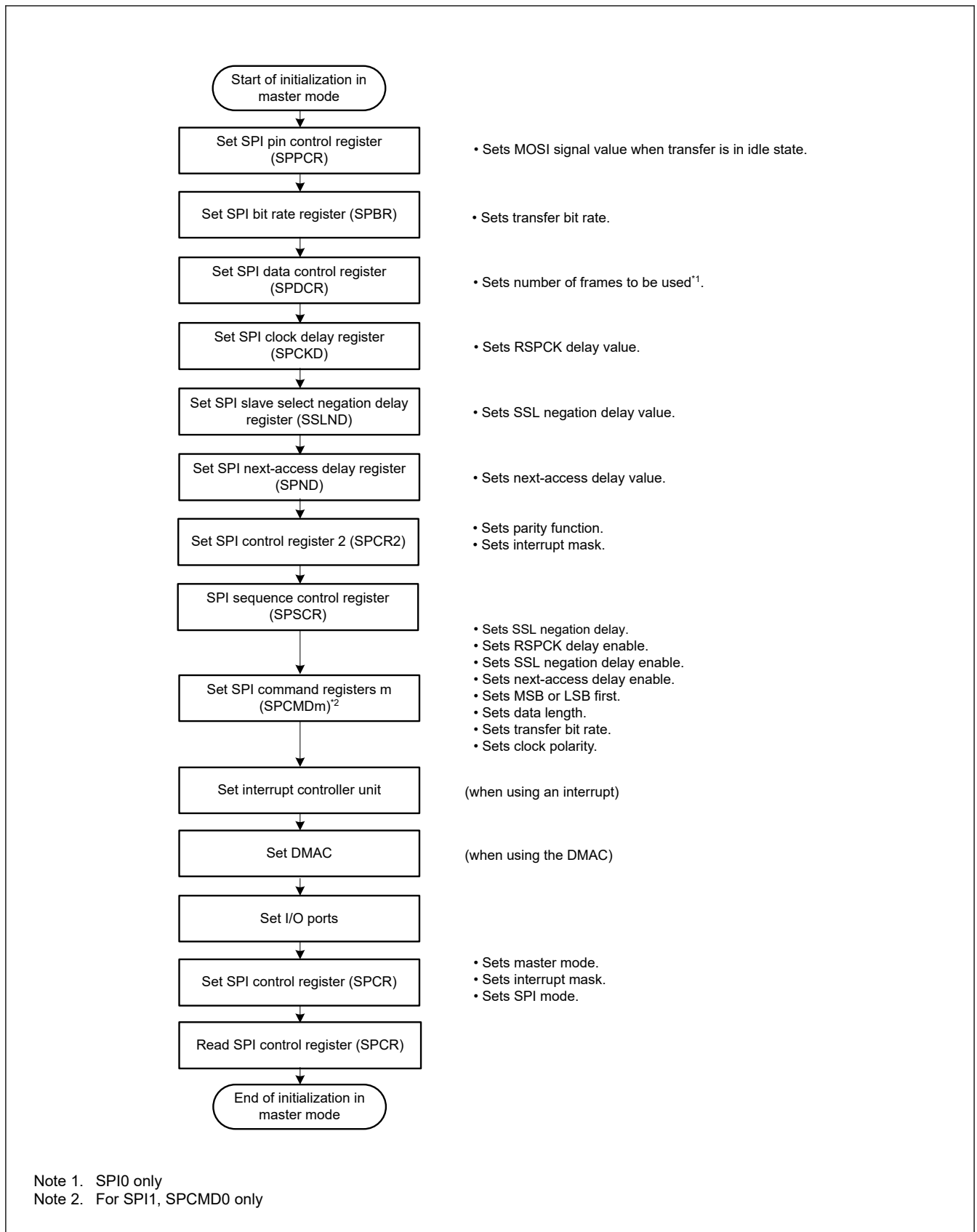


Figure 35.47 Example of initialization flow in master mode for clock synchronous operation

(5) Software processing flow

Software processing during clock synchronous master operation is the same as that for SPI master operation. For details, see (9) Software processing flow in [section 35.3.10.1. Master mode operation](#). Mode fault errors do not occur in clock synchronous operation.

35.3.11.2 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 MISO_n output signal. The SSL0 input signal is not used in clock synchronous operation. For details on the SPI transfer format, see [section 35.3.5. Transfer Formats](#).

(2) Terminating serial transfer

The SPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPSR.SPRF flag is 0), on termination of serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR/SPDR_HA register. On termination of a serial transfer, the SPI changes the status of the shift register to empty regardless of the receive buffer.

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length depends on the SPCMD0.SPB[3:0] bits setting. For details on the SPI transfer format, see [section 35.3.5. Transfer Formats](#).

(3) Initialization flow

[Figure 35.48](#) shows an example of initialization flow for clock synchronous operation when the SPI is used in slave mode. For a description of how to set up the ICU, DTC, and I/O ports, see the descriptions given in the individual blocks.

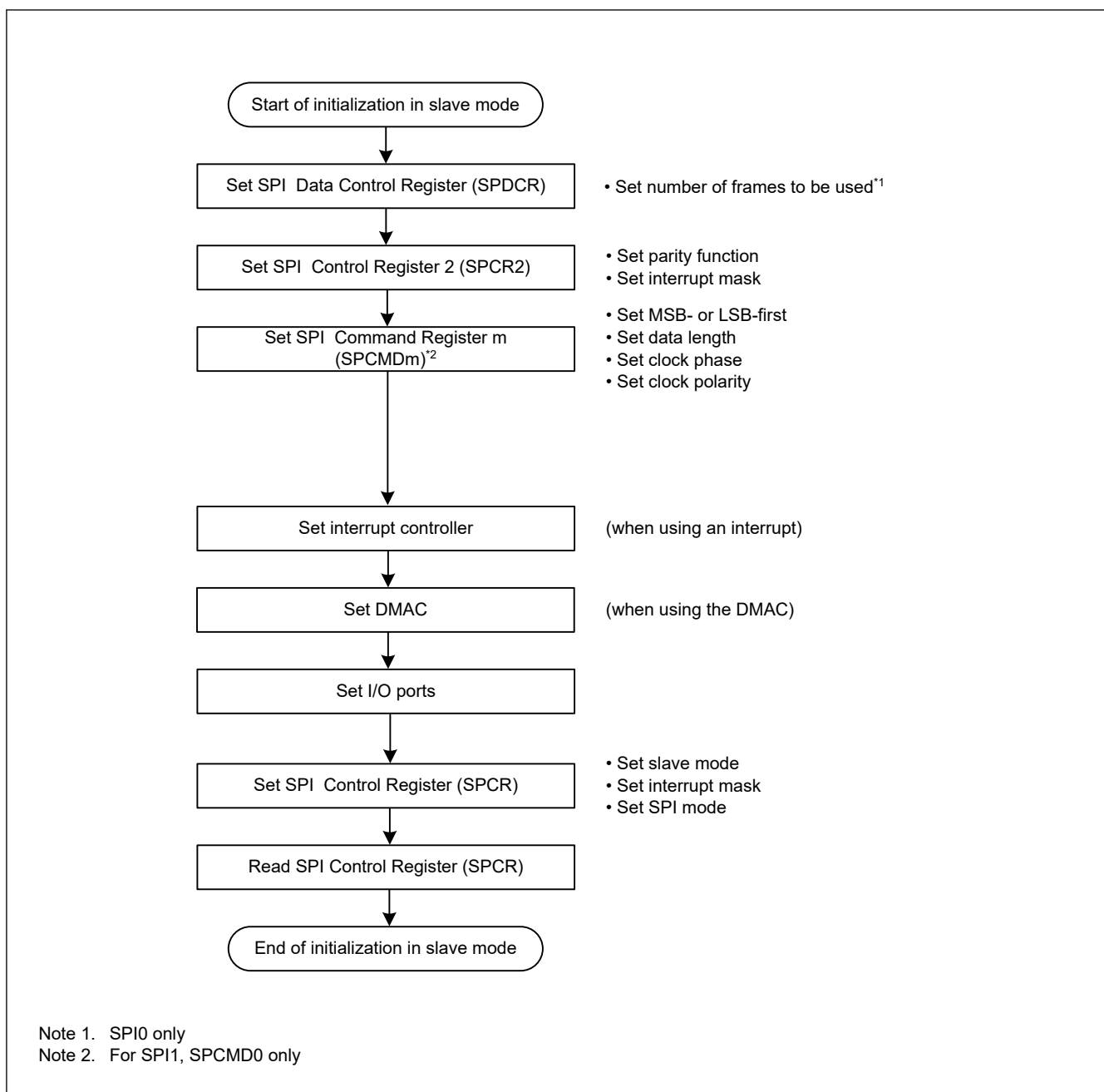


Figure 35.48 Example of initialization flow in slave mode for clock synchronous operation

(4) Software processing flow

Software processing during clock synchronous slave operation is the same as that for SPI slave operation. For details, see [\(6\)Software processing flow](#). Mode fault errors do not occur in clock synchronous mode.

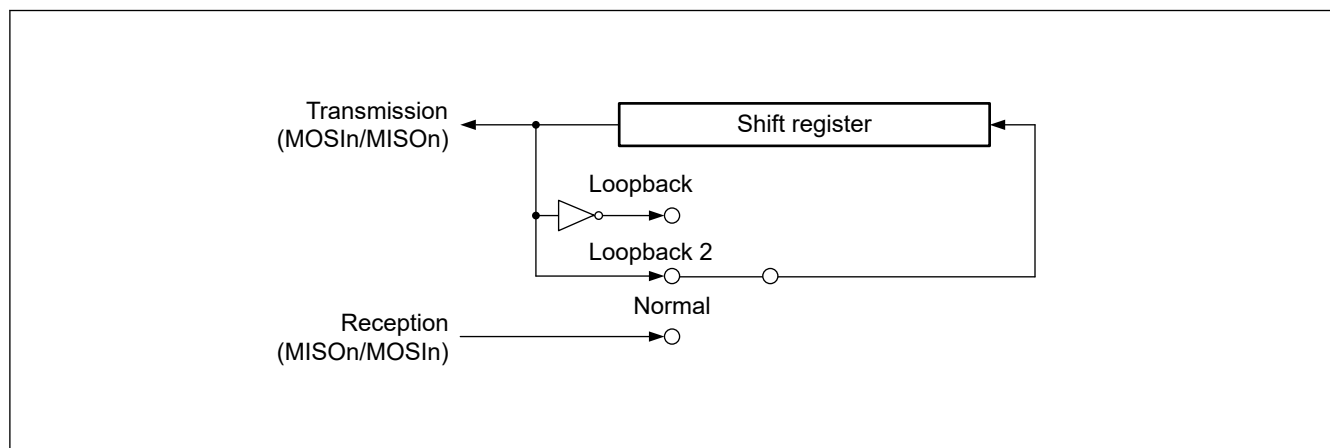
35.3.12 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the SPI shuts off the path between the MISO pin and the shift register if the SPCR.MSTR bit is 1, or between the MOSI pin and the shift register if the SPCR.MSTR bit is 0, and connects the input and output paths of the shift register, establishing a loopback mode. The SPI does not shut off the path between the MOSI pin and the shift register if the SPCR.MSTR bit is 1, or between the MISO pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the SPI or the reversed transmit data becomes the received data for the SPI.

[Table 35.12](#) lists the relationship between the SPLP2 and SPLP bits and the received data. [Figure 35.49](#) shows the configuration of the shift register I/O paths when the SPI in master mode is set to loopback mode (SPPCR.SPLP2 = 0, SPPCR.SPLP = 1).

Table 35.12 SPLP2 and SPLP bit settings and received data

SPPCR.SPLP2 bit	SPPCR.SPLP bit	Received data
0	0	Input data from the MOSIn pin or MISO pin
0	1	Inverted transmit data
1	0	Transmit data
1	1	Transmit data

**Figure 35.49** Configuration of shift register I/O paths in loopback mode for master mode

35.3.13 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. To detect defects in the parity bit adding unit and error detecting unit, the parity circuit performs self-diagnosis as shown in [Figure 35.50](#).

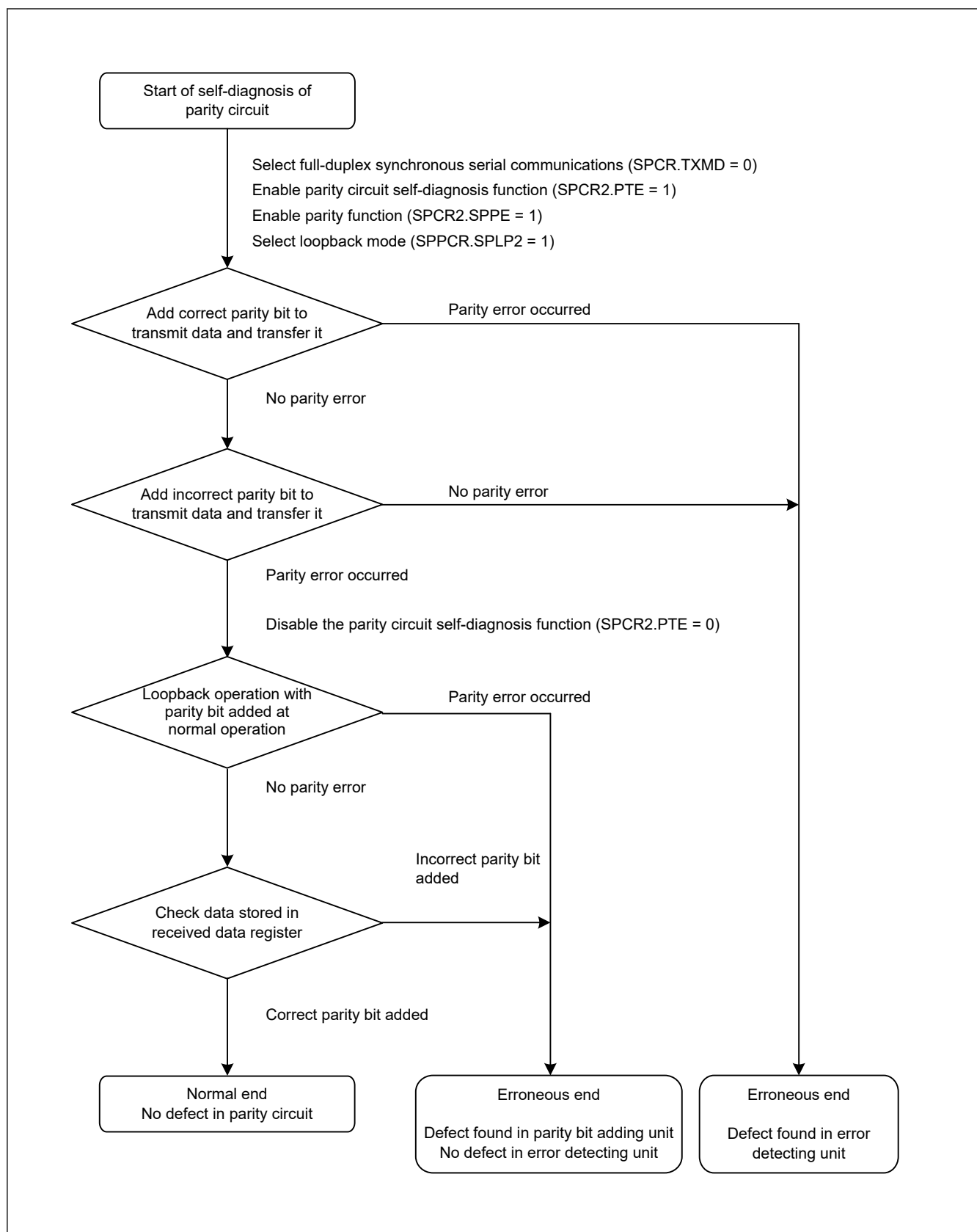


Figure 35.50 Self-diagnosis flow for parity circuit

35.3.14 Interrupt Sources

The SPI has the following interrupt sources:

- Receive buffer full
- Transmit buffer empty
- SPI error (mode-fault, underrun, overrun, or parity error)
- SPI idle
- Transmission-complete

The DMAC or DTC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Because the vector address for the SPIi_SPEI (SPI error interrupt) is allocated to interrupt requests on mode-fault, underrun, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the SPI are listed in [Table 35.13](#). An interrupt is generated on satisfaction of one of the interrupt conditions in [Table 35.13](#). Clear the receive buffer full and transmit buffer empty sources through a data transfer.

When using the DMAC or DTC to perform data transmission and reception, you must first set up the DMAC or DTC to be in a transfer-enabled status before setting the SPI. For information on setting up the DMAC or DTC, see [section 19, DMA Controller \(DMAC\)](#) and [section 20, Data Transfer Controller \(DTC\)](#).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt occur while the ICU.IELSRn.IR flag is 1, the interrupt is not output as a request for the ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IELSRn.IR flag becomes 0. A retained interrupt request is automatically discarded when it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be set to 0.

Table 35.13 SPI interrupt sources

Interrupt source	Symbol	Interrupt condition	DMAC/DTC activation
Receive buffer full	SPIi_SPRI	The receive buffer becomes full (SPSR.SPRF flag is 1) while the SPCR.SPRIE bit is 1	Possible
Transmit buffer empty	SPIi_SPTI	The transmit buffer becomes empty (SPSR.SPTEF flag is 1) while the SPCR.SPTIE bit is 1	Possible
SPI error (mode-fault, underrun, overrun, or parity error)	SPIi_SPEI	The SPSR.MODF, OVRF, UDRF or PERF flag sets to 1 while the SPCR.SPEIE bit is 1	Impossible
SPI idle	SPIi_SPII	The SPSR.IDLNF flag sets to 0 while the SPCR2.SPIIE bit is 1	Impossible
Transmission-complete	SPIi_SPTEND	• Master mode: an interrupt is generated when the IDLNF flag (SPI idle flag) changes from 1 to 0 • Slave mode: an interrupt occurs on conditions shown in Table 35.15	Impossible

35.4 Event Link Controller Event Output

The Event Link Controller (ELC) can produce the following event output signals: (SPI0 only)

- Receive buffer full event output
- Transmit buffer empty event output
- Mode-fault, underrun, overrun, or parity error event output
- SPI idle event output
- Transmission-completed event output

The event link output signal is output regardless of the interrupt enable bit setting.

35.4.1 Receive Buffer Full Event Output

This event signal is output when received data is transferred from the shift register to the SPDR/SPDR_HA on completion of serial transfer.

35.4.2 Transmit Buffer Empty Event Output

This event signal is output when data for transmission is transferred from the transmit buffer to the shift register and when the value of the SPE bit changes from 0 to 1.

35.4.3 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 35.5.4. Constraints on Mode-Fault, Underrun, Overrun, or Parity Error Event Output](#) if using this event signal.

(1) Mode-fault

[Table 35.14](#) lists the conditions for occurrence of a mode-fault event.

Table 35.14 Conditions for mode-fault occurrence

SPI mode	SPCR.MODFEN bit	SSLn0 pin	Remarks
SPI operation (SPMS = 0) Slave (SPCR.MSTR = 0)	1	Not active	Event is output only when the SSLn0 pin is deactivated during transmission

(2) Underrun

This event signal is output in response to an underrun when a serial transfer starts while the transmission data is not ready, and the value of the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1. Under these conditions, the MODF and UDRF flags are set to 1.

(3) Overrun

This event signal is output in response to an overrun when a serial transfer completes while the receive buffer contains unread data and the value of the SPCR.TXMD bit is 0. Under these conditions, the OVRF flag is set to 1.

(4) Parity error

This event signal is output in response to a parity error detected on completion of a serial transfer while the value of the TXMD bit in SPCR is 0 and the value of the SPPE bit in SPCR2 is 1.

35.4.4 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).

35.4.5 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 35.15](#) lists the conditions for occurrence of a transmission-completed event in slave mode.

Table 35.15 Conditions for generation of transmission-complete event in slave mode

Conditions	Transmit buffer state	Shift register state	Other
SPI operation (SPMS = 0)	Empty	Empty	Negation of SSLn0 input
Clock synchronous operation (SPMS = 1)	Empty	Empty	Edge detection of the last RSPCKn

Whether the operation is in master mode or slave mode, an event is not output if 0 is written to the SPCR.SPE bit in transmission or the SPCR.SPE bit is cleared by the mode-fault error or the underrun error.

35.5 Usage Notes

35.5.1 Settings for the Module-Stop State

The Module Stop Control Register B (MSTPCRB) can enable or disable the SPI operation. The SPI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details on the Module Stop Control Register B, see [section 13, Power-Saving Functions](#).

35.5.2 Constraint on Low-Power Functions

When using the module-stop function and entering a low-power mode other than Sleep mode, set the SPCR.SPE bit to 0 before completing communication.

35.5.3 Constraints on Starting Transfer

If the ICU.IELSRn.IR flag is 1 when transfer starts, the interrupt request is internally retained, which can lead to unanticipated behavior of the ICU.IELSRn.IR flag.

To prevent this, use the following procedure to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1):

1. Confirm that transfer stopped (the SPCR.SPE bit is 0).
2. Set the associated interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) to 0.
3. Read the associated interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) and confirm that its value is 0.
4. Set the ICU.IELSRn.IR flag to 0.

35.5.4 Constraints on Mode-Fault, Underrun, Overrun, or Parity Error Event Output

Using the mode-fault, underrun, overrun, or parity error event is prohibited if the SPI is in multi-master mode (when the SPCR.SPMS bit is 0, the SPCR.MSTR bit is 1, and the SPCR.MODFEN bit is 1).

35.5.5 Constraints on the SPSR.SPRF and SPSR.SPTEF Flags

If the polling flags, SPRF and SPTEF, are used, using the interrupts is prohibited, and you must set the SPCR.SPRIE and SPCR.SPTIE bits to 0. Either the interrupts or the flags can be used, but not both.

35.5.6 Constraints on Register Settings

Set registers in the SPI while the SPCR.SPE bit is 0 (disabling the SPI).

35.5.7 Note on Resuming Communications in Slave Mode

Caution should be taken to set the SPCMDm.CPHA bit to the value 0 if operation is to be as a slave. Re-enabling of the SPI function (changing the SPCR.SPE bit from 0 to 1) within the last half cycle of the clock signal (RSPCKn) after disabling the function during serial transfer will lead to errors in detection in the form of recognizing the transition in the value of RSPCKn when the transfer is over (from 0 to 1 or from 1 to 0) as a transition at the start of the next transfer as shown in [Figure 35.22](#).

Note that enabling the SPI function for the resumption of communications while the SPE bit is 0 must be after the transition of RSPCKn at the completion of transfer.

36. Quad Serial Peripheral Interface (QSPI)

36.1 Overview

The QSPI is a memory controller for connecting a serial ROM (nonvolatile memory such as a serial flash memory, serial EEPROM, or serial FeRAM) that has an SPI-compatible interface.

Note that booting up of this device through the QSPI is not possible. Be sure to boot it up from the on-chip ROM.

Table 36.1 lists the QSPI specifications, Figure 36.1 shows a block diagram, and Table 36.2 lists the I/O pins.

Table 36.1 QSPI specifications

Parameter	Specifications
Number of channels	1 channel
SPI protocols	- Single SPI protocol, extended SPI protocol to achieve full-duplex communications Note: Standard or fast reading can only be used in single SPI operation. Four-wire communications with the serial flash memory by using the QSSL, QSPCLK, QIO0, and QIO1 pins (QIO0, QSSL, and QSPCLK for output, and QIO1 for input) - Dual SPI protocol to achieve half-duplex communications Four-wire communications with the serial flash memory by using the QSSL, QSPCLK, QIO0, and QIO1 pins (QSSL and QSPCLK for output, and QIO0 and QIO1 for input and output) - Quad SPI protocol to achieve half-duplex communications Six-wire communications with the serial flash memory by using the QSSL, QSPCLK, and QIO0 to QIO3 pins (QSSL and QSPCLK for output, and QIO0 to QIO3 for input and output)
SPI mode	- SPI mode 0: The QSPCLK signal is driven low when the SPI bus is not active. - SPI mode 3: The QSPCLK signal is driven high when the SPI bus is not active.
SPI timing adjustment function	The following settings are possible to suit various types of serial flash memory device: - SPI bus reference cycle (SFMSKC.SFMDV[4:0]) - Duty cycle correction (SFMSKC.SFMDTY) - Adjustment of the number of dummy cycles (SFMSDC.SFMDN[3:0]) - Minimum width at high level for the QSSL signal (SFMSSC.SFMSW[3:0]) - QSSL signal setup time (SFMSSC.SFMSLD) - QSSL signal hold time (SFMSSC.SFMSHD) - Serial data output enable hold time (SFMSMD.SFMOEX)
ROM access mode	- Support for Read, Fast Read, Fast Read Dual Output, Fast Read Dual I/O, Fast Read Quad Output, and Fast Read Quad I/O instructions - Substitutable instruction code - Prefetch function (data are sequentially stored in a buffer after one request without waiting for further requests to read the serial flash memory) - Polling processing - SPI bus cycle extension function - XIP mode (allowing skipping of the reception of an instruction code to read the serial flash memory) Note: ROM access mode is only possible with reading .
Direct communication mode	Flexible support for a wide variety of serial flash memory instructions and functions through software control, including erase, ID read, and power-down control
Interrupt source	Error interrupts
Module-stop function	Module-stop state can be set to reduce power consumption.

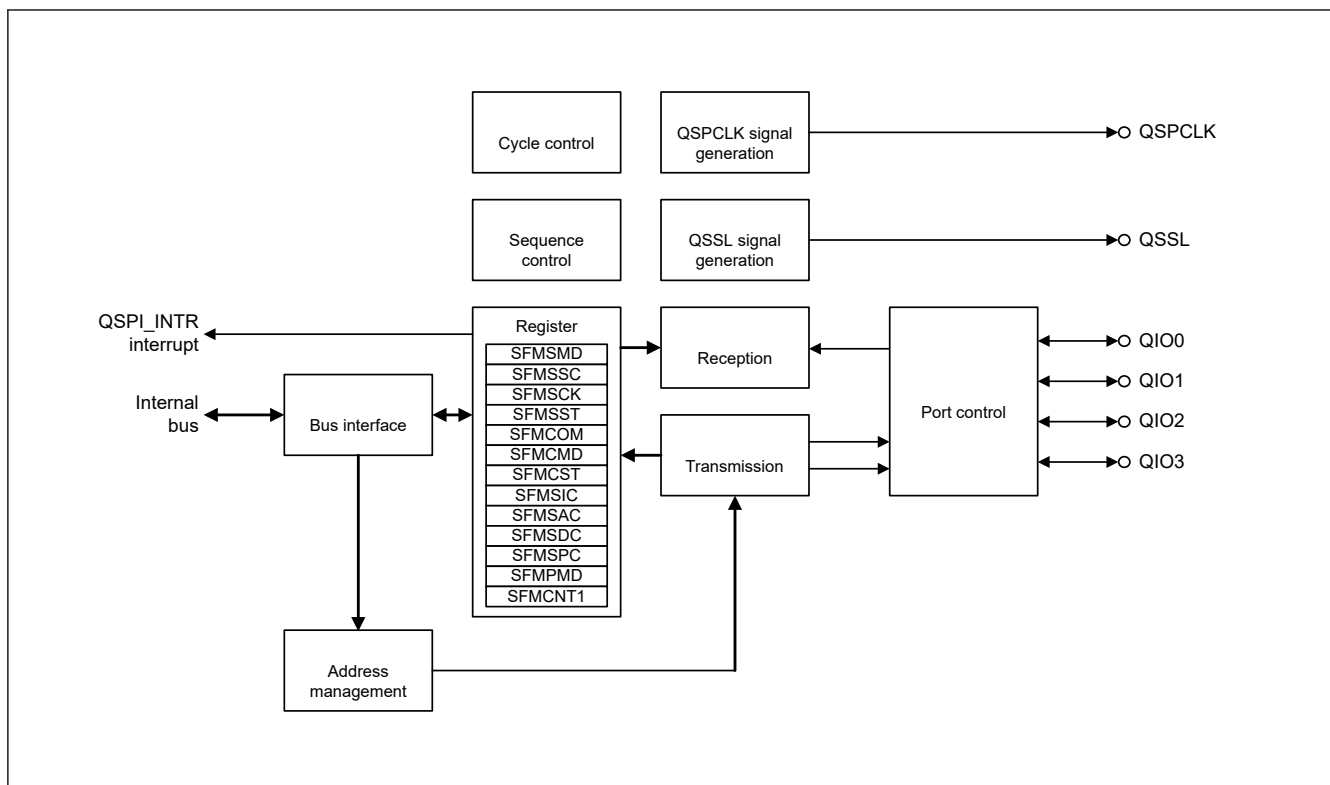


Figure 36.1 QSPI block diagram

Table 36.2 QSPI I/O pins

Function	Pin name	I/O	Description
QSPI	QSPCLK	Output	QSPI clock output pin.
	QSSL	Output	QSPI slave output pin.
	QIO0 to QIO3	I/O	Data0 to Data3

36.2 Register Descriptions

36.2.1 SFMSMD : Transfer Mode Control Register

Base address: QSPI = 0x6400_0000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SFMC CE	—	—	—	—	—	SFMO EX	SFMM D3	—	SFMP FE	SFMSE[1:0]	—	—	SFMRM[2:0]	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SFMRM[2:0]	Serial interface read mode select 0 0 0: Standard Read 0 0 1: Fast Read 0 1 0: Fast Read Dual Output 0 1 1: Fast Read Dual I/O 1 0 0: Fast Read Quad Output 1 0 1: Fast Read Quad I/O Others: Setting prohibited	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	SFMSE[1:0]	QSSL extension function select after SPI bus access 0 0: Do not extend QSSL 0 1: Extend QSSL by 33 QSPCLK 1 0: Extend QSSL by 129 QSPCLK 1 1: Extend QSSL infinitely	R/W
6	SFMPFE	Prefetch function select 0: Disable function 1: Enable function	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	SFMMD3	SPI mode select. 0: SPI mode 0 1: SPI mode 3	R/W
9	SFMOEX	Extension select for the I/O buffer output enable signal for the serial interface 0: Do not extend 1: Extend by 1 QSPCLK	R/W
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	SFMCCE	Read instruction code select 0: Uses automatically generated SPI instruction code*1 1: Use instruction code in the SFMSIC register	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When QSPI accesses serial flash memory, the instruction code is based on the SFMSAC register and SFMSMD register settings.
See [section 36.6.1. SPI Instructions That Are Automatically Generated](#).

36.2.2 SFMSSC : Chip Selection Control Register

Base address: QSPI = 0x6400_0000

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	SFMS LD	SFMS HD	SFMSW[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1

Bit	Symbol	Function	R/W
3:0	SFMSW[3:0]	Minimum high-level width select for QSSL signal 0x0: 1 QSPCLK 0x1: 2 QSPCLK 0x2: 3 QSPCLK 0x3: 4 QSPCLK 0x4: 5 QSPCLK 0x5: 6 QSPCLK 0x6: 7 QSPCLK 0x7: 8 QSPCLK 0x8: 9 QSPCLK 0x9: 10 QSPCLK 0xA: 11 QSPCLK 0xB: 12 QSPCLK 0xC: 13 QSPCLK 0xD: 14 QSPCLK 0xE: 15 QSPCLK 0xF: 16 QSPCLK	R/W
4	SFMSHD	QSSL Signal Hold Time 0: QSSL outputs high after 0.5 QSPCLK cycles from the last rising edge of QSPCLK. 1: QSSL outputs high after 1.5 QSPCLK cycles from the last rising edge of QSPCLK.	R/W
5	SFMSLD	QSSL Signal Setup Time 0: QSSL outputs low before 0.5 QSPCLK cycles from the first rising edge of QSPCLK. 1: QSSL outputs low before 1.5 QSPCLK cycles from the first rising edge of QSPCLK.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

36.2.3 SFMSKC : Clock Control Register

Base address: QSPI = 0x6400_0000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	SFMD TY	SFMDV[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
4:0	SFMDV[4:0]	Serial interface reference cycle select. (Pay attention to irregularities.) 0x00: 2 PCLKA 0x01: 3 PCLKA (divided by an odd number)*1*2 0x02: 4 PCLKA 0x03: 5 PCLKA (divided by an odd number)*1*2 0x04: 6 PCLKA 0x05: 7 PCLKA (divided by an odd number)*1*2 0x06: 8 PCLKA 0x07: 9 PCLKA (divided by an odd number)*1*2 0x08: 10 PCLKA 0x09: 11 PCLKA (divided by an odd number)*1*2 0x0A: 12 PCLKA 0x0B: 13 PCLKA (divided by an odd number)*1*2 0x0C: 14 PCLKA 0x0D: 15 PCLKA (divided by an odd number)*1*2 0x0E: 16 PCLKA 0x0F: 17 PCLKA (divided by an odd number)*1*2 0x10: 18 PCLKA 0x11: 20 PCLKA 0x12: 22 PCLKA 0x13: 24 PCLKA 0x14: 26 PCLKA 0x15: 28 PCLKA 0x16: 30 PCLKA 0x17: 32 PCLKA 0x18: 34 PCLKA 0x19: 36 PCLKA 0x1A: 38 PCLKA 0x1B: 40 PCLKA 0x1C: 42 PCLKA 0x1D: 44 PCLKA 0x1E: 46 PCLKA 0x1F: 48 PCLKA	R/W
5	SFMDTY*2	Duty ratio correction function select for the QSPCLK signal when divided by an odd number 0: Make no correction 1: Make correction	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Set the SFMDTY bit to 1 when PCLKA is to be divided by an odd number.

Note 2. Two cycles of PCLKA cannot be set when the frequency of PCLKA is higher than 48 MHz in the boost mode.

36.2.4 SFMSST : Status Register

Base address: QSPI = 0x6400_0000

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	PFOF F	PFFUL	—	PFCNT[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	PFCNT[4:0]	Number of bytes of prefetched data 0x00: 0 byte 0x01: 1 byte 0x02: 2 bytes 0x03: 3 bytes 0x04: 4 bytes 0x05: 5 bytes 0x06: 6 bytes 0x07: 7 bytes 0x08: 8 bytes 0x09: 9 bytes 0x0A: 10 bytes 0x0B: 11 bytes 0x0C: 12 bytes 0x0D: 13 bytes 0x0E: 14 bytes 0x0F: 15 bytes 0x10: 16 bytes 0x11: 17 bytes 0x12: 18 bytes Others: Reserved	R
5	—	This bit is read as 0.	R
6	PFFUL	Prefetch buffer state 0: Prefetch buffer has free space 1: Prefetch buffer is full	R
7	PFOFF	Prefetch function operating state 0: Prefetch function operating 1: Prefetch function not enabled or not operating	R
31:8	—	These bits are read as 0.	R

36.2.5 SFMCOM : Communication Port Register

Base address: QSPI = 0x6400_0000

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SFMD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	SFMD[7:0]	Port for direct communication with the SPI bus Input and output from this port are converted to an SPI bus cycle in direct communications mode (DCOM = 1). Access to this port is ignored in ROM access mode.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

36.2.6 SPMCMD : Communication Mode Control Register

Base address: QSPI = 0x6400_0000

Offset address: 0x014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DCOM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DCOM	Mode select for communication with the SPI bus 0: ROM access mode 1: Direct communication mode*1	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SPMCMD.DCOM = 1 must be written when the transaction ends. For details, see [section 36.10. Direct Communication Mode](#).

36.2.7 SPMCST : Communication Status Register

Base address: QSPI = 0x6400_0000

Offset address: 0x018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	EROM R	—	—	—	—	—	—	COMB SY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	COMBSY	SPI bus cycle completion state in direct communication 0: No serial transfer being processed 1: Serial transfer being processed	R
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	EROMR	ROM access detection status in direct communication mode 0: ROM access not detected 1: ROM access detected	R/(W)*1
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit.

36.2.8 SFMSIC : Instruction Code Register

Base address: QSPI = 0x6400_0000

Offset address: 0x020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SFMCIC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SFMCIC[7:0]	Serial flash instruction code to substitute	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

36.2.9 SFMSAC : Address Mode Control Register

Base address: QSPI = 0x6400_0000

Offset address: 0x024

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	SFM4 BC	—	—	SFMAS[1:0]	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
1:0	SFMAS[1:0]	Number of address bytes select for the serial interface 0 0: 1 byte 0 1: 2 bytes 1 0: 3 bytes 1 1: 4 bytes	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	SFM4BC	Default instruction code select, when the serial interface address width is 4 bytes 0: Do not use 4-byte address read instruction code 1: Use 4-byte address read instruction code	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

36.2.10 SFMSDC : Dummy Cycle Control Register

Base address: QSPI = 0x6400_0000

Offset address: 0x028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SFMXD[7:0]								SFMX EN	SFMX ST	—	—	SFMDN[3:0]			
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	SFMDN[3:0]	Number of dummy cycles select for Fast Read instructions 0x0: Default dummy cycles for each instruction: - Fast Read Quad I/O: 6 QSPCLK - Fast Read Quad Output: 8 QSPCLK - Fast Read Dual I/O: 4 QSPCLK - Fast Read Dual Output: 8 QSPCLK - Fast Read: 8 QSPCLK 0x1: 3 QSPCLK* ¹ 0x2: 4 QSPCLK 0x3: 5 QSPCLK 0x4: 6 QSPCLK 0x5: 7 QSPCLK 0x6: 8 QSPCLK 0x7: 9 QSPCLK 0x8: 10 QSPCLK 0x9: 11 QSPCLK 0xA: 12 QSPCLK 0xB: 13 QSPCLK 0xC: 14 QSPCLK 0xD: 15 QSPCLK 0xE: 16 QSPCLK 0xF: 17 QSPCLK	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	SFMXST	XIP mode status 0: Normal (non-XIP) mode 1: XIP mode	R
7	SFMXEN	XIP mode permission 0: Prohibit XIP mode 1: Permit XIP mode	R/W
15:8	SFMXD[7:0]	Mode data for serial flash (Controls XIP mode.)* ²	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. To avoid a conflict with the input/output switch of the serial flash memory pin connected to QIO0 pin, select more than four cycles of QSPCLK as the number of dummy cycles for the fast read instruction when the output enable signal is extended by setting the SFMOEX bit in the SFMSMD register to 1.

Note 2. As the mode data for serial flash memory, specify the XIP mode setting data set in actual serial flash memory.

36.2.11 SFMSPC : SPI Protocol Control Register

Base address: QSPI = 0x6400_0000

Offset address: 0x030

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	SFMS DE	—	—	SFMSPC[1:0]	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SFMSPC[1:0]	SPI protocol select*1 0 0: Single SPI Protocol, Extended SPI protocol 0 1: Dual SPI protocol 1 0: Quad SPI protocol 1 1: Setting prohibited	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	SFMSDE	QSPCLK extended selection bit when switching I/O of QIOn pin 0: No QSPCLK extension 1: QSPCLK expansion when switching I/O direction of QIOn pin	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The states of the QIO2 and QIO3 pins change depending on the settings of the SFMSMD.SFMRM[2:0] and SFMSPC[1:0] bits. For details, see [section 36.9. QIO2 and QIO3 Pin States](#).

36.2.12 SFMPMD : Port Control Register

Base address: QSPI = 0x6400_0000

Offset address: 0x034

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SFMW PL	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	SFMWPL	WP pin level specification 0: Low level 1: High level	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

36.2.13 SPMCNT1 : External QSPI Address Register

Base address: QSPI = 0x6400_0000

Offset address: 0x804

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	QSPI_EXT[5:0]						—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
25:0	—	These bits are read as 0. The write value should be 0.	R/W
31:26	QSPI_EXT[5:0]	Bank switching address When accessing from 0x60000000 to 0x63FFFFFF, the address bus is set from QSPI_EXT[5:0] to the upper 6 bits of the internal bus address for the address bus. 0x00: QSPI bank 00 0x01: QSPI bank 01 0x02: QSPI bank 02 ⋮ 0x3C: QSPI bank 60 0x3D: QSPI bank 61 0x3E: QSPI bank 62 0x3F: Setting prohibited	R/W

36.3 Memory Map

36.3.1 External Bus Space

The locations of a serial flash memory and control register on the address space are determined by the address range of the area set in the configuration.

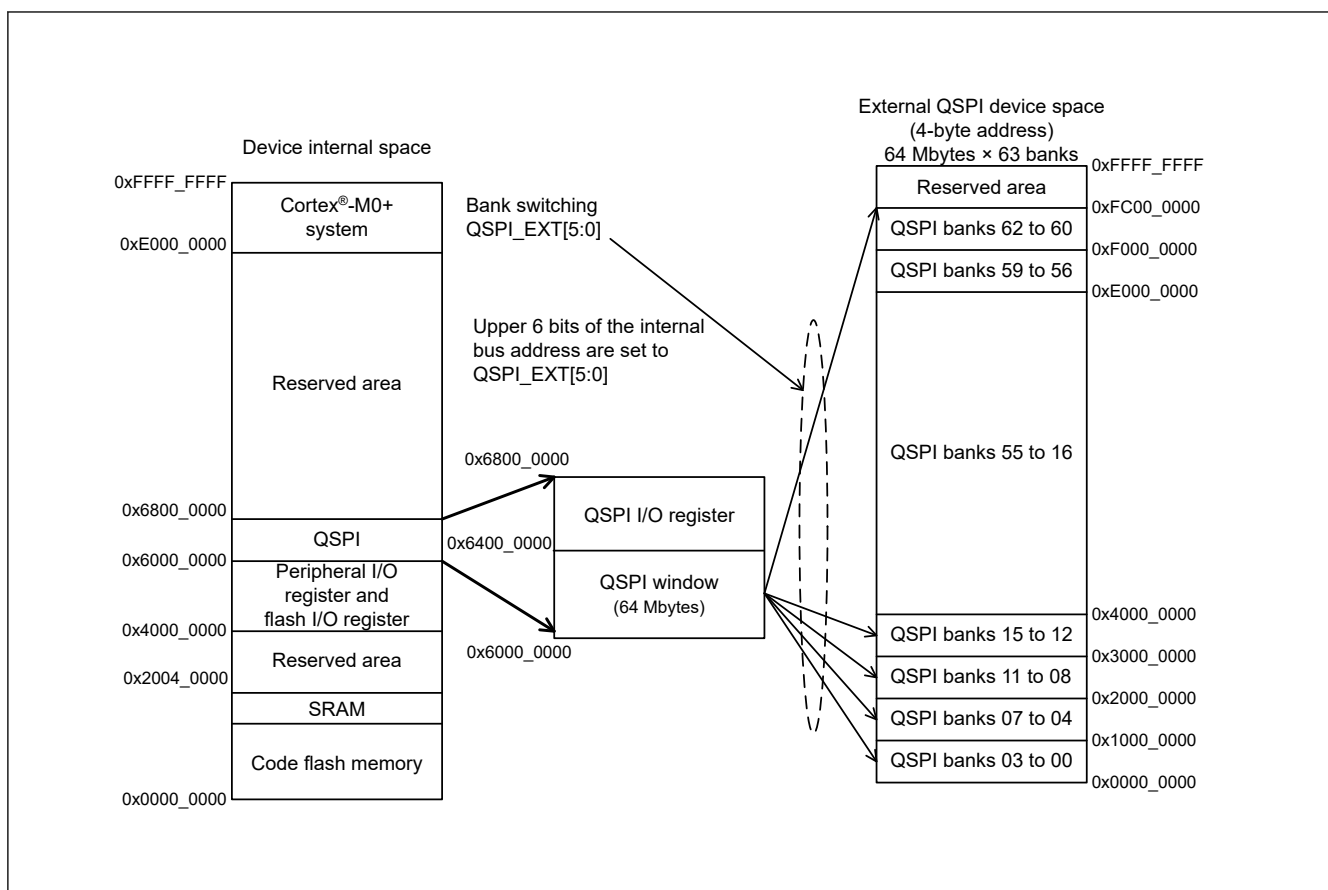


Figure 36.2 Default area setting and memory map

36.3.2 Address Width of the SPI Space and SPI Bus

The SPI space has a 32-bit address width for referencing the serial flash memory. When the SPI space is accessed for a read, an SPI bus cycle starts automatically, and data read from the serial flash is returned.

The address width of the SPI space is fixed at 32 bits. However, the address width of the SPI bus is selectable to 8, 16, 24, or 32 bits in the SFMAS[1:0] bits in the Address Mode Control Register (SFMSAC) register. If 8, 16, or 24 bits is selected as the address width of the SPI bus, only the lower part of the address used to access the SPI space is posted to the serial flash memory through the SPI bus. As a result, the mirror image of the serial flash corresponding to the address width of the SPI bus repeatedly appears in the SPI space.

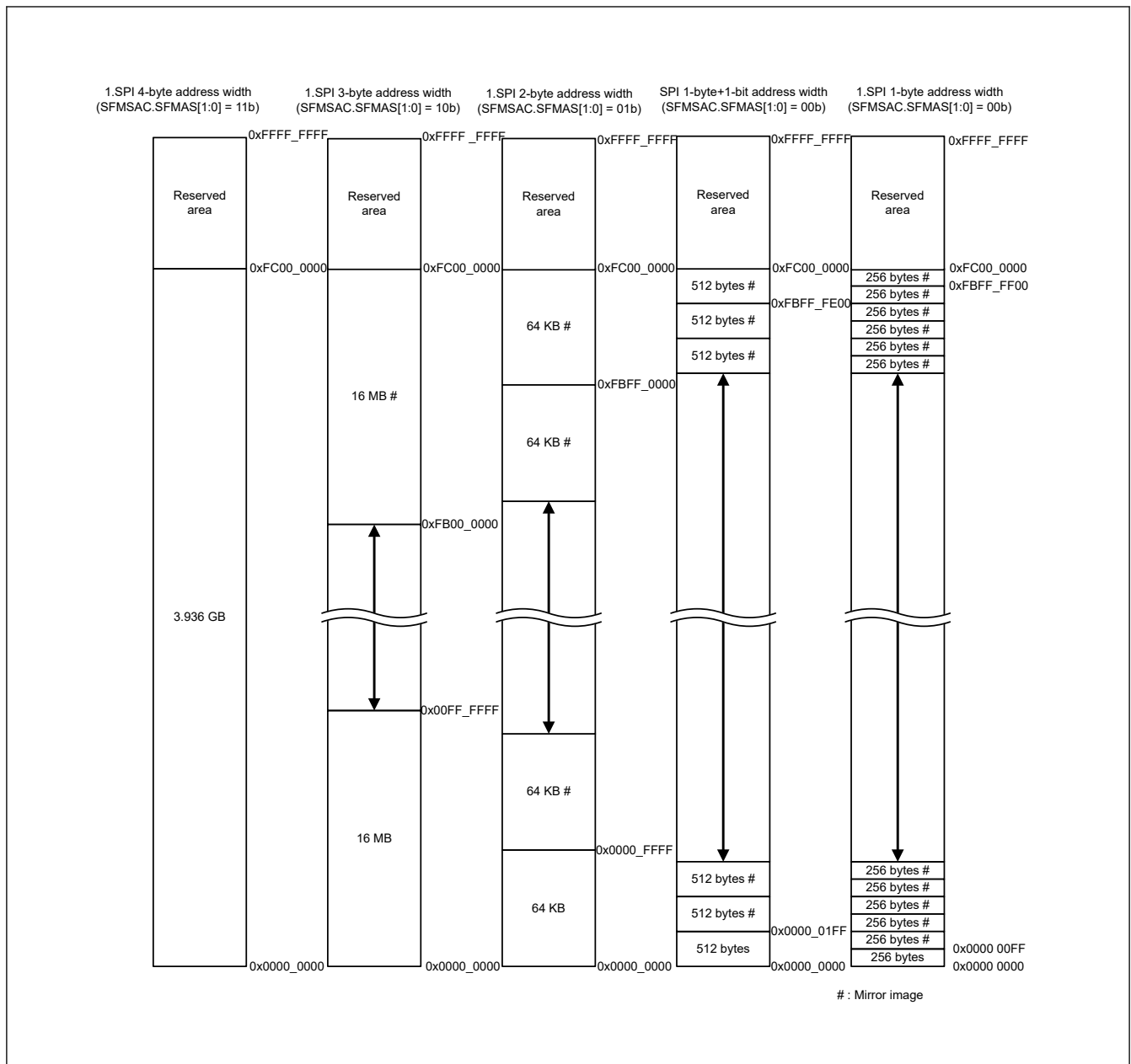


Figure 36.3 Memory map of SPI space

Note: The SPI bus address width is selectable to 8, 16, 24, or 32 bits in the SFMAS[1:0] bits in the SFMSAC register (cases 1 to 3 and 5 in the figure correspond to the respective address widths). When an 8-bit address width is selected, the address information of the ninth bit can be embedded in the Read instruction code. The memory map in case 4 in the figure is for the 9-bit address width. For details on the Read instruction, see [section 36.6.2. Standard Read Instruction](#).

36.4 SPI Bus

36.4.1 SPI Protocol

Single SPI, extended SPI, dual SPI, and quad SPI are supported in addition to the SPI protocol used for serial flash memory connection.

The initial state of the SPI protocol is Single SPI, extended SPI and can be changed with the SFMSPI[1:0] bits in the SPI Protocol Control Register (SFMSPC) register.

The address and data pins used in the Single SPI, extended SPI protocol change depending on the setting of the serial interface read mode select bits SFMRM[2:0] in Transfer Mode Control Register (SFMSMD). Table 36.3 and Table 36.4 list the pins used for instruction code, addresses, and data in each of the SPI protocols.

Table 36.3 List of SPI Protocols (1)

SPI Protocol (SFMSPC.SFMSP[1:0])	Single SPI Protocol, Extended SPI Protocol					
Serial interface read mode select (SFMSMD.SFMRM[2:0])	Standard read	Fast read	Fast read dual output	Fast read dual I/O	Fast read quad output	Fast read quad I/O
All pins used	QSPCLK, QSSL, QIO0, QIO1	QSPCLK, QSSL, QIO0, QIO1	QSPCLK, QSSL, QIO0, QIO1	QSPCLK, QSSL, QIO0, QIO1	QSPCLK, QSSL, QIO0, QIO1, QIO2, QIO3	QSPCLK, QSSL, QIO0, QIO1, QIO2, QIO3
Pins used for instruction code	QIO0	QIO0	QIO0	QIO0	QIO0	QIO0
Pins used for addresses	QIO0	QIO0	QIO0	QIO0, QIO1	QIO0	QIO0, QIO1, QIO2, QIO3
Pins used for data	QIO0/QIO1	QIO0/QIO1	QIO0, QIO1	QIO0, QIO1	QIO0, QIO1, QIO2, QIO3	QIO0, QIO1, QIO2, QIO3

Note: Single SPI protocol operation is for standard read and fast read. Extended SPI protocol operation is fast read dual output, fast read dual I/O, fast read quad output, and fast read quad I/O.

Table 36.4 List of SPI Protocols (2)

SPI Protocol (SFMSPC.SFMSP[1:0])	Dual-SPI Protocol		Quad-SPI Protocol	
Serial interface read mode select (SFMSMD.SFMRM[2:0])	Fast read dual output	Fast read dual I/O	Fast read quad output	Fast read quad I/O
All pins used	QSPCLK, QSSL, QIO0, QIO1	QSPCLK, QSSL, QIO0, QIO1	QSPCLK, QSSL, QIO0, QIO1, QIO2, QIO3	QSPCLK, QSSL, QIO0, QIO1, QIO2, QIO3
Pins used for instruction code	QIO0, QIO1	QIO0, QIO1	QIO0, QIO1, QIO2, QIO3	QIO0, QIO1, QIO2, QIO3
Pins used for addresses	QIO0, QIO1	QIO0, QIO1	QIO0, QIO1, QIO2, QIO3	QIO0, QIO1, QIO2, QIO3
Pins used for data	QIO0, QIO1	QIO0, QIO1	QIO0, QIO1, QIO2, QIO3	QIO0, QIO1, QIO2, QIO3

In single SPI protocol and extended SPI protocol, the instruction code is always output from the QIO0 pin. Address and data input/output operations are performed according to the settings in SFMSMD.SFMRM[2:0].

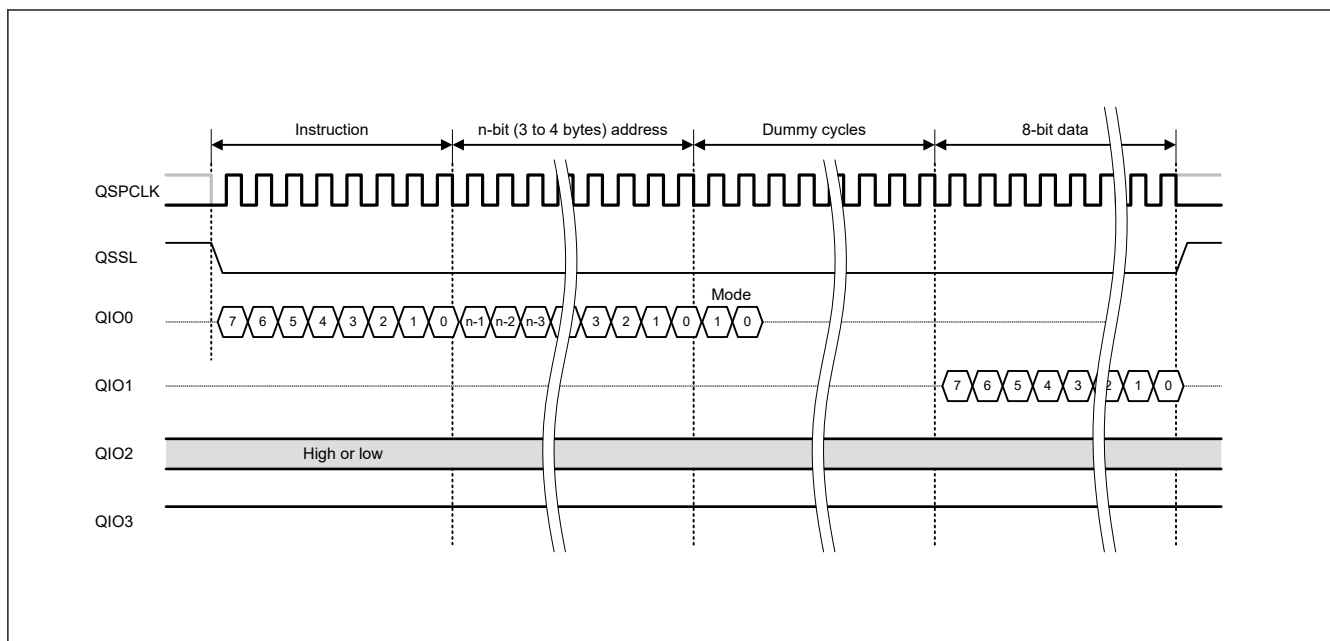


Figure 36.4 Single SPI Protocol, Extended SPI Protocol example 1 for Fast Read

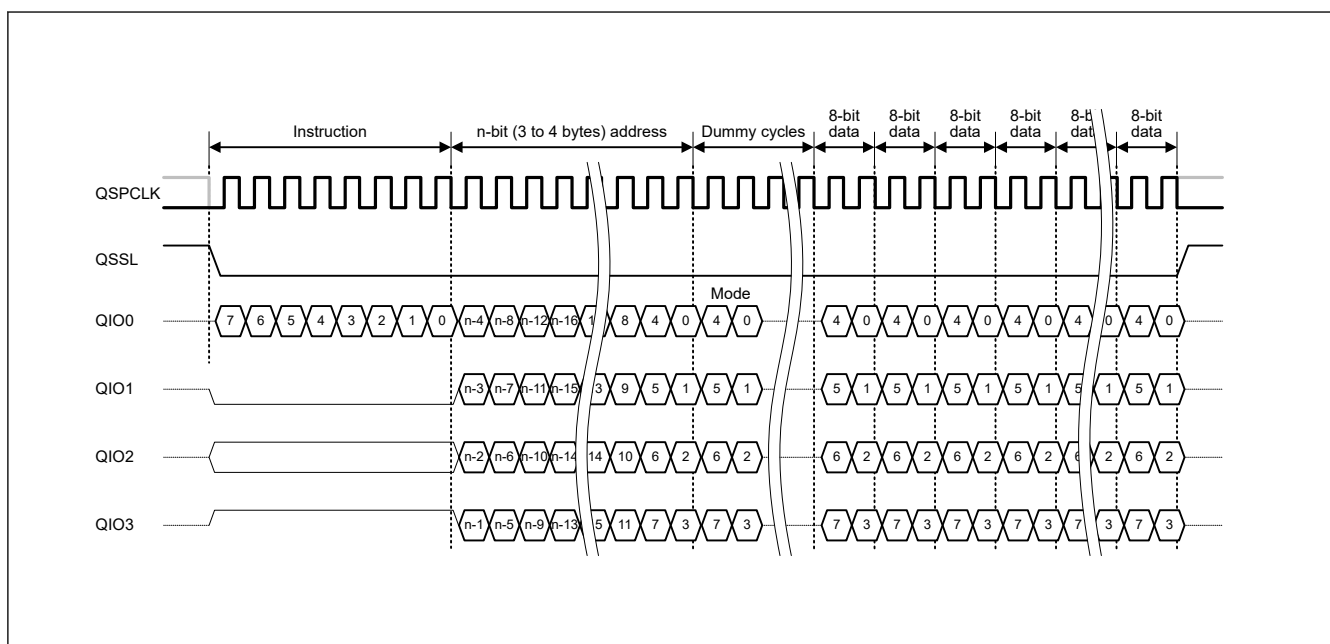


Figure 36.5 Single SPI Protocol, Extended SPI Protocol example 2 for Fast Read Quad I/O

The Dual SPI protocol performs I/O operation of all signals such as instruction codes, addresses, and data using two pins, QIO0 and QIO1.

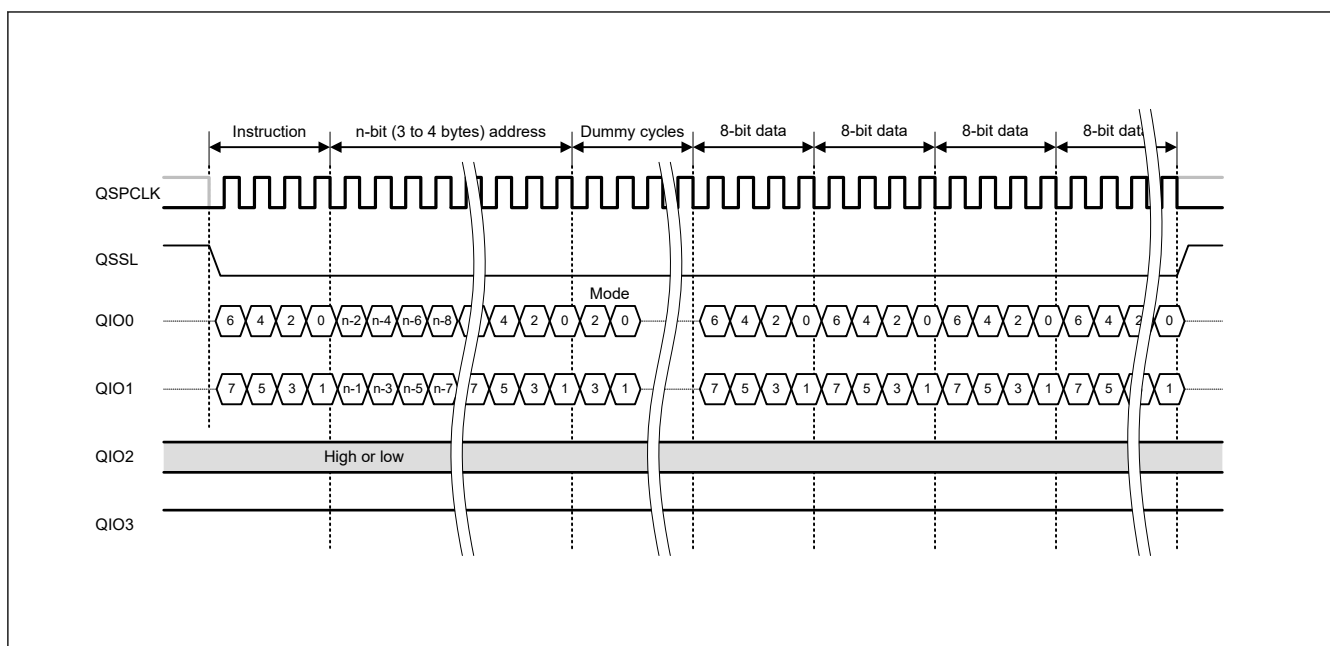


Figure 36.6 Dual SPI protocol example for Fast Read Quad I/O

The Quad SPI protocol performs I/O operation of all signals such as instruction codes, addresses, and data using four pins, QIO0, QIO1, QIO2, and QIO3.

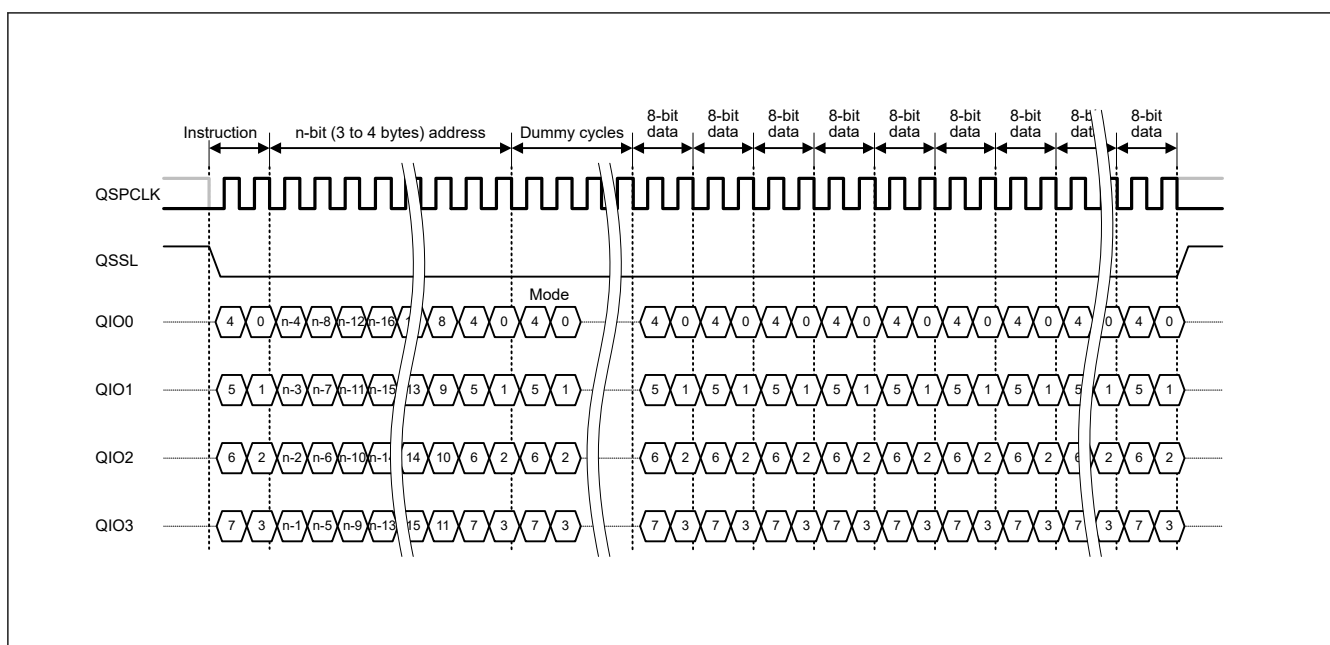


Figure 36.7 Quad SPI protocol example for Fast Read Quad I/O

36.4.2 SPI Mode

Either SPI mode 0 or SPI mode 3 can be selected as the SPI mode. This can be switched by changing the register setting during operation. The difference between SPI modes 0 and 3 is the state of the QSPCLK signal when it is inactive. The standby level of the QSPCLK signal in SPI mode 0 is low, and high in SPI mode 3.

Serial data is output from the QSPI on a falling edge of the serial clock and is read into the serial flash memory on a rising edge of the serial clock. Serial data is output from the serial flash memory on a falling edge of the serial clock and is read into the QSPI on the next falling edge of the serial clock.

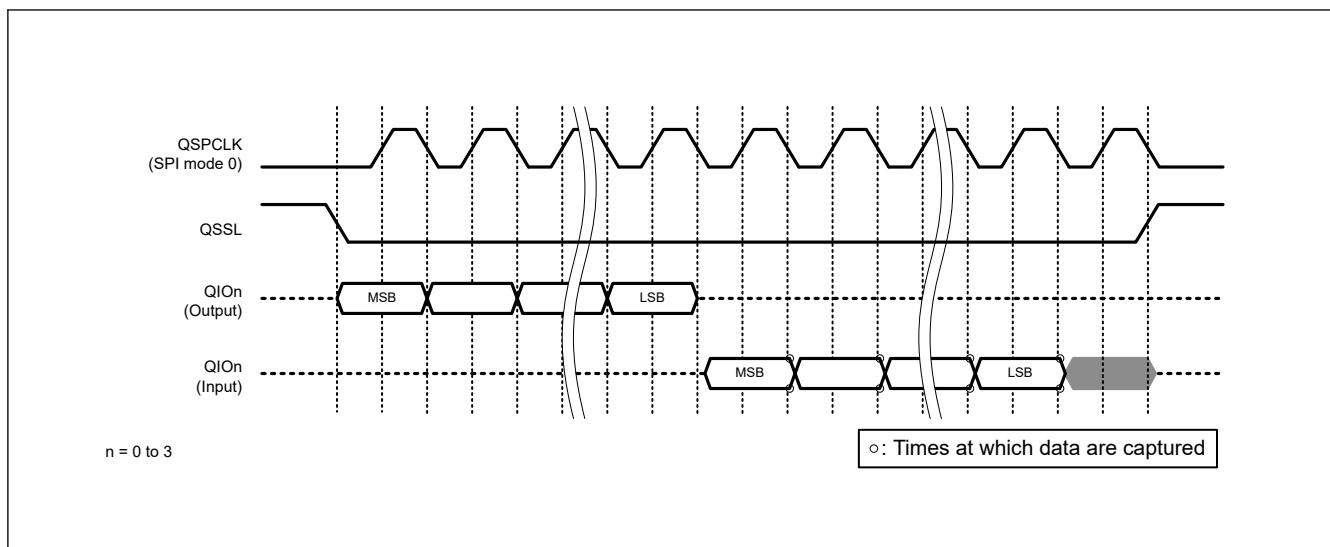


Figure 36.8 Basic serial interface timing (SPI mode 0)

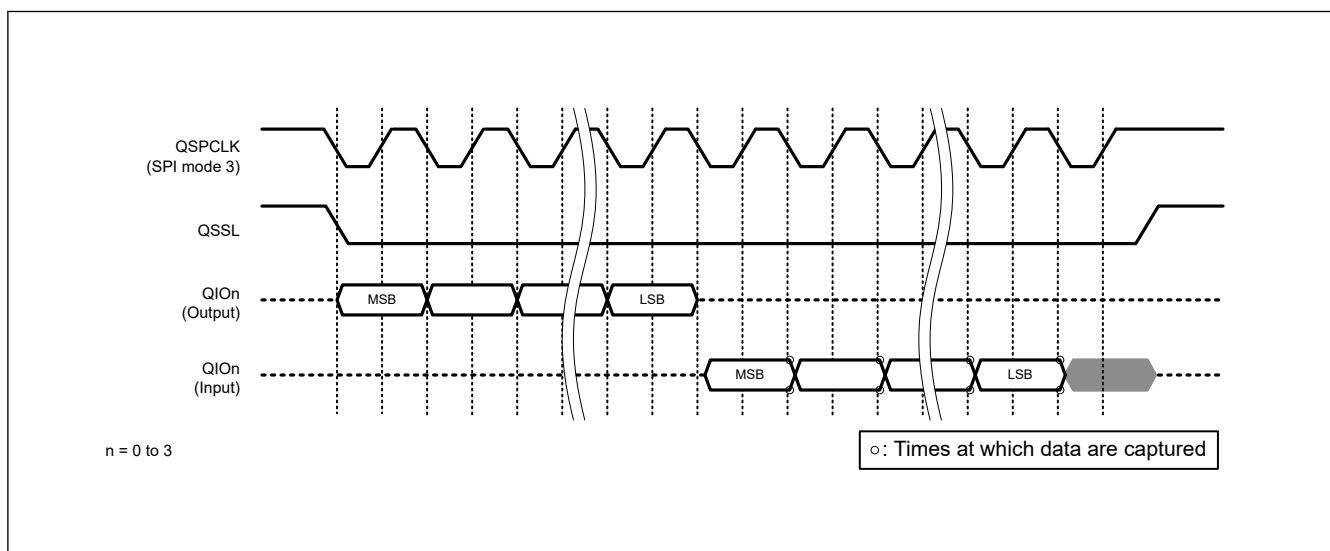


Figure 36.9 Basic serial interface timing (SPI mode 3)

36.5 SPI Bus Timing Adjustment

The timing of the SPI bus signal can be adjusted in the registers. The configured timing is applied to all SPI bus accesses, for both ROM access and direct communication.

36.5.1 SPI Bus Reference Cycles

The SPI bus operates on reference cycles obtained by multiplying PCLKA by an integer. The reference cycles are selectable within the range of PCLKA multiplied by 2 to 48 in the SFMDV[4:0] bits in the Transfer Mode Control Register (SFMSKC) register.

Table 36.5 Relationship among SFMDV[4:0] bits, cycle multiplier, and serial clock frequencies (1 of 2)

SFMDV[4:0]	Cycle multiplier	PCLKA = 64 [MHz]	PCLKA = 32 [MHz]
11111b	48	1.33	0.67
11110b	46	1.39	0.70
11101b	44	1.45	0.73
11100b	42	1.52	0.76
11011b	40	1.60	0.80

Table 36.5 Relationship among SFMDV[4:0] bits, cycle multiplier, and serial clock frequencies (2 of 2)

SFMDV[4:0]	Cycle multiplier	PCLKA = 64 [MHz]	PCLKA = 32 [MHz]
11010b	38	1.68	0.84
11001b	36	1.78	0.89
11000b	34	1.88	0.94
10111b	32	2.00	1.00
10110b	30	2.13	1.07
10101b	28	2.29	1.14
10100b	26	2.46	1.23
10011b	24	2.67	1.33
10010b	22	2.91	1.45
10001b	20	3.20	1.60
10000b	18	3.56	1.78
01111b	17	3.76	1.88
01110b	16	4.00	2.00
01101b	15	4.27	2.13
01100b	14	4.57	2.29
01011b	13	4.92	2.46
01010b	12	5.33	2.67
01001b	11	5.82	2.91
01000b	10	6.40	3.20
00111b	9	7.11	3.56
00110b	8	8.00	4.00
00101b	7	9.14	4.57
00100b	6	10.67	5.33
00011b	5	12.80	6.40
00010b	4	16.00	8.00
00001b	3	21.33	10.67
00000b	2	—	16.00

36.5.2 QSPCLK Signal Duty Ratio

When the reference clock is configured as PCLKA divided by an odd number without duty ratio correction, the duty ratio of the QSPCLK signal will not be 50%. When the reference clock is PCLKA divided by an odd number, be sure to enable the duty ratio correction function (SFMSKC.SFMDTY = 1).

When the reference clock is PCLKA divided by an even number, the SFMDTY setting in the SFMSKC register is ignored.

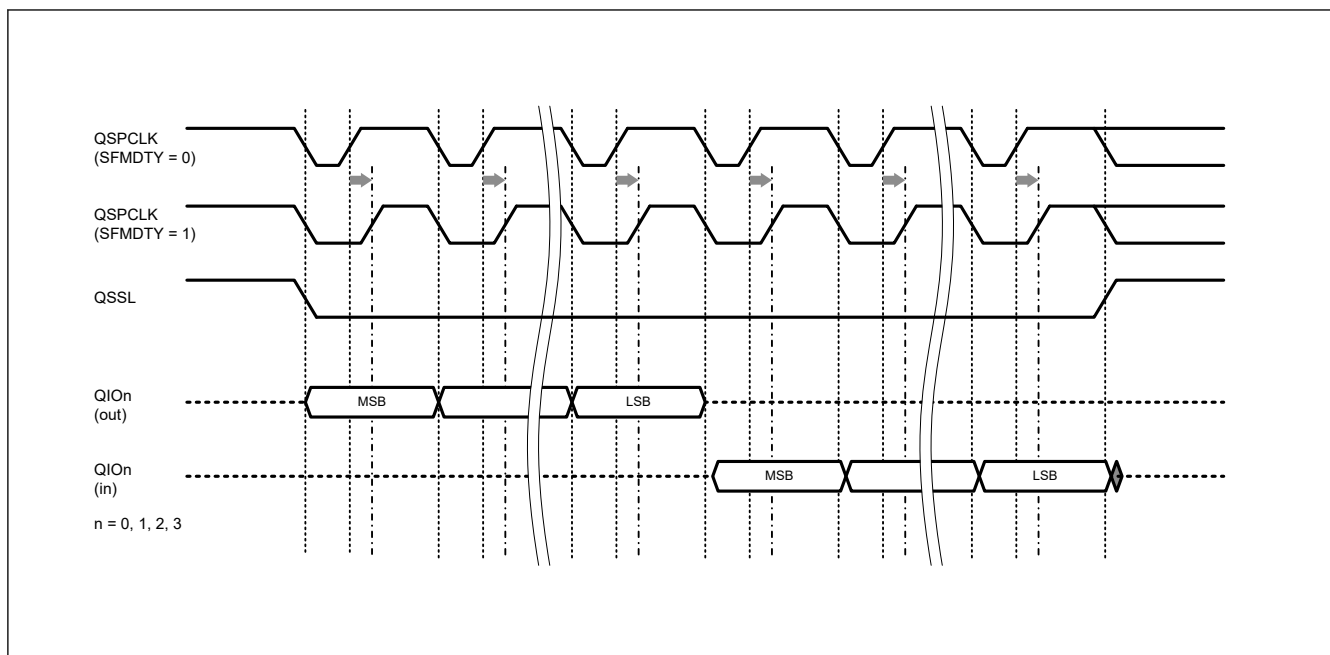


Figure 36.10 Example correction of the QSPCLK signal duty ratio using the SFMDTY bit, when PCLKA is multiplied by 3

36.5.3 Minimum High-Level Width for the QSSL Signal

Between adjacent SPI bus cycles, the QSSL signal must be held high (inactive) for a sufficient time to satisfy the deselect time required by the serial flash memory. The minimum high-level width of the QSSL output signal is selectable as the reference cycle multiplied by an integer from 1 to 16 in the SFMSW[3:0] bits in the Instruction Code Register (SFMSSC) register.

36.5.4 QSSL Signal Setup Time

The QSSL signal setup time that the serial flash memory requires after the QSSL signal is driven active low until the first rising edge of the QSPCLK signal can be configured. The setup time can be selected as 0.5 or 1.5 cycles of QSPCLK in the SFMSLD bit of the SFMSSC register.

Set a value that meets the most constrained timing condition for your application.

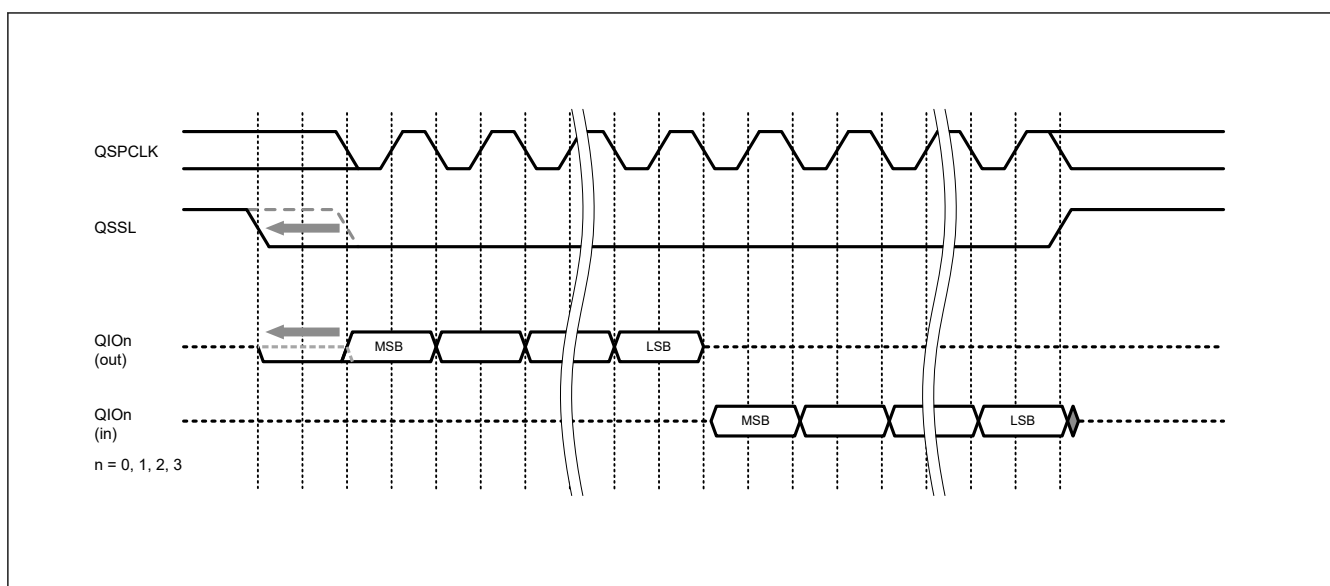


Figure 36.11 Setup time adjustment for the QSSL signal using the SFMSLD bit

36.5.5 QSSL Signal Hold Time

The QSSL signal hold time that the serial flash memory requires until the QSSL signal is driven high after the last rising edge of the QSPCLK signal can be configured. The hold time can be selected as 0.5 or 1.5 cycles of QSPCLK in the SFMSHD bit of the SFMSSC register.

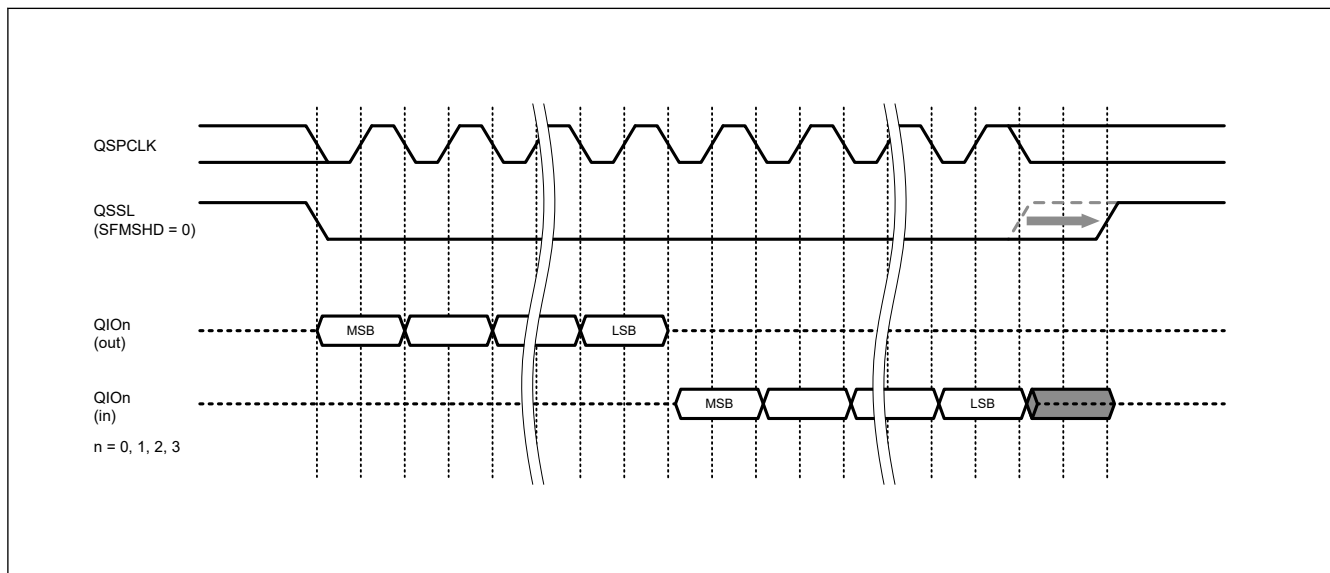


Figure 36.12 Hold time adjustment for the QSSL signal using the SFMSSD bit

36.5.6 Hold Time of the Serial Data Output Enable

The buffer output enable of the QIO0, QIO1, QIO2, or QIO3 pin can be extended by 1 QSPCLK using the SFMOEX bit in the SFMSMD register.

For a standard read instruction, it is extended immediately after an address code. For other read instructions, it is extended after two cycles of mode data (XIP mode control) of the serial flash memory in dummy cycles.

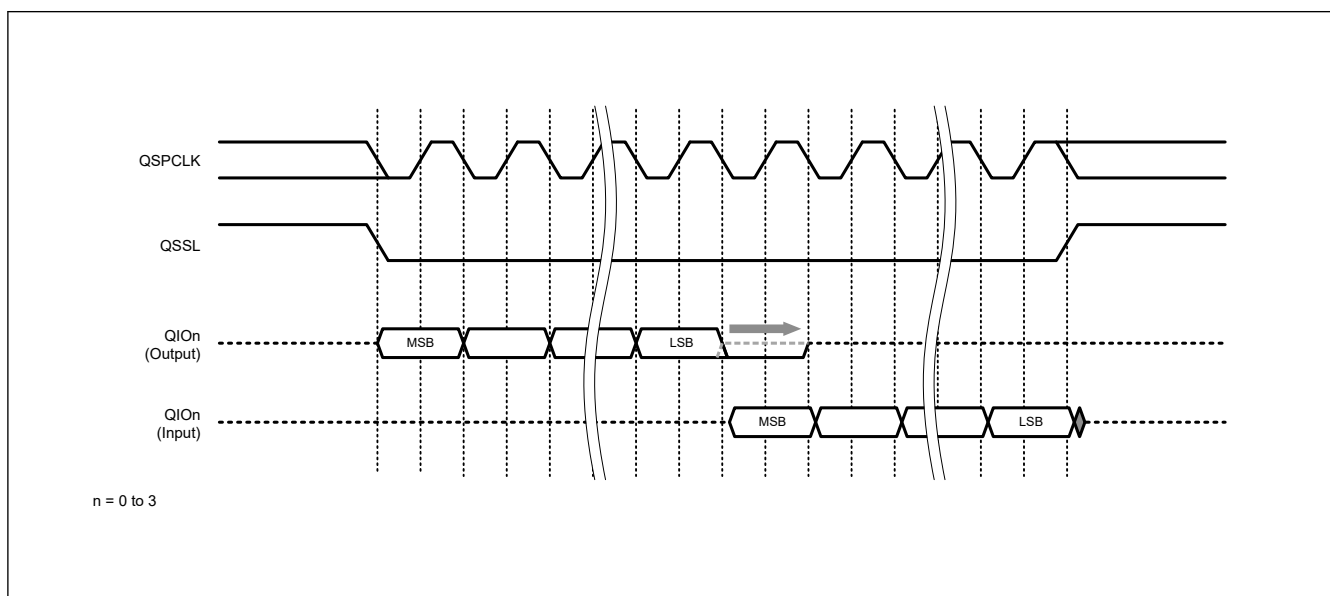


Figure 36.13 Hold time adjustment for output enable using the SFMOEX bit (Standard Read)

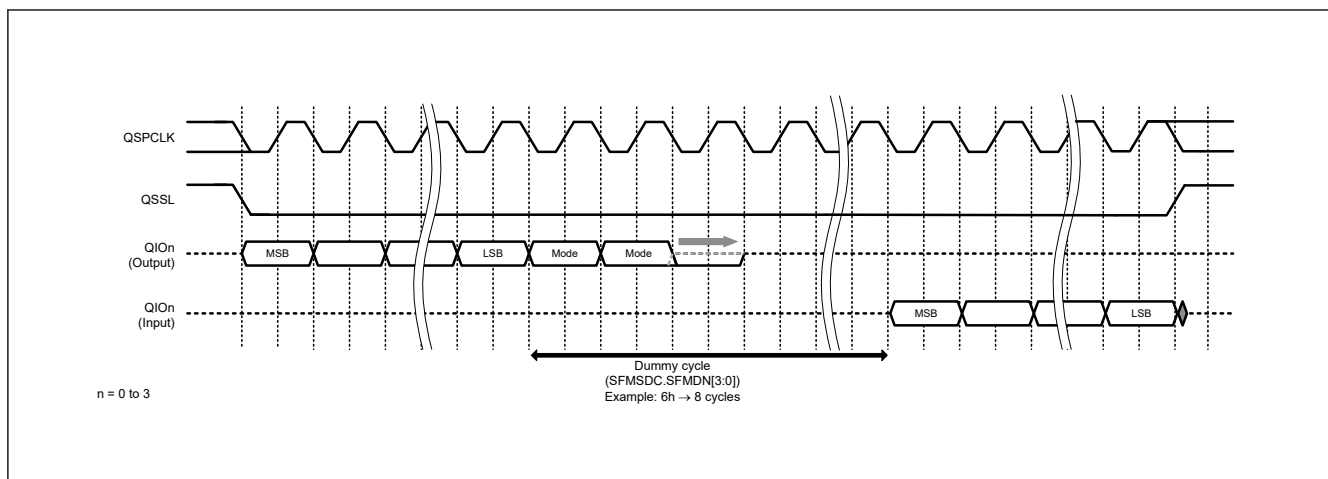


Figure 36.14 Hold Time Adjustment of Output Enabling Using the SFMOEX Bit (Fast Read)

36.5.7 Setup Time for Serial Data Output

When a command or address is transmitted to the serial flash memory, the setup time begins on serial data output and ends when the QSPCLK signal rises.

36.5.8 Hold Time for Serial Data Output

When a command or address is transmitted to the serial flash memory, the hold time begins on the rising edge of QSPCLK and ends when the serial data makes another transmission.

36.6 SPI Instruction Set Used for Serial Flash Memory Access

36.6.1 SPI Instructions That Are Automatically Generated

When the serial flash memory is accessed, an SPI bus cycle using one of the instructions described in [Table 36.6](#) to [Table 36.10](#) is automatically generated based on the settings in the SFMSAC register and in the SFMSMD register.

Table 36.6 SPI instructions automatically generated when SFMAS[1:0] = 00b

SPI instruction	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Read	0x03 ^{*1}	1	—	1 to ∞	Required: SFMRM[2:0] = 000b, A8 = 0
	0x0B ^{*1}	1	—	1 to ∞	Required: SFMRM[2:0] = 000b, A8 = 1

Note 1. If the SFMSMD.SFMCCE bit is set to 1, the SFMCIC[7:0] bits in the Instruction Code Register (SFMSIC) setting is used as an instruction code.

Table 36.7 SPI instructions automatically generated when SFMAS[1:0] = 01b

SPI instruction	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Read	0x03 ^{*1}	2	—	1 to ∞	Required: SFMRM[2:0] = 000b

Note 1. If the SFMSMD.SFMCCE bit is set to 1, the SFMSIC.SFMCIC[7:0] setting is used as an instruction code.

Table 36.8 SPI instructions automatically generated when SFMAS[1:0] = 10b (1 of 2)

SPI instruction	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Read	0x03 ^{*1}	3	—	1 to ∞	Required: SFMRM[2:0] = 000b
Fast Read	0x0B ^{*1}	3	8 ^{*2}	1 to ∞	Selectable: SFMRM[2:0] = 001b
Fast Read Dual Output	0x3B ^{*1}	3	8 ^{*2}	1 to ∞	Selectable: SFMRM[2:0] = 010b

Table 36.8 SPI instructions automatically generated when SFMAS[1:0] = 10b (2 of 2)

SPI instruction	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Fast Read Dual I/O	0xBB ^{*1}	3	4 ^{*2}	1 to ∞	Selectable: SFMRM[2:0] = 011b
Fast Read Quad Output	0x6B ^{*1}	3	8 ^{*2}	1 to ∞	Selectable: SFMRM[2:0] = 100b
Fast Read Quad I/O	0xEB ^{*1}	3	6 ^{*2}	1 to ∞	Selectable: SFMRM[2:0] = 101b

Note 1. If the SFMSMD.SFMCCE bit is set to 1, the SFMSIC.SFMCIC[7:0] setting is used as an instruction code.

Note 2. The number of dummy cycles is configurable by using SFMDN[3:0] bits in the Dummy Cycle Control Register (SFMSDC).

Table 36.9 SPI instructions automatically generated when SFMAS[1:0] = 11b and SFM4BC = 0

SPI instruction	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Read	0x03 ^{*1}	4	—	1 to ∞	Required: SFMRM[2:0] = 000b
Fast Read	0x0B ^{*1}	4	8 ^{*2}	1 to ∞	Selectable: SFMRM[2:0] = 001b
Fast Read Dual Output	0x3B ^{*1}	4	8 ^{*2}	1 to ∞	Selectable: SFMRM[2:0] = 010b
Fast Read Dual I/O	0xBB ^{*1}	4	4 ^{*2}	1 to ∞	Selectable: SFMRM[2:0] = 011b
Fast Read Quad Output	0x6B ^{*1}	4	8 ^{*2}	1 to ∞	Selectable: SFMRM[2:0] = 100b
Fast Read Quad I/O	0xEB ^{*1}	4	6 ^{*2}	1 to ∞	Selectable: SFMRM[2:0] = 101b

Note 1. If the SFMSMD.SFMCCE bit is set to 1, the SFMSIC.SFMCIC[7:0] setting is used as an instruction code.

Note 2. The number of dummy cycles is configurable by using the SFMSDC.SFMDN[3:0] bits.

Table 36.10 SPI instructions automatically generated when SFMAS[1:0] = 11b and SFM4BC = 1

SPI instruction	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Read	0x13 ^{*1}	4	—	1 to ∞	Required: SFMRM[2:0] = 000b
Fast Read	0x0C ^{*1}	4	8 ^{*2}	1 to ∞	Selectable: SFMRM[2:0] = 001b
Fast Read Dual Output	0x3C ^{*1}	4	8 ^{*2}	1 to ∞	Selectable: SFMRM[2:0] = 010b
Fast Read Dual I/O	0xBC ^{*1}	4	4 ^{*2}	1 to ∞	Selectable: SFMRM[2:0] = 011b
Fast Read Quad Output	0x6C ^{*1}	4	8 ^{*2}	1 to ∞	Selectable: SFMRM[2:0] = 100b
Fast Read Quad I/O	0xEC ^{*1}	4	6 ^{*2}	1 to ∞	Selectable: SFMRM[2:0] = 101b

Note 1. If the SFMSMD.SFMCCE bit is set to 1, the SFMSIC.SFMCIC[7:0] setting is used as an instruction code.

Note 2. The number of dummy cycles is configurable by using the SFMSDC.SFMDN[3:0] bits.

36.6.2 Standard Read Instruction

The standard Read instruction is a common read instruction supported by most serial flash memory. When an SPI bus cycle starts, the QSSL signal (serial flash memory select) is asserted, and the instruction code (0x03 or 0x13)^{*1} is output. Next, an address with a width of 1 to 4 bytes, specified in the SFMAS[1:0] bits in the SFMSAC register, is transmitted. Data is then received.

This standard Read instruction is selected in the initial QSPI settings.

Note 1. Many 4-KB serial flash memory devices have an address field not larger than 1 byte (A7-A0) to minimize the overhead and to receive A8 information from bit 3 of the Read instruction code. To support these devices, the QSPI only outputs A8 (address bit 8) to bit [3] of the standard Read instruction code when an address width of 1 byte is specified (SFMAS[1:0] = 00). This means that 0x0B might be output instead of 0x03 as the standard Read instruction code. This code duplicates the Fast Read instruction code. However, for most of the 2-KB or smaller serial flash memory, with an address width of 1 byte, bit 3 of a command is designed to be excluded from decoding as a don't-care bit, so such a Read instruction code is recognized correctly as the standard Read instruction code. In rare cases, some serial flash memory allow bit 3 to be decoded. When such a serial flash memory is connected, configure your application to avoid access resulting in A8 = 1.

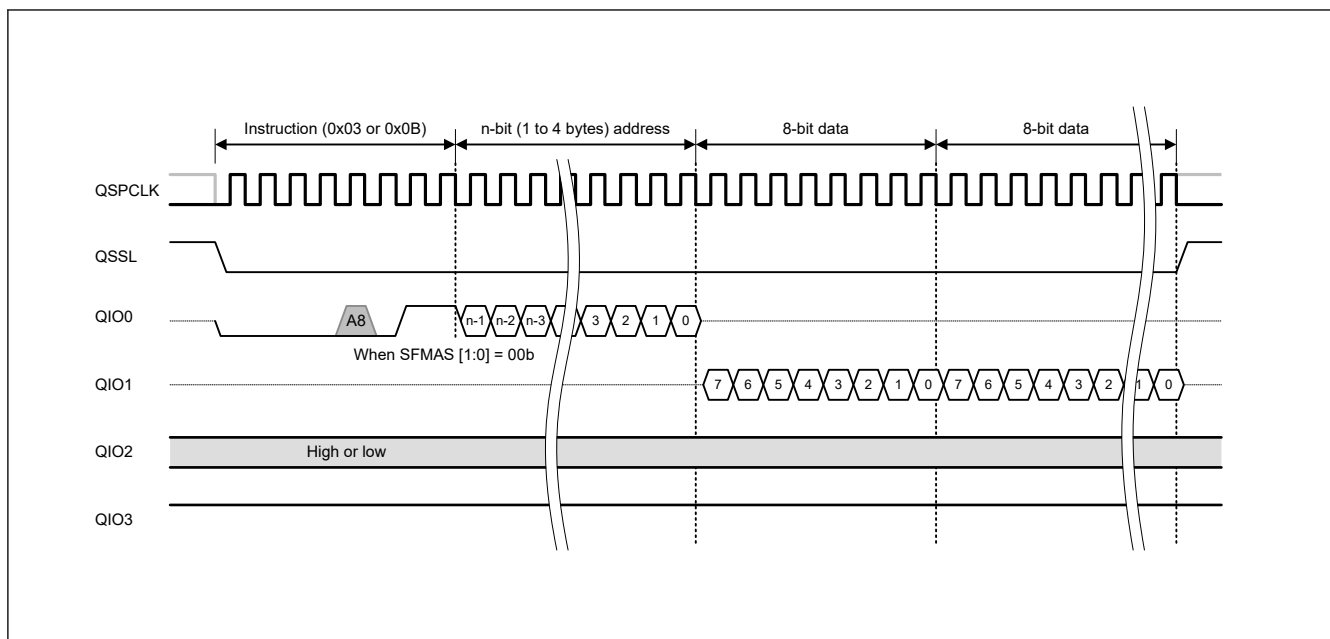


Figure 36.15 Standard Read bus cycle

36.6.3 Fast Read Instruction

The Fast Read instruction is a read instruction that supports a higher communication clock speed than the standard Read instruction. When an SPI bus cycle starts, the QSSL signal is asserted, and the instruction code (0x0B or 0x0C) is output. Next, an address with a width of 1 to 4 bytes, specified in the SFMAS[1:0] bits in SFMSAC, and a certain number of dummy cycles, specified in the SFMSDC register, are transmitted. Data is then received.

The first two dummy cycles are used to select the XIP mode. When the XIP mode is selected, the same instruction used this time is applied to the next SPI bus cycle, and the instruction code is not output the next SPI bus cycle. For details on the XIP mode, see [section 36.8. XIP Control](#).

Switching to the Fast Read instruction is controlled in the SFMSMD register.

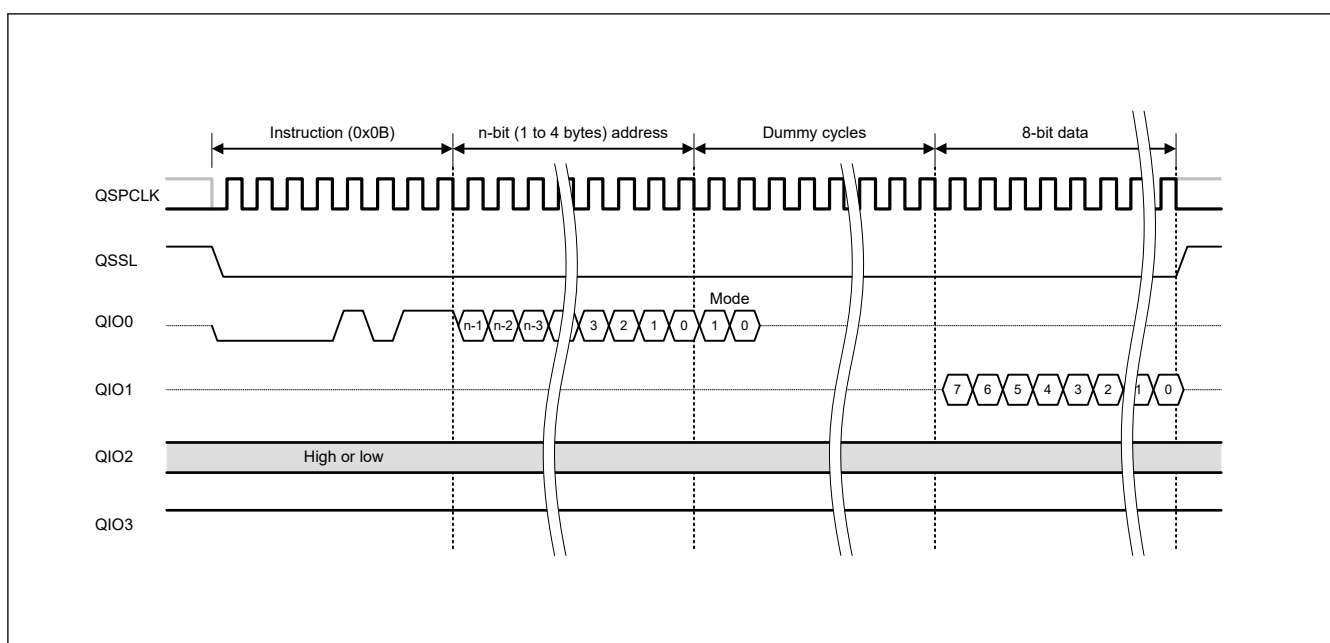


Figure 36.16 Fast Read bus cycle

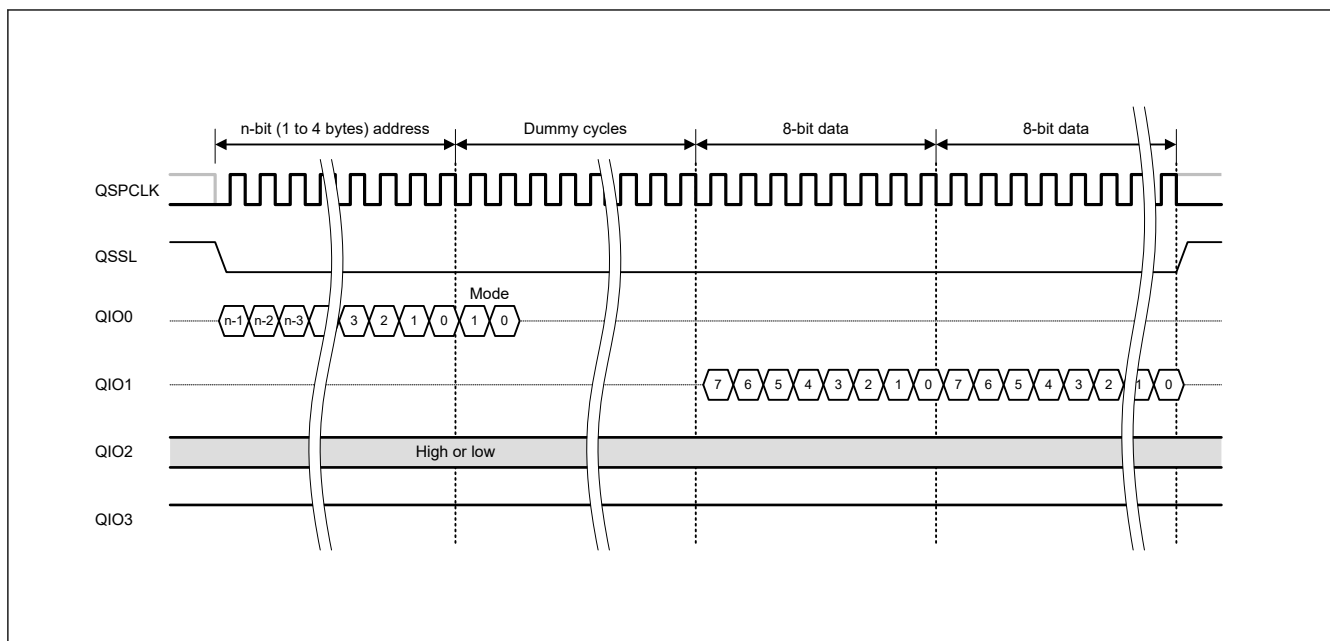


Figure 36.17 Fast Read bus cycle in XIP mode

Note: To use the Fast Read instruction, a serial flash memory that supports Fast Read transfers is required.

36.6.4 Fast Read Dual Output Instruction

The Fast Read Dual Output instruction is a read instruction that uses two signal lines to receive data. When the SPI bus cycle starts, the QSSL signal is asserted. The instruction code (0x3B or 0x3C) and an address with a width of 1 to 4 bytes, specified in the SFMAS[1:0] bits in the SFMSAC register, are transmitted from the QIO0 pin. Next, a certain number of dummy cycles, specified in the SFMSDC register, is generated. Data is then received through the QIO0 and QIO1 pins. Even bit data is received from the QIO0 pin and odd bit data is received from the QIO1 pin.

The first two dummy cycles are used to select the XIP mode. When the XIP mode is selected, the same instruction used this time is applied to the next SPI bus cycle, and the instruction code is not output the next SPI bus cycle. For details on the XIP mode, see [section 36.8. XIP Control](#).

Switching to Fast Read Dual Output is controlled in the SFMSMD register.

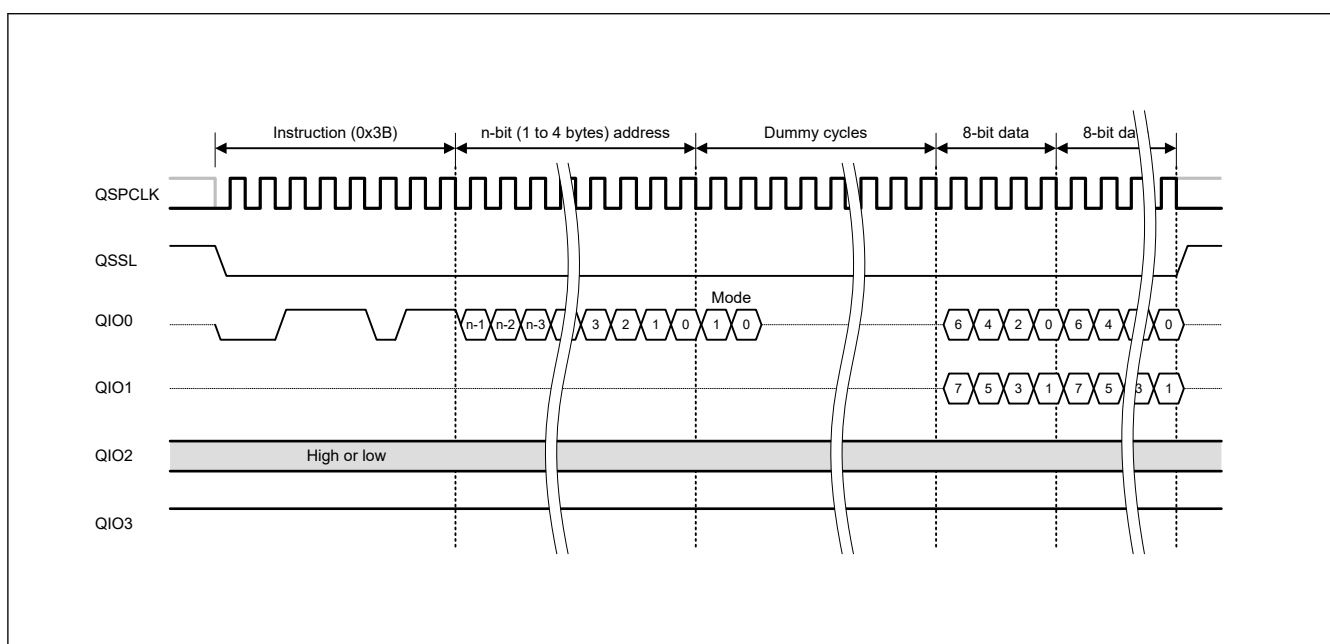


Figure 36.18 Fast Read Dual Output bus cycle

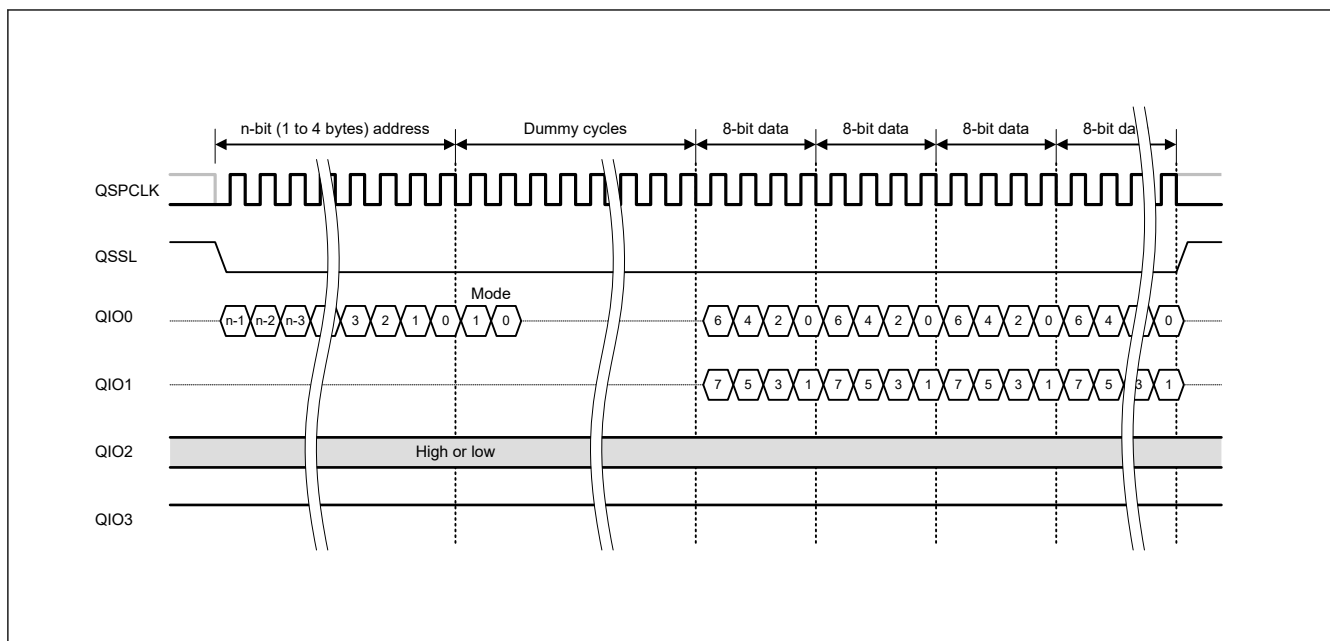


Figure 36.19 Fast Read Dual Output bus cycle in XIP mode

Note: To use the Fast Read Dual Output instruction, a serial flash memory that supports Fast Read Dual Output transfers is required.

36.6.5 Fast Read Dual I/O Instruction

The Fast Read Dual I/O instruction is a read instruction that uses two signal lines to transmit an address and receive data. When the SPI bus cycle starts, the QSSL signal is asserted, and the instruction code (0xBB or 0xBC) is output from the QIO0 pin. Next, an address with a width of 1 to 4 bytes, specified in the SFMAS[1:0] bits in the SFMSAC register, is transmitted through the QIO0 and QIO1 pins, and a certain number of dummy cycles, specified in the SFMSDC register, is generated. Data is then received through the QIO0 and QIO1 pins. Address and dummy cycle transmission and data reception are performed through the QIO0 pin for even bits and through the QIO1 pin for odd bits.

The first two dummy cycles are used to select the XIP mode. When the XIP mode is selected, the same instruction used this time is applied to the next SPI bus cycle, and the instruction code is not output the next SPI bus cycle. For details on the XIP mode, see [section 36.8. XIP Control](#).

Switching to Fast Read Dual I/O is controlled in the SFMSMD register.

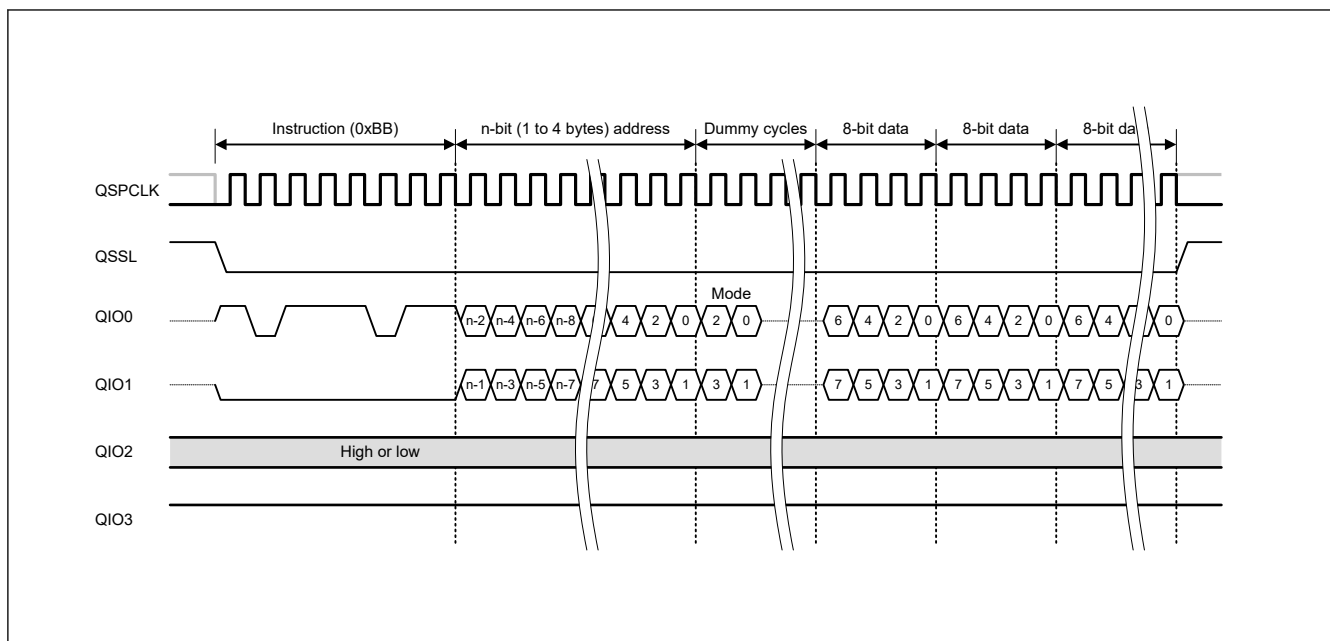


Figure 36.20 Fast Read Dual I/O bus cycle

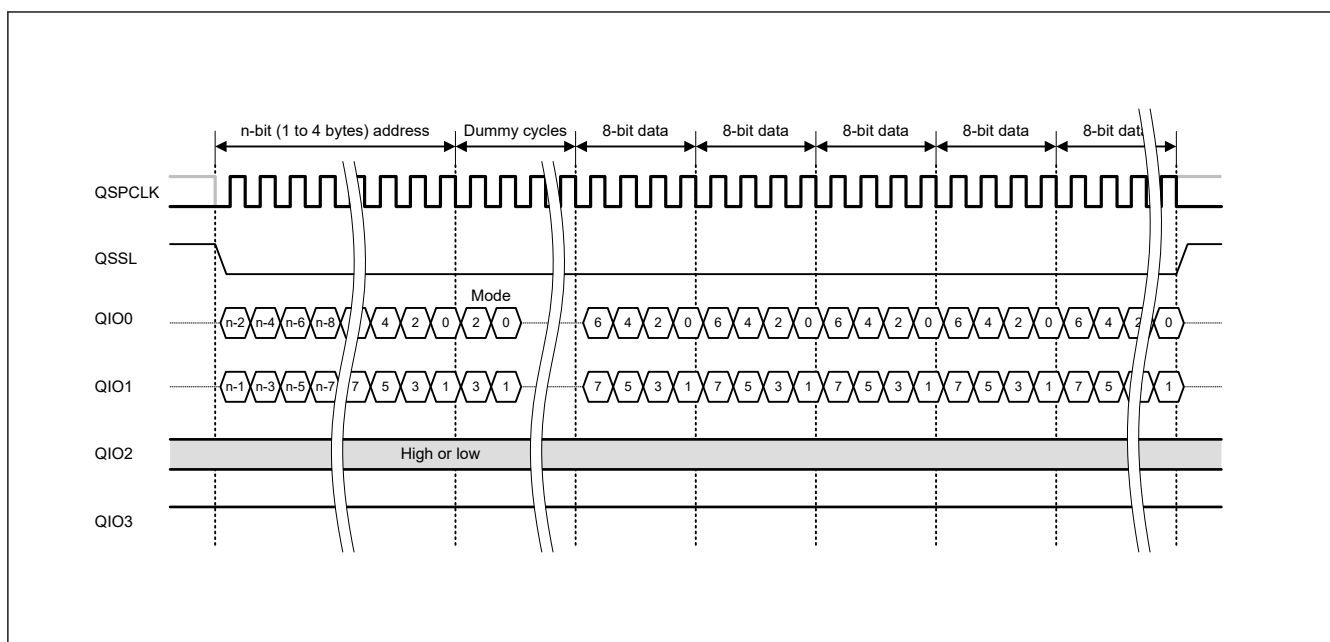


Figure 36.21 Fast Read Dual I/O bus cycle in XIP mode

Note: To use the Fast Read Dual I/O instruction, a serial flash memory that supports Fast Read Dual I/O transfers is required.

36.6.6 Fast Read Quad Output Instruction

The Fast Read Quad Output instruction is a read instruction that uses four signal lines to receive data. When the SPI bus cycle starts, the QSSL signal is asserted. The instruction code (0x6B or 0x6C) and an address with a width of 1 to 4 bytes, specified in the SFMAS[1:0] bits in the SFMSAC register, are output from the QIO0 pin. Next, a certain number of dummy cycles, specified in the SFMDN[3:0] bits in the SFMSMD register, are generated. Data is then received through the QIO0, QIO1, QIO2, and QIO3 pins.

The first two dummy cycles are used to select the XIP mode. When the XIP mode is selected, the same instruction used this time is applied to the next SPI bus cycle, and the instruction code is not output the next SPI bus cycle. For details on the XIP mode, see [section 36.8. XIP Control](#).

Switching to Fast Read Quad Output is controlled in the SFMSMD register.

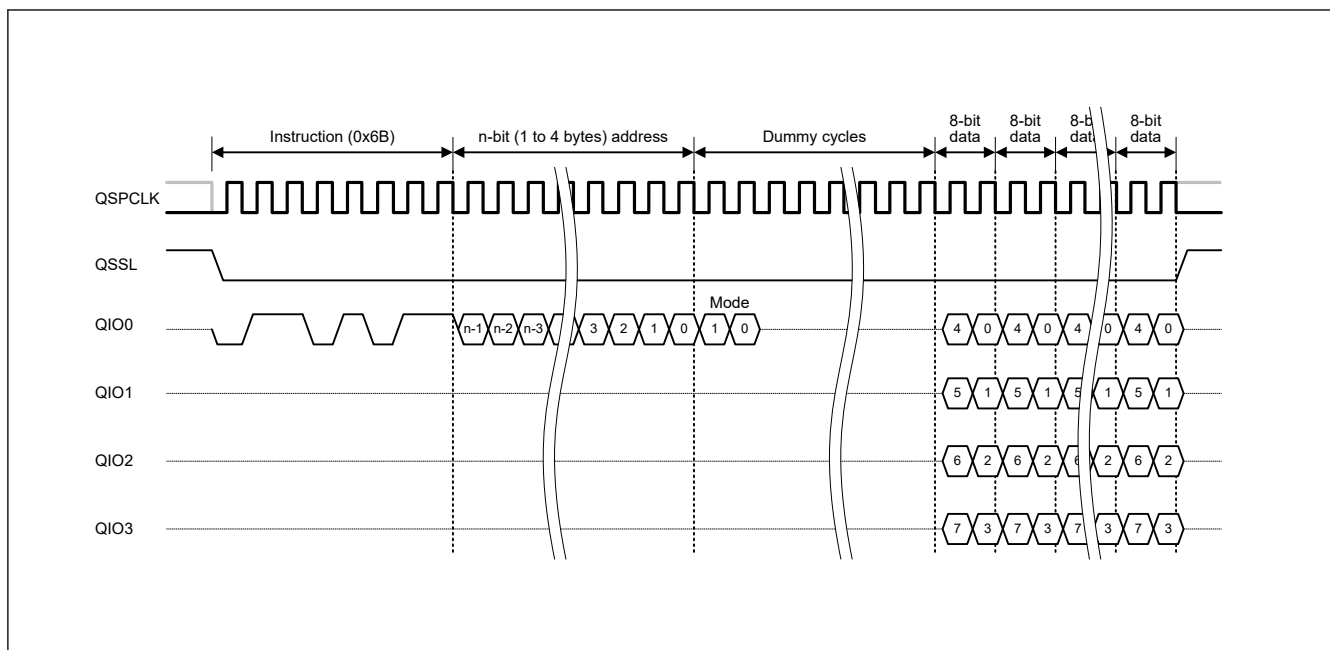


Figure 36.22 Fast Read Quad Output bus cycle

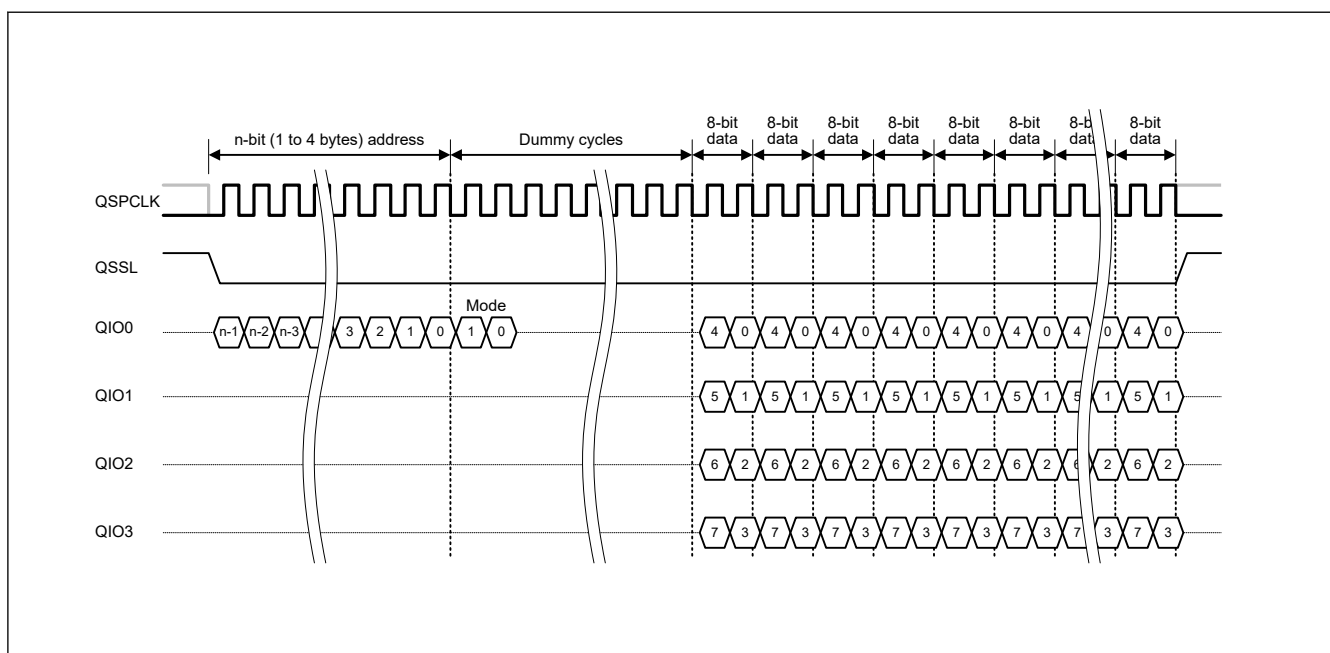


Figure 36.23 Fast Read Quad Output bus cycle in XIP mode

Note: To use Fast Read Quad Output, a serial flash memory that supports Fast Read Quad Output transfer is required.

36.6.7 Fast Read Quad I/O Instruction

The Fast Read Quad I/O instruction is a read instruction that uses four signal lines to transmit an address and receive data. When the SPI bus cycle starts, the QSSL signal is asserted, and the instruction code (0xEB or 0xEC) is output. Next, an address with a width of 1 to 4 bytes, specified in the SFMAS[1:0] bits in the SFMSAC register, is transmitted through the QIO0, QIO1, QIO2, and QIO3 pins, and a certain number of dummy cycles, specified in the SFMDN[3:0] bits in the SFMSMD register, is generated. Data is then received through the QIO0, QIO1, QIO2, and QIO3 pins.

The first two dummy cycles are used to select the XIP mode. When the XIP mode is selected, the same instruction used this time is applied to the next SPI bus cycle, and the instruction code is not output the next SPI bus cycle. For details on the XIP mode, see [section 36.8. XIP Control](#).

Switching to Fast Read Quad I/O is controlled in the SFMSMD register.

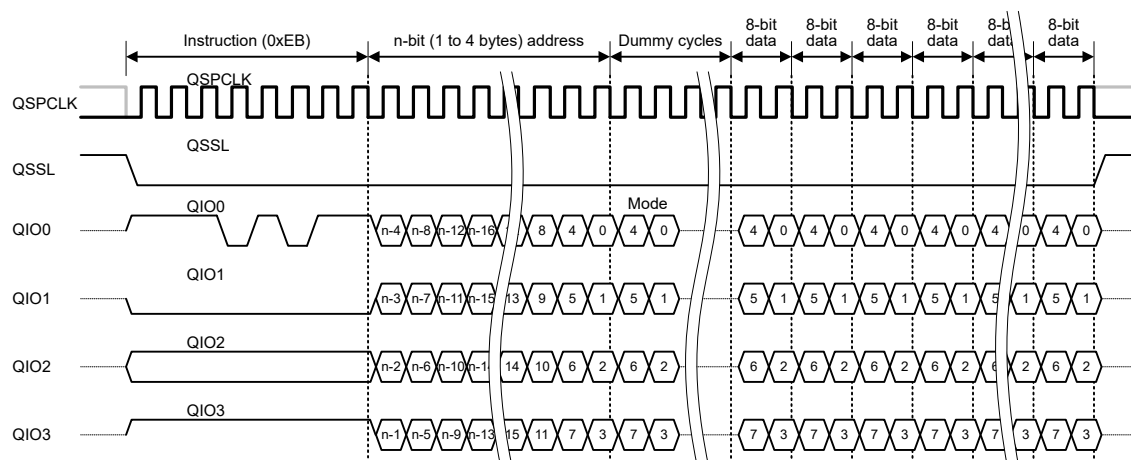


Figure 36.24 Fast Read Quad I/O bus cycle

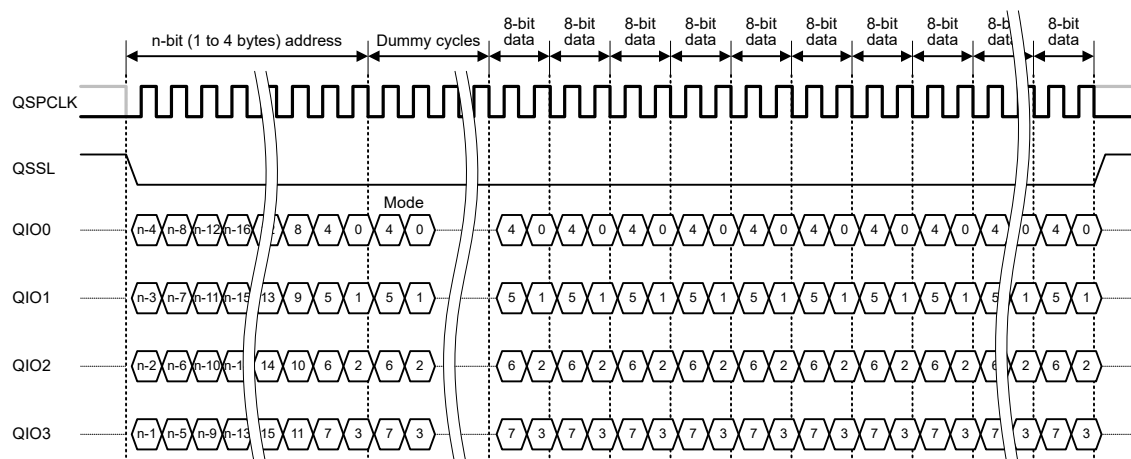


Figure 36.25 Fast Read Quad I/O bus cycle in XIP mode

Note: To use the Fast Read Quad I/O instruction, a serial flash memory that supports Fast Read Quad I/O transfers is required.

36.7 SPI Bus Cycle Arrangement

36.7.1 Serial Flash Memory Read Based on Individual Conversion

ROM read bus cycles are individually converted to SPI bus cycles on a one-to-one basis. When a ROM read bus cycle is detected, the QSSL signal is asserted, and an SPI bus cycle starts. When the data receiving is finished from the serial flash memory, the QSSL signal is negated and the SPI bus cycle is complete.

When the next ROM read bus cycle is detected, the QSSL signal is asserted again after ensuring the minimum high level width (inactive) of the QSSL signal (set in the SFMSSC.SFMSW[3:0] bits) is reached. Then the next SPI bus cycle starts.

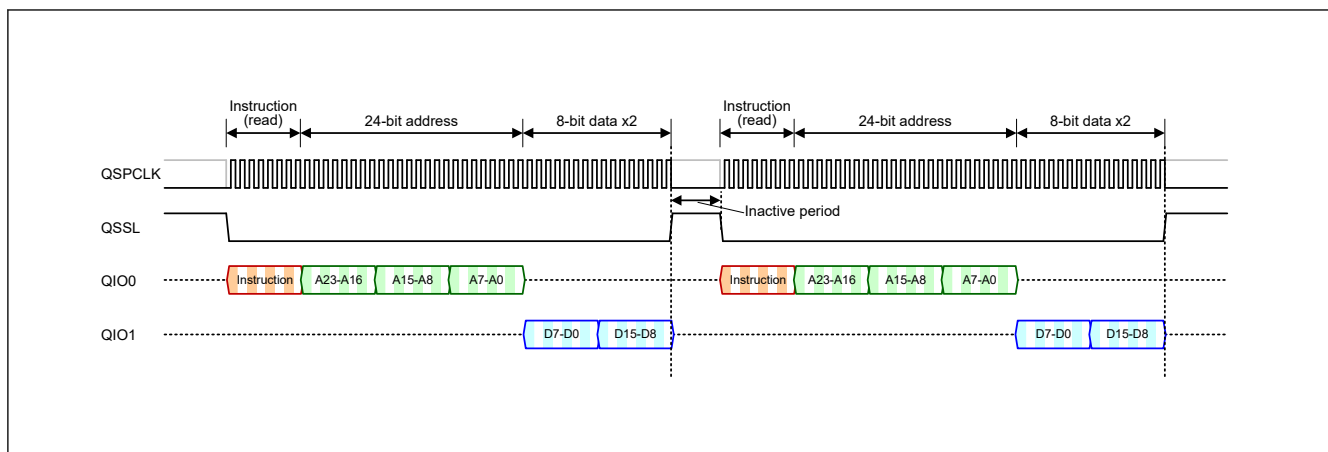


Figure 36.26 Successive data read operations based on individual conversion

36.7.2 Serial Flash Memory Read Using the Prefetch Function

In operations such as CPU instruction execution and block data transfer, data is often read in ascending order from contiguous addresses. Serial flash memory provides the ability to repeat data reception without reissuing an instruction code and address. To work with this function, the QSPI has a prefetch function for continuous data reception. However, if the CPU issues a flash read request for discontinuous flash addresses, SPI bus cycles are separated from each other, disabling the prefetch function.

To enable the prefetch function of the QSPI, set the SFMPFE bit in the SFMSMD register to 1. When the prefetch function is enabled, data is received continuously and stored in the prefetch buffer of the QSPI, without waiting for another flash read request. When the CPU issues a flash read request, an address check is made. If an address match is confirmed, the data in the buffer is passed to the CPU. If an address mismatch is detected, the data in the buffer is discarded and a new SPI bus cycle is issued.

The buffer for prefetching is 18 bytes long. When this buffer is full, the SPI bus cycle is ended. When the buffer data is read to create free space, a new SPI bus cycle is automatically started to resume prefetching.

The prefetch function allows for efficient transfer operations when data is read in ascending order from contiguous addresses, as in instruction fetch and block data transfer.

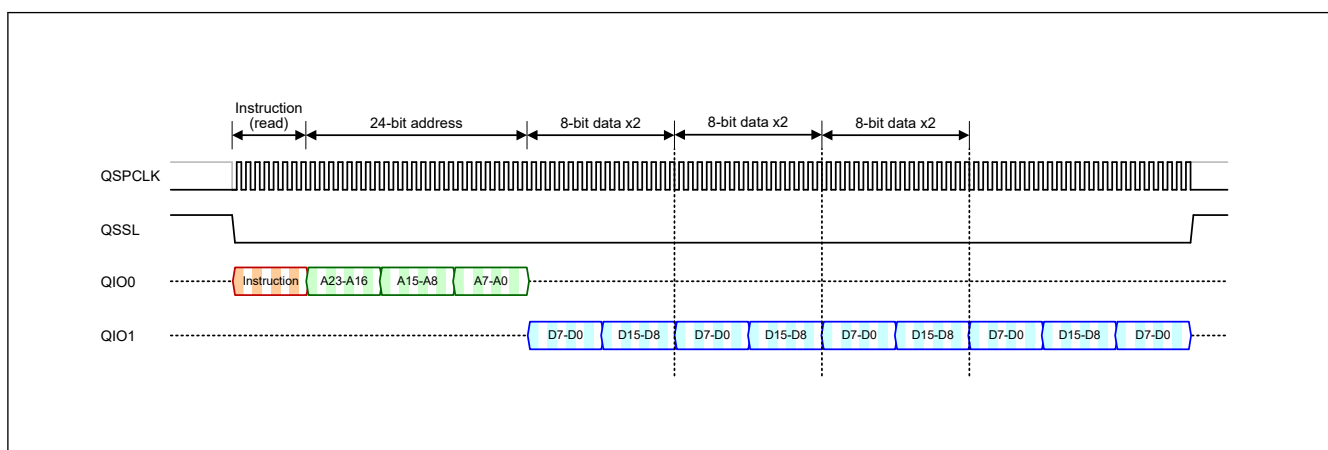


Figure 36.27 Successive data read operations using the prefetch function

36.7.3 Halt of Prefetching

If a serial flash memory read request for discontinuous addresses is issued when continuous data is being received by the prefetch function, the transfer of continuous data being made is halted and a new SPI bus cycle is started.

36.7.4 Direct Specification of Prefetch Destination

When the prefetch function is enabled (SFMSMD.SFMPFE = 1), when writing to the QSPI window area occurs, after the writing is completed, prefetching starts from the write start address. Writes to serial flash memory cannot be performed.

Combining this function with described in [section 36.7.5. Prefetch State Polling](#), can reduce the load on the internal bus when data is read from a low-speed serial flash memory.

Note: Writing to the QSPI window area with a data size of 2 bytes or more causes a hardfault.

36.7.5 Prefetch State Polling

A read by CPU from a low-speed serial flash memory causes the CPU system bus to be occupied until completion of the SPI reception bus cycle. The prefetch state polling function is provided to reduce this load.

The PFOFF bit in the Status Register (SFMSST) register indicates the state of the prefetch function, and the PFCNT[4:0] bits in the SFMSST register indicate the number of data bytes already prefetched. Place the polling program in the SRAM of this device.

```
//
// copy 1K byte (32bit x 256 word) data from serial flash to internal SRAM
//
unsigned long *sptr; // pointer for the serial flash
unsigned long *dptr; // pointer for the destination
int i;

SFMSMD |= 0x0040; // set SFMPFE bit to enable prefetch
*( (volatile unsigned char *) sptr ) = 0; // make the TAG valid to start prefetch

for ( i = 0 ; i < 256 ; i++ ){
while ( ( SFMSST & 0x00FF ) < 0x04 ){}; // waiting for 4-byte data to be received
*(dptr++) = *(sptr++);
}
```

Note: When executing a polling program, place the program outside of the serial flash memory. If the polling program is executed when the program is placed on the serial flash memory, the prefetch target frequently switches to an instruction code. This eliminates the effect of polling, and an infinite loop can result because the prefetch buffer is not filled.

36.7.6 SPI Bus Cycle Extension Function

If the SFMSE[1:0] bits in the SFMSMD register are set to a value other than 00b, the QSPI waits for the next flash read. At this time, the QSPCLK signal stops, the QSSL signal is kept active low even after data is obtained from the serial flash memory, and the SPI bus cycle is suspended.

If the address of the next flash read is contiguous in ascending order, the toggling of the QSPCLK signal is restarted to continue reception of subsequent data. If the address of the next flash read is not contiguous in ascending order, the QSSL signal is driven high once to end the SPI bus cycle being suspended. A new SPI bus cycle is then started.

When data is read intermittently from ascending order contiguous addresses, this function enables an efficient transfer operation to be performed by reducing the overhead for instruction code and address transmission.

The SPI bus cycle extension time is selectable in the SFMSE[1:0] bits in the SFMSMD register. When the specified extension time elapses, the QSSL signal returns to the high level to automatically end the SPI bus cycle being suspended. If the SFMSE[1:0] bits are set to 11b, QSSL is extended infinitely. This increases the power consumption of the serial flash memory.

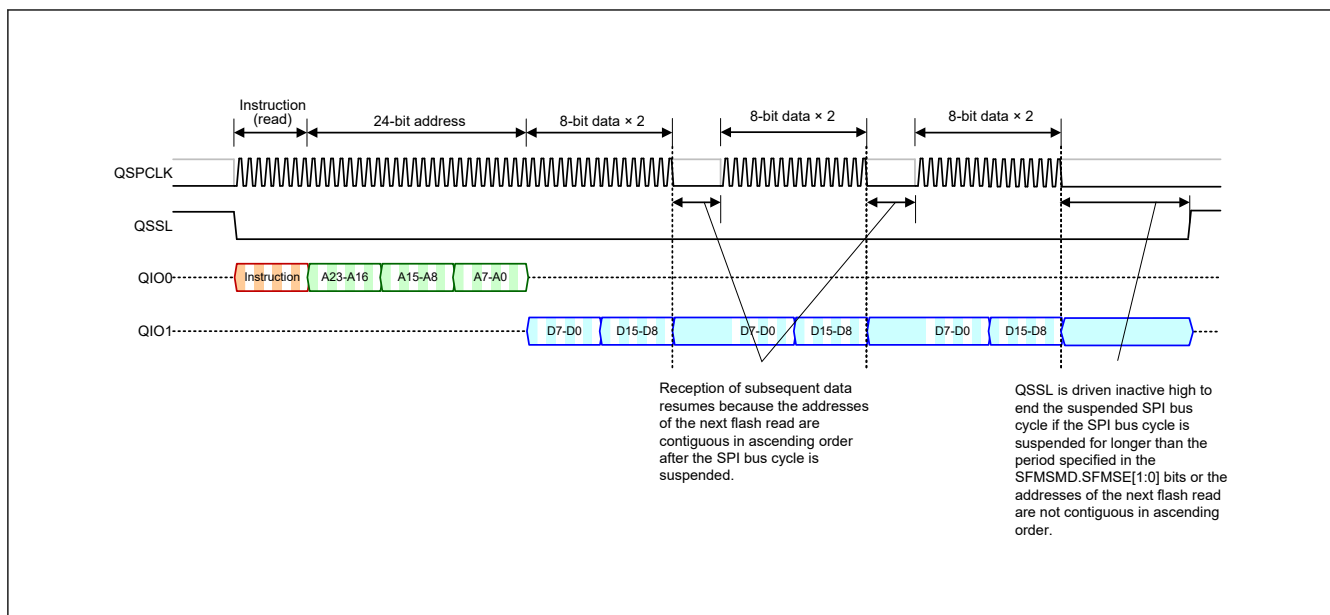


Figure 36.28 Successive data read operations using the SPI bus cycle extension

36.8 XIP Control

Some serial flash memory devices allow latencies to be reduced by skipping instruction code reception for flash reads. This instruction code skip function is selected in mode data received during the dummy cycle period of the previous serial bus cycle.

In the dummy cycle of the Fast Read instructions, the QSPI controls the XIP mode of the serial flash memory by using the serial data signal to send the mode data set in the SFMXD[7:0] bits in the SFMSDC register during the first 2 cycles, as shown in [Figure 36.29](#).

The mode data to enable the XIP mode differs for each serial flash memory. Accordingly, set the appropriate mode data in the SFMXD[7:0] bits.

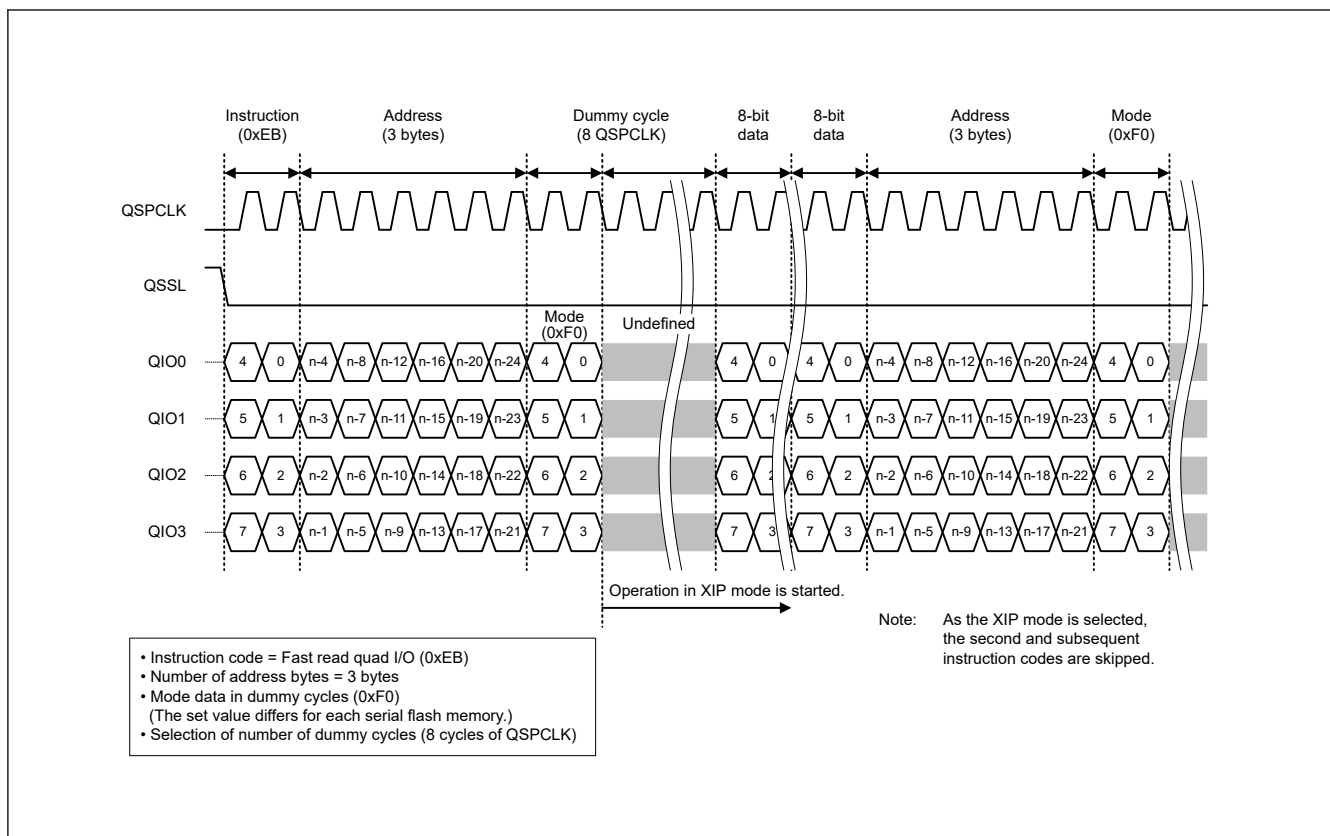


Figure 36.29 XIP mode control data

36.8.1 Setting XIP Mode

To start XIP mode in serial flash memory, perform the following register settings:

- Set a mode data value in the SFMXD[7:0] bits in the SFMSDC register.*¹
- Set the SFMXEN bit in the SFMSDC register to 1.

In the dummy cycle of the first fast read cycle after these registers are set, the mode data value set in the register is transferred. From that point, XIP mode is enabled in the serial flash memory. To confirm the current XIP mode status, read the SFMXST flag in the SFMSDC register.

Note 1. In the SFMXD[7:0] bits in the SFMSDC register, set the mode data that follows the specifications for the actual serial flash memory.

The following figure shows an example of the XIP mode setting procedure.

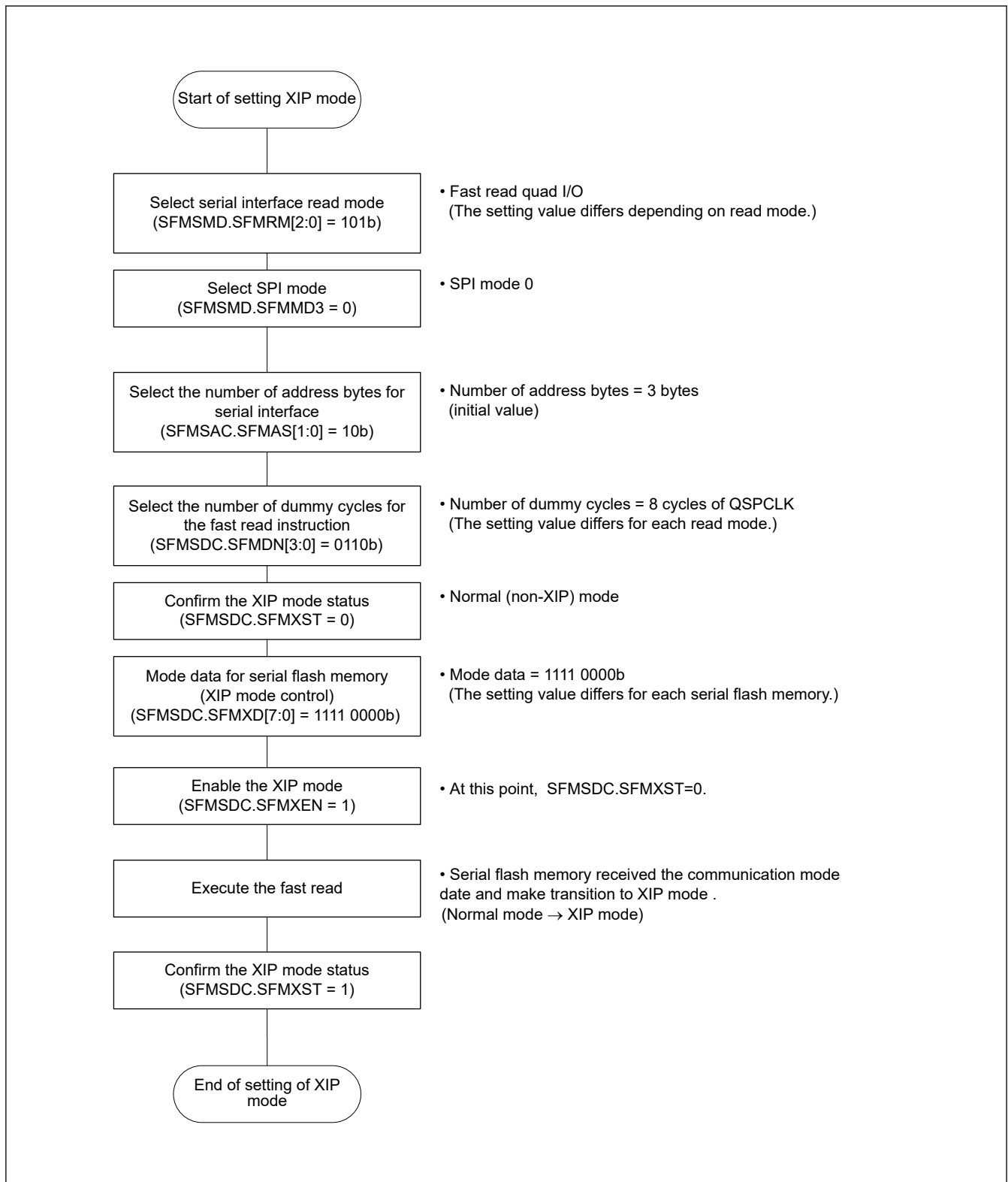


Figure 36.30 Flowchart of XIP Mode

36.8.2 Releasing the XIP Mode

To release XIP mode in serial flash memory, perform the following register setting:

- Set the mode data value to disable XIP mode in SFMSDC.SFMXD [7: 0] bits^{*1}
- Set the SFMXEN bit in the SFMSDC register to 0.

In the dummy cycle of the first fast read cycle after this register is set, The mode data value that disables the XIP mode set in the register is transferred. From that point, XIP mode is disabled in the serial flash memory. To confirm the current XIP mode status, read the SFMXST flag in the SFMSDC register.

Note 1. Set the mode data in the SFMSDC.SFMXD [7: 0] bits according to the specifications of the serial flash memory.

Figure 36.31 shows an example of the procedure for releasing XIP mode.

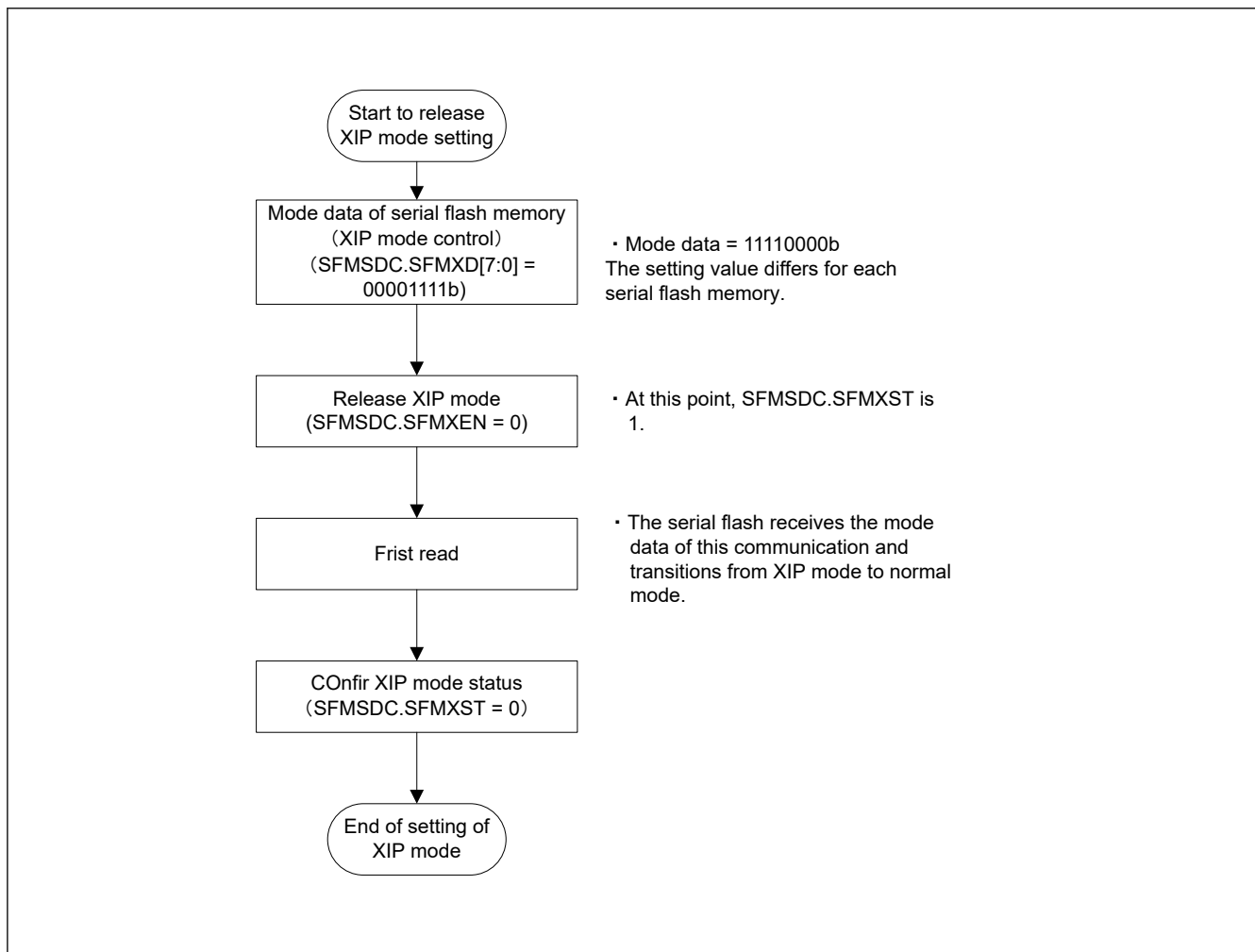


Figure 36.31 Releasing XIP mode (flowchart)

36.9 QIO2 and QIO3 Pin States

The QIO2 and QIO3 pin states depend on the serial interface read mode specified in the SFMRM[2:0] bits in the SFMSMD register.

Table 36.11 QIO2 and QIO3 pin states (1 of 2)

SFMSMD.SFMRM[2:0] bits	QIO2 pin state*1	QIO3 pin state*2	Remarks
111	Setting prohibited		
110			
101	Input or output as serial data signal (The pin is in the Hi-Z state when it is inactive.)	Input or output as serial data signal (The pin is in the Hi-Z state when it is inactive.)	Fast Read Quad I/O
100			Fast Read Quad Output

Table 36.11 QIO2 and QIO3 pin states (2 of 2)

SFMSMD.SFMRM[2:0] bits	QIO2 pin state* ¹	QIO3 pin state* ²	Remarks
011	Output SFMWPL bit variable of the Port Control Register (SFMPMD) (initial value is low level)	Output high level	Fast Read Dual I/O
010			Fast Read Dual Output
001			Fast Read
000			Read (Initial State)

Note 1. The serial flash memory can also use the QIO2 pin for the write protect (WP) function. The WP function prohibits writes to the status registers. (The function is available in mode other than Quad-SPI mode.)

Note 2. The serial flash memory can also use the QIO3 pin for the HOLD or RESET function.

The hold function places the I/O pin in an inactive state without deselecting the chip. (The function is available in mode other than Quad-SPI mode.)

The reset function resets the serial flash memory. (The function is available when the QSSL pin function is disabled or in a mode in which the QIO3 pin is not used.)

36.10 Direct Communication Mode

36.10.1 About Direct Communication

The QSPI can read the serial flash memory contents by automatically converting from reading the QSPI window area to SPI bus cycles. However, serial flash memory have many different functions in addition to memory data read, including ID information read, erase, programming, and status information read. There is no standardized instruction set for using these functions, and more functions are being added rapidly by different vendors to different devices. Therefore, to support these functions, the software can create any required SPI bus cycle by communicating directly with serial flash memory.

36.10.2 Using Direct Communication Mode

To communicate directly with serial flash memory, transition to direct communication mode by setting the DCOM bit in the Communication Mode Control Register (SFMCMMD) register to 1. While direct communications mode is selected, the read operation to the serial flash memory by the QSPI window is invalid.

Note: If the QSPI is set to the XIP mode, you must terminate the XIP mode before starting direct communication mode.

36.10.3 Generating the SPI Bus Cycle during Direct Communication

The SPI bus cycle in direct communications starts on the first access to the SFMCOM register, and after a series of input / output operations are executed via the SFMCOM register, the bus cycle ends when 1 is written to the SFMCMMD register. At that point, a write to the SFMCOM port is converted to a one-byte transmission to the SPI bus, and a read from the SFMCOM register is converted to a one-byte reception from the SPI bus.

During the period from the first access to the SFMCOM register to the last write operation to the SFMCMMD register, the QSSL signal is held active to notify the serial flash memory that a series of SPI bus cycles is in progress.

Note: In direct communication mode, all writes to registers other than SFMCMMD and SFMCOM (including SFMSMD, SFMSSC, SFMSKC, SFMSST, SFMCST, SFMSIC, SFMSAC, SFMSDC, SFMSPC, and SFMPMD) are disabled.

Figure 36.32 to Figure 36.34 show direct communication program examples, and Figure 36.35 shows ID read direct communication timing examples.

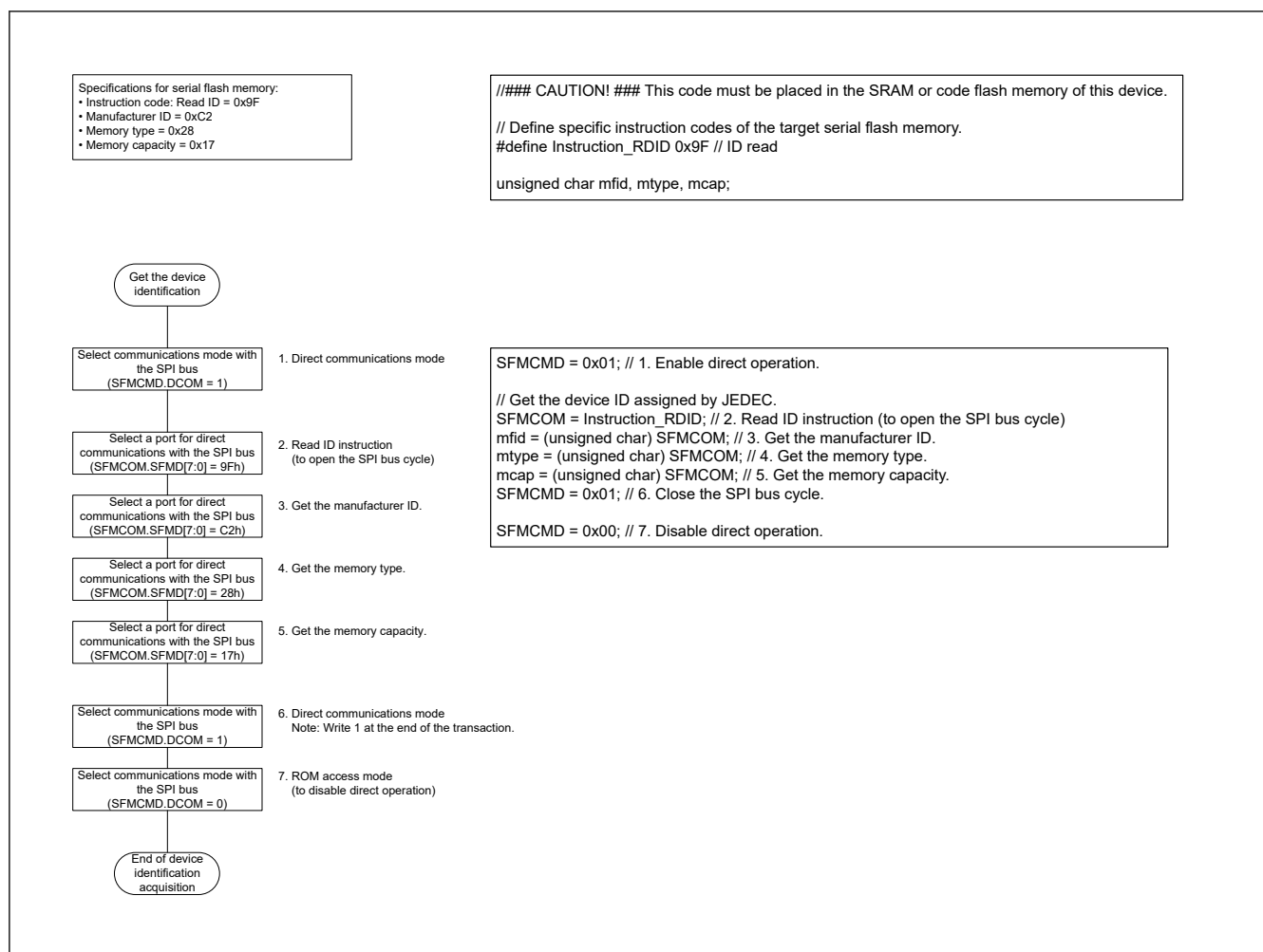


Figure 36.32 Flowchart of Device ID Acquisition

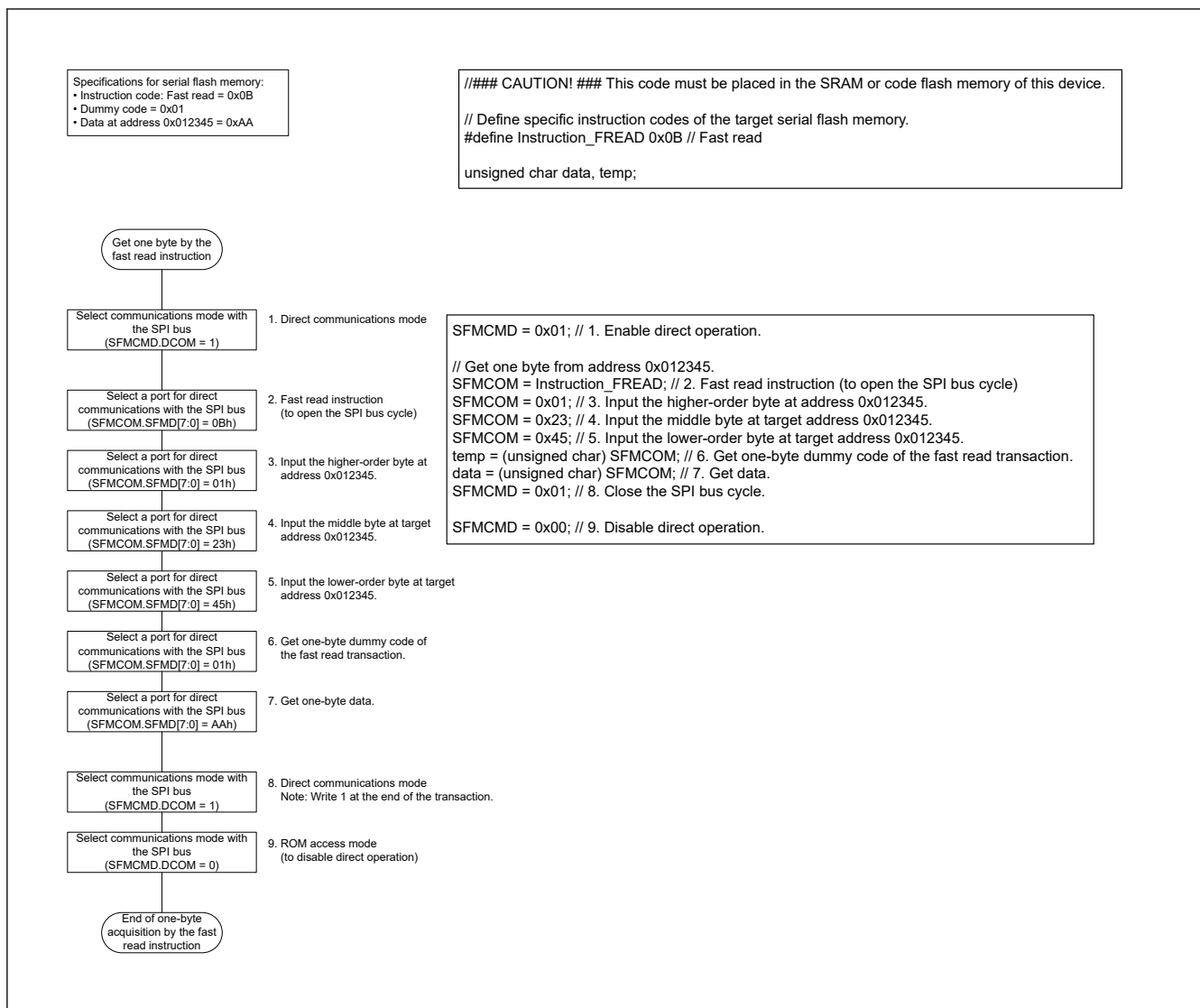


Figure 36.33 Flowchart of One-byte Acquisition by the Fast Read Instruction

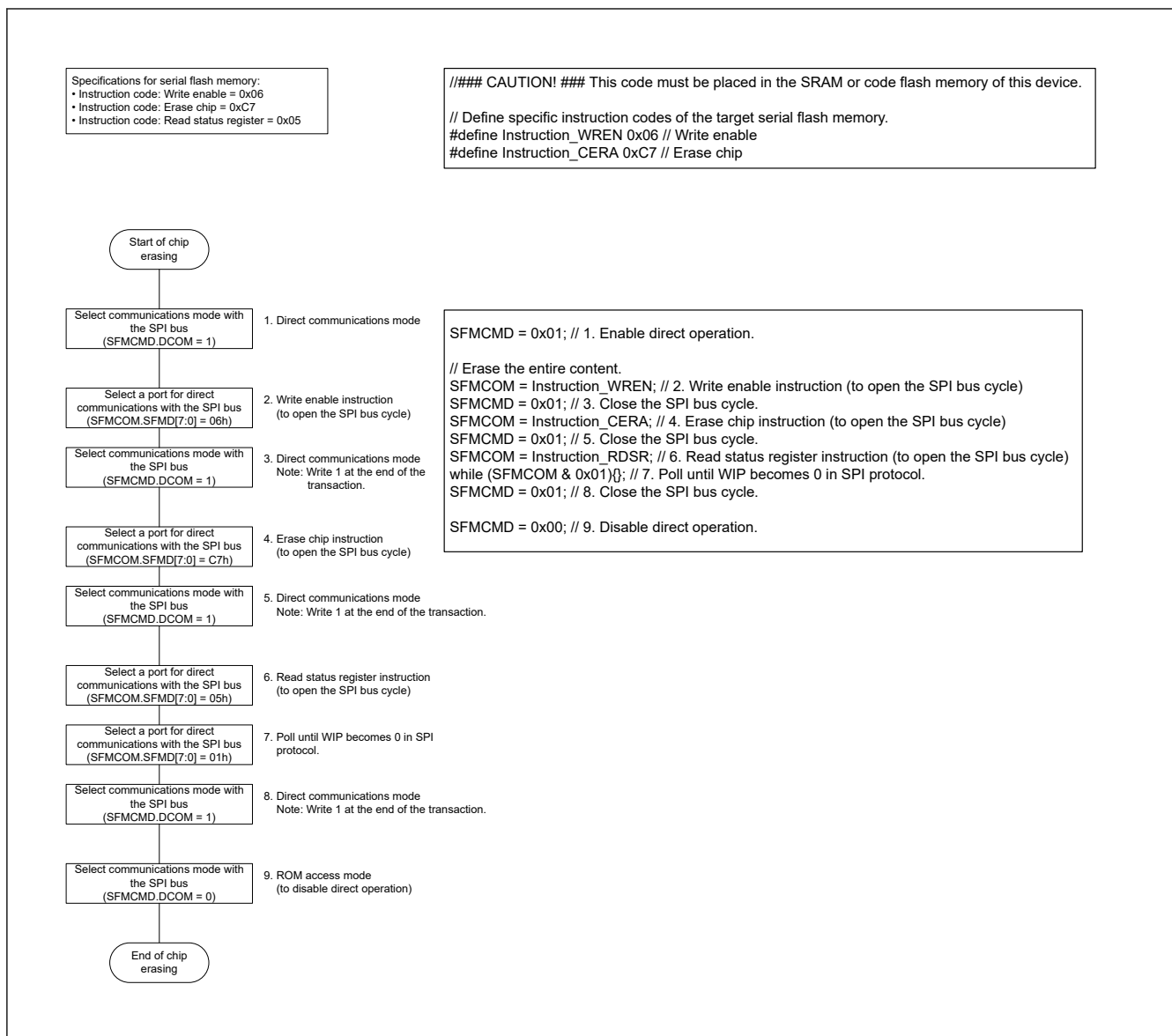


Figure 36.34 Flowchart of Chip Erasing

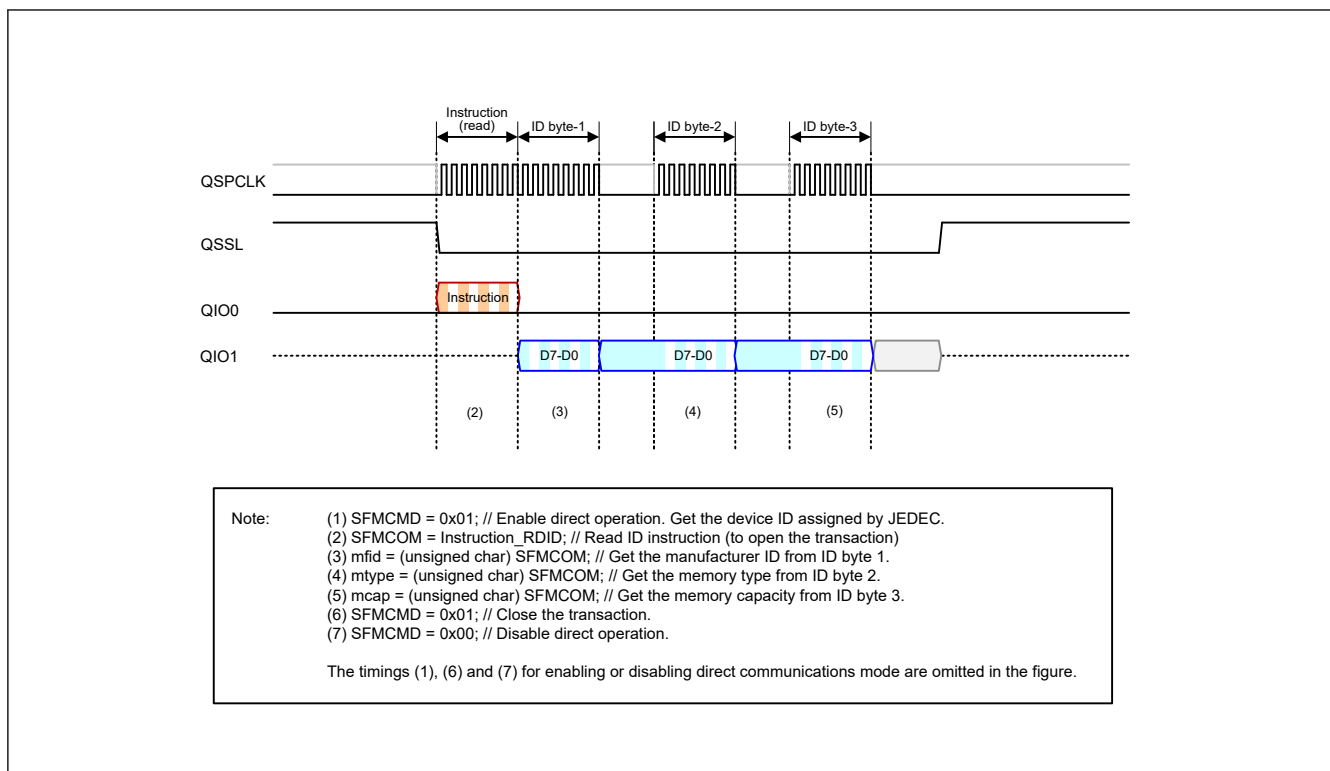


Figure 36.35 Example of direct communication timing for ID read

Note: When the Single SPI Protocol, Extended SPI Protocol is used in direct communication mode, the standard Read or Fast Read instruction must be used to reference the contents of the serial flash memory. The QSPI does not support Fast Read Dual Output, Fast Read Dual I/O, Fast Read Quad Output, or Fast Read Quad I/O transfers in this configuration. When these high-speed read operations are required, use ROM access memory.

36.11 Interrupts

When ROM read access is detected in direct communication mode, the SFMCST.EROMR flag is set to 1 and QSPI generates an interrupt request. Interrupt requests are retained until the EROMR bit is cleared by a 0 write. For details, see [section 16, Interrupt Controller Unit \(ICU\)](#).

36.12 Usage Note

36.12.1 Settings for the Module-Stop Function

QSPI operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The QSPI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

36.12.2 Procedure for Changing Settings in Multiple Control Registers

The settings of the QSPI control registers can be modified dynamically during system operation. However, when the settings of multiple control registers are changed sequentially, an SPI bus cycle might occur before all of the registers are updated. The register setting sequence must be designed so that the SPI bus timing specifications are satisfied at all stages of register setting modification.

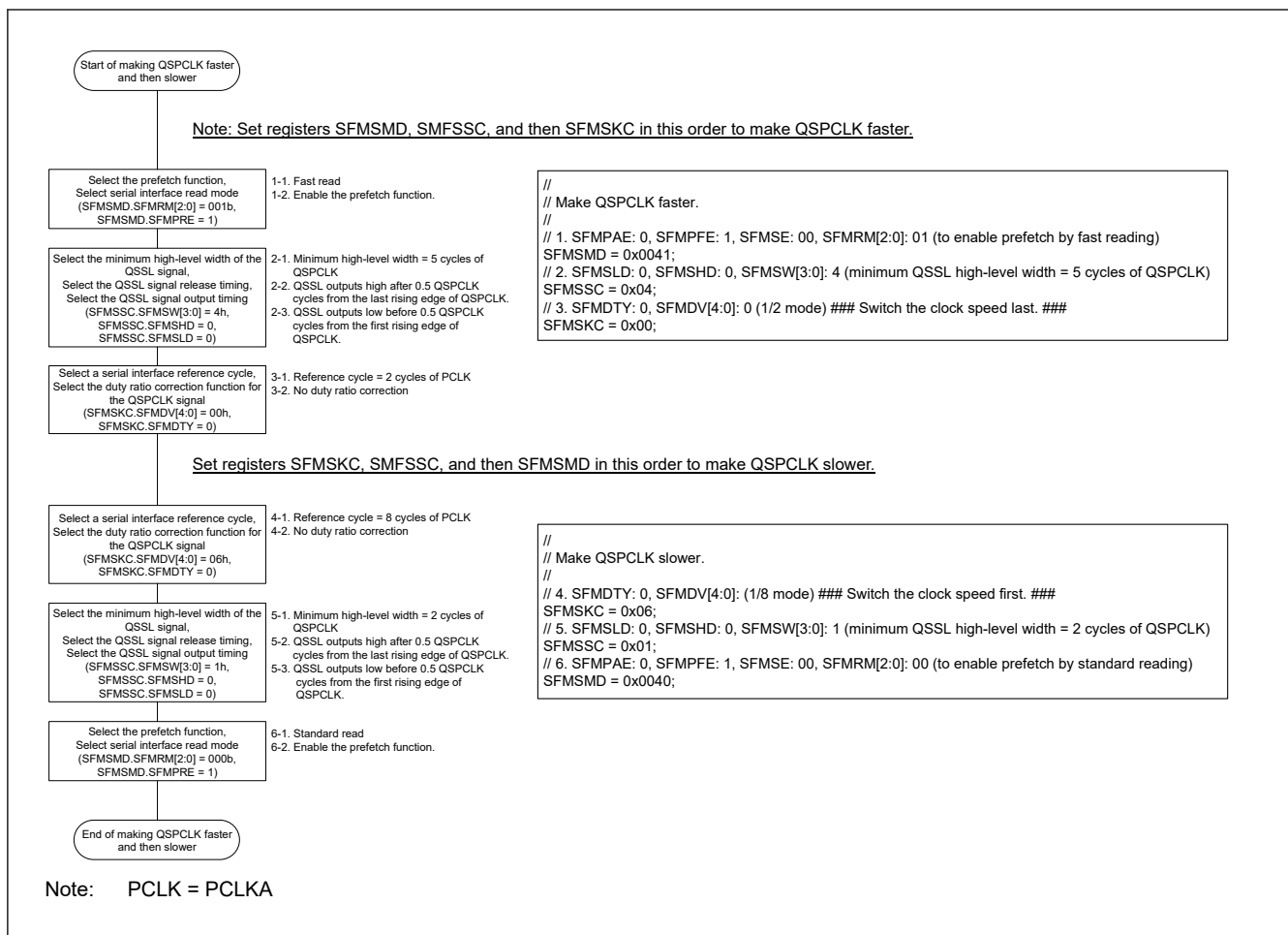


Figure 36.36 Flowchart of Making QSPCLK Faster and Slower

36.12.3 Setting the Serial Flash Memory

For the mode data and pins of serial flash memory, see the manual for the serial flash memory you are using because they differ for each device.

37. Cyclic Redundancy Check (CRC) Calculator

37.1 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 37.1 lists the CRC calculator specifications and Figure 37.1 shows a block diagram.

Table 37.1 CRC calculator specifications

Parameter	Specifications for 8-bit data	Specifications for 32-bit data
Data size	8-bit	32-bit
Data for CRC calculation*1	CRC code generated for data in 8n-bit units (where n is a natural number)	CRC code generated for data in 32n-bit units (where n is a natural number)
CRC processor unit	Operation executed on 8 bits in parallel	Operation executed on 32 bits in parallel
CRC generating polynomial	One of three generating polynomials that is selectable: [8-bit CRC] $X^8 + X^2 + X + 1$ (CRC-8) [16-bit CRC] $X^{16} + X^{15} + X^2 + 1$ (CRC-16) $X^{16} + X^{12} + X^5 + 1$ (CRC-CCITT).	One of two generating polynomials that is selectable: [32-bit CRC] $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ (CRC-32) $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$ (CRC-32C).
CRC calculation switching	The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication	
Module-stop function	Module-stop state can be set to reduce power consumption	
CRC snoop	Monitor reads from and writes to a certain register address	—

Note 1. This function cannot divide data used in CRC calculations. Write data in 8-bit or 32-bit units.

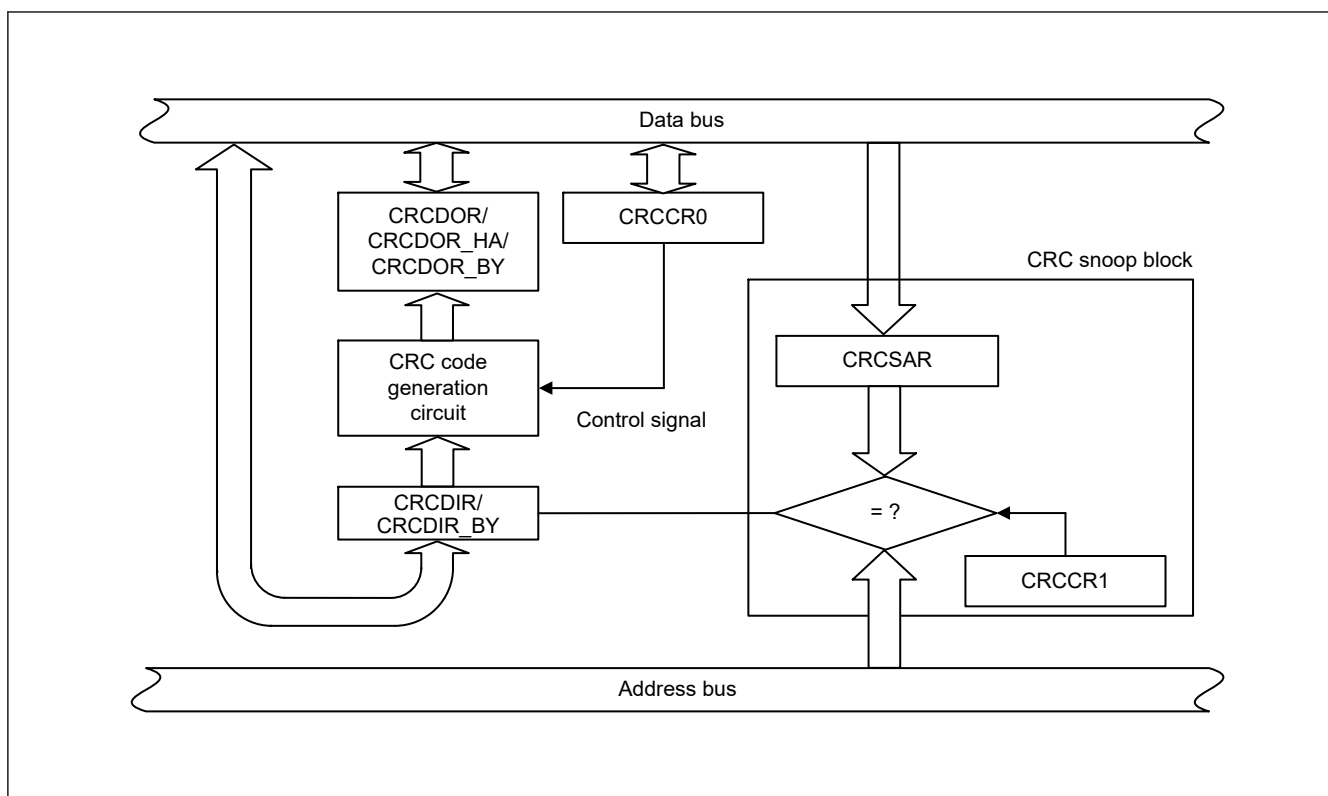


Figure 37.1 CRC calculator block diagram

37.2 Register Descriptions

37.2.1 CRCCR0 : CRC Control Register 0

Base address: CRC = 0x4007_4000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DORCLR	LMS	—	—	—	GPS[2:0]		

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	GPS[2:0]	CRC Generating Polynomial Switching 0 0 1: 8-bit CRC-8 ($X^8 + X^2 + X + 1$) 0 1 0: 16-bit CRC-16 ($X^{16} + X^{15} + X^2 + 1$) 0 1 1: 16-bit CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) 1 0 0: 32-bit CRC-32 ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$) 1 0 1: 32-bit CRC-32C ($X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$) Others: No calculation is executed	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	LMS	CRC Calculation Switching 0: Generate CRC code for LSB-first communication 1: Generate CRC code for MSB-first communication	R/W
7	DORCLR	CRCDOR/CRCDOR_HA/CRCDOR_BY Register Clear 0: No effect 1: Clear the CRCDOR/CRCDOR_HA/CRCDOR_BY register	W

GPS[2:0] bits (CRC Generating Polynomial Switching)

The GPS[2:0] bits select the CRC generating polynomial.

LMS bit (CRC Calculation Switching)

The LMS bit selects the bit order of generated CRC code. Transmit the lower byte of the CRC code first for LSB-first communication and the upper byte first for MSB-first communication. For details on transmitting and receiving CRC code, see [section 37.3. Operation](#).

DORCLR bit (CRCDOR/CRCDOR_HA/CRCDOR_BY Register Clear)

Write 1 to the DORCLR bit to set the CRCDOR/CRCDOR_HA/CRCDOR_BY register to 0x00000000. This bit is read as 0. Only 1 can be written to it.

37.2.2 CRCCR1 : CRC Control Register 1

Base address: CRC = 0x4007_4000

Offset address: 0x01

Bit position: 7 6 5 4 3 2 1 0

Bit field:	CRCS EN	CRCS WR	—	—	—	—	—	—

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	CRCSWR	Snoop-On-Write/Read Switch 0: Snoop-on-read 1: Snoop-on-write	R/W
7	CRCSEN	Snoop Enable 0: Disabled 1: Enabled	R/W

CRCSWR bit (Snoop-On-Write/Read Switch)

The CRCSWR bit selects the direction of access in the CRC snoop function.

When this bit is set to 0 (initial value), the CRC snoop operation to read a specific register is enabled. Similarly, when this bit is set to 1, the CRC snoop operation to write a specific register is enabled.

CRCSEN bit (Snoop Enable)

When the CRCSEN bit is set to 1, the CRC snoop operation is enabled. When this bit is set to 0, the CRC snoop operation is disabled.

37.2.3 CRCDIR/CRCDIR BY : CRC Data Input Register

Base address: CRC = 0x4007 4000

Offset address: 0x04

Bit position: 31 0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	CRC input data The CRCDIR register is a 32-bit read/write register to write data for CRC-32 or CRC-32C calculation. The CRCDIR_BY (CRCDIR[31:24]) is an 8-bit read/write register to write data for CRC-8, CRC-16, or CRC-CCITT calculation.	R/W

37.2.4 CRCDOR/CRCDOR_HA/CRCDOR_BY : CRC Data Output Register

Base address: CRC = 0x4007_4000

Offset address: 0x08

Bit position: 31

0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	CRC output data The CRCDOR register is a 32-bit read/write register for CRC-32 or CRC-32C calculation. The CRCDOR_HA (CRCDOR[31:16]) register is a 16-bit read/write register for CRC-16 or CRC-CCITT calculation. The CRCDOR_BY (CRCDOR[31:24]) register is an 8-bit read/write register for CRC-8 calculation. Because its initial value is 0x00000000, rewrite the CRCDOR/CRCDOR_HA/CRCDOR_BY register to perform the calculations using a value other than the initial value. Data written to the CRCDIR/CRCDIR_BY register is CRC calculated and the result is stored in the CRCDOR/CRCDOR_HA/CRCDOR_BY register. If the CRC code is calculated following the transferred data and the result is 0x00000000, there is no CRC error.	R/W

37.2.5 CRCSAR : Snoop Address Register

Base address: CRC = 0x4007_4000

Offset address: 0x0C

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:

CRCSA[13:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
13:0	CRCSA[13:0]	Register Snoop Address These bits store the TDR or RDR address in the SCI module to snoop	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

CRCSA[13:0] bits (Register Snoop Address)

The CRCSA[13:0] bits specify the lower address 14 bits of the register monitored by the CRC snoop operation.

Only the following addresses can be used for the CRCSA[13:0] bits:

- 0x4007_0003: SCI0.TDR, 0x4007_0005:SCI0.RDR
- 0x4007_0023: SCI1.TDR, 0x4007_0025:SCI1.RDR
- 0x4007_000F: SCI0.FTDRL, 0x4007_0011: SCI0.FRDR
- 0x4007_002F: SCI1.FTDRL, 0x4007_0031: SCI1.FRDR

37.3 Operation

37.3.1 Basic Operation

The CRC calculator generates CRC codes for use in LSB-first or MSB-first transfer.

The following examples show CRC code generation for input data (0xF0) using the 16-bit CRC-CCITT generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC Data Output Register (CRCDOR_HA) is cleared before CRC calculation.

When an 8-bit CRC (with the polynomial $X^8 + X^2 + X + 1$) is in use, the valid bits of the CRC code are obtained in CRCDOR_BY. When a 32-bit CRC is in use, the valid bits of the CRC code are obtained in CRCDOR.

Figure 37.2 and Figure 37.3 show the LSB-first and MSB-first data transmission examples respectively. Figure 37.4 and Figure 37.5 show the LSB-first and MSB-first data reception examples.

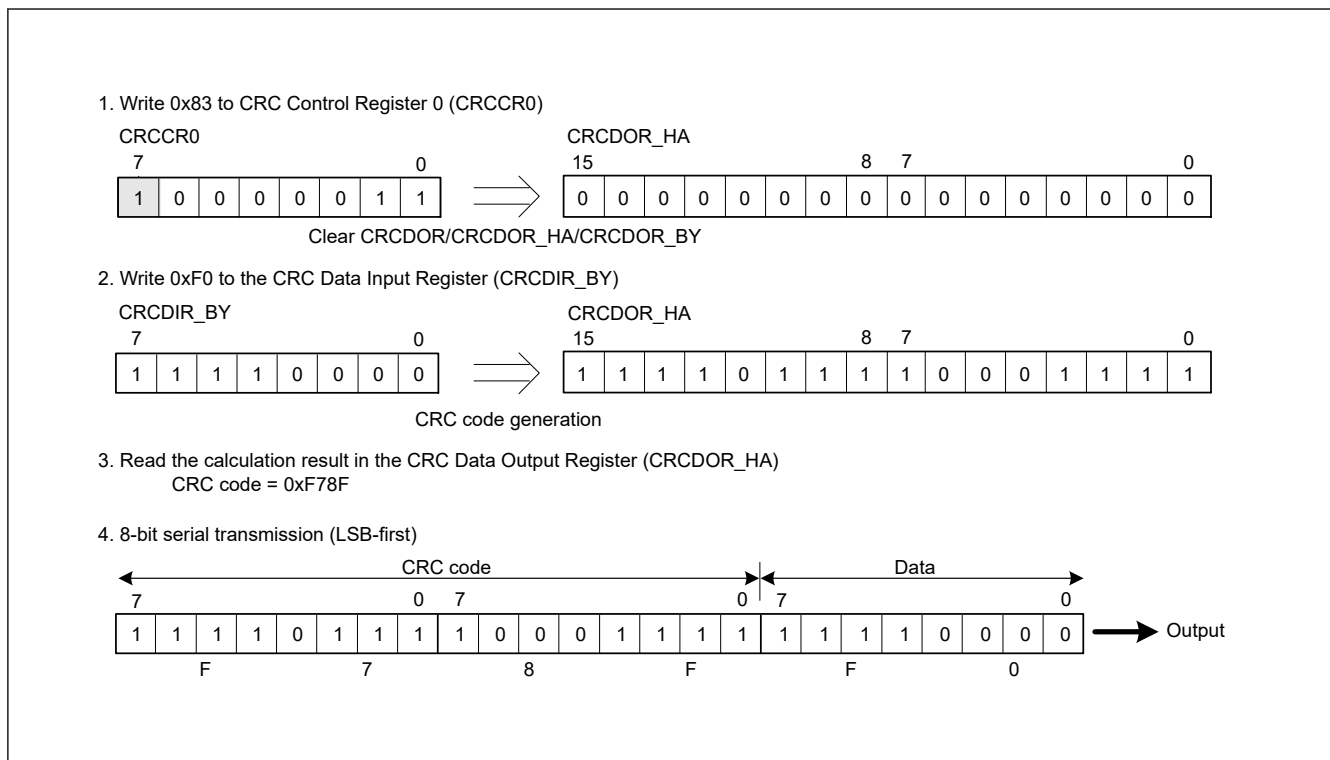


Figure 37.2 LSB-first data transmission

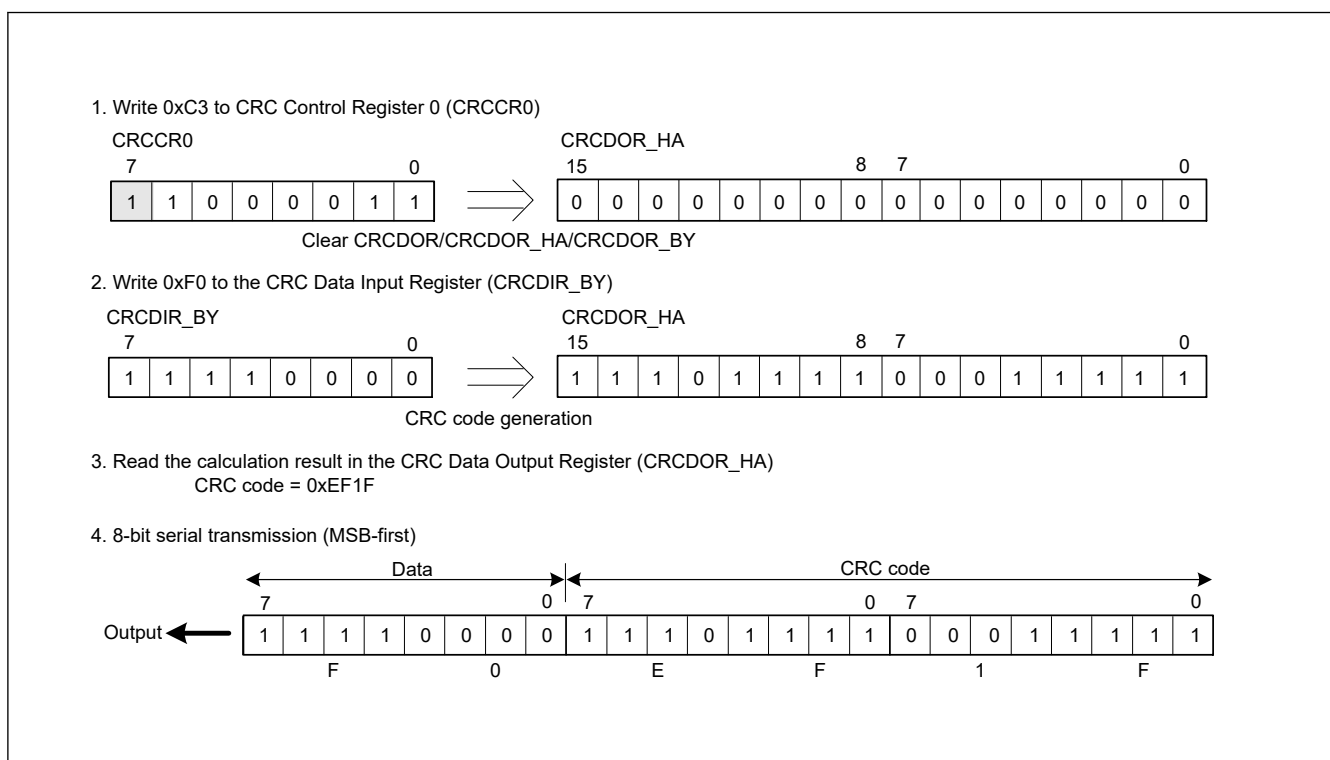
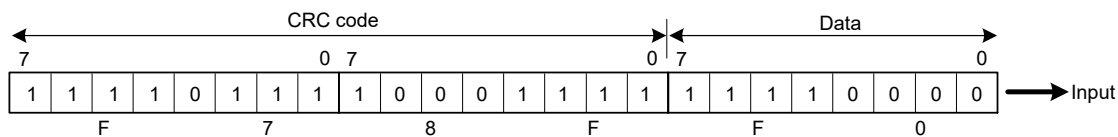


Figure 37.3 MSB-first data transmission

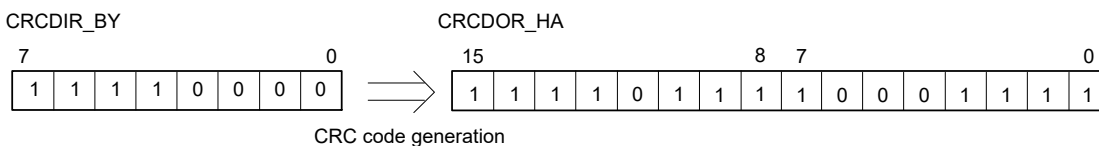
1. 8-bit serial reception (LSB-first)



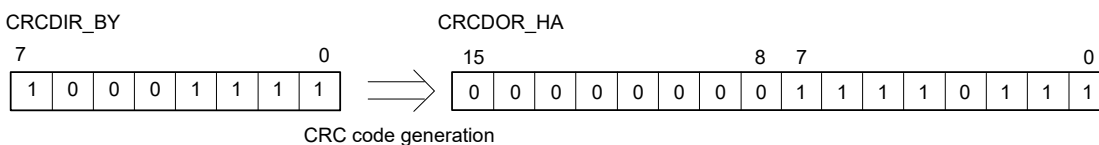
2. Write 0x83 to the CRC Control Register 0 (CRCCR0)



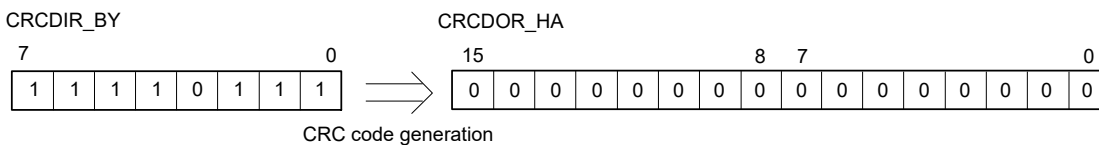
3. Write 0xF0 to the CRC Data Input Register (CRCDIR_BY)



4. Write 0x8F to the CRC Data Input Register (CRCDIR_BY)



5. Write 0xF7 to the CRC Data Input Register (CRCDIR_BY)



6. Read the calculation result in the CRC Data Output Register (CRCDOR_HA)

CRC code = 0x0000 → no error

Figure 37.4 LSB-first data reception

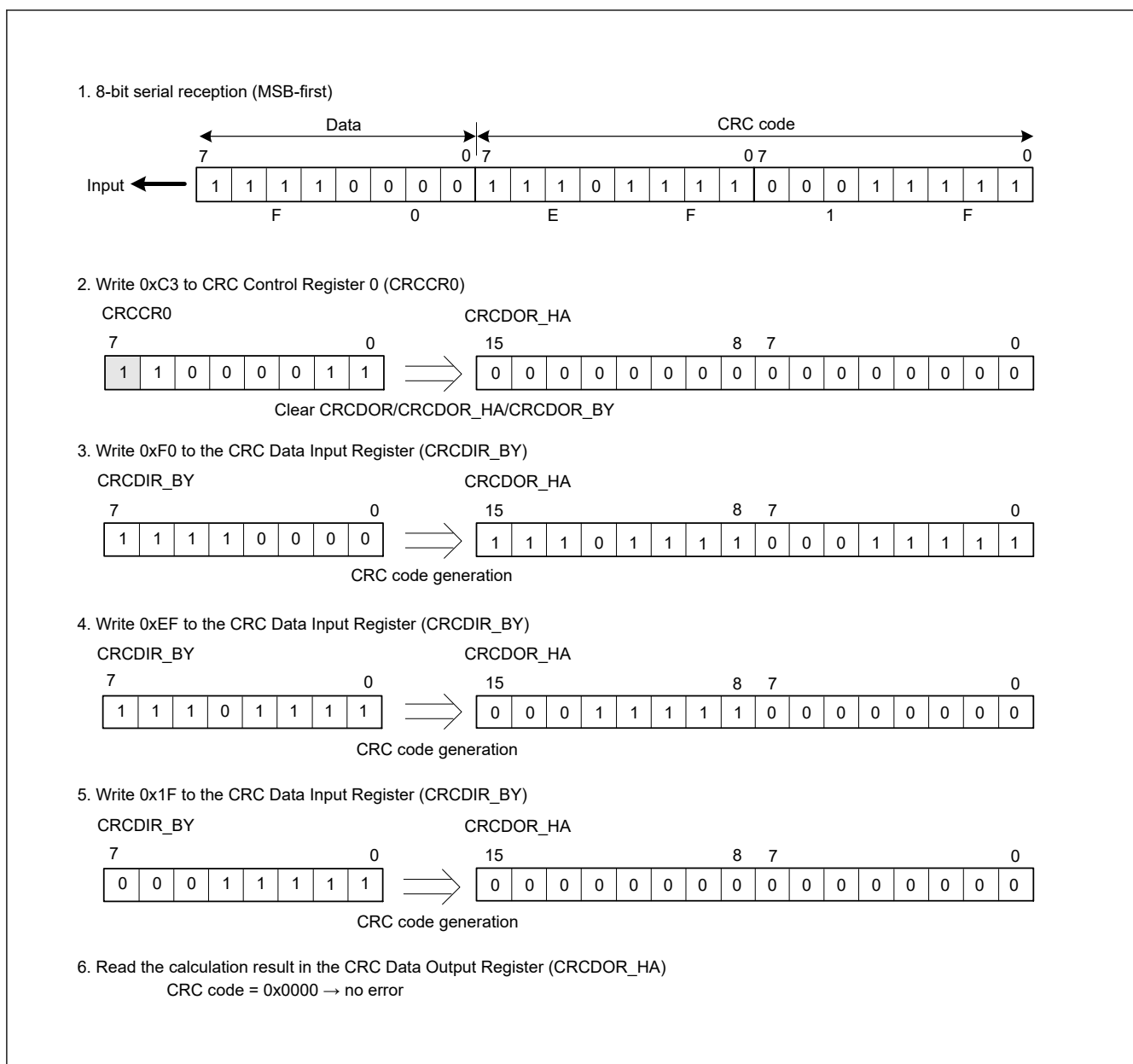


Figure 37.5 MSB-first data reception

37.3.2 CRC Snoop Function

The CRC snoop function monitors reads from and writes to a specific register and performs CRC calculation on the monitored data automatically. Because the CRC snoop function recognizes writes to and reads from a specific register address as a trigger to automatically perform CRC calculation, there is no need to write data to the CRCDIR_BY register. All I/O register specified in the [section 37.2.5. CRCSAR : Snoop Address Register](#) are subject to the CRC snoop. The CRC snoop is useful in monitoring writes to the SCIn.TDR register, and reads from the SCIn.RDR register.

To use this function, write the lower address 14 bits of a specific register to bits CRCSA13 to CRCSA0 in the CRCSAR register, and set CRCSSEN bit in the CRCCR1 register to 1. Then, set the CRCSWR bit in the CRCCR1 register to determine the access direction.

When both the CRCSSEN and CRCSWR bits are set to 1, and data is written to a target register in a bus master module such as the CPU, DMAC, and DTC, the CRC calculator stores the data in the CRCDIR_BY register and performs CRC calculation. Similarly, when the CRCSSEN bit is set to 1, CRCSWR bit to 0, and data is read from a target register in a bus master module such as the CPU, DMAC, and DTC, the CRC calculator stores the data in the CRCDIR_BY register and performs CRC calculations.

When the CRC code is generated by using CRC-8, CRC-16, and CRC-CCITT generating polynomial, the target register is accessed in 1 byte (8 bits). Similarly, when the CRC code is generated by using CRC-32 and CRC-32C generating polynomial, the target register is accessed in words (32 bits).

37.4 Usage Notes

37.4.1 Settings for the Module-Stop State

The Module Stop Control Register C (MSTPCRC) can enable or disable CRC calculator operation. The CRC calculator is initially stopped after a reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

37.4.2 Note on Transmission

The transmission sequence for the CRC code differs based on whether the transmission is LSB-first or MSB-first. [Figure 37.6](#) shows an LSB-first and MSB-first data transmission.

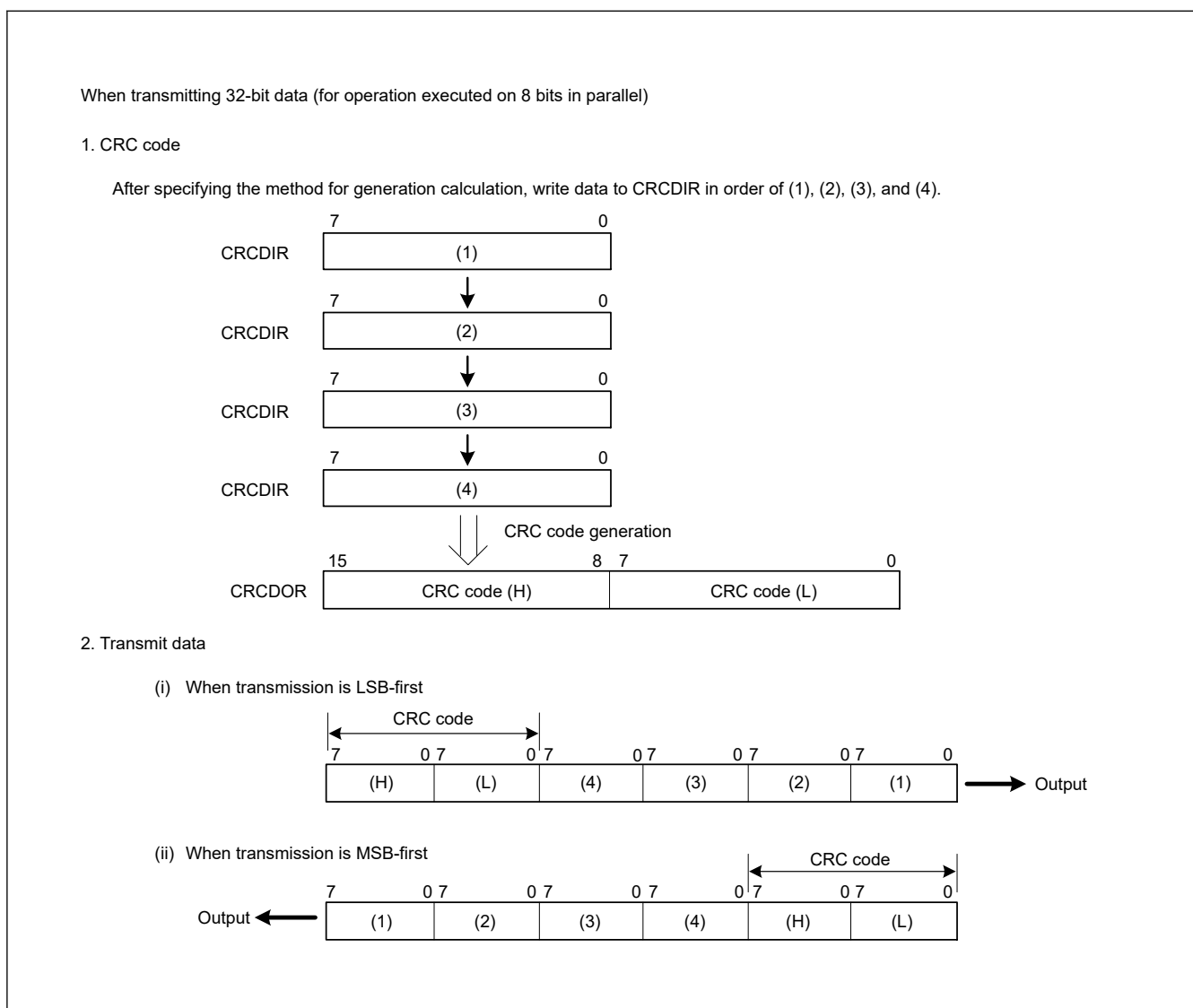


Figure 37.6 LSB-first and MSB-first data transmission

38. Division Circuit (DIV)

38.1 Overview

The divider (DIV) is a circuit that divides signed 32-bit fixed-point data at high speed in 16 cycles. When the operation is complete, an operation completion interrupt signal is output.

Table 38.1 lists the DIV specifications and Figure 38.1 shows DIV block diagram.

Table 38.1 DIV specifications

Parameter	Specifications
Division operation	The quotient and remainder of dividend / divisor are obtained. • Dividend: Signed 32-bit data • Divisor: Signed 32-bit data • Quotient: Signed 32-bit data • Remainder: Signed 32-bit data
Format	Fixed point number
Function for reducing power consumption	Module-stop state can be set.
Interrupt source	Division operation complete interrupt

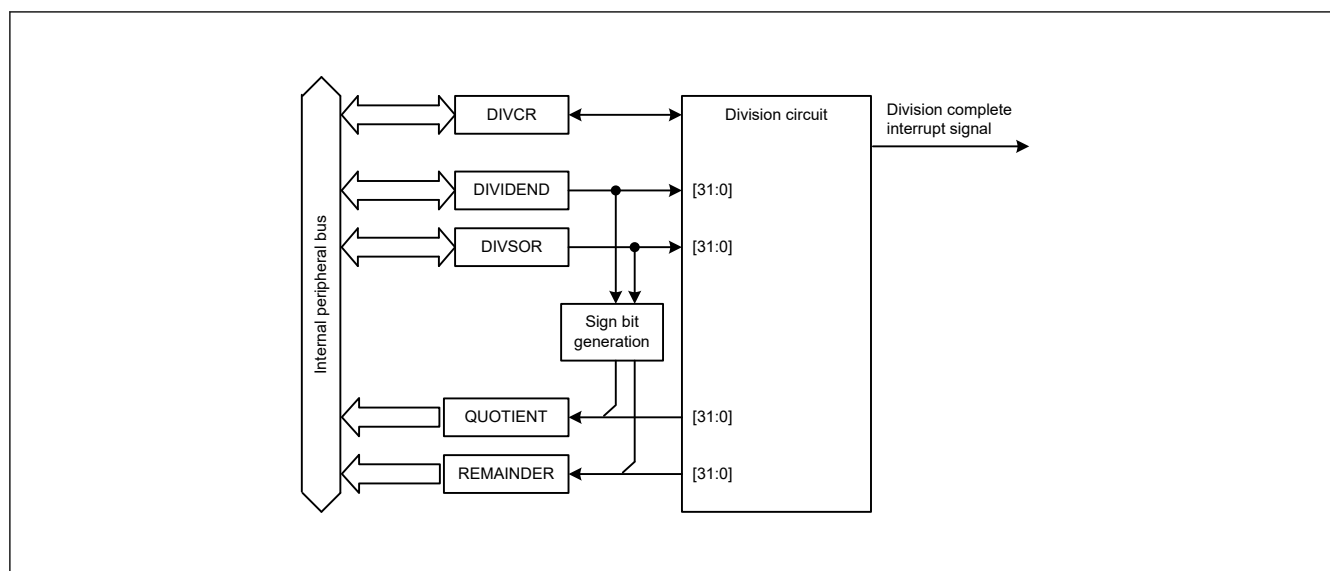


Figure 38.1 DIV block diagram

38.2 Register Descriptions

38.2.1 DIVCR : Division Operation Control Register

Base address: DIV = 0x4008_4600

Offset address: 0x80

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	DIVER	CALB	—	—	—	—	—	—	—	EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EN	Division Operation Start 0: No effect 1: Start a division operation	W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	CALBUSY	Division Operation Status Flag 0: Idle (next division operation can be started). 1: Division operation is in progress (next operation cannot be started).	R
9	DIVERR	Division by Zero Flag 0: The division operation is started after writing any value except 0 to the Divisor Setting Register (DIVSOR). 1: The division operation is started after writing 0 to the Divisor Setting Register (DIVSOR).	R
31:10	—	These bits are read as 0. The write value should be 0.	R/W

The DIVCR register contains writable only bit. For this bit, initial value is read regardless of the former setting value. Writing the value that is operated by the read value back to this register is prohibited.

EN bit (Division Operation Start)

Writing 1 to this bit starts division operation. Overwriting 1 during the operation starts division operation from the beginning.

CALBUSY flag (Division Operation Status Flag)

The CALBUSY indicates whether division operation is in progress.

[Setting condition]

- When writing 1 to the EN bit.

[Clearing condition]

- When the division operation has completed.

DIVERR flag (Division by Zero Flag)

The DIVERR flag indicates an error due to division by zero. Include the process of checking this flag if a user program may set 0 to the DIVSOR register.

[Setting condition]

- When an division operation is started after writing 0 to bit [30:0] of the DIVSOR register.

[Clearing condition]

- When an division operation is started after writing any value except 0 to bit [30:0] of the DIVSOR register.

38.2.2 DIVIDEND : Dividend Setting Register

Base address: DIV = 0x4008_4600

Offset address: 0x84

Bit position: 31

0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	The DIVIDEND register sets the dividend. Bit 31 is the sign bit. Set this register when the DIV is idle (DIVCR.CALBUSY = 0). Match the number of bits in the integer and fraction parts of this register with the associated number of bits in the integer and fraction parts of the DIVSOR register.	R/W

38.2.3 DIVSOR : Divisor Setting Register

Base address: DIV = 0x4008_4600

Offset address: 0x88

Bit position: 31

0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	The DIVSOR sets the divisor. Bit 31 is the sign bit. Set this register when the DIV is idle (DIVCR.CALBUSY = 0). Match the number of bits in the integer and fraction parts of this register with the associated number of bits in the integer and fraction parts of the DIVIDEND.	R/W

38.2.4 QUOTIENT : Quotient Indication Register

Base address: DIV = 0x4008_4600

Offset address: 0x8C

Bit position: 31

0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	The QUOTIENT is stored the quotient of the operation result as an integer. Bit 31 is the sign bit. If the fixed point positions of the DIVIDEND and DIVSOR registers are not aligned, correct results can not be obtained.	R

38.2.5 REMAINDER : Remainder Register

Base address: DIV = 0x4008_4600

Offset address: 0x90

Bit position: 31

0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	The REMAINDER stores the remainder of the operation result. Bit 31 is the sign bit. If the fixed point positions of the DIVIDEND and DIVSOR registers are not aligned, correct results can not be obtained.	R

38.3 Operation

38.3.1 Operation Procedure

The following shows the procedure for setting the divider, and [Figure 38.2](#) shows an operation example of the divider.

1. Check that the DIV is idle state (DIVCR.CALBUSY = 0).
2. Set the dividend in the DIVIDEND register.
3. Set the divisor in the DIVSOR register. For the DIVSOR register, set a value that is not 0. The DIVIDEND and DIVSOR registers can be set in any order.

4. Division operation starts when 1 is written to the DIVCR.EN bit. When division operation starts, the DIVCR.CALBUSY flag of the DIVCR register is set to 1.
5. When division operation finishes, the operation results are stored in the QUOTIENT and REMAINDER registers. Also, the DIVCR.CALBUSY flag is cleared and the division complete interrupt signal, DIV_CALCCOMP, is output.

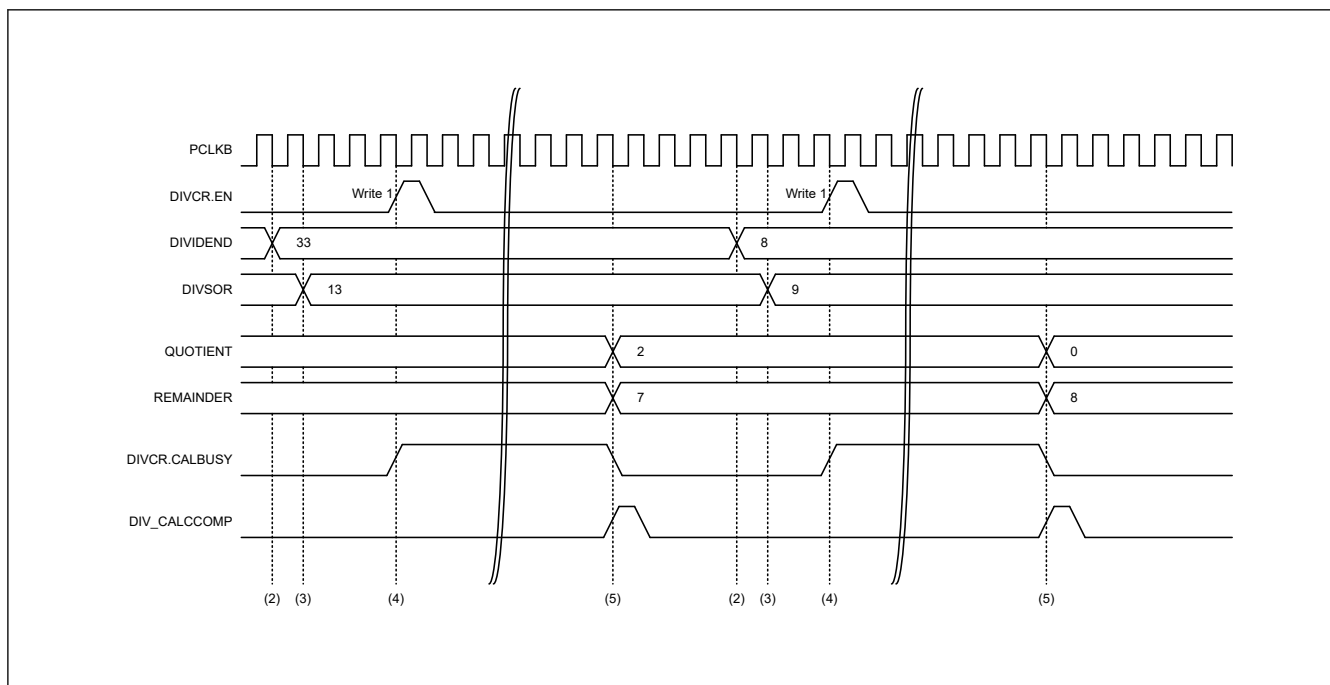


Figure 38.2 DIV Operation example

38.3.2 Interrupt Sources

The DIV can generate an interrupt signal when the division operation is complete.

Table 38.2 Interrupt source

Name	Interrupt source	CPU interrupt	DTC activation	DMAC activation
DIV_CALCCOMP	Division operation termination interrupt	Possible	Possible	Possible

If the DTC or DMAC is used to transfer data, set up and enable the DTC or DMAC before setting up the DIV. For details on the DTC settings, see [section 20, Data Transfer Controller \(DTC\)](#), and for details on the DMAC settings, see [section 19, DMA Controller \(DMAC\)](#).

38.4 Usage Notes

38.4.1 Settings for Module-Stop Function

DIV operation can be disabled or enabled using the Module Stop Control Register C (MSTPCRC). The DIV is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

39. Data Inversion and Logical Operation (DIL)

39.1 Overview

Data inversion and logical operation (DIL) provides a function to perform bitwise operation, byte swap, and bit order inversion every 8 bits of 32-bit input data.

Table 39.1 lists the DIL specifications and Figure 39.1 shows a block diagram.

Table 39.1 DIL specifications

Parameter	Specifications
Data conversion	- Data inversion - The bit inversion value of input data is output - AND, OR, and XOR operations of two input data - Data inversion enables NAND, NOR, and XNOR operations - Conversion of data alignment per byte width (byte swap) - Bit order inversion of MSB and LSB every 8 bits
Function for reducing power consumption	The module-stop state can be set

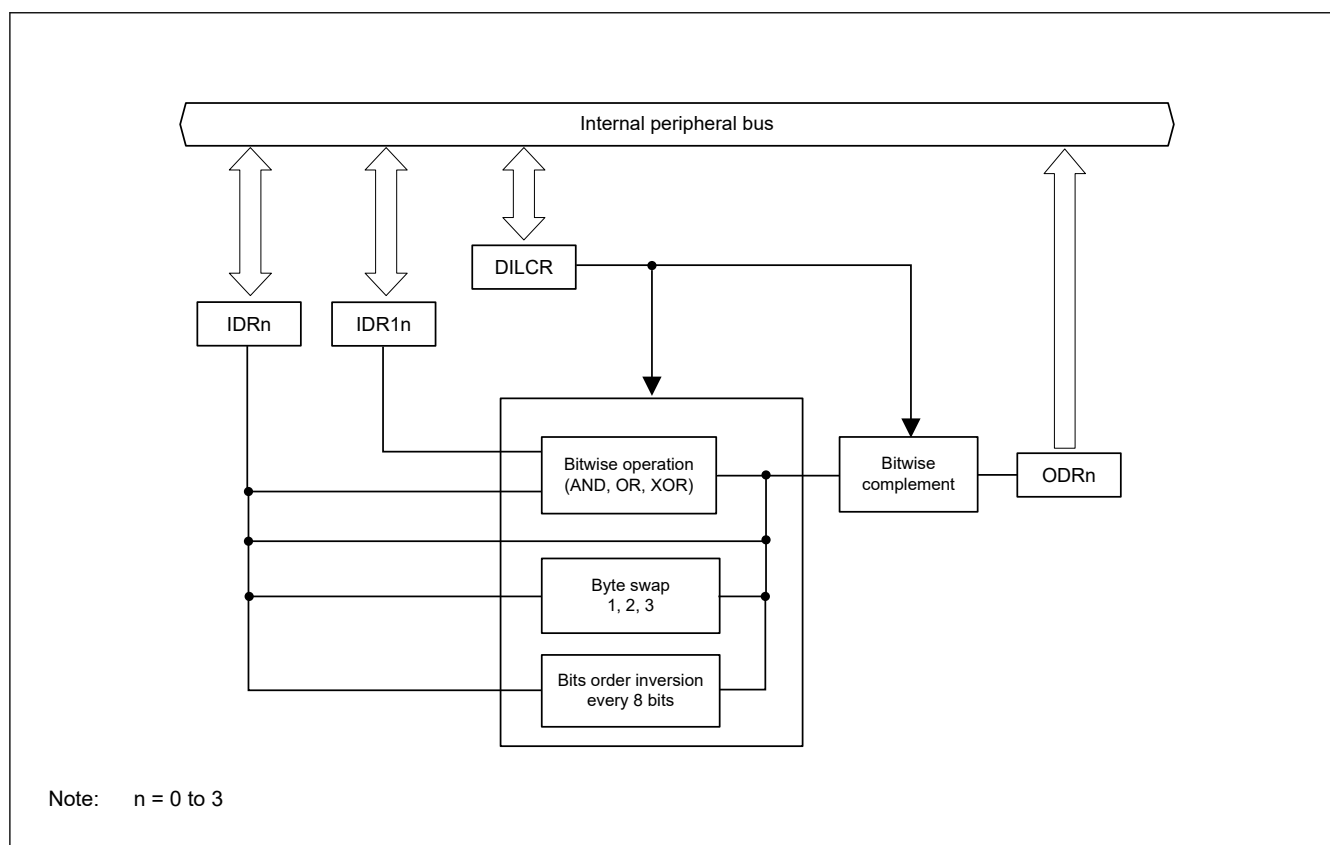


Figure 39.1 DIL block diagram

39.2 Register Descriptions

39.2.1 IDRn : DIL Input Data Registers n (n = 0 to 3)

Base address: DIL = 0x4008_4500

Offset address: 0x00 + 0x04 × n

Bit position: 31 0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	When the AND operation, OR operation, or XOR operation is selected, the first operand is stored. When byte swap or bit order inversion every 8 bits is selected, the data to be converted is stored.	R/W

39.2.2 IDR1n : DIL Input Data Registers 1n (n = 0 to 3)

Base address: DIL = 0x4008_4500

Offset address: 0x20 + 0x04 × n

Bit position: 31 0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	When the AND operation, OR operation, or XOR operation is selected, the second operand is stored.	R/W

39.2.3 ODRn : DIL Output Data Registers n (n = 0 to 3)

Base address: DIL = 0x4008_4500

Offset address: 0x10 + 0x04 × n

Bit position: 31 0

Bit field:

Value after reset: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	n/a	The operation result or conversion result that is selected by the DILCR register is stored.	R

39.2.4 DILCR : DIL Control Register

Base address: DIL = 0x4008_4500

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	DILMODE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DILEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DILEN	Bit Inversion Function Enable 0: Bit inversion function enabled 1: Bit inversion function disabled	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
19:16	DILMODE[3:0]	Data Conversion Mode Select 0x0: No data conversion 0x1: AND operation 0x2: OR operation 0x3: XOR operation 0x8: Byte swap 1 0x9: Byte swap 2 0xA: Byte swap 3 0xB: Bit order inversion every 8 bits Others: Setting prohibited	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

DILEN bit (Bit Inversion Function Enable)

The DILEN bit enables or disables the bit inversion function.

DILMODE[3:0] bits (Data Conversion Mode Select)

The DILMODE[3:0] bits set the data conversion mode. If you selected the AND operation, OR operation, or XOR operation, the result of the logical operation of the IDR_n and IDR1_n registers are stored in the ODR_n register. If you selected byte swap or bit order inversion every 8 bits, the value of the IDR_n register is converted and stored in the ODR_n register. [Figure 39.2](#) shows the data alignment for byte swap and bit order inversion every 8 bits.

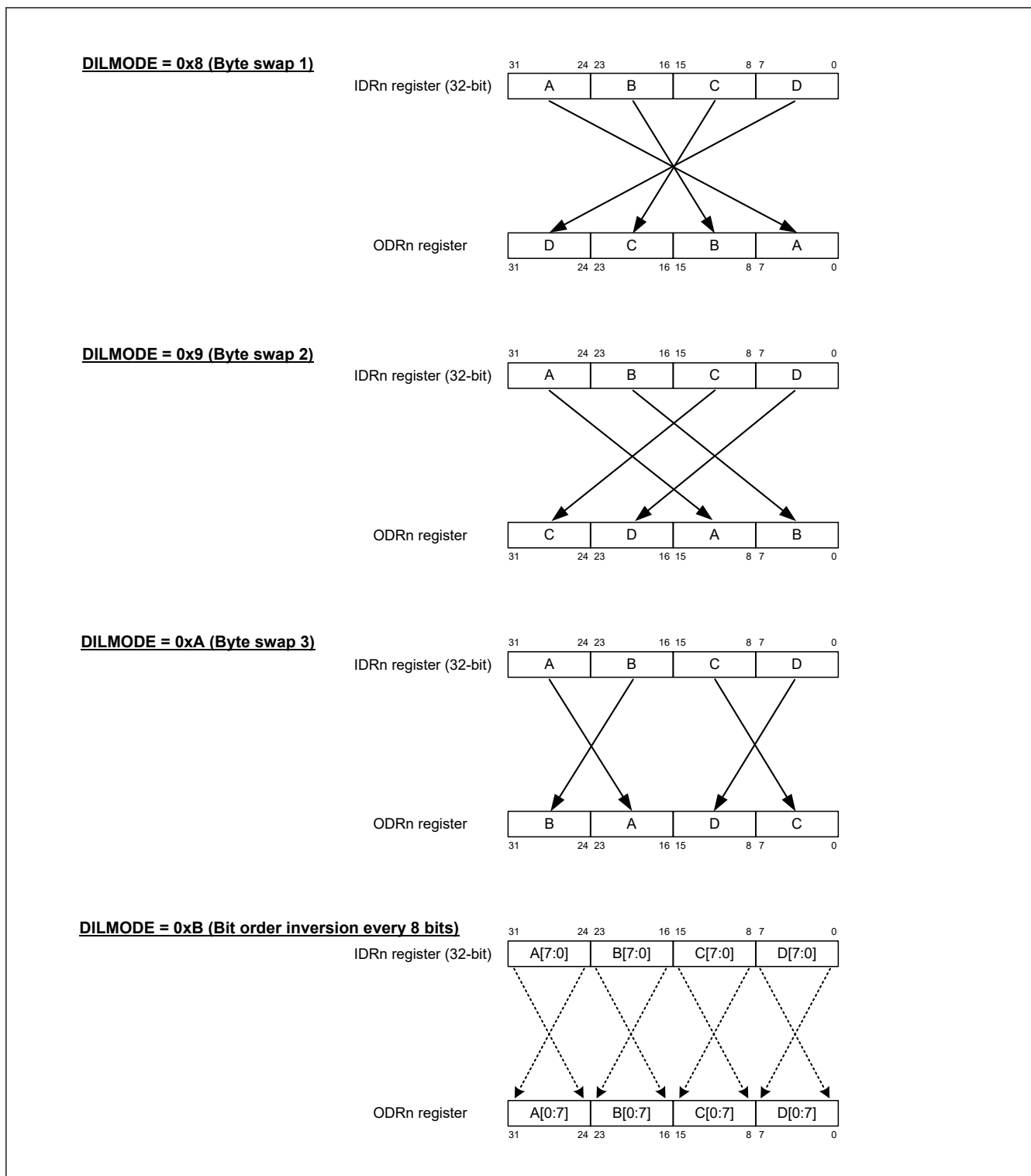


Figure 39.2 Data alignment for byte swap and bit order inversion every 8 bits

39.3 Operation

39.3.1 Basic Operation

DIL starts an operation when the module-stop state is canceled. When data to be converted is input from memory to the DIL Input Data Register (IDRn and IDR1n) after the module-stop state is canceled, a conversion value is generated. The value is stored in the DIL Output Data Register (ODRn).

Table 39.2 shows the relationship between the IDRn, IDR1n, and ODRn registers for each conversion mode.

Table 39.2 Conversion operation

Function	Conversion mode	Operation	
	DILCR.DILMODE[3:0]	DILCR.DILEN = 0	DILCR.DILEN = 1
AND	0x1	$ODR_n = \sim(IDR_n \& IDR_{1n})$	$ODR_n = IDR_n \& IDR_{1n}$
OR	0x2	$ODR_n = \sim(IDR_n IDR_{1n})$	$ODR_n = IDR_n IDR_{1n}$
XOR	0x3	$ODR_n = \sim(IDR_n \wedge IDR_{1n})$	$ODR_n = IDR_n \wedge IDR_{1n}$
No conversion	0x0	$ODR_n = \sim IDR_n$	$ODR_n = IDR_n$
Byte swap 1	0x8	$ODR_0 = \sim IDR_3$ $ODR_1 = \sim IDR_2$ $ODR_2 = \sim IDR_1$ $ODR_3 = \sim IDR_0$	$ODR_0 = IDR_3$ $ODR_1 = IDR_2$ $ODR_2 = IDR_1$ $ODR_3 = IDR_0$
Byte swap 2	0x9	$ODR_0 = \sim IDR_2$ $ODR_1 = \sim IDR_3$ $ODR_2 = \sim IDR_0$ $ODR_3 = \sim IDR_1$	$ODR_0 = IDR_2$ $ODR_1 = IDR_3$ $ODR_2 = IDR_0$ $ODR_3 = IDR_1$
Byte swap 3	0xA	$ODR_0 = \sim IDR_1$ $ODR_1 = \sim IDR_0$ $ODR_2 = \sim IDR_3$ $ODR_3 = \sim IDR_2$	$ODR_0 = IDR_1$ $ODR_1 = IDR_0$ $ODR_2 = IDR_3$ $ODR_3 = IDR_2$
Bit order inversion every 8 bits	0xB	$ODR_n[31:24] = \sim IDR_n[24:31]$ $ODR_n[23:16] = \sim IDR_n[16:23]$ $ODR_n[15:8] = \sim IDR_n[8:15]$ $ODR_n[7:0] = \sim IDR_n[0:7]$	$ODR_n[31:24] = IDR_n[24:31]$ $ODR_n[23:16] = IDR_n[16:23]$ $ODR_n[15:8] = IDR_n[8:15]$ $ODR_n[7:0] = IDR_n[0:7]$

39.4 Usage Notes

39.4.1 Settings for the Module-Stop Function

The Module Stop Control Register C (MSTPCRC) can enable or disable the DIL operation. The data conversion circuit is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

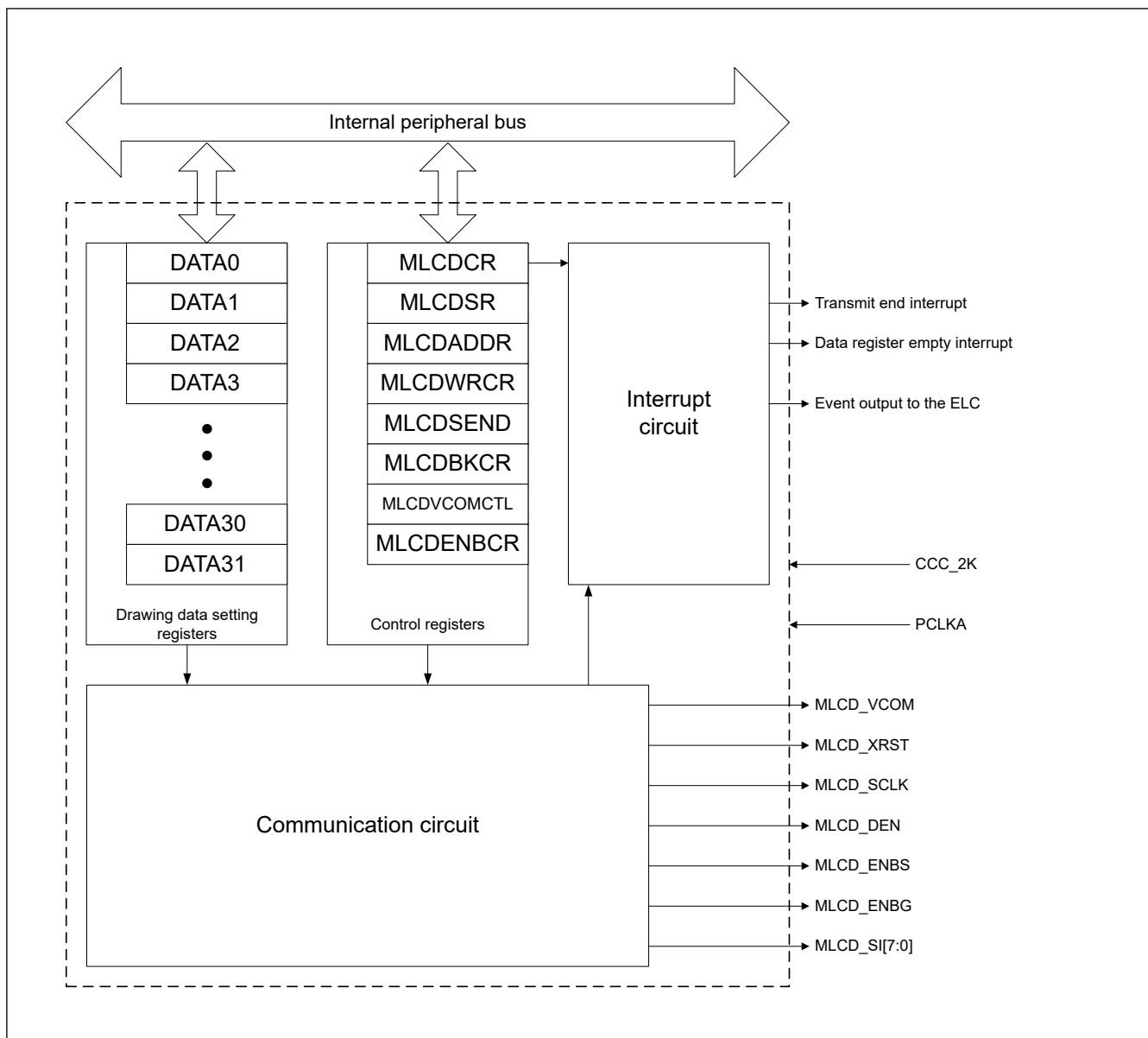
40. MIP LCD Controller (MLCD)

40.1 Overview

The MIP LCD controller (MLCD) is a circuit that controls parallel interface type MIP LCD panels. [Table 40.1](#) lists MLCD specifications and [Figure 40.1](#) shows MLCD block diagram. [Table 40.2](#) lists MLCD I/O pins.

Table 40.1 MLCD specifications

Parameter	Specifications
Communication method	Parallel interface supporting the MIP LCD panel
Functions	- Continuous rewriting of specified rectangular area in 8-bit units - Continuous rewriting enabled for the range specified with the number of horizontal bytes and the number of vertical lines - Vertical address increment and decrement setting enabled - Horizontal address increment and decrement setting enabled - Block transfer rewriting in 16-bit × 16-bit or 8-bit × 8-bit units (setting of the number of horizontal and vertical blocks and scanning method enabled) - Endian conversion function (MSB/LSB first switchable for 8-bit data)
Clocks	- PCLKA (bus clock) - VCLK (for generating the MLCD_VCOM signal. 2.048 kHz (CCC_2K clock))
Interrupts	- Transmit end interrupt (MLCD_TEI) - Data register empty interrupt (MLCD_TEMI)
Event link function (output)	- Two event signals can be output to the ELC. - Transmission end (MLCD_ELCTEND) - Data register empty (MLCD_ELCTEM)
Functions for reducing power consumption	The module-stop state can be set.

**Figure 40.1 MLCD block diagram****Table 40.2 MLCD I/O pins**

Pin name	I/O	Function
MLCD_VCOM	Output	Common electrode polarity signal pin
MLCD_XRST	Output	LCD control output pin
MLCD_SCLK	Output	Serial output clock pin for communications
MLCD_DEN	Output	Data identification signal pin
MLCD_ENBS	Output	Horizontal data enable pin
MLCD_ENBG	Output	Vertical data enable pin
MLCD_SI[7:0]	Output	Image data signal pin

40.2 Register Descriptions

40.2.1 MLCDCR : MLCD Control Register

Base address: MLCD = 0x4007_0200

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	VADDRDEC	HADDRDEC	—	—	—	—	SCKCR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	BITSW	—	—	—	XRST	—	TEMIE	TEIE	—	—	—	—	TE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TE	MLCD Transmit Enable 0: Disable data transmission 1: Enable data transmission	R/W
4:1	—	These bits are read as 0. The write value should be 0.	R/W
5	TEIE	Transmit End Interrupt Enable 0: Disable MLCD_TEI interrupt requests 1: Enable MLCD_TEI interrupt requests	R/W
6	TEMIE	Data Register Empty Interrupt Enable 0: Disable MLCD_TEMI interrupt requests 1: Enable MLCD_TEMI interrupt requests	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	XRST	Liquid Crystal Display 0: Disable MIP LCD panel display 1: Enable MIP LCD panel display	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	BITSW	Endian Conversion 0: MSB first 1: LSB first	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
23:16	SCKCR[7:0]	Transmit Clock Control* ¹ These bits specify the width at high level of the transmit clock. A width of 1 to 255 cycles (of PCLKA) can be set depending on the setting value. Setting 0 is prohibited.	R/W
27:24	—	These bits are read as 0. The write value should be 0.	R/W
28	HADDRDEC	Horizontal Address Control* ² This bit specifies the automatic change method of the horizontal direction address. 0: The address is incremented 1: The address is decremented	R/W
29	VADDRDEC	Vertical Address Control This bit specifies the automatic change method of the vertical direction address. 0: The address is incremented 1: The address is decremented	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the SCKCR[7:0] is set to 0x01, the width at high level of the first pulse is one cycle, and that of the second or more pulse is two cycle. For details, see [section 51, Electrical Characteristics](#), MLCD timing.

Note 2. In 8 × 8 block transfer mode, the setting of this bit does not affect the horizontal address update operation.

TE bit (MLCD Transmit Enable)

The TE bit enables or disables the data transmit operation. When the transmit data is written in the MLCD drawing data setting register after setting the TE bit to 1, it is sent. Note that the SCKCR[7:0] bits must be set before the TE bit is set to 1. It is automatically cleared to 0 after the transfer is completed.

TEIE bit (Transmit End Interrupt Enable)

The TEIE bit enables or disables transmit end interrupt requests.

TEMIE bit (Data Register Empty Interrupt Enable)

The TEMIE bit enables or disables data register empty interrupt requests.

XRST bit (Liquid Crystal Display)

The XRST bit enables or disables the MIP LCD panel display. After setting the MLCDVCOMCTL.VCOME bit, wait for 1 ms or more before executing.

BITSW bit (Endian Conversion)

The BITSW bit specifies the bit arrangement used when transmitting data to MSB or LSB first. When BITSW = 0 of MSB transfer, the DATAm[7:0] bits are transmitted to the MLCD_SI[7:0]. When BITSW = 1 of LSB transfer, the DATAm[0:7] bits are transmitted to the MLCD_SI[7:0]. For the DATA_HWn and DATA_Wk registers, this bit also specifies the bit arrangement in units of eight bits.

SCKCR[7:0] bits (Transmit Clock Control)

The SCKCR[7:0] bits specify the width at high level of the transmit clock. Set these bits before the TE bit is set to 1. Set them 1 μ s or more.

HADDRDEC bit (Horizontal Address Control)

The HADDRDEC bit sets whether the horizontal direction address is incremented or decremented when updated. Controlling this bit and the BITSW bit allows horizontally flipping the image data in the memory and displaying it on an MIP LCD panel. When the setting of the HADDRDEC register is 0, transfer starts with the first data stored in the MLCD drawing data setting register as shown in Figure 40.2. When the setting of the HADDRDEC register is 1, transfer starts with the last data stored in the MLCD drawing data setting register as shown in Figure 40.3.

VADDRDEC bit (Vertical Address Control)

The VADDRDEC bit sets whether the vertical direction address is incremented or decremented when updated. Controlling this bit allows vertically flipping the image data in the memory and displaying it on the MIP LCD panel.

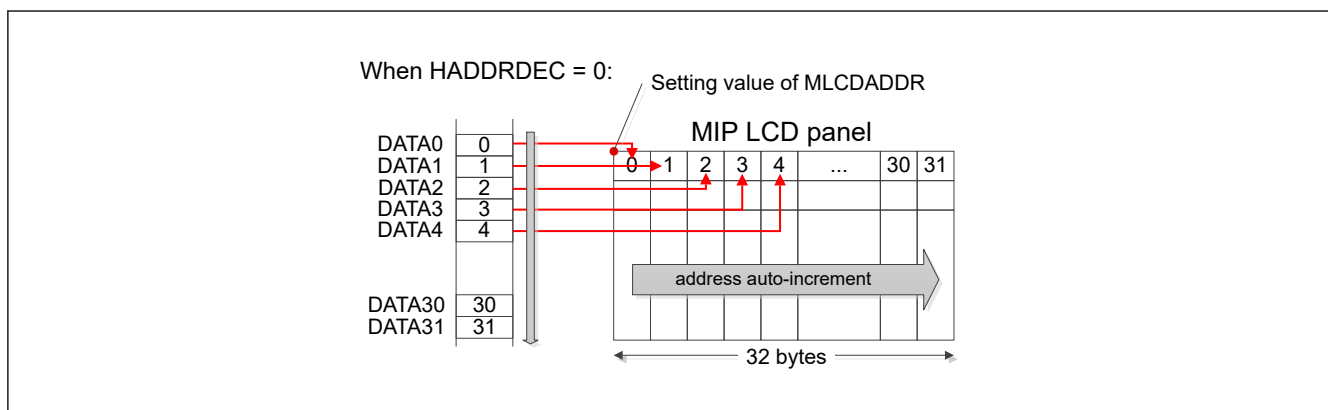


Figure 40.2 Transfer operation when HADDRDEC = 0

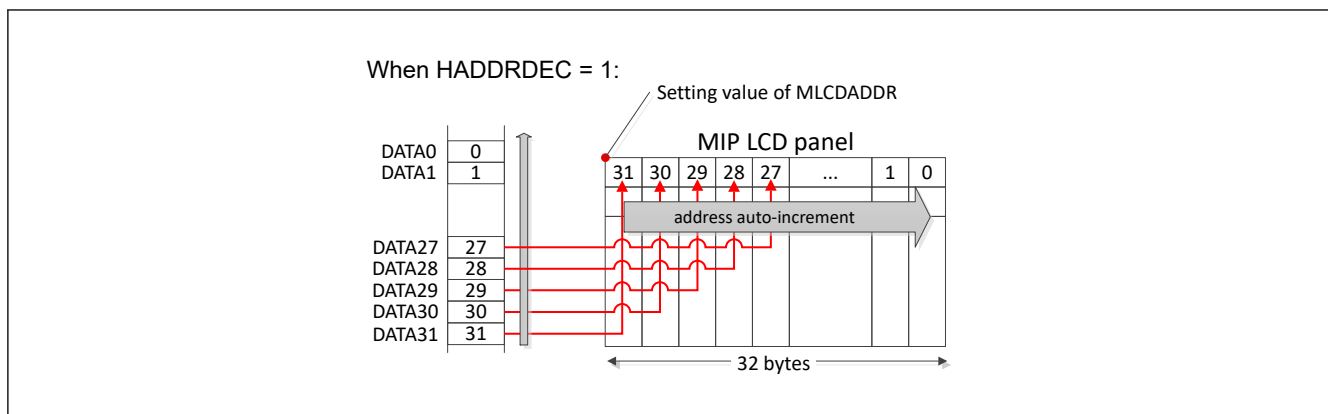


Figure 40.3 Transfer operation when HADDRDEC = 1

40.2.2 MLCDSR : MLCD Status Register

Base address: MLCD = 0x4007_0200

Offset address: 0x04

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

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

Bit	Symbol	Function	R/W
0	TEND	Transmission Flag 0: Data is being transmitted. 1: Data has been transmitted (transmission stopped).	R
1	TEF	Empty Flag 0: There is data to be transmitted. 1: Data has been transmitted (the MLCD drawing data setting register is empty).	R
31:2	—	These bits are read as 0.	R

TEND flag (Transmission Flag)

The TEND flag indicates completion of transmission.

[Setting condition]

- Completing the transmission of data that is set by the MLCDWRCR register for the basic or same image transmission
- Completing the transmission of data that is set by the MLCDBKCR register for the block transmission

[Clearing conditions]

- Writing of data for transmission to the MLCD drawing data setting register while the MLCDCR.TE bit is 1
- Writing of 1 to the MLCDCR.TE bit while the setting of the same image transmit setting bit (MLCDSEND.EN) is set to 1

TEF flag (Empty Flag)

The TEF flag indicates that the MLCD drawing data setting register is emptied.

[Setting conditions]

- Writing of 1 to the MLCDCR.TE bit

- When the MLCD drawing data setting register is emptied

[Clearing condition]

- Writing of data for transmission to the MLCD drawing data setting register while the MLCD_CR.TE bit is 1

40.2.3 MLCDADDR : MLCD Address Setting Register

Base address: MLCD = 0x4007_0200

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	VADDR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	HADDR[4:0]					—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
7:3	HADDR[4:0]	Horizontal Address Setting These bits specify the address in the horizontal direction.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	VADDR[7:0]	Vertical Address Setting These bits specify the address in the vertical direction.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

HADDR[4:0] bits (Horizontal Address Setting)

The HADDR[4:0] bits specify the address in the horizontal direction in units of bytes. Set it before writing to the MLCD drawing data setting register.

When the automatic change method for the horizontal address is address decrement (MLCD_CR.HADDRDEC = 1), set the Horizontal Address Setting bit to the following.

$$\text{HADDR}[4:0] = (\text{horizontal address of the first transmission data}) - (\text{horizontal direction size setting bit}) + 1$$

For example, when a 3-byte image in the horizontal direction size is placed at 0x00 to 0x02 of the horizontal direction address, set the following:

$$\text{HADDR}[4:0] = 0x02 - 0x03 + 1 = 0x00$$

VADDR[7:0] bits (Vertical Address Setting)

The VADDR[7:0] bits specify the address in the vertical direction. Set it before writing to the MLCD drawing data setting register.

Figure 40.4 shows the coordinate image of the LCD.

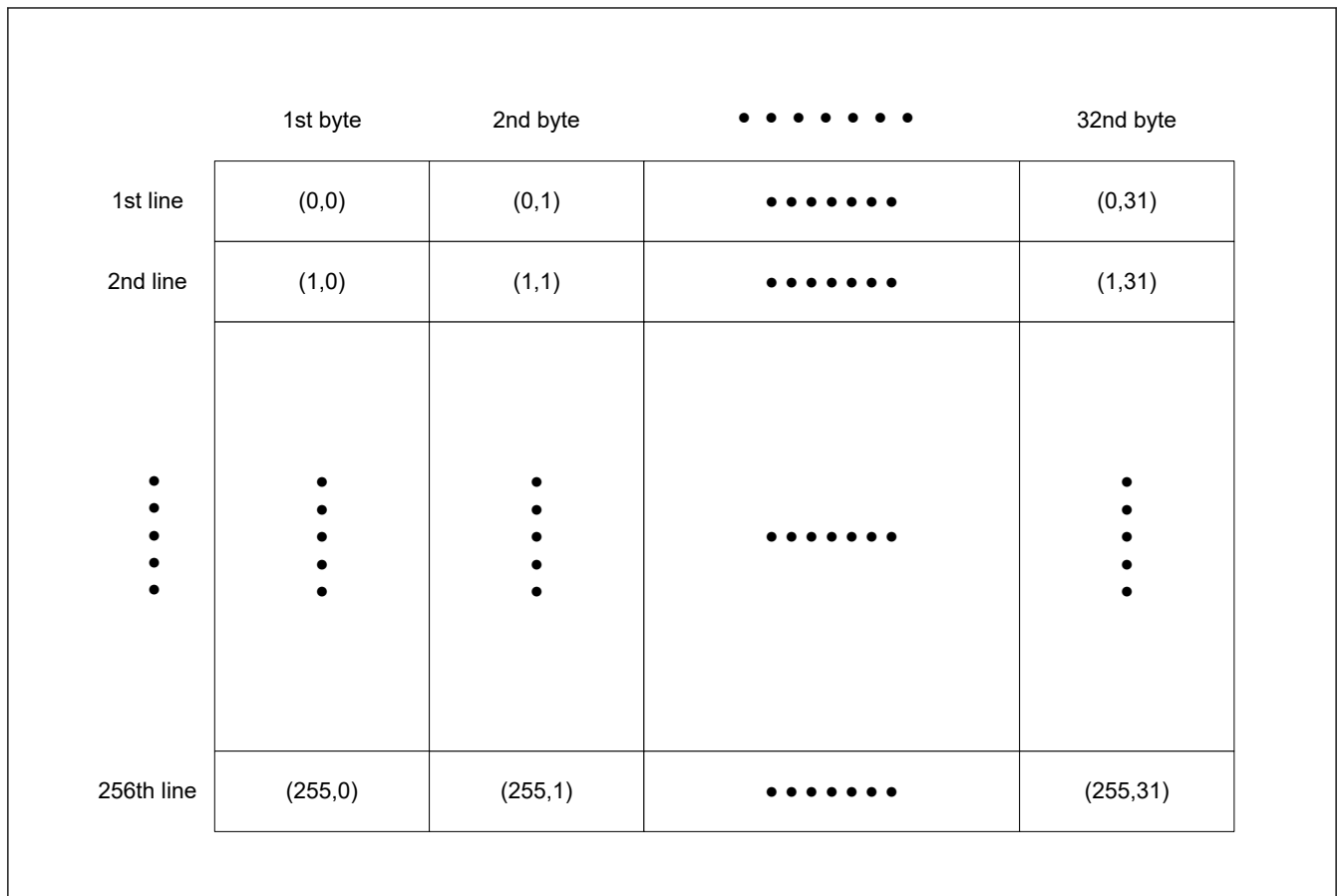


Figure 40.4 Coordinate image of the MIP LCD panel

40.2.4 MLCDWRCR : MLCD Transmit Data Size Setting Register

Base address: MLCD = 0x4007_0200

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	LINE[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	BYTE[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
5:0	BYTE[5:0]	Horizontal Size Setting These bits specify the size in the horizontal direction (1 to 32 bytes). The setting range is from 0x01 to 0x20. The other settings are prohibited.	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
24:16	LINE[8:0]	Vertical Size Setting These bits specify the size in the vertical direction (1 to 256 lines). The setting range is from 0x001 to 0x100. The other settings are prohibited.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

BYTE[5:0] bits (Horizontal Size Setting)

The BYTE[5:0] bits specify the size of the transmit data in the horizontal direction. Set it in the range from 1 to 32 bytes. To specify the size to 32 bytes, set BYTE[5:0] = 0x20. Set it before writing to the MLCD drawing data setting register. Note that these bits are valid when the MLCDBKCR.BKEN bit is 0 (block transfer disabled).

LINE[8:0] bits (Vertical Size Setting)

The LINE[8:0] bits specify the size of the transmit data in the vertical direction. Set it in the range from 1 to 256 lines. To specify the size to 256 lines, set LINE[8:0] = 0x100. Set it before writing to the MLCD drawing data setting register. Note that these bits are valid when the MLCDBKCR.BKEN bit is 0 (block transfer disabled).

40.2.5 MLCDSEND : MLCD Same Image Transmit Register

Base address: MLCD = 0x4007_0200

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EN	Same Image Transmit Setting 0: Disable the same image transmit function.*1 1: Enable the same image transmit function.*1	W
31:1	—	The write value should be 0.	W

Note 1. Can be rewritten when the TEND flag in the MLCD Status Register (MLCDSR) is 1 (transmission stopped).

EN bit (Same Image Transmit Setting)

The EN bit allows continuous transmission of image data without updating the values in the MLCD drawing data setting register (DATAm/DATAn_HWn/DATAn_Wk).

40.2.6 MLCDBKCR : MLCD Block Transfer Setting Register

Base address: MLCD = 0x4007_0200

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	BKVNUM[5:0]						—	—	BKHNUM[5:0]					
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	BKMODE[1:0]		—	—	—	—	—	—	BKSIZ E	BKEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BKEN	Block Transfer Enable 0: Disable block transfer 1: Enable block transfer.	R/W

Bit	Symbol	Function	R/W
1	BKSIZE	Block Transfer Size 0: 16 × 16 bit 1: 8 × 8 bit	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	BKMODE[1:0]	Block Transfer Mode Setting 0 0: Transfer mode 1 0 1: Transfer mode 2 1 0: Transfer mode 3 1 1: Transfer mode 4	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
21:16	BKHNUM[5:0]	Horizontal Transfer Count Setting These bits specify the number of blocks in the horizontal direction (1 to 32 blocks) for transfer. The setting range is from 0x01 to 0x20. The other settings are prohibited.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
29:24	BKVNUM[5:0]	Vertical Transfer Count Setting These bits specify the number of blocks in the vertical direction (1 to 32 blocks) for transfer. The setting range is from 0x01 to 0x20. The other settings are prohibited.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

BKEN bit (Block Transfer Enable)

The BKEN bit enables or disables block transfer. One block is 8-bit × 8-bit or 16-bit × 16-bit image data.

BKSIZE bit (Block Transfer Size)

The BKSIZE bit sets the block transfer size. Set this bit before the transfer count of the horizontal direction and that of the vertical direction are set. During the MLCD transfer, do not change this bit.

BKMODE[1:0] bits (Block Transfer Mode Setting)

The BKMODE[1:0] specify block transfer mode of the MIP LCD panel. [Figure 40.5](#) shows the case where both the BKHNUM[5:0] and BKVNUM[5:0] bits set the number of blocks to 3. Before setting this register, set the origin coordinates by using the MLCD Address Setting Register (MLCDADDR).

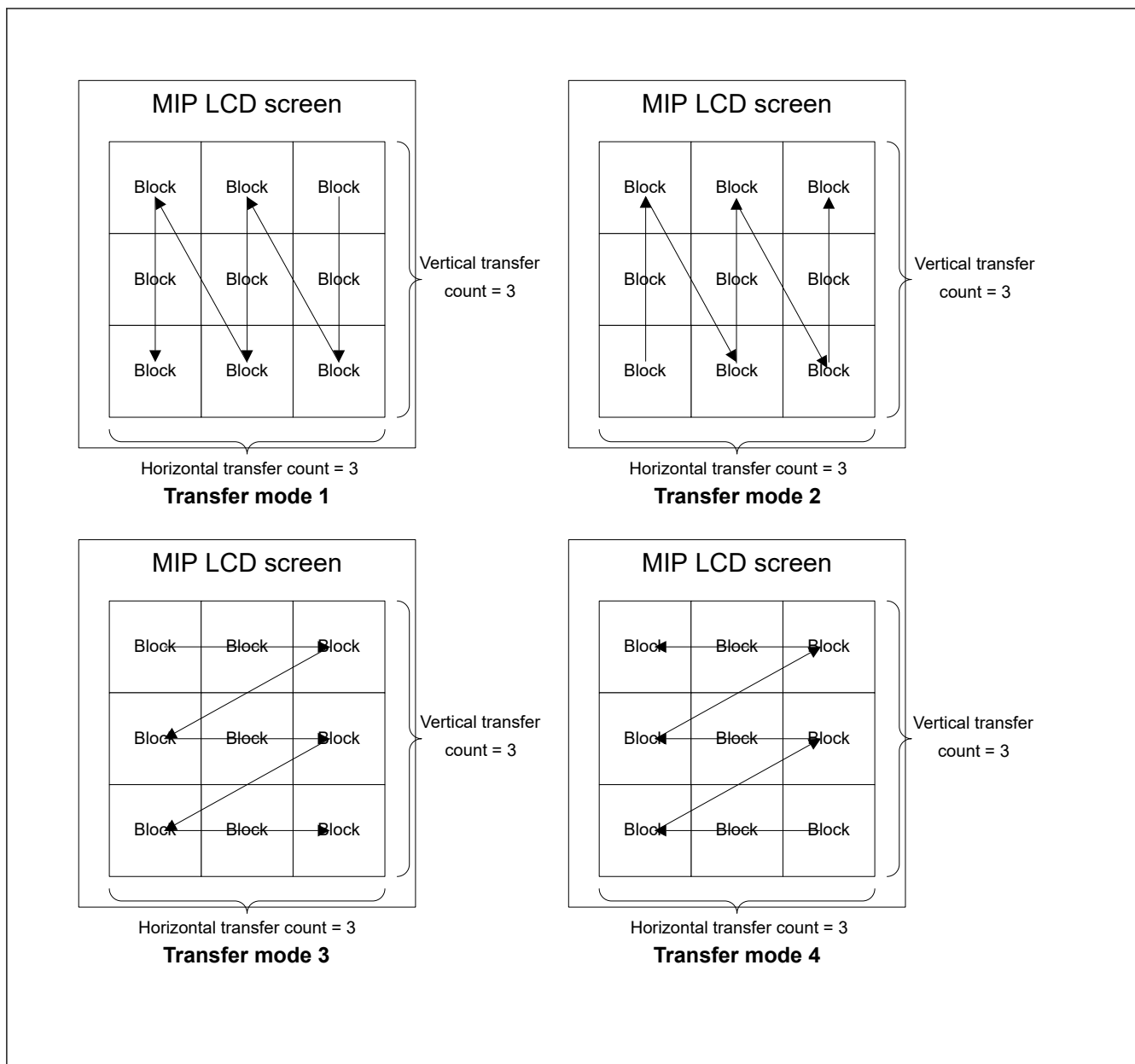


Figure 40.5 Block transfer modes (when the number of blocks in the horizontal direction for transfer = 3 and the number of blocks in the vertical direction for transfer = 3)

BKHNUM[5:0] bits (Horizontal Transfer Count Setting)

The BKHNUM[5:0] bits specify the number of blocks in the horizontal direction for transfer for the MIP LCD panel. When BKSIZ = 1 (8×8 block transfer), set it in the range from 1 to 32 blocks. When BKSIZ = 0 (16×16 block transfer), set it in the range from 1 to 16 blocks. Set this bit before setting the MLCD drawing data setting register.

BKVNUM[5:0] bits (Vertical Transfer Count Setting)

The BKVNUM[5:0] bits specify the number of blocks in the vertical direction for transfer for the MIP LCD panel. When BKSIZ = 1 (8×8 block transfer), set it in the range from 1 to 32 blocks. When BKSIZ = 0 (16×16 block transfer), set it in the range from 1 to 16 blocks. If the setting is out of the range, then the operation is not guaranteed. Set this bit before setting the MLCD drawing data setting register.

40.2.7 MLCDVCOMCTL : MLCD VCOM Control Register

Base address: MLCD = 0x4007_0200

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BMASK[7:0]								FMASK[7:0]							
Value after reset:	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	HWMS KF	HWMS KEN	—	—	—	—	—	—	VCOMW[1:0]	VCOM E	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VCOME	VCOM Output Enable 0: Disable MLCD_VCOM output. 1: Enable MLCD_VCOM output.	R/W
2:1	VCOMW[1:0]	VCOM Output High-Width Setting* ¹ These bits specify the high-level width (t_{cVCOM}) of the MLCD_VCOM output signal. 0 0: 1000 ms 0 1: 500 ms 1 0: 2000 ms 1 1: 5000 ms	R/W
8:3	—	These bits are read as 0. The write value should be 0.	R/W
9	HWMSKEN	HW VCOM Mask* ³ 0: Mask function in transmission disabled 1: Mask function in transmission enabled	R/W
10	HWMSKF	HW Mask This bit shows the HW mask status. 0: HW mask is stopped 1: HW mask is in proceed	R
15:11	—	These bits are read as 0. The write value should be 0.	R/W
23:16	FMASK[7:0]	VCOM Hardware Mask Range Bits Before Data Transmission* ² 0x00 to 0x09: Settings prohibited 0x0A to 0xFF: Mask time before the VCOM signal edge (10 to 255: number of PCLKA cycles)	R/W
31:24	BMASK[7:0]	VCOM Hardware Mask Range Bits After Data Transmission* ² 0x00 to 0x03: Settings prohibited 0x04 to 0xFF: Mask time after the VCOM signal edge (4 to 255: number of PCLKA cycles)	R/W

Note 1. Normally, the high and low widths of the MLCD_VCOM output signal are the same, but when the HW mask is enabled, the duty ratio of MLCD_VCOM may change. Set the VCOMW[1:0] bits when VCOME = 0. Set the VCOME bit before MLCD is transmitted. In the middle of the MLCD transmission, the change of the VCOMW[1:0] bits and the VCOME bit are prohibited.

Note 2. Do not change the setting of this bit when HWMSKEN = 1.

Note 3. When using the VCOM HW mask, set PCLKA to 8 kHz or higher.

VCOME bit (VCOM Output Enable)

The VCOME bit enables or disables MLCD_VCOM output. When the MLCD_CR.TE bit is 0, set the VCOME bit.

VCOMW[1:0] bits (VCOM Output High-Width Setting)

The VCOMW[1:0] bits specify the width at high level (t_{cVCOM}) of the MLCD_VCOM output signal. For details, see [section 51.3.13. MLCD Timing](#) in [section 51, Electrical Characteristics](#)

HWMSKEN bit (HW VCOM Mask)

The HWMSKEN bit is the VCOM mask control bit by hardware. When this bit is 1, VCOM retains the original output state during BMASK and FMASK register setting period before and after MLCD transfer, and during MLCD data transfer. Set this bit after setting VCOME before enabling MLCDE. HWMSKEN setting is prohibited while HWMSKF is 1.

HWMSKF flag (HW Mask)

The HWMSKF flag is a hardware mask status flag. When HWMSKEN is enabled, HWMSKF is automatically set to 1 after MLCDE is enabled. It automatically becomes 0 after the hardware mask ends. HWMSKEN setting is prohibited while HWMSKF is 1.

FMASK[7:0] bits (VCOM Hardware Mask Range Bits Before Data Transmission)

The FMASK[7:0] bits set the VCOM hardware mask time range before data transmission. Due to limitations of the MIP LCD specification, drawing data cannot be transmitted before the VCOM signal change point. By setting the FMASK[7:0] bits, it is possible to mask the drawing data not to be transmitted before the VCOM signal change point. Set this bit after checking MIP LCD panel data sheet. Figure 40.6 shows the mask period.

Set the FMASK[7:0] bits according to the following formula.

Masked period set by FMASK[7:0] bits = (FMASK[7:0] - 1) × 0.488 ms

(Setting range : FMASK[7:0] ≥ 2 + (MLCDENBCR.ENBW[9:0] + (MLCDENBCR.ENBEG[7:0] × 2 + 2) × (2048 ÷ PCLKA frequency))

Note: Round up to the decimal point.

BMASK[7:0] bits (VCOM Hardware Mask Range Bits After Data Transmission)

The BMASK[7:0] bits set the VCOM hardware mask time range after data transmission. Due to limitations of the MIP LCD specification, drawing data cannot be transmitted after the VCOM signal change point. By setting the BMASK[7:0] bits, it is possible to mask the drawing data not to be transmitted after the VCOM signal change point. Set this bit after checking MIP LCD panel data sheet. Figure 40.6 shows the mask period.

Set the BMASK[7:0] bits according to the following formula.

Masked period set by BMASK[7:0] bits = (BMASK[7:0] - 1) × 0.488 ms

(Setting range : BMASK[7:0] ≥ 4)

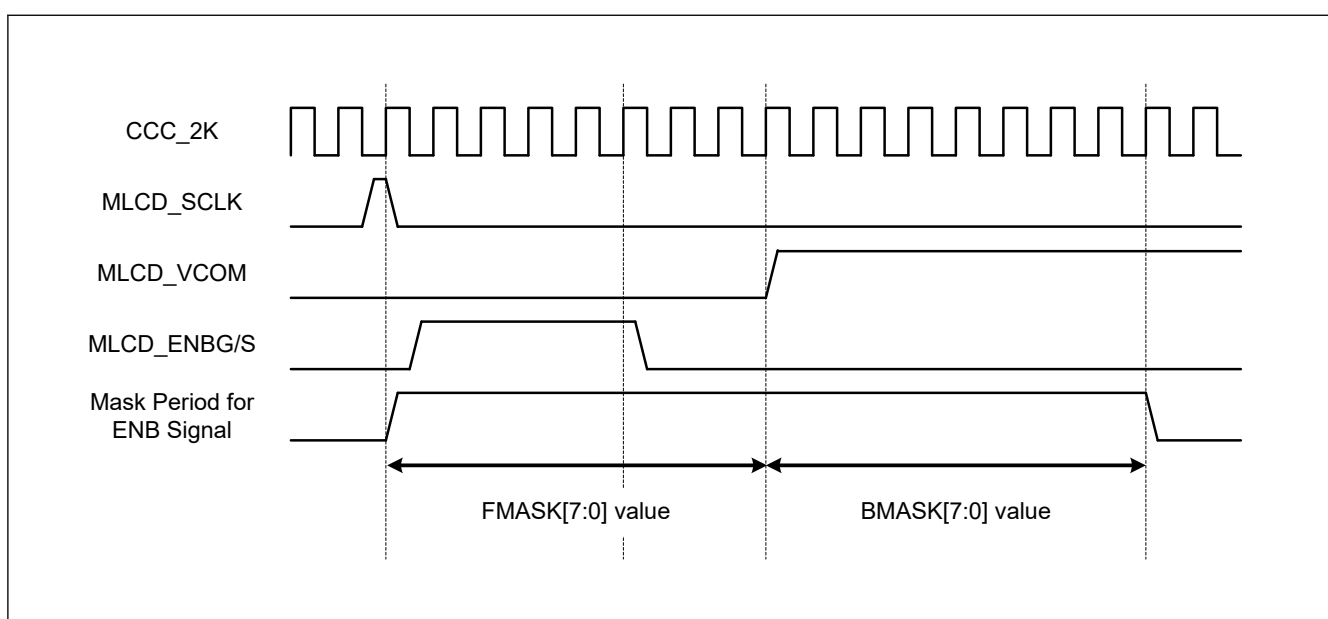


Figure 40.6 Drawing data mask period before and after VCOM signal change

40.2.8 MLCDENBCR : MLCD Enable Signal Control Register

Base address: MLCD = 0x4007_0200

Offset address: 0x1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	ENBEG[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ENBW[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bit	Symbol	Function	R/W
9:0	ENBW[9:0]	High-Width Setting These bits specify the high-level width (t_{wENBH}) of the MLCD_ENBG and MLCD_ENBS signals.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
23:16	ENBEG[7:0]	Timing Setting These bits specify the timing of the MLCD_ENBG and MLCD_ENBS signals.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: Make a setting so that the ENBW[9:0] setting value + (ENBEG[7:0] setting value × 2) ≤ 500 μs.

ENBW[9:0] bits (High-Width Setting)

The ENBW[9:0] bits specify the high-level width (t_{wENBH}) of the horizontal/vertical direction data enable signal. 2 to 1023 cycles (of PCLKA) can be set. Set this bit after checking MIP LCD panel data sheet. Since the low width of the MLCD_SCLK signal changes depending on the setting value of this bit, if this bit is set larger, the data transmission time will increase.

ENBEG[7:0] bits (Timing Setting)

The ENBEG[7:0] bits specify the transition timing of the horizontal/vertical direction data enable signal. 2 to 255 cycles (of PCLKA) can be set. Set this bit after checking MIP LCD panel data sheet. Since the low width of the MLCD_SCLK signal changes depending on the setting value of this bit, if this bit is set larger, the data transmission time will increase.

40.2.9 DATAm/DATa_HWn/DATa_Wk : MLCD Drawing Data Setting Register (m = 0 to 31, n = 0 to 15, k = 0 to 7)

Base address: MLCD = 0x4007_0200

Offset address: 0x20 + 0x01 × m (DATAm)
 0x20 + 0x02 × n (DATa_HWn)
 0x20 + 0x04 × k (DATa_Wk)

Bit position:	31	24	23	16	15	8	7	0
Bit field:								
Value after reset:	0	0	0	0	0	0	0	0

MLCD.DATAm register is the 8-bit R/W register that sets the drawing transmission data.

MLCD.DATa_HWn register is the 16-bit R/W register that sets the drawing transmission data.

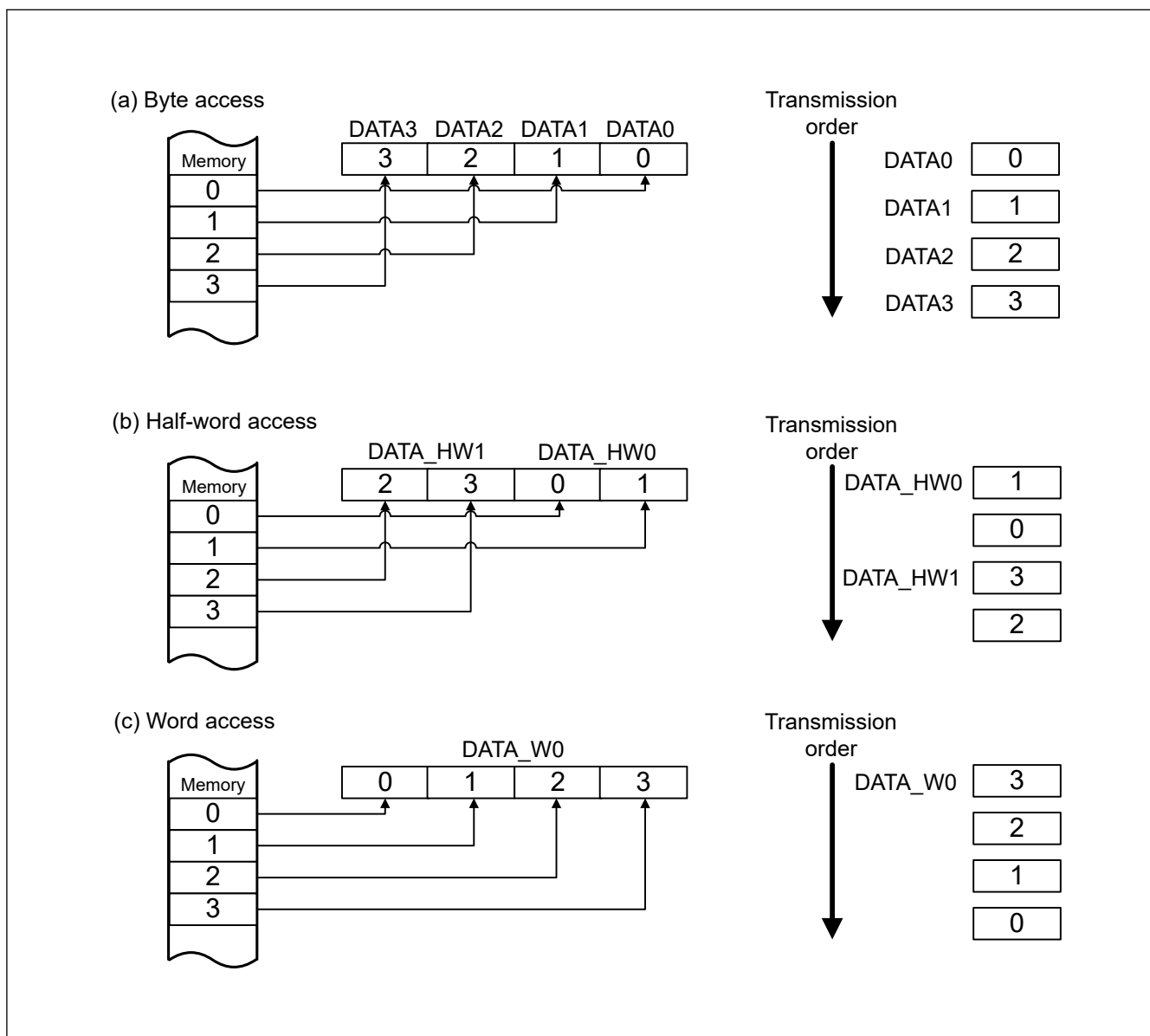
MLCD.DATa_Wk register is the 32-bit R/W register that sets the drawing transmission data.

Table 40.3 shows the allocation of the MLCD Drawing Data Setting Register.

Table 40.3 Allocation of the MLCD Drawing Data Setting Register

Address	Upper 16 bits		Lower 16 bits	
	Upper 8 bits	Lower 8 bits	Upper 8 bits	Lower 8 bits
0x4007_0220	DATA3 (0x4007_0223)	DATA2 (0x4007_0222)	DATA1 (0x4007_0221)	DATA0 (0x4007_0220)
0x4007_0224	DATA7 (0x4007_0227)	DATA6 (0x4007_0226)	DATA5 (0x4007_0225)	DATA4 (0x4007_0224)
⋮				
0x4007_023C	DATA31 (0x4007_023F)	DATA30 (0x4007_023E)	DATA29 (0x4007_023D)	DATA28 (0x4007_023C)

Figure 40.7 shows the relationship between data access width and data storage position.

**Figure 40.7 Relationship between data access width and data storage position**

40.3 Operation

40.3.1 Timing waveform

Figure 40.8 and Figure 40.9 show the relationship between the MLCD register setting value and the timing waveform.

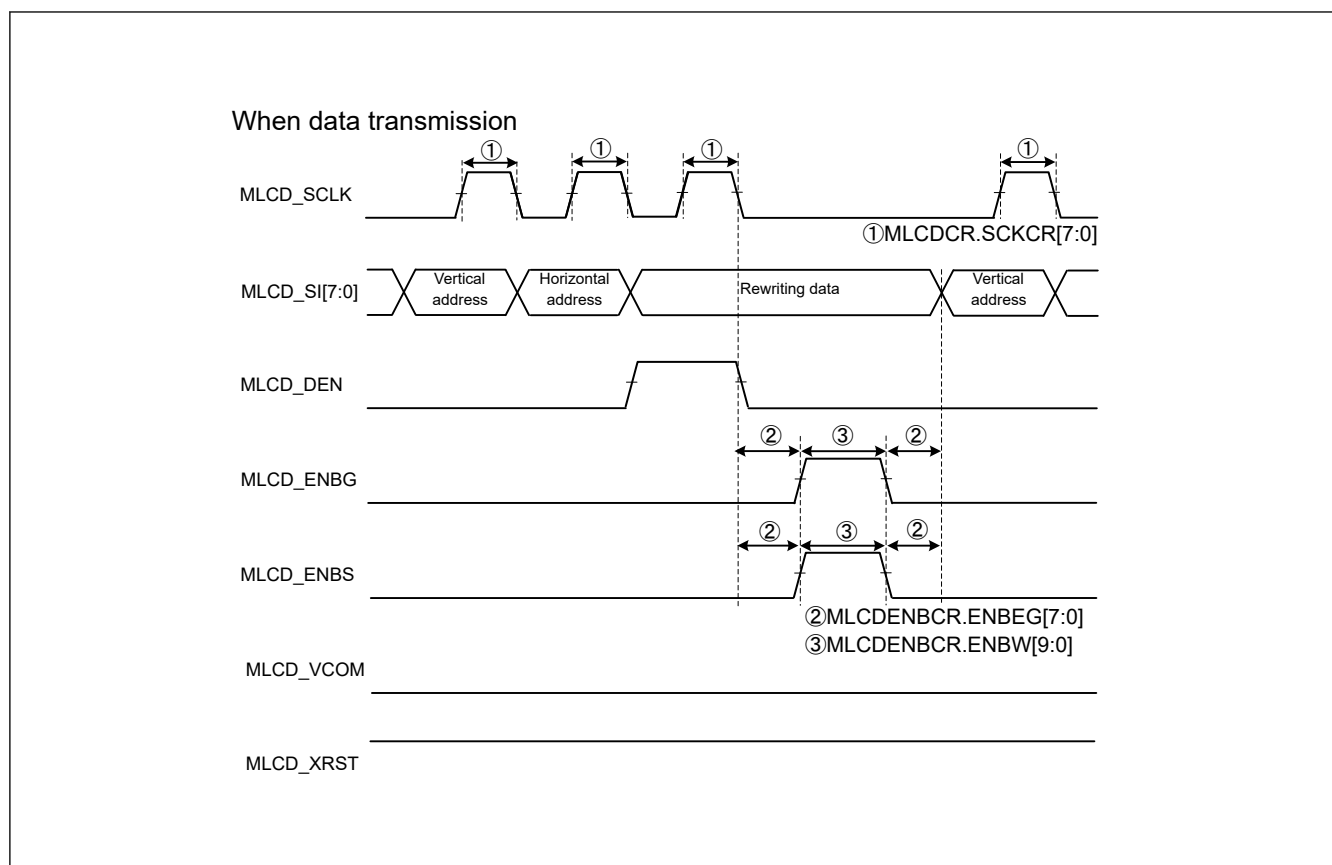


Figure 40.8 Relationship between MLCD register setting value and timing waveform (during data transmission)

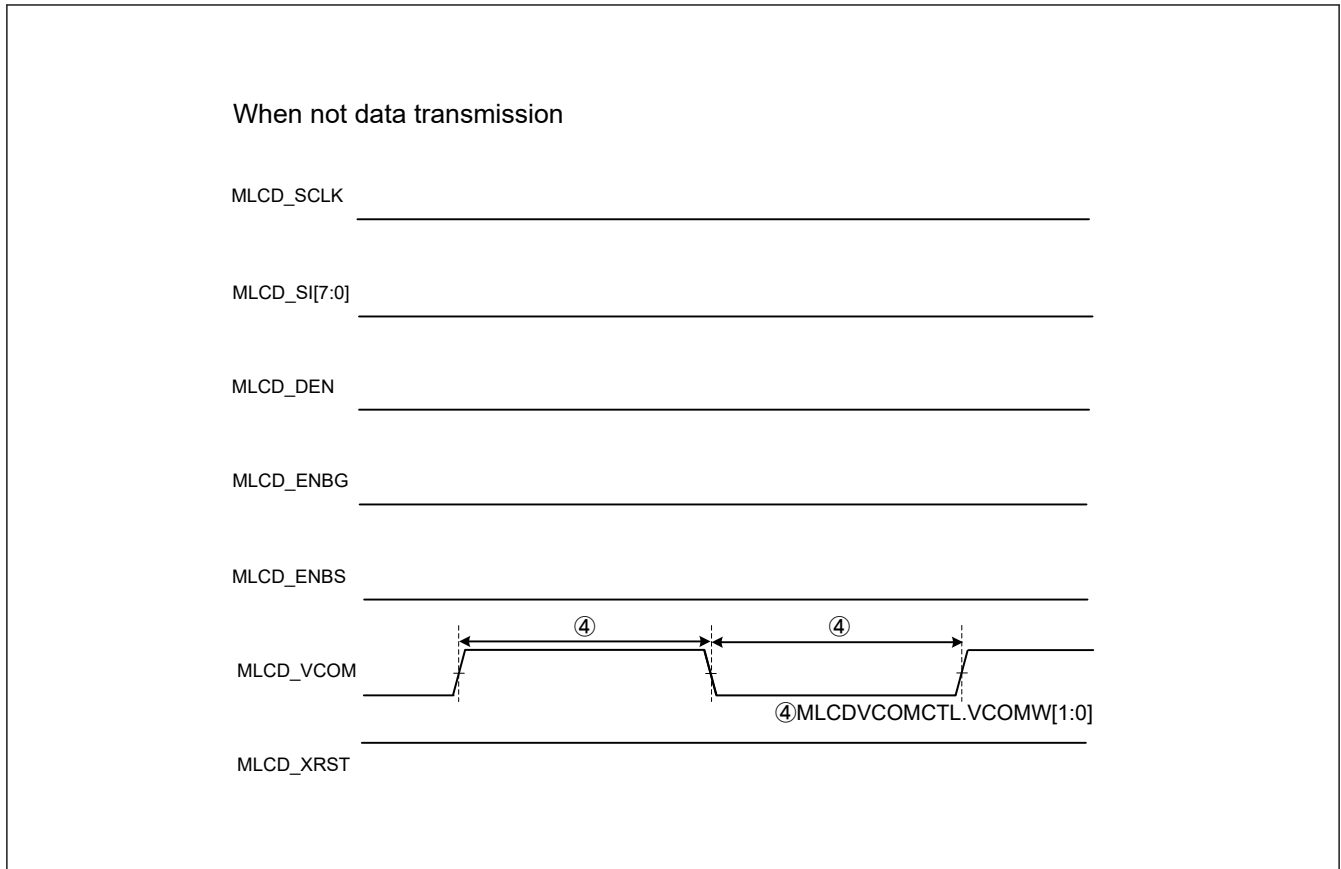


Figure 40.9 Relationship between MLCD register setting value and timing waveform (when data is not transmitted)

40.3.2 Setting and Timing of MLCD_VCOM

The signal output from the MLCD_VCOM pin is a periodic signal (VCOM signal) that can be used for the screen burn-in prevention of the LCD. The period is set by the MLCDVCOMCTL.VCOMW bit. Set the register as follows:

1. Set the period of the output signal by the MLCDVCOMCTL.VCOMW[1:0] bit.
2. Set the MLCDVCOMCTL.VCOME bit to 1 and output the periodic signal (VCOM signal) from the MLCD_VCOM pin. As the limitation of the MIP LCD, because it is necessary to stop the periodic change of the VCOM signal during MLCD data transfer, a function to suppress the signal change exists.

40.3.2.1 Mask Function by Hardware Control

During MLCD data transfer, the periodic change of the VCOM signal can be stopped to maintain the original output state depending on the register setting value. Set the register as follows:

1. Set the change suppression period of the VCOM signal before and after the MLCD data transfer. The suppression period before the MLCD data transfer is set by the MLCDVCOMCTL.FMASK[7:0] bits. In addition, that after the MLCD data transfer is set by the MLCDVCOMCTL.BMASK[7:0] bits.
2. Write 1 to the MLCDVCOMCTL.HWMSKEN bit. This enables the mask control function of the VCOM signal by hardware control. While the HWMSKEN bit is 1, do not change both the FMASK [7: 0] and BMASK [7: 0] bits.
3. Whether or not the change of the VCOM signal are suppressed, you can check the MLCDVCOMCTL.HWMSKF flag.

Figure 40.10 and Figure 40.11 show the example when the mask function by hardware control is enabled (MLCDVCOMCTL.HWMSKEN = 1). Figure 40.10 shows the example of the timing of MLCD data transfer and VCOM output when FMASK = 10 and BMASK = 4. For the period (1), (setting value of the FMASK bit) - 1 is obtained. For the period (2), (setting value of the BMASK bit) - 1 is obtained. The dotted line of the MLCD_VCOM is the operation when the function is disabled.

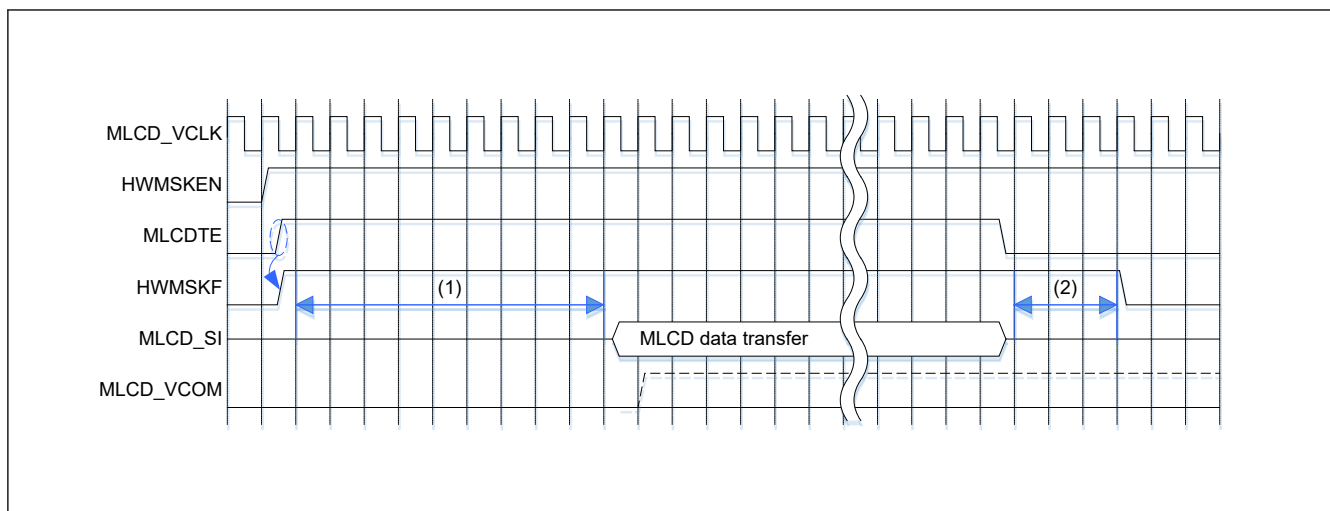


Figure 40.10 MLCD data transfer and VCOM output setting procedure example (part1)

Figure 40.11 shows the example of the timing of resuming from the MLCD data transfer completion when FMASK = 10 and BMASK = 4. After the image transfer is complete, when you transfer a new MLCD data in the BMASK bit setting period (2), you can transfer the data immediately omitting the mask time before VCOM change point.

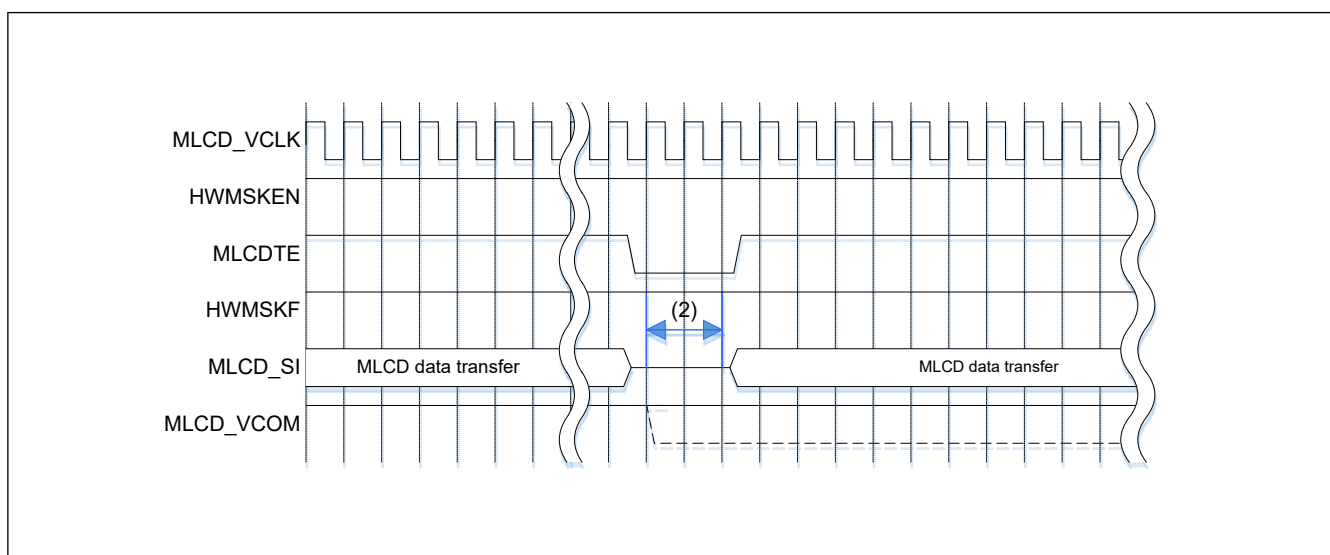


Figure 40.11 MLCD data transfer and VCOM output setting procedure example (part2)

40.3.2.2 Mask function comparison

Table 40.4 shows the differences in operation when the HW mask function is used or not.

Table 40.4 Mask function comparison

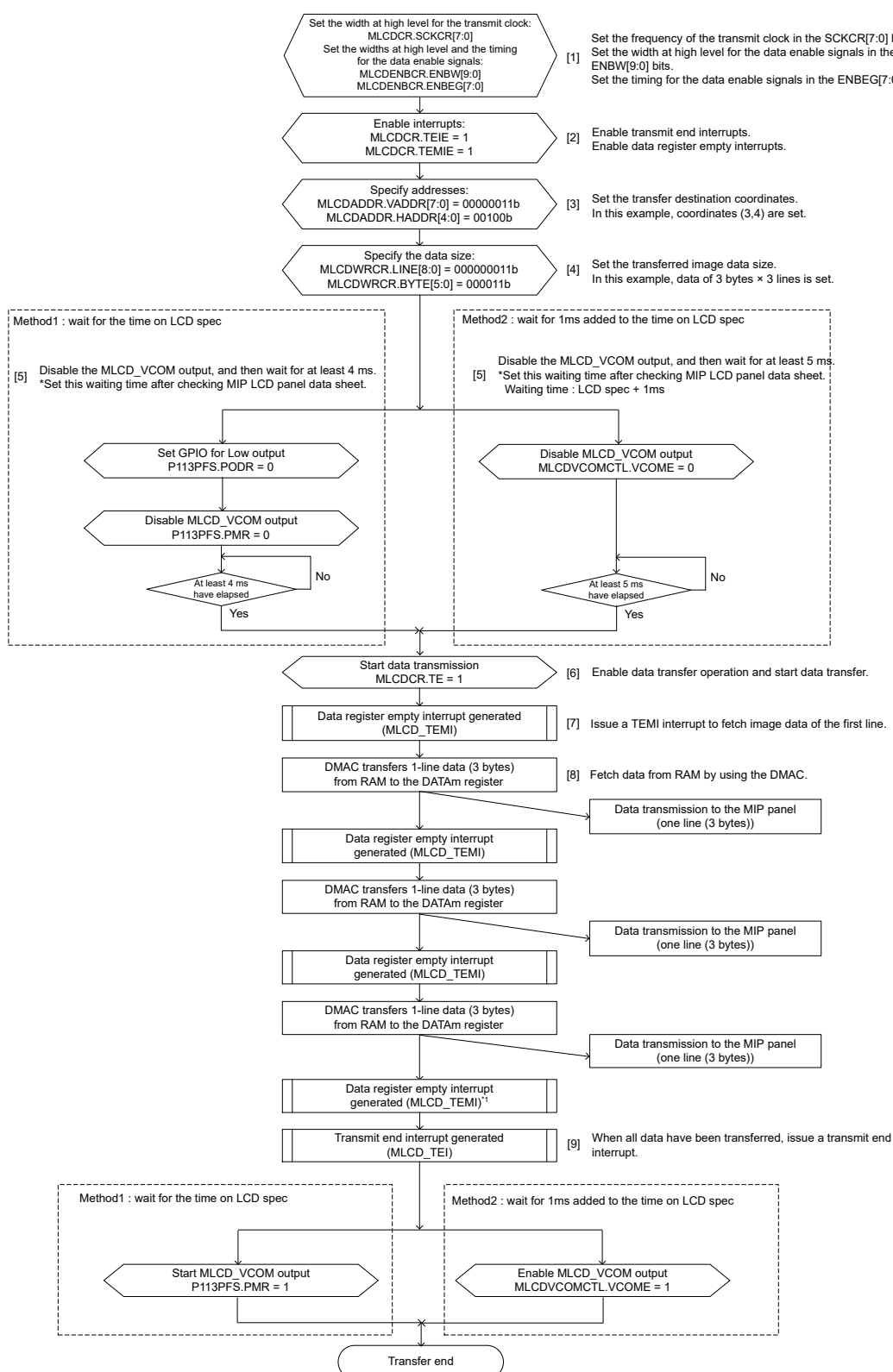
	When the HW mask function is not used	When the HW mask function is used
How to mask VCOM ^{*1}	Run by user software	Run by hardware
Register setting when mask VCOM	MLCDVCOMCTL.VCOME = 0 (Output prohibited)	MLCDVCOMCTL.HWMSKEN = 1 (Mask function enabled)
VCOM value when mask VCOM	Low	Keep the value before stopping
Operation when MLCDCR.TE is set to 1	Start data transmission immediately	Start data transmission after the time which is set by MLCDVCOMCTL.FMASK
Omitting condition of mask time before VCOM change point	—	Start next data transmission during masking time after VCOM change point of previous data transmission

Note 1. Regarding VCOM mask, see [section 40.3.2. Setting and Timing of MLCD_VCOM](#)

40.3.3 Basic Transmission

Basically, it is assumed that data is sent byte by byte (8 bits). Continuous transmission of basic data is enabled by defining the transmission size by using the MLCDWRCR register after setting the address as the origin of drawing in the MLCDADDR register (see [Figure 40.4](#)). [Figure 40.12](#) and [Figure 40.13](#) show the setting flow of basic transmission (sending 3-byte × 3-line data to coordinates (3, 4)).

[Figure 40.12](#) shows the procedure when the HW mask function is not used. [Figure 40.13](#) shows the procedure when using the HW mask function.



Note 1. (Number of lines + 1)th of the interrupt is generated. When using CPU interrupt and DTC, do not write data to the MLCD drawing data setting register with (Number of lines + 1)th `MLCD_TEMI` interrupt request (final request). When using DMAC, (Number of lines + 1)th `MLCD_TEMI` interrupt request is held in the DMA transfer request bit (`DELSRn.IR` bit). Clear the `DELSRn.IR` bit before transferring the next data.

Figure 40.12 Setting procedure example for basic transmission (the HW mask function is not used)

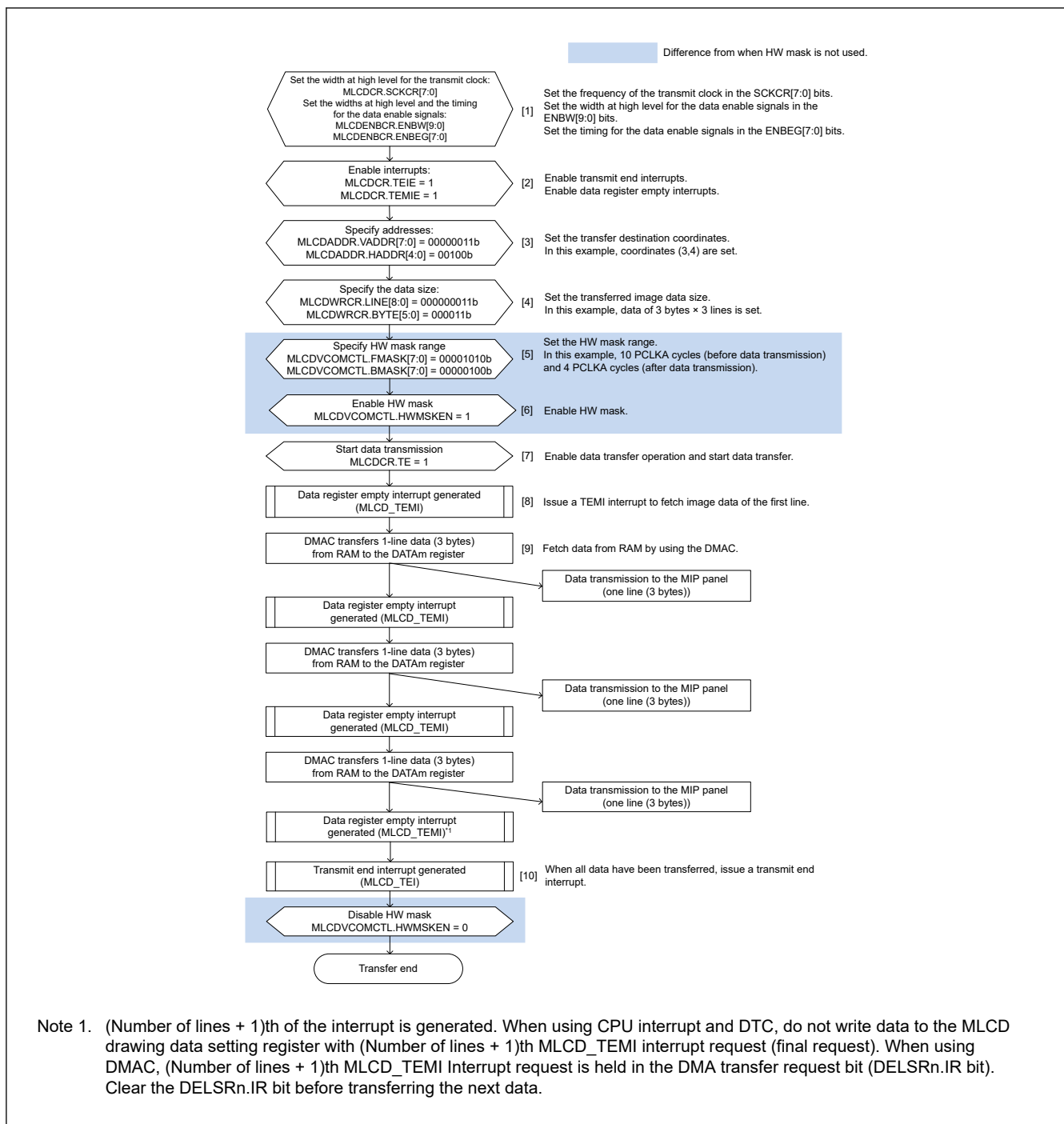


Figure 40.13 Setting procedure example for basic transmission (the HW mask function is used)

40.3.4 Same Image Transmission

To display all 1s or 0s over a wide range of the screen, it is possible to draw the MIP LCD panel multiple times via transfer of one-line data from the RAM by using the forced transmission function. Figure 40.14 and Figure 40.15 show the case where 22-byte data is copied. Figure 40.14 shows the procedure when the HW mask function is not used, and Figure 40.15 is the procedure when using the HW mask function.

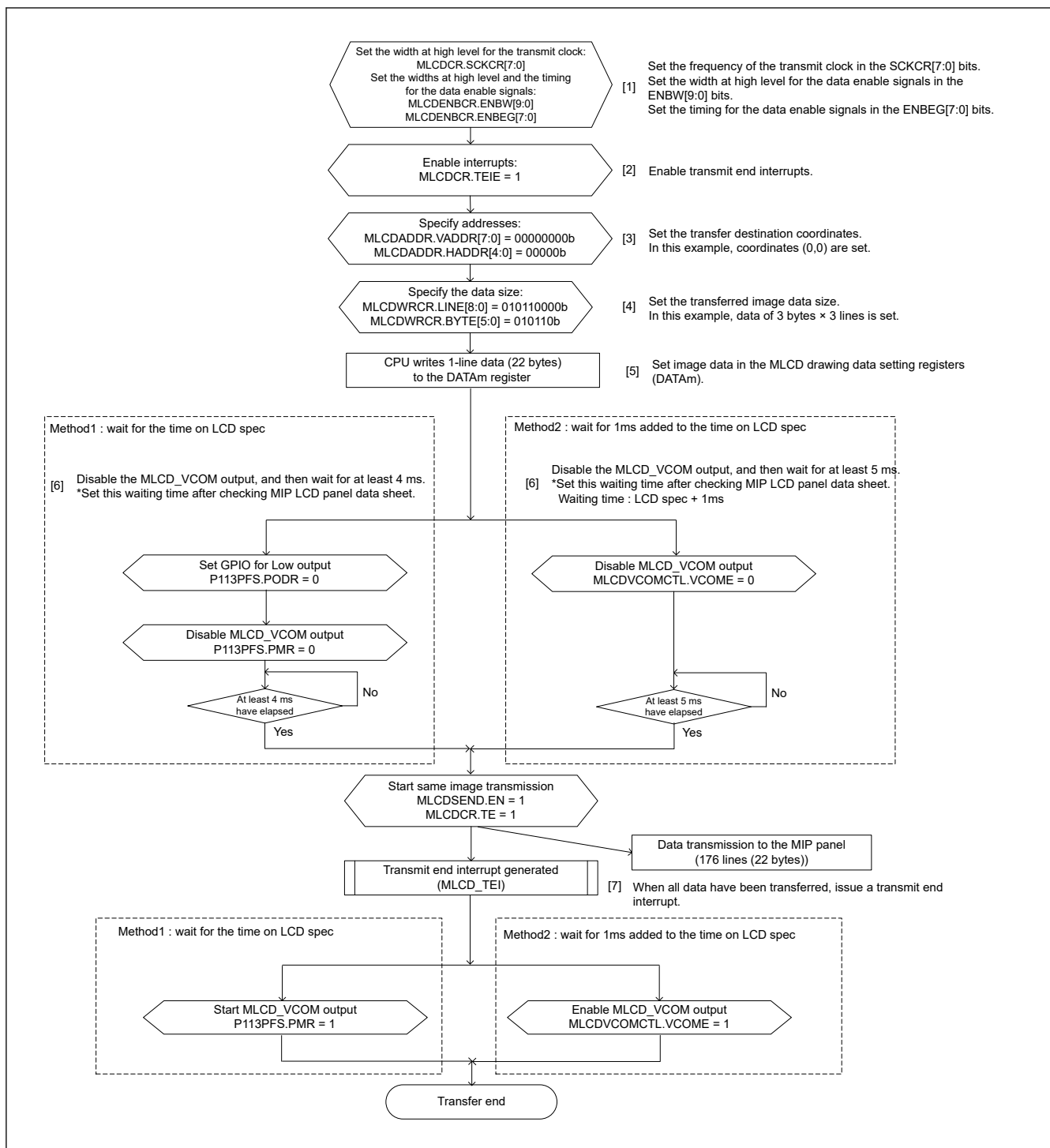


Figure 40.14 Setting procedure example for same image transmission (the HW mask function is not used)

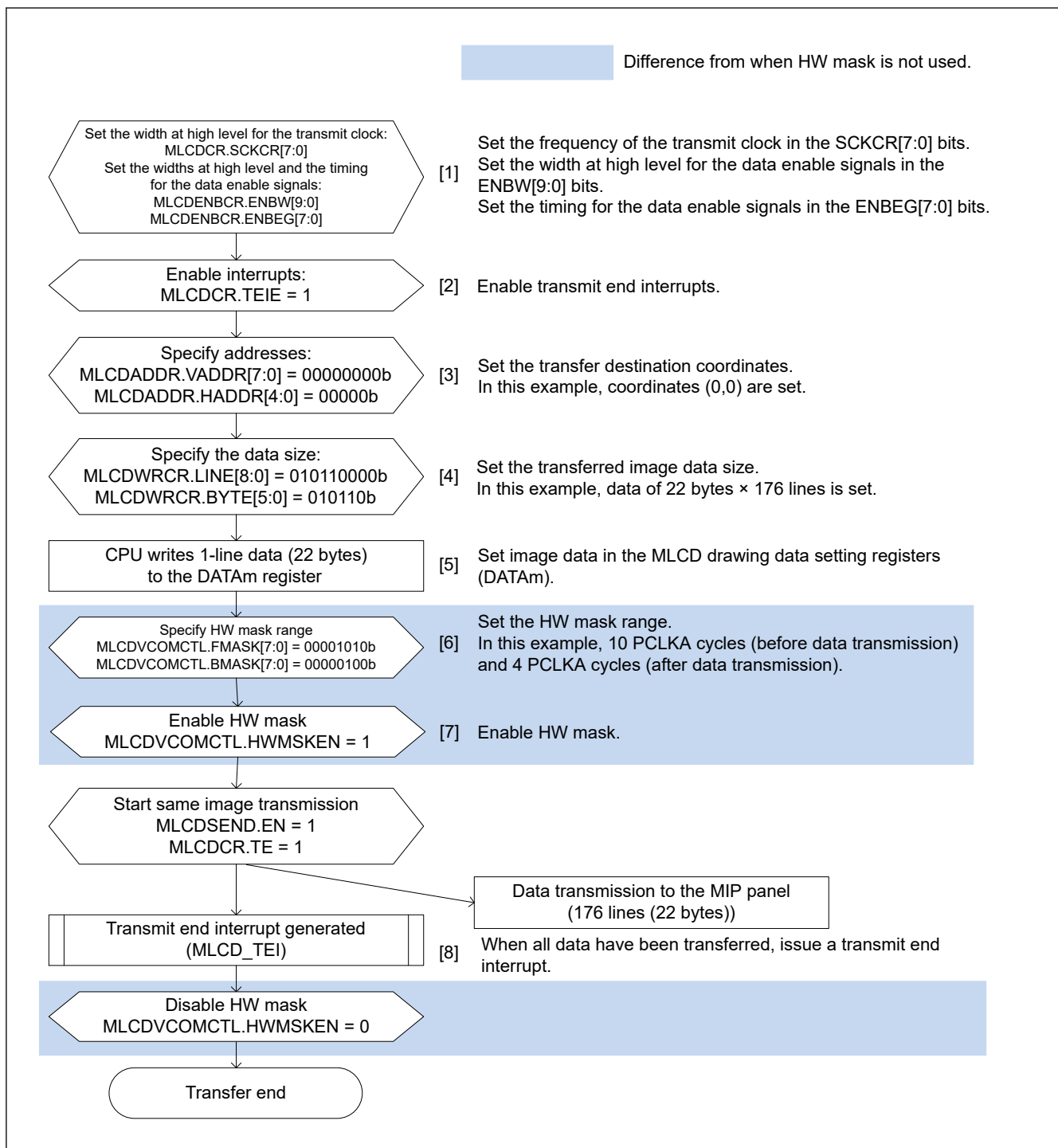
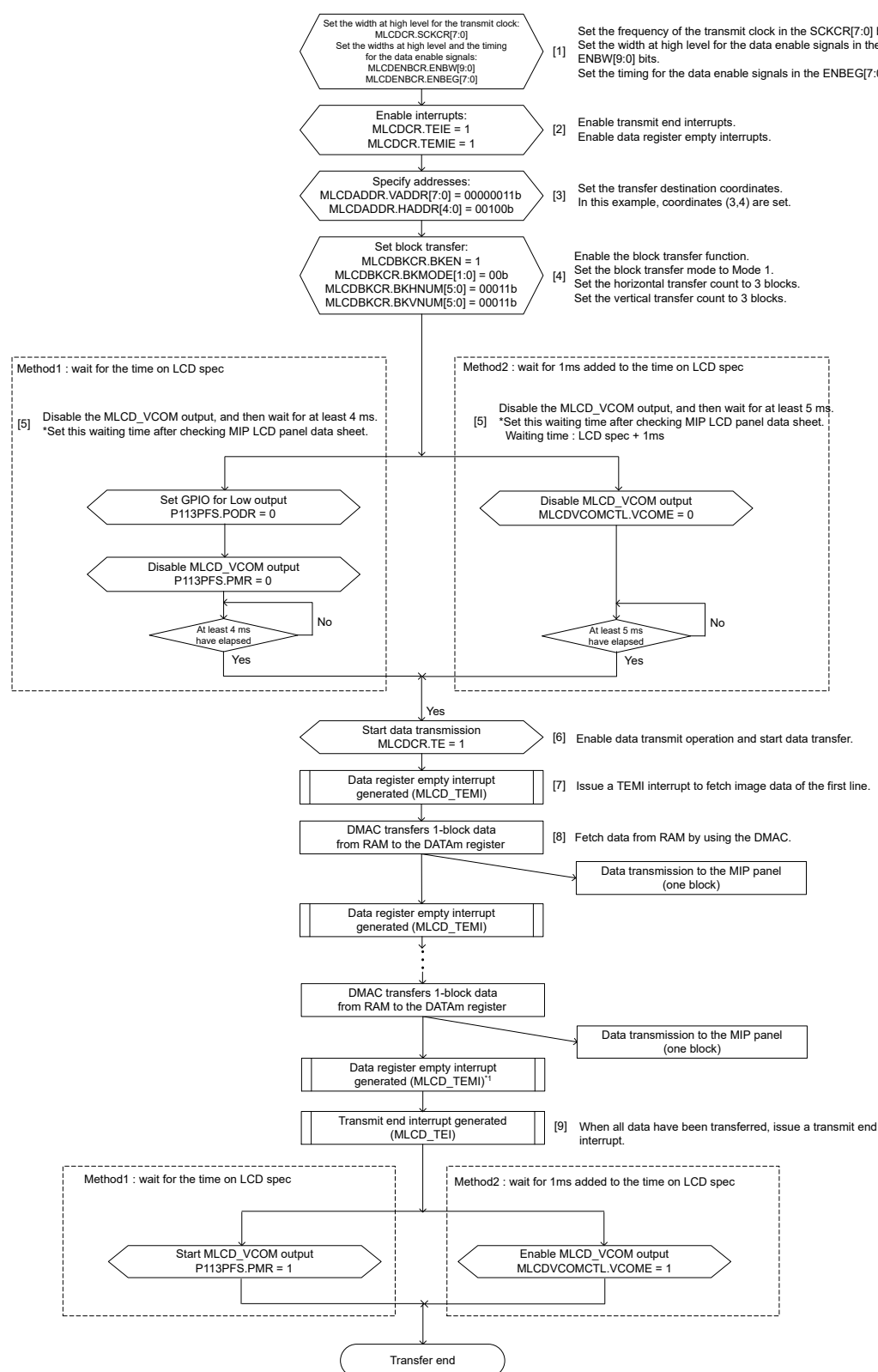


Figure 40.15 Setting procedure example for same image transmission (the HW mask function is used)

40.3.5 Block Transmission

Data can be sent in 8×8 bits (8 bytes) or 16×16 bits (32 bytes) blocks from the coordinates specified as the origin. [Figure 40.16](#) and [Figure 40.17](#) show the transfer flow. [Figure 40.16](#) shows the procedure when the HW mask function is not used. [Figure 40.17](#) shows the procedure when using the HW mask function.



Note 1. (Number of blocks + 1)th of the interrupt is generated. When using CPU interrupt and DTC, do not write data to the MLCD drawing data setting register with (Number of blocks + 1)th MLCD_TEMI interrupt request (final request). When using DMAC, (Number of blocks + 1)th MLCD_TEMI Interrupt request is held in the DMA transfer request bit (DELSRn.IR bit). Clear the DELSRn.IR bit before transferring the next data.

Figure 40.16 Transfer procedure example for block transmission (the HW mask function is not used)

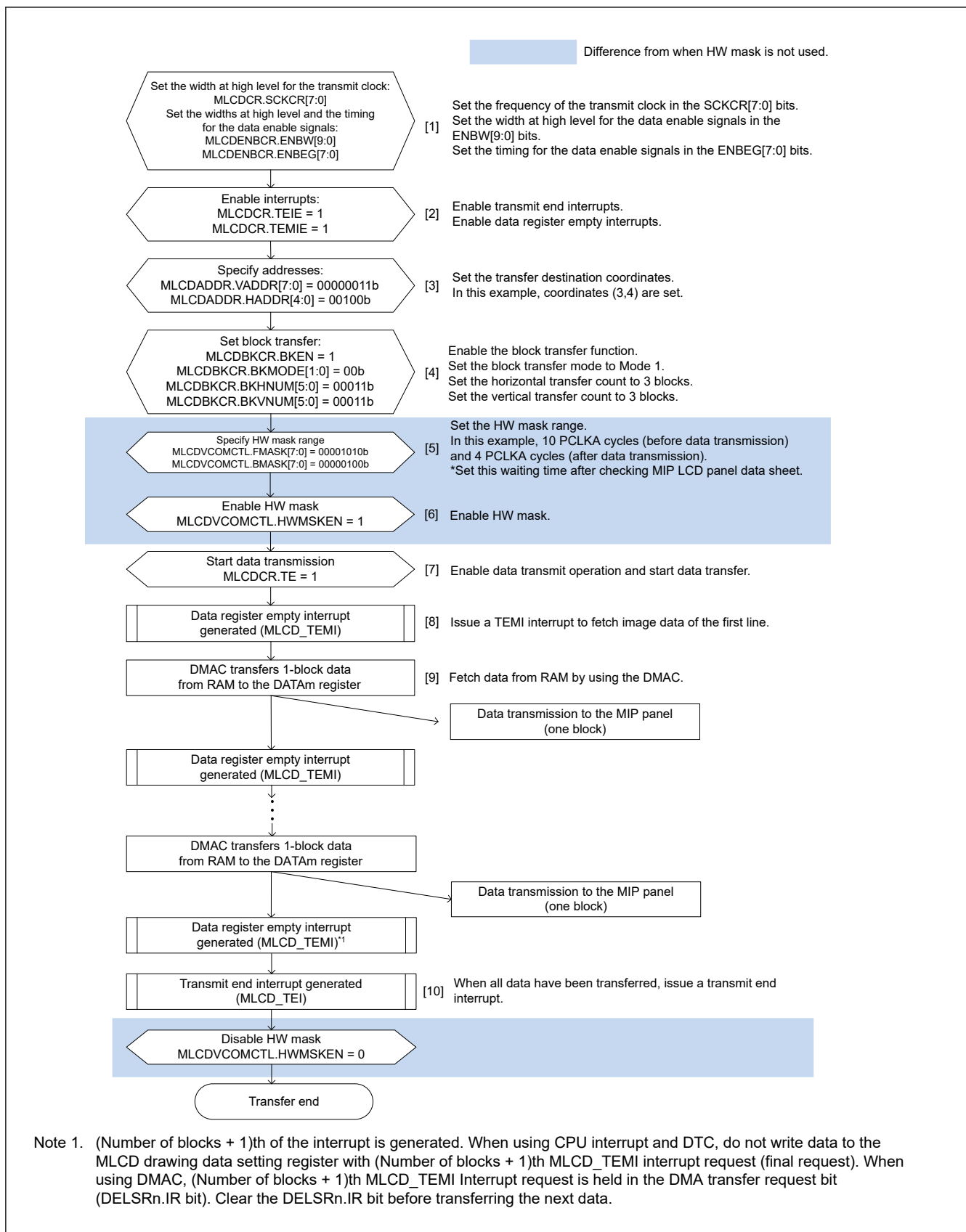


Figure 40.17 Transfer procedure example for block transmission (the HW mask function is used)

40.4 Interrupt Sources and Event Links

40.4.1 Interrupt Requests

[Table 40.5](#) lists the interrupt sources in the MLCD. There are two interrupt sources for the MLCD: the transmit end interrupt and data register empty interrupt.

Table 40.5 Interrupt sources

Name	Interrupt source	Interrupt generation condition	Interrupt to the CPU	DTC activation	DMAC activation
MLCD_TEI	Transmit end interrupt	(TEIE = 1) and (TEND = 1)	Possible	Possible	Possible
MLCD_TEMI	Data register empty interrupt	(TEMIE = 1) and (TEF = 1)	Possible	Possible	Possible

When the DTC or DMAC is to handle data transfer, whichever is selected should be set up first. After enabling the module to handle transfer, set up the MLCD. For the method for setting the DTC or DMAC, see [section 19, DMA Controller \(DMAC\)](#) and [section 20, Data Transfer Controller \(DTC\)](#).

40.4.2 Event Link Operation

[Table 40.6](#) lists the MLCD event link sources. The MLCD uses the event link controller (ELC) to perform a link operation to a specified module using an interrupt request as the event signal. The MLCD outputs the event signal regardless of the setting of the corresponding interrupt request enable bit in the MLCD Control Register (MLCDCR). For the detailed setting procedure for the event link function, see [section 21, Event Link Controller \(ELC\)](#).

Table 40.6 Event link sources

Name	Event link output source
MLCD_ELCTEND	Transmission end
MLCD_ELCTEM	Data register empty

40.5 Usage Notes

40.5.1 Setting for the Module-Stop Function

MLCD operation can be disabled or enabled using the Module Stop Control Register C (MSTPCRC). MLCD is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

40.5.2 Setting an Area for Drawing Data

Set the drawing data so that it does not exceed the maximum size of the MIP LCD panel.

40.5.3 MLCD_VCOM Output Settings

When data are to be transmitted, be sure to disable the MLCD_VCOM output (MLCDVCOMCTL.VCOME = 0).

41. 2D Graphics Data Conversion Circuit (GDT)

41.1 Overview

The 2D graphics data conversion circuit (GDT) handles image processing such as compositing and scaling down of the sizes of two-dimensional images. The image size that can be handled in a single operation is up to 32 bytes. The GDT can also convert various glyph data into image data and compose them with the background image. Note that for color images, the R, G, and B images must be stored in separate areas of the memory.

Table 41.1 GDT specifications

Parameter	Specifications
Rotation	The unit of image processing is 16×16 bits or 8×8 bits 90-degree clockwise 90-degree counterclockwise - Vertical flip - Horizontal flip
Scaling down	The unit of image processing is 8×8 bits - Pixel averaging method (reduction size can be selected from $1/8$, $2/8$, $3/8$, $4/8$, $5/8$, $6/8$, $7/8$)¹ - Pixel skipping (images can only be scaled down to $1/2$)
Inversion	The unit of image processing is 16×16 bits Bit inversion (1 is inverted to 0, and vice versa)
Monochrome compositing	The units of monochrome compositing are 16×16 bits and 8×8 bits. Combinations of a foreground image, background image, and border image that should be trimmed away can be composited.
Color compositing	The units of color compositing are 16×16 bits and 8×8 bits. Setting priority to the specified color of the foreground Handling a specified color in the foreground as being in the top layer or setting parts of the foreground with the specified color to the background colors in the same position is possible in the compositing of images.
Scrolling	The unit of image processing is two 8×16 -bit images An 8×16-bit image can be extracted from combinations of data for two consecutive images with the position set in 1-bit units.
Conversion of glyph data into image data	Glyph data can be converted into image data. Applicable to glyph sizes from 7×7 bits to 63×64 bits²
Colorization	The unit of image processing is 16×16 bits Black or white pixels can be colorized by 3-bit RGB values; any of eight colors is specifiable for each of them.
Color data sorting	The unit of color data sorting is 16×16 bits. - The separate R, G, and B images in memory can be sorted into a single area in order of R, G, and B for each pixel. - Sorting can be specified to result in a sequence of 3-bit units (RGB) or 4-bit units (RGB and a padding bit).
Endian conversion	The unit of endian conversion is 16×16 bits. The MSB and LSB can be swapped.
Power-saving function	The module-stop state can be set
Interrupt sources	- Image data input request interrupt - Image data output request interrupt - Glyph data conversion complete interrupt

Note: The numbers of bits represent the heights and widths of the glyphs.

Note 1. This function is only applicable to monochrome images.

Note 2. If the dimensions in bits are multiples of 8, conversion is not necessary because the glyph data can simply be handled as image data.

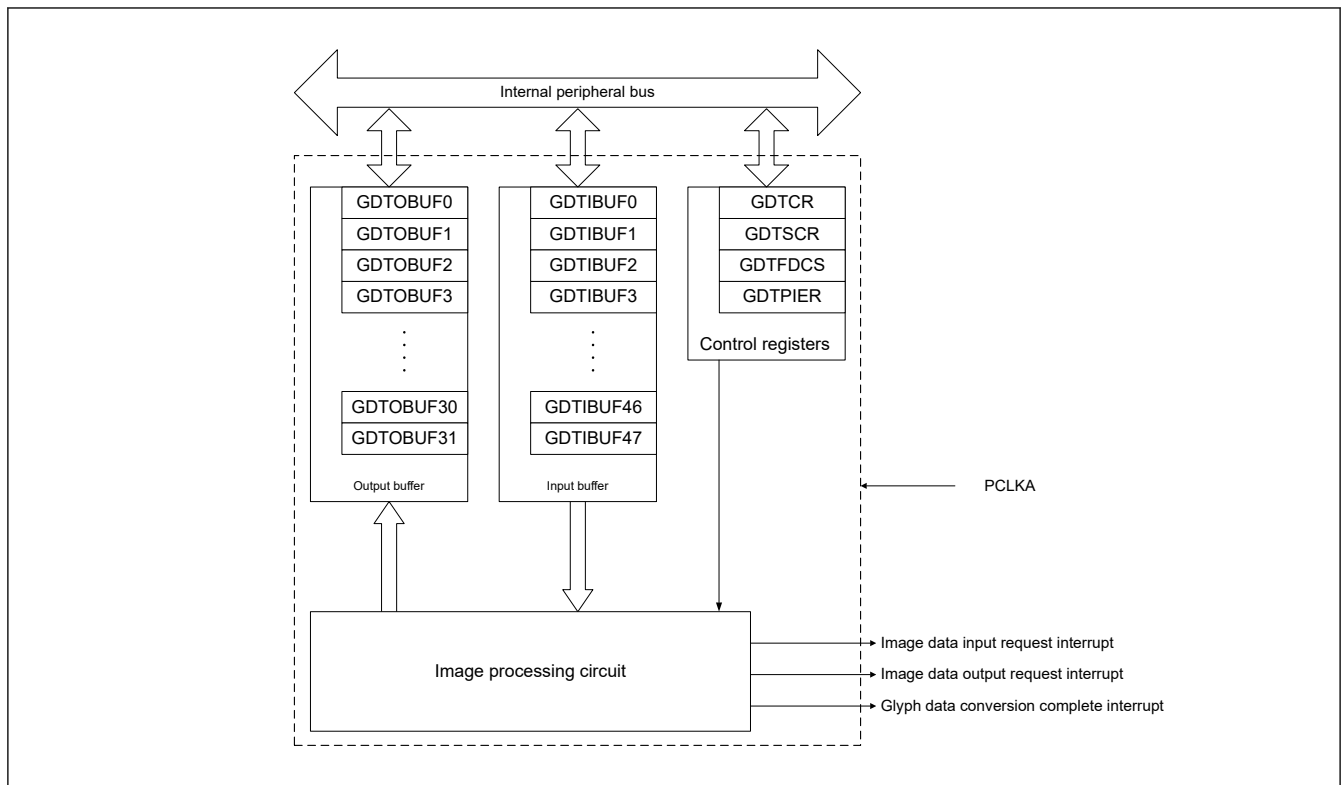


Figure 41.1 2D graphics data conversion circuit (GDT) block diagram

41.2 Register Descriptions

41.2.1 GDTCR : GDT Control Register

Base address: GDT = 0x4007_0800

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CLRDS1[2:0]			CLRDS0[2:0]			CLREN	—	CDCS[2:0]			CPTS	CSYEN	MBRDEN	MPCS	MSYEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ISCREN[2:0]			RTTFC[1:0]		RTTEN	CIALGSL	CIALGEN	ENDIAN	—	GDTDSZ	IFLPEN	—	—	—	GDTSTART
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GDTSTART	GDT Processing Start 0: GDT is stopped 1: GDT processing is started	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	IFLPEN	Inversion Enable 0: Bit inversion is disabled*1 1: Bit inversion is enabled	R/W
5	GDTDSZ	Image Processing Data Size Select*2 0: The processable data size is 16 × 16 bits 1: The processable data size is 8 × 8 bits	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	ENDIAN	Endian Conversion 0: Endian conversion is disabled*3 1: Endian conversion is enabled	R/W
8	CIALGEN	Color Data Sorting Enable 0: The sorting of color data is disabled*3 1: The sorting of color data is enabled	R/W
9	CIALGSL	Color Data Sorting Select 0: The color data are sorted in order of R, G, and B represented by three bits. 1: The color data are sorted in order of R, G, and B represented by three bits, with one bit of padding, for a total of four bits.	R/W
10	RTTEN	Rotation Enable 0: Rotation is disabled*3 1: Rotation is enabled	R/W
12:11	RTTFC[1:0]	Rotation Type Select 0 0: 90-degree clockwise 0 1: 90-degree counterclockwise 1 0: Vertical flip 1 1: Horizontal flip	R/W
15:13	ISCREN[2:0]	Image Scrolling Enable 0 0 0: Image scrolling is disabled.*3 0 0 1: The image is displayed after right-shifting by 1 bit 0 1 0: The image is displayed after right-shifting by 2 bits 0 1 1: The image is displayed after right-shifting by 3 bits 1 0 0: The image is displayed after right-shifting by 4 bits 1 0 1: The image is displayed after right-shifting by 5 bits 1 1 0: The image is displayed after right-shifting by 6 bits 1 1 1: The image is displayed after right-shifting by 7 bits	R/W
16	MSYNEN	Monochrome Image Compositing Enable 0: The compositing of monochrome images is disabled*3 1: The compositing of monochrome images is enabled	R/W
17	MPCS	Monochrome Priority Color Setting 0: Priority in compositing is given to pixels having the value 1 in the foreground image data. 1: Priority in compositing is given to pixels having the value 0 in the foreground image data.	R/W
18	MBRDEN	Image Trimming Enable 0: Image trimming is disabled 1: Image trimming is enabled	R/W
19	CSYNEN	Color Image Compositing Enable 0: The compositing of color images is disabled*3 1: The compositing of color images is enabled	R/W
20	CPTS	Color Image Priority/Transparency Setting 0: The specified color of the foreground on the forefront is displayed (priority color mode). 1: The same color as the background color in the parts of the foreground image with the specified color is displayed (transparent color mode).	R/W
23:21	CDCS[2:0]	Specified Color Setting These bits specify an RGB color by setting a value from 000b to 111b.*4 Bits [23:21] respectively correspond to R, G, and B.	R/W
24	—	This bit is read as 0. The write value should be 0.	R/W
25	CLREN	Image Colorization Function Enable 0: Colorization of the image is disabled*3 1: Colorization of the image is enabled	R/W
28:26	CLRDS0[2:0]	0 Data Color Setting These bits specify the RGB color for pixels having the value 0 by setting a value from 000b to 111b.*4 Bits [28:26] respectively correspond to R, G, and B.	R/W

Bit	Symbol	Function	R/W
31:29	CLRDS1[2:0]	1 Data Color Setting These bits specify the RGB color for pixels having the value 1 by setting a value from 000b to 111b.*4 Bits [31:29] respectively correspond to R, G, and B.	R/W

Note 1. Bit inversion can be enabled at the same time as other functions.

Note 2. This bit can only be used with compositing and rotation; its setting should be 0 when other functions are to be used.

Note 3. Setting bits [7], [8], [10], [15:13], [16], [19], and [25] to enabled is mutually exclusive. If any of these bits is enabled at the same time as the other bit, operation cannot be guaranteed. Similarly, these bits cannot be enabled at the same time as when the SHRNKEN bit in GDTSCR or FDCEN bit in GDTFDCS is enabled.

Note 4. The following eight RGB colors are specifiable:

- [000b: Black]
- [100b: Red]
- [110b: Yellow]
- [010b: Green]
- [011b: Cyan]
- [001b: Blue]
- [101b: Magenta]
- [111b: White]

GDTSTART bit (GDT Processing Start)

The GDTSTART bit starts data processing after a function is enabled.

IFLPEN bit (Inversion Enable)

The IFLPEN bit enables or disables the bit reversal function.

GDTDSZ bit (Image Processing Data Size Select)

The GDTDSZ bit specifies the image size when image compositing or rotation is enabled. Write 0 to this bit when neither image compositing nor rotation is to be used.

ENDIAN bit (Endian Conversion)

The ENDIAN bit enables or disables the MSB/LSB reversal function for the bit array of output data.

CIALGEN bit (Color Data Sorting Enable)

The CIALGEN bit enables or disables the color data sorting function.

CIALGSL bit (Color Data Sorting Select)

The CIALGSL bit selects whether a dummy bit is added to sorted color data.

RTTEN bit (Rotation Enable)

The RTTEN bit enables or disables the image rotation function.

RTTFC[1:0] bits (Rotation Type Select)

The RTTFC[1:0] bits select the rotation type of the image rotation function.

ISCREN[2:0] bits (Image Scrolling Enable)

The ISCREN[2:0] bits specify the amount of movement for the image scroll function.

MSYNEN bit (Monochrome Image Compositing Enable)

The MSYNEN bit enables or disables the compositing of monochrome images.

MPCS bit (Monochrome Priority Color Setting)

The MPCS bit selects whether priority is given to 0 or 1 in the compositing of the foreground image.

MBRDEN bit (Image Trimming Enable)

The MBRDEN bit enables or disables the image trimming function.

CSYNEN bit (Color Image Compositing Enable)

The CSYNEN bit enables or disables the compositing of color images.

CPTS bit (Color Image Priority/Transparency Setting)

The CPTS bit selects whether the specified color (specified in the CDCS[2:0] bits) of the foreground image is displayed on the forefront, or an image with the background image transmitted through only the part of the specified color is displayed.

CDCS[2:0] bits (Specified Color Setting)

The CDCS[2:0] bits specify the color of a priority or transparent image generated by the CPTS bit.

CLREN bit (Image Colorization Function Enable)

The CLREN bit enables or disables the monochrome image colorization function.

CLRDS0[2:0] bits (0 Data Color Setting)

The CLRDS0[2:0] bits specify the color for 0 data of a monochrome image.

CLRDS1[2:0] bits (1 Data Color Setting)

The CLRDS1[2:0] bits specify the color for 1 data of a monochrome image.

41.2.2 GDTSCR : Reduction Function Control Register

Base address: GDT = 0x4007_0800

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SHRNKNUM[2:0]			—	—	—	—	—	—	—	SBCS
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SHRNKSZ[2:0]			—	—	—	—	—	—	—	SHRNKEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SHRNKEN	Scaling Down Enable 0: Scaling down is disabled.*1 *3 1: Scaling down is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
10:8	SHRNKSZ[2:0]	Scaling Down Ratio Select 0 0 0: Images are scaled down to 1/8 by pixel averaging.*2 0 0 1: Images are scaled down to 2/8 by pixel averaging.*2 0 1 0: Images are scaled down to 3/8 by pixel averaging.*2 0 1 1: Images are scaled down to 4/8 by pixel averaging.*2 1 0 0: Images are scaled down to 5/8 by pixel averaging.*2 1 0 1: Images are scaled down to 6/8 by pixel averaging.*2 1 1 0: Images are scaled down to 7/8 by pixel averaging.*2 1 1 1: Images are scaled down to 1/2 by pixel averaging.*2	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	SBCS	Unused Background Color Setting 0: 0 is written to the unused bits after scaling down.*1 1: 1 is written to the unused bits after scaling down.	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
26:24	SHRNKNUM[2:0]	Number of Images for Scaling Down 0 0 0: 1 0 0 1: 2 0 1 0: 3 0 1 1: 4 1 0 0: 5 1 0 1: 6 1 1 0: 7 1 1 1: 8	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In the image reduction function, a redundant area is generated in the reduced image except when eight units of the 8 × 8-bit image data (minimum unit of image processing) are processed. The redundant area after reduction can be set to the same color as the background data by setting the SBCS bit.

Note 2. Image reduction by the pixel averaging method is only enabled for monochrome images.

Note 3. This bit cannot be enabled at the same time as the ENDIAN, CIALGEN, RTTEN, ISCREN[2:0], MSYNEN, CSYNEN, or CLREN bits in the GDTCR register. If two or more of these bits are enabled at the same time, operation cannot be guaranteed. Likewise, this bit cannot be enabled at the same time as the FDCEN bit in the GDTFDCS register.

SHRNKEN bit (Scaling Down Enable)

The SHRNKEN bit enables or disables the scaling down of images.

SHRNKSZ[2:0] bits (Scaling Down Ratio Select)

The SHRNKSZ[2:0] bits specify the algorithm and ratio for use in scaling down.

SBCS bit (Unused Background Color Setting)

The SBCS bit specifies 0 or 1 for the value to be written to the unused bits after scaling down.

SHRNKNUM[2:0] bits (Number of Images for Scaling Down)

The SHRNKNUM[2:0] bits specify the number of images to be consecutively scaled down.

41.2.3 GDTFDCS : Glyph Data Image Conversion Setting Register

Base address: GDT = 0x4007_0800

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Bit field:	—	FDLNGSZ[6:0]								—	—	FDLTDSZ[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Bit field:	—	—	FDIR[5:0]						FDH A D	SAC[2:0]			—	—	—	FDCE N	
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	

Bit	Symbol	Function	R/W
0	FDCEN	Glyph Data Conversion Enable*1 0: Conversion of glyph data into image data is disabled. 1: Conversion of glyph data into image data is enabled.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6:4	SAC[2:0]	Start Address Change 0 0 0: No shifting 0 0 1: Right-shifted by 1 bit 0 1 0: Right-shifted by 2 bits 0 1 1: Right-shifted by 3 bits 1 0 0: Right-shifted by 4 bits 1 0 1: Right-shifted by 5 bits 1 1 0: Right-shifted by 6 bits 1 1 1: Right-shifted by 7 bits	R/W
7	FDHAD	Unused Bit Setting 0: 0 is written to the unused bits after glyph conversion. 1: 1 is written to the unused bits after glyph conversion.	R/W
13:8	FDIR[5:0]	Number of Rounds of Data Processing Setting These bits specify the number of rounds of glyph data to be processed.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
21:16	FDLTDSZ[5:0]	Glyph Data Horizontal Size Setting These bits specify the number of bits of glyph data in the horizontal direction. The setting range is from 7 to 63 bits.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
30:24	FDLNGSZ[6:0]	Glyph Data Vertical Size Setting These bits specify the number of bits of glyph data in the vertical direction. The setting range is from 7 to 64 bits.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. This bit cannot be enabled at the same time as the ENDIAN, CIALGEN, RTTEN, ISCREN[2:0], MSYNEN, CSYNEN, or CLREN bits in the GDTCSR register. If two or more of these bits are enabled at the same time, operation cannot be guaranteed. Likewise, this bit cannot be enabled at the same time as the SHRKNEN bit in the GDTSCR register.

FDCEN bit (Glyph Data Conversion Enable)

The FDCEN bit enables or disables the conversion of glyph data into image data.

SAC[2:0] bits (Start Address Change)

The SAC[2:0] bits specify the number of bits of right-shifting at the address where conversion into image data is to start.

FDHAD bit (Unused Bit Setting)

The FDHAD bit specifies the value with which to fill the unused bits resulting from setting of the start address change bit (SAC[2:0]).

FDIR[5:0] bits (Number of Rounds of Data Processing Setting)

The FDIR[5:0] bits specify the number of rounds of processing, with the type of setting depending on the size of the glyph data to be processed. The unit of processing is 8×8 bits (64 bits). Set the required number of cycles in terms of the basic processing unit from the size of the glyph data to be processed.

- If the remainder obtained by dividing the total of all the bits of the glyph data by 64 is 0, set the quotient value in the FDIR[5:0] bits.
- If the remainder obtained by dividing the total of all the bits of the glyph data by 64 is not 0, set the value of (quotient + 1) in the FDIR[5:0] bits.

For example, when 20×20 -bit (400-bit) glyph data is to be processed, $400 \div 64 = 6$, R 16. That is, the remainder is not 0, therefore set the FDIR[5:0] bits to 7 (quotient + 1).

FDLTDSZ[5:0] bits (Glyph Data Horizontal Size Setting)

The FDLTDSZ[5:0] bits specify the number of horizontal-direction bits of the glyph data. The setting range is from 7 to 63 bits. For example, if you want to set 7 bits, set FDLTDSZ[5:0] = 0x07.

Note: For glyph data whose pixel size is a multiple of 8, GDT conversion is not necessary because the glyph data is equal to the image data.

The FDLNGSZ[6:0] bits specify the number of vertical-direction bits of the glyph data. The setting range is from 7 to 64 bits. For example, if you want to set 7 bits, set FDLNGSZ[6:0] bits = 0x07.

Base address: GDT = 0x4007_0800
Offset address: 0x00C

Bit	Symbol	Function	R/W
0	DATIIE	Image Data Input Request Interrupt Enable 0: Interrupt requests are disabled 1: Interrupt requests are enabled	R/W
1	DATOIE	Image Data Output Request Interrupt Enable 0: Interrupt requests are disabled 1: Interrupt requests are enabled	R/W
2	FDCENDIE	Glyph Data Conversion Complete Interrupt Enable 0: Interrupt requests are disabled 1: Interrupt requests are enabled	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	INTMODE	GDT Interrupt Mode Select* ¹ 0: Selects the conveying of interrupt signals to the DMAC 1: Selects the conveying of interrupt signals to the DTC	R/W

Base address: GDT = 0x4007_0800
Offset address: $0x100 + 0x4 \times n$

The GDTIBUFn register (n = 0 to 47) is 32-bit register that functions as a buffer to hold input image data. This register is only writable. When writing input image data to this register, write the data in order from the register with the lowest

address among the registers to be used. For byte access or halfword access, write the data in order from the lower-order bits as indicated in [Table 41.2](#).

Table 41.2 Allocation of GDT image data input register

Register	Higher-order 16 bits		Lower-order 16 bits	
	Higher-order 8 bits	Lower-order 8 bits	Higher-order 8 bits	Lower-order 8 bits
GDTIBUF0 (0x4007_0900)	0x4007_0903	0x4007_0902	0x4007_0901	0x4007_0900
GDTIBUF1 (0x4007_0904)	0x4007_0907	0x4007_0906	0x4007_0905	0x4007_0904
⋮				
GDTIBUF47 (0x4007_09BC)	0x4007_09BF	0x4007_09BE	0x4007_09BD	0x4007_09BC

41.2.6 GDTIBUF_n : GDT Image Data Output Register (n = 0 to 31)

Base address: GDT = 0x4007_0800

Offset address: 0x200 + 0x4 × n

Bit position: 31

0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

The GDTIBUF_n register (n = 0 to 31) is a 32-bit register that functions as a buffer to hold output image data. This register is only readable. When reading output image data from this register, read the data in order from the register with the lowest address among the registers to be used. For byte access or halfword access, read the data in order from the lower-order bits as indicated in [Table 41.3](#).

Table 41.3 Allocation of GDT image data output register

Register	Higher-order 16 bits		Lower-order bits	
	Higher-order 8 bits	Lower-order 8 bits	Higher-order 8 bits	Lower-order 8 bits
GDTIBUF0 (0x4007_0A00)	0x4007_0A03	0x4007_0A02	0x4007_0A01	0x4007_0A00
GDTIBUF1 (0x4007_0A04)	0x4007_0A07	0x4007_0A06	0x4007_0A05	0x4007_0A04
⋮				
GDTIBUF31 (0x4007_0A7C)	0x4007_0A7F	0x4007_0A7E	0x4007_0A7D	0x4007_0A7C

41.3 Operation

41.3.1 GDT Processing Flow

The basic processing by the GDT involves the transfer of the image data stored in the memory to the GDT image data input registers, processing of the data as desired, storage of the processed data in the GDT image data output registers, and writing back the processed data to the memory. Data transfer between the memory and the GDT is mainly handled by the DMAC or DTC.

[Figure 41.2](#) to [Figure 41.4](#) are examples of flows of processing according to the means of transfer. [Figure 41.2](#) shows the flow of bit inversion with the use of the DMAC for data transfer. Processing is repeated to execute processing to invert 256 bits × 256 bits (one image).

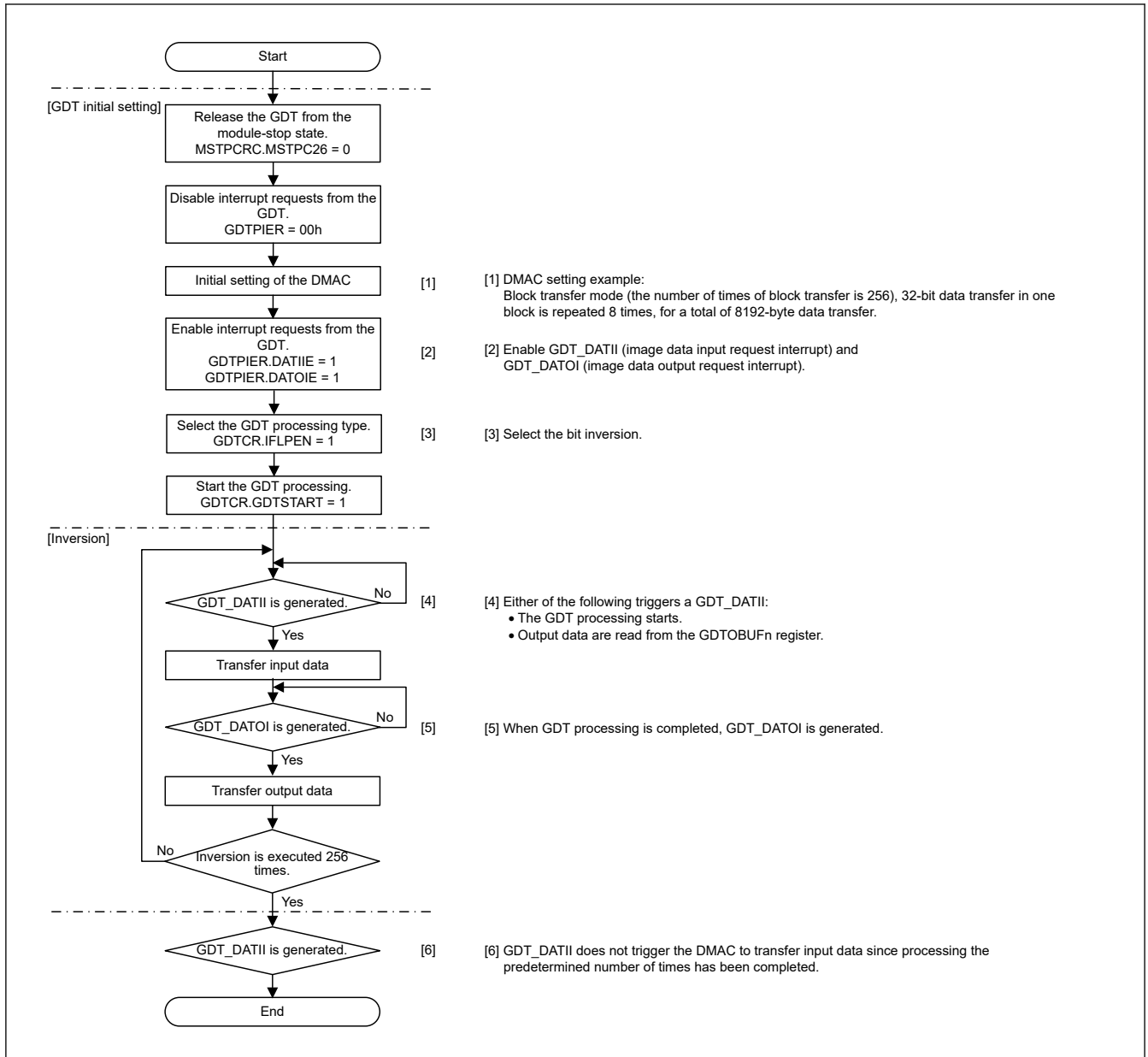


Figure 41.2 Example of a flow of bit inversion with the use of the DMAC for data transfer

Figure 41.3 shows a flow of color compositing with the use of the DMAC for data transfer. The image size handled in a single round of processing is 16 bits × 16 bits, and color compositing is executed for 256 bits × 256 bits (one image) by repeating the processing. DMAC channel 0 is used for transferring input image data from the memory to the GDTIBUFn registers, and DMAC channel 1 is used for transferring output image data from the GDTIBUFn registers to the memory. In color compositing, six planes of input image data and three planes of output image data are handled in each round of processing. Consequently, the transfer source address, transfer destination address, or both are changed in response to every image data input request interrupt and image data output request interrupt in the transfer of image data that have been allocated to a non-contiguous area in the memory or registers. When the settings of the DMAC are changed as described above, the CPU between the GDT and DMAC re-sets the information required for transfer and re-activates DMAC. For details on setting the DMAC, see [section 19, DMA Controller \(DMAC\)](#).

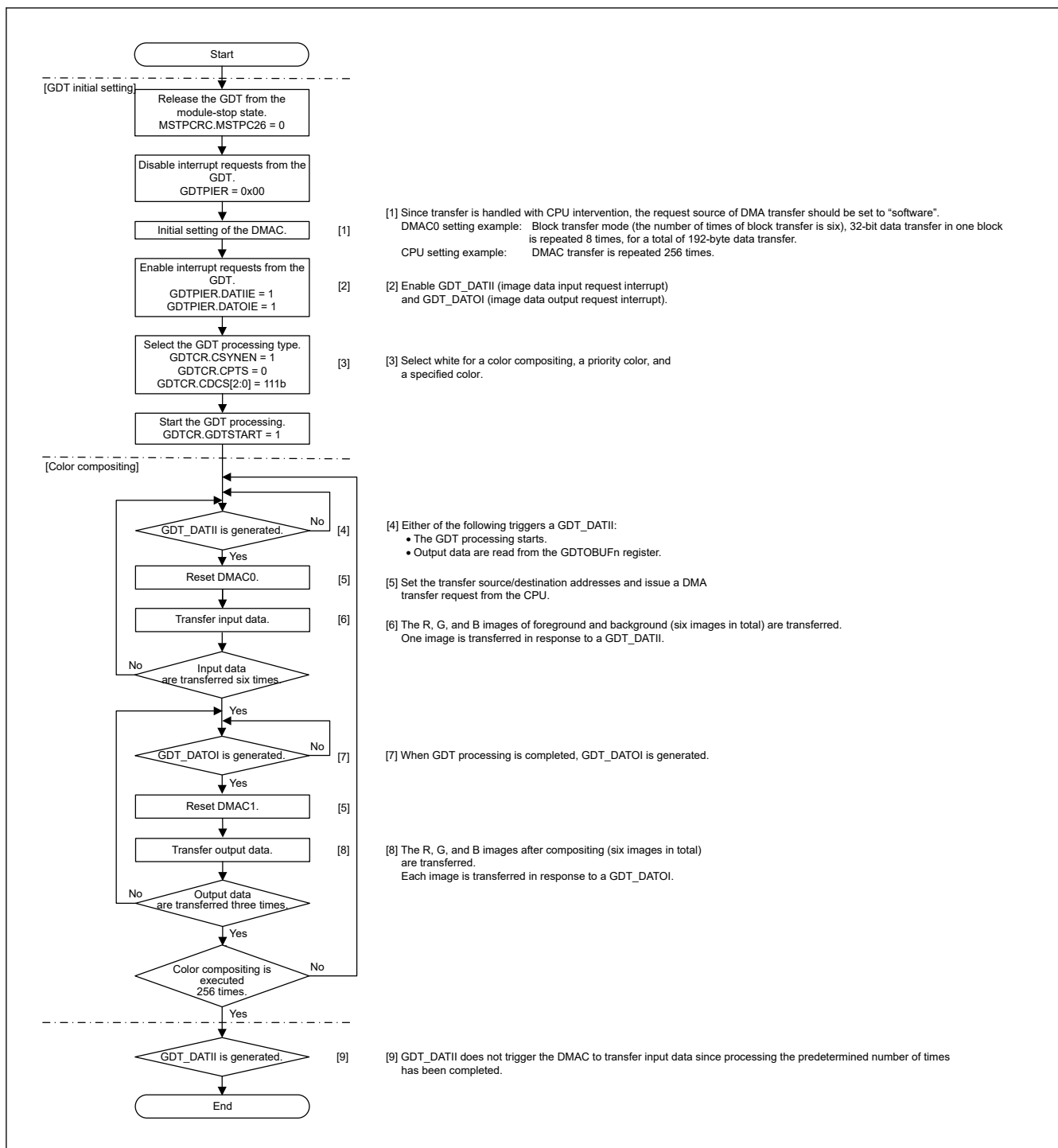


Figure 41.3 Example of a flow of color compositing with the use of the DMAC for data transfer requiring CPU intervention for control

Figure 41.4 shows a flow of a color compositing with the use of the DTC for data transfer. The image size handled in a single round of processing is 16 bits × 16 bits, and color compositing is executed for 256 bits × 256 bits (one image) by repeating processing. For details on setting the DTC, see [section 20, Data Transfer Controller \(DTC\)](#).

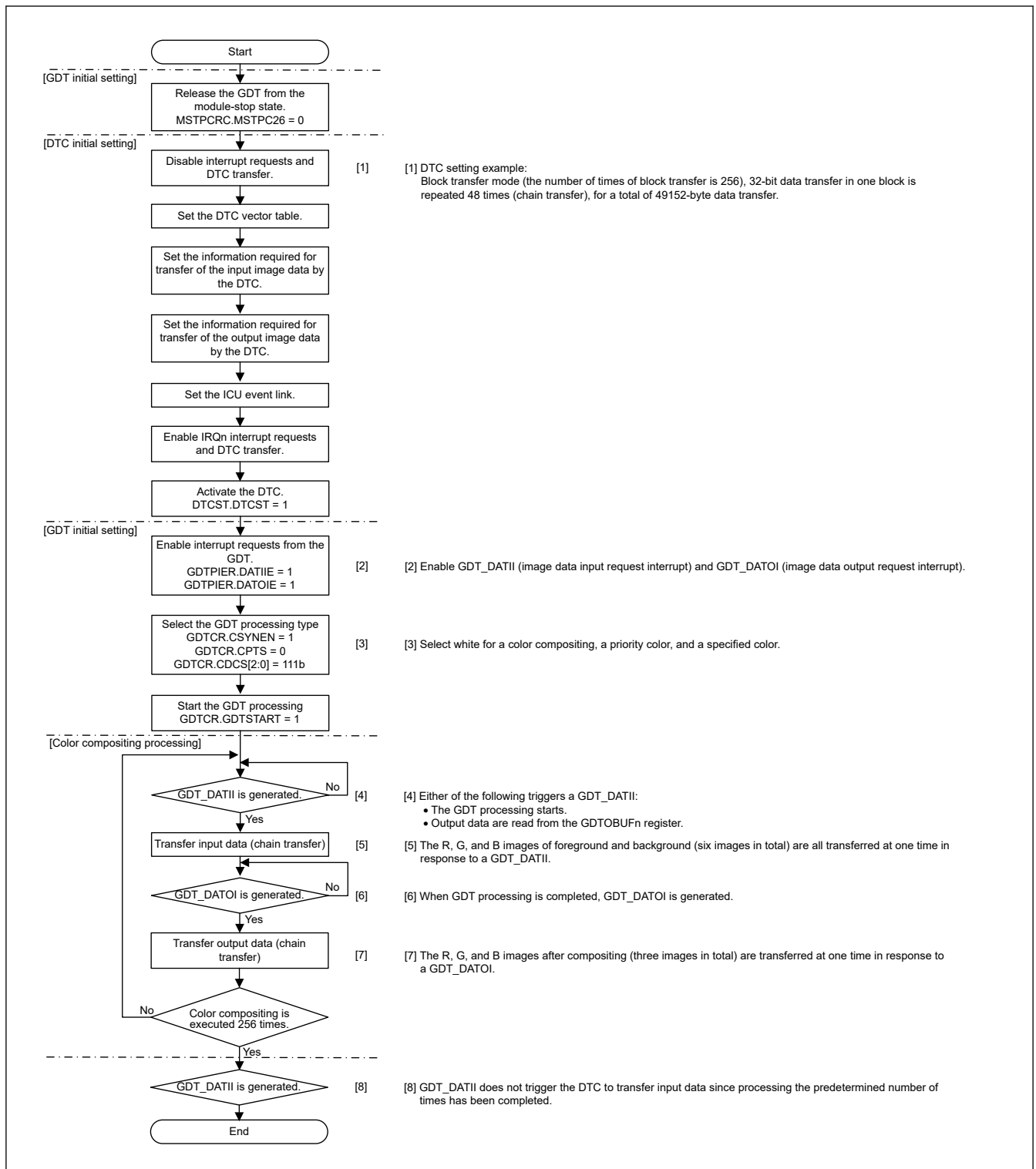


Figure 41.4 Example of a flow of color compositing with the use of the DTC for data transfer

Figure 41.5 and Figure 41.6 show examples of the transfer of image data. In these examples, the compositing of monochrome images is enabled, the size of the input data is 16 bits × 16 bits, and trimming is enabled. As shown in Figure 41.5, the transfer of input data by using the DMAC with individual transfer requests requires three image data input request interrupts for completion of transfer of each section. The transfer of input data by using chain transfer by the DTC requires one image data input request interrupt for completion of the transfer.

Figure 41.6 shows an example of the transfer of output data. This requires one interrupt for completion of the transfer regardless of whether transfer is by the DMAC or DTC. In scrolling and glyph conversion, allocation of the data to the registers depends on the type of transfer.

For more details, see [Figure 41.33](#) to [Figure 41.35](#) with regard to scrolling, and [Figure 41.40](#) and [Figure 41.41](#) with regard to glyph conversion.

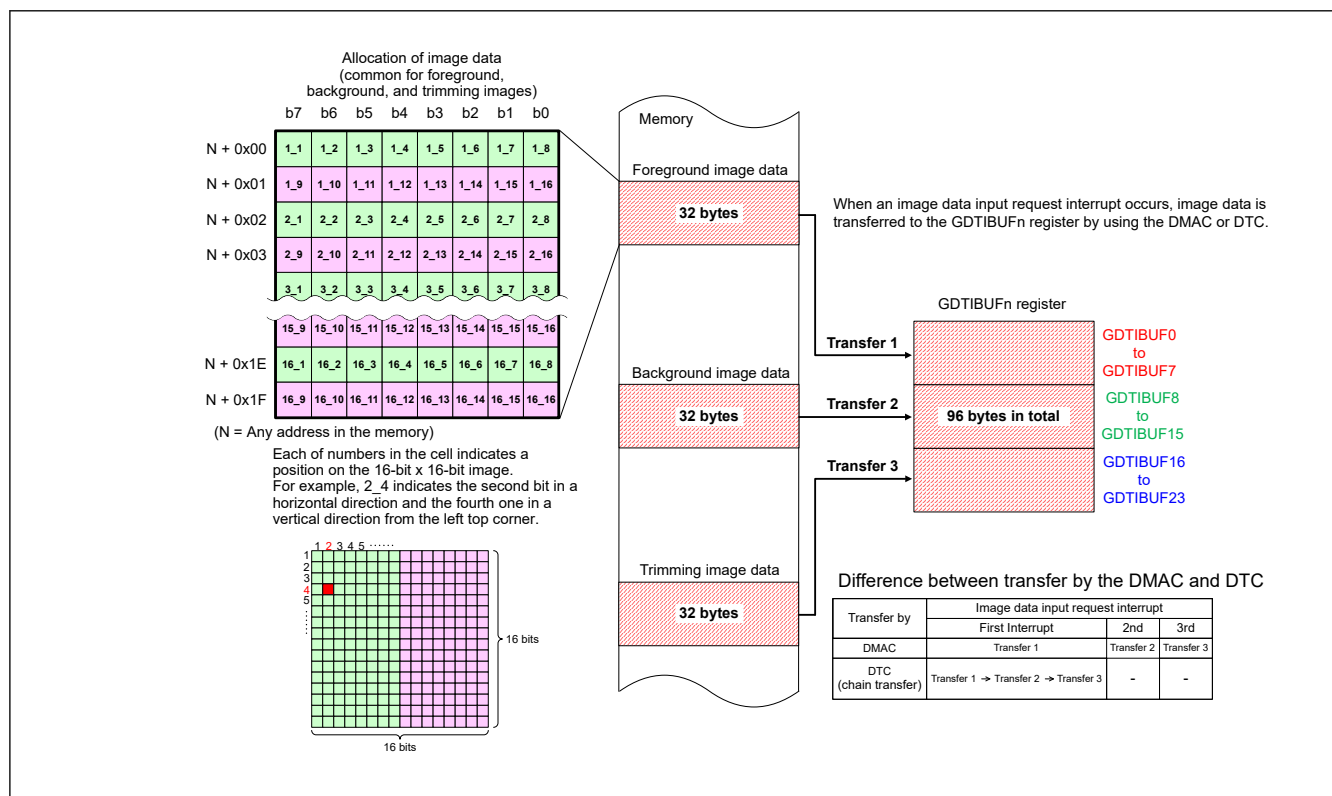


Figure 41.5 Example of the transfer of input image data in monochrome compositing

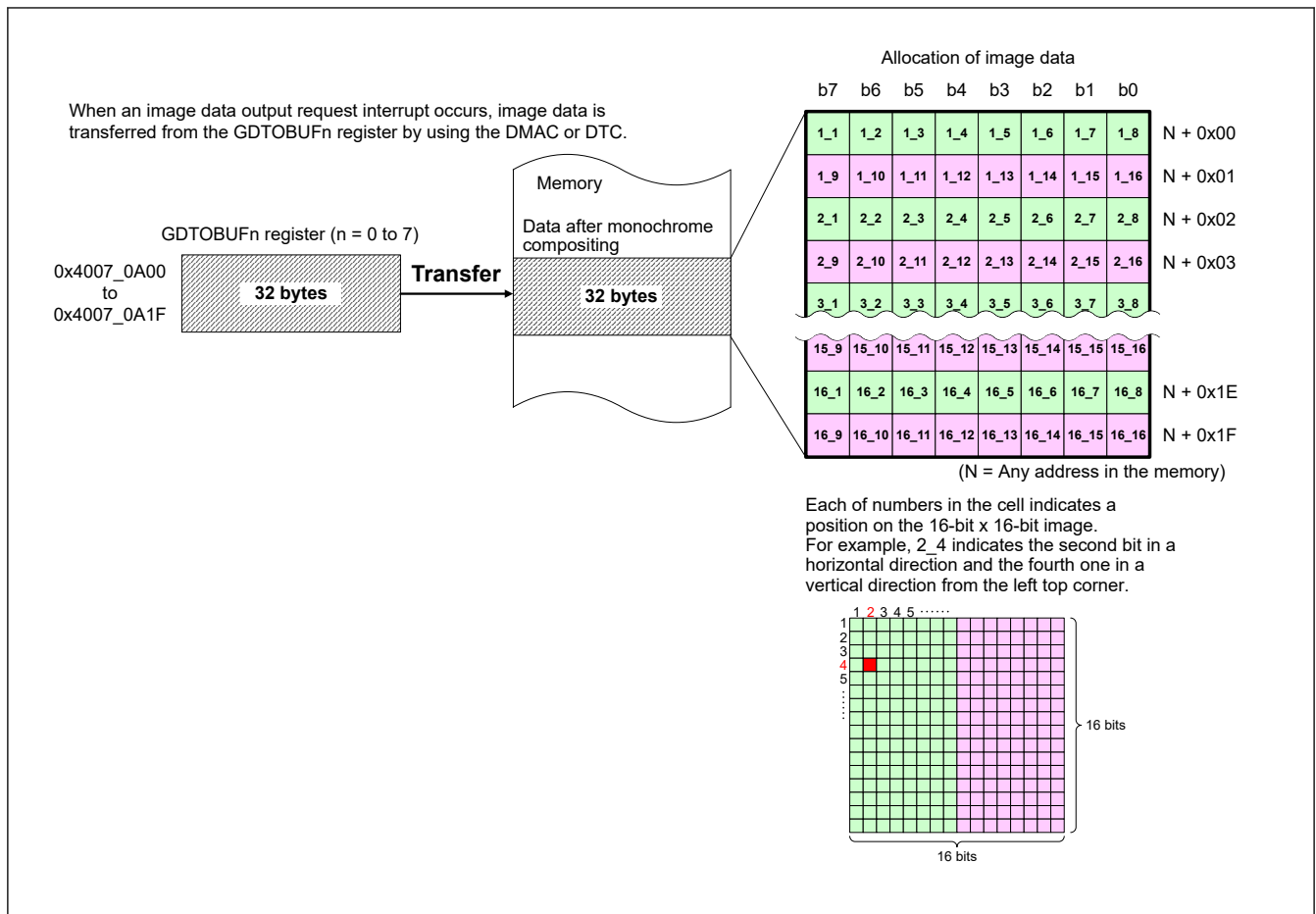


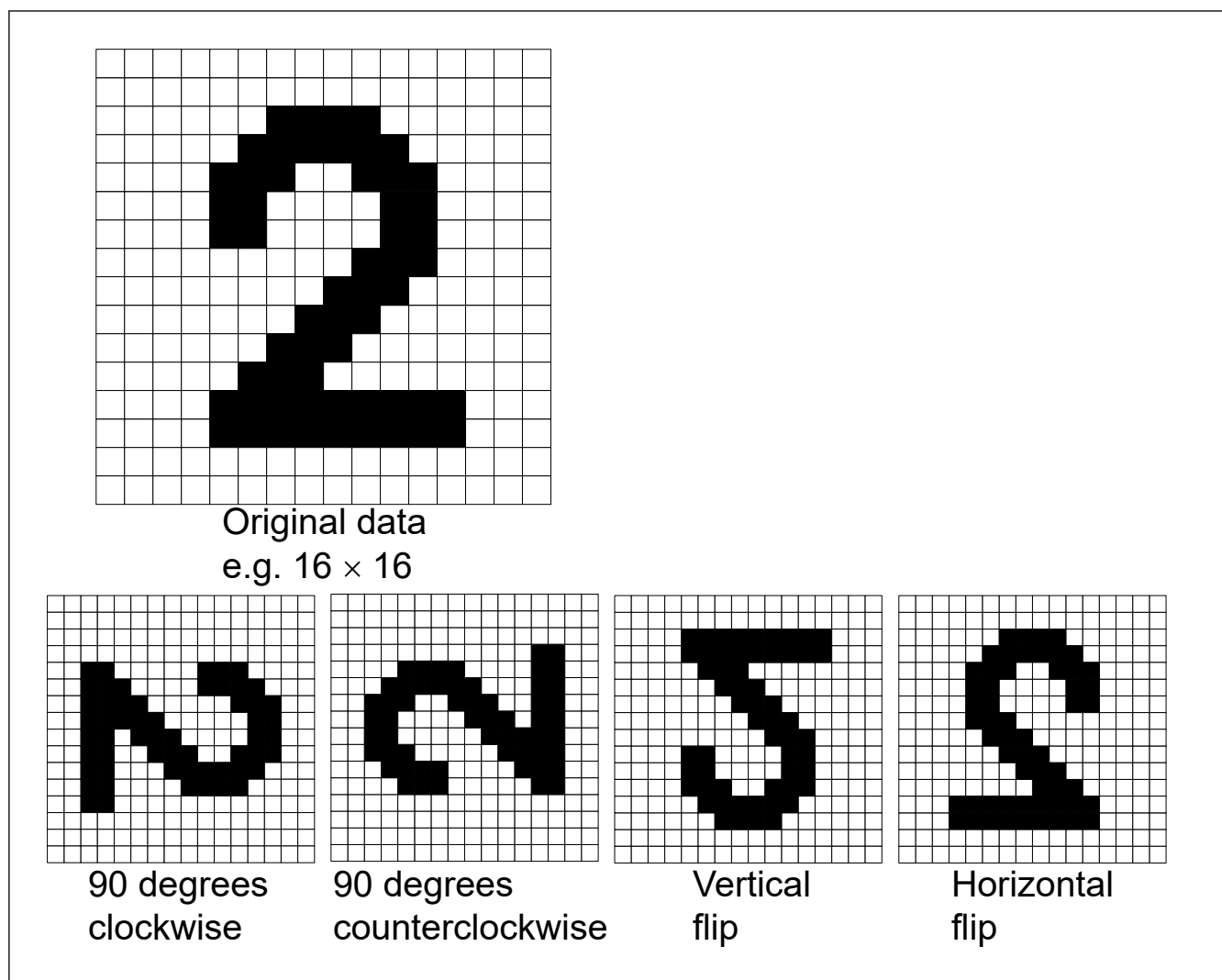
Figure 41.6 Example of the transfer of output image data in monochrome compositing

41.3.2 Rotation

The GDT provides an image rotation function with the following four types of rotation:

- 90-degree clockwise
- 90-degree counterclockwise
- Vertical flip
- Horizontal flip

The 180-degree rotation and the inside-out can be achieved by combining the basic four types of rotation. [Figure 41.7](#) shows a concrete rotation image.

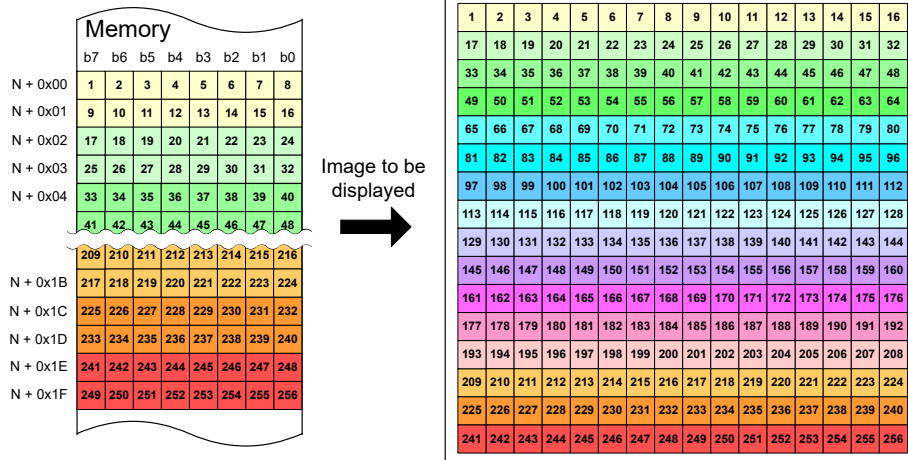
**Figure 41.7** Rotation image

This function involves the reading of data from the GDT image data input registers, rotating of the data that were read as specified in the GDTCR.RTTFC[1:0] bits, and storage of the rotated data in the GDT image data output registers. [Figure 41.8](#) and [Figure 41.9](#) are examples of the allocation of data in the memory and on the LCD before and after the rotation when handling the image data as 16 × 16 bits. N in the given figures indicate any address in the memory. [Table 41.4](#) shows the correspondence between the data before rotation, type of rotation, and the data after rotation.

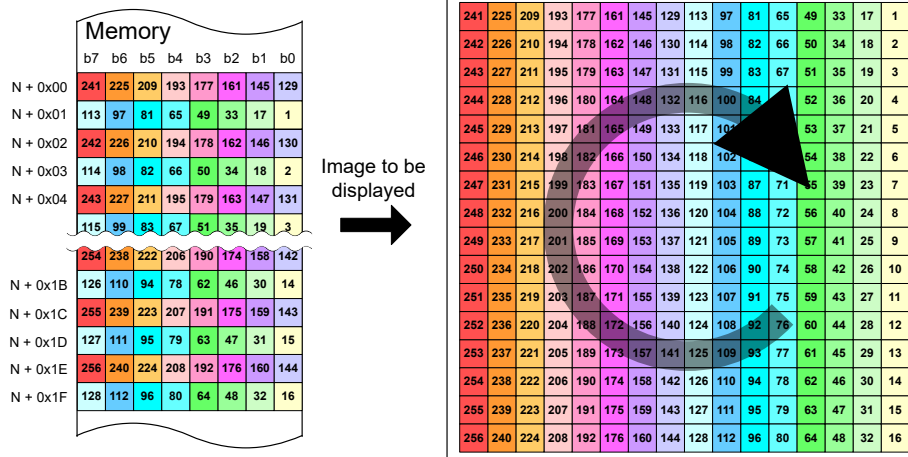
Table 41.4 Correspondence between the data before rotation, type of rotation, and data after rotation

Data before image processing	Details of image processing	Data after image processing
Figure 41.8 (a)	90-degree clockwise	Figure 41.8 (b)
	90-degree counterclockwise	Figure 41.8 (c)
	Horizontal flip	Figure 41.9 (d)
	Vertical flip	Figure 41.9 (e)

(a) Before rotation by the GDT



(b) Rotation by 90 degrees clockwise



(c) Rotation by 90 degrees counterclockwise

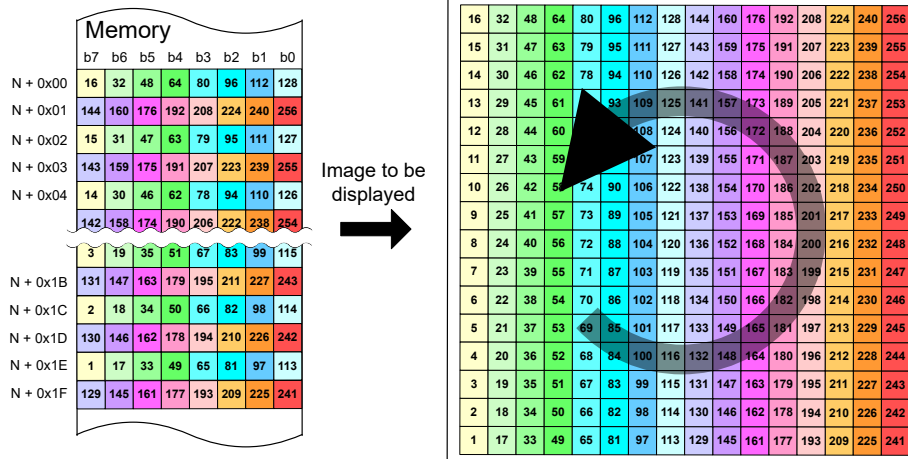


Figure 41.8 Example of the allocation of data in the memory and on the LCD before and after rotation (1/2)

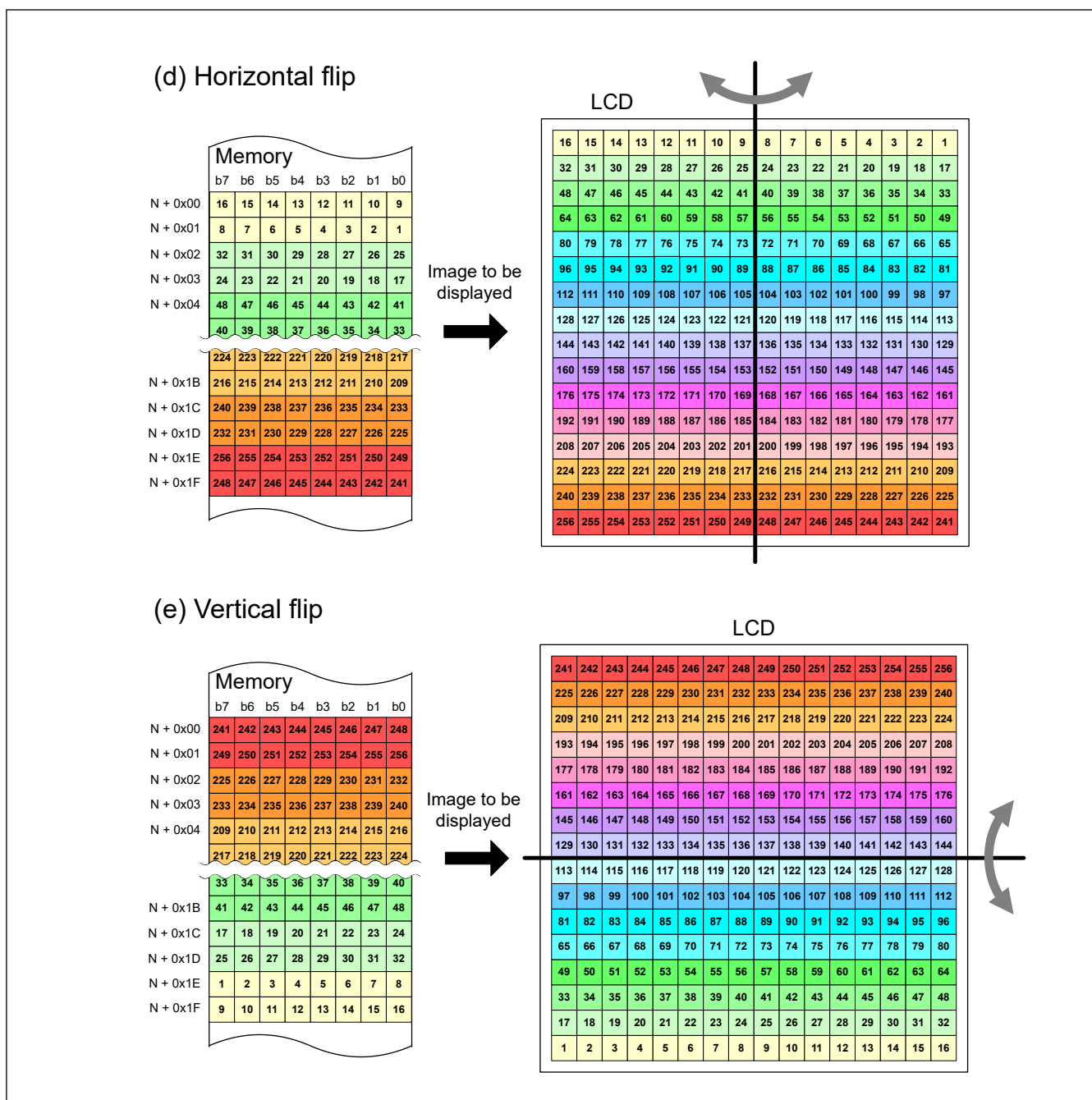


Figure 41.9 Example of the allocation of data in the memory and on the LCD before and after rotation (2/2)

The following shows the procedure for rotating an image by 90 degrees clockwise (as in Figure 41.8 (a)) and displaying it on the LCD (on the right side of Figure 41.8 (b)).

1. The image data (on the left side of Figure 41.8 (a)) are written to the GDT image data input registers by using the DMAC or DTC.
2. The GDT reads the data from the GDT image data input registers, and outputs data for an image which has been rotated by 90 degrees clockwise (on the left side of Figure 41.8 (b)) to the GDT image data output registers.
3. The data in the GDT image data output registers are read by using the DMAC or DTC. When the image is displayed, the image on the LCD will have been rotated.

Table 41.5 lists the GDT image data input registers and GDT image data output registers to be used in rotation.

Table 41.5 Registers used in rotation

Input image size (bit)	Image data input request interrupt				Output image size (bit)	Image data output request interrupt		
	Interrupt mode	Number of interrupts	Transfer for one interrupt (bytes)	GDT Image Data Input Register		Number of interrupts	Transfer for one interrupt (byte)	GDT Image Data Output Register
8 × 8	—*1	1	8	First interrupt: GDTIBUF0 and GDTIBUF1	8 × 8	1	8	First interrupt: GDTIBUF0 and GDTIBUF1
16 × 16	—*1	1	32	First interrupt: GDTIBUF0 to GDTIBUF7	16 × 16	1	32	First interrupt: GDTIBUF0 to GDTIBUF7

Note 1. The number of interrupts, amount transferred per interrupt, and registers used are the same regardless of the interrupt mode setting.

Figure 41.10 shows an example of a flow of rotating the 16 × 16-bit images N consecutive times with data transfer handled by the DMAC. In this example, DMAC channel 0 is used to transfer the input image data from the memory to the GDTIBUFn registers. DMAC channel 1 is used to transfer the output image data from the GDTIBUFn registers to the memory. Each of the shaded parts in the figure shows the flow of transfers for a single round of rotation. A single round of rotation is handled in the following order from step 1 to step 5.

1. The GDT issues an image data input request interrupt for the CPU.
2. The CPU re-sets the information required for transfer by the DMAC and then issues a DMA transfer request.
3. The DMAC transfers the input image data from the memory to the GDTIBUFn registers.
4. The GDT applies rotation to the data stored in the GDTIBUFn registers, stores the rotated data in the GDTIBUFn registers, and issues an image data output request interrupt as a trigger for the DMAC.
5. The DMAC transfers the output image data from the GDTIBUFn registers to the memory.

When rotations have been completed N times, the DMAC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the CPU, which does not, however, trigger the DMAC to transfer input data since processing the predetermined number of times has been completed.

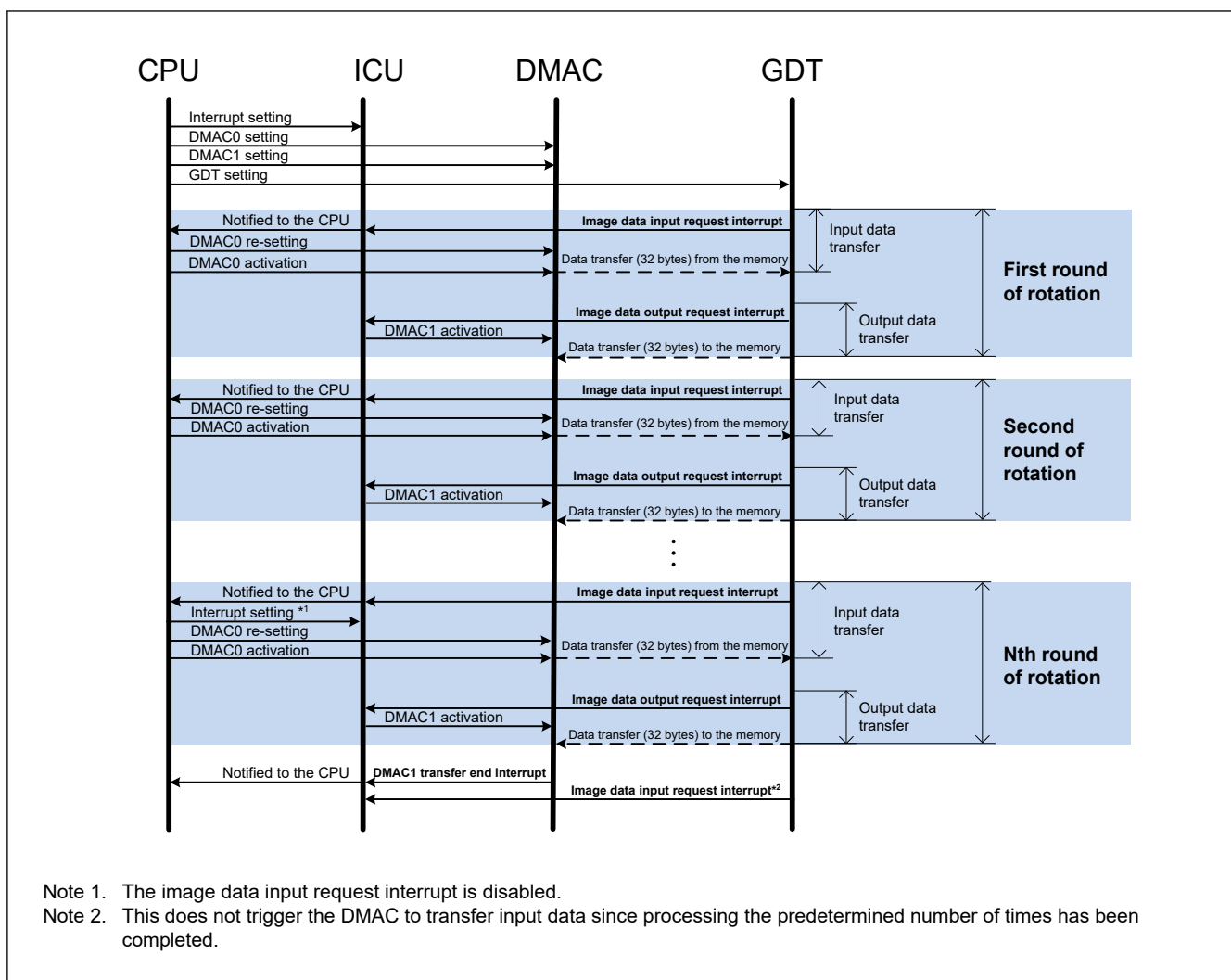


Figure 41.10 Example of a flow of rotation with the use of the DMAC for data transfer requiring CPU intervention for control

Figure 41.11 shows an example of a flow of rotating the 16×16 -bit images N consecutive times when data transfer is handled by the DTC. Each of the shaded parts in the figure shows the flow of transfers for a single round of rotation. A single round of rotation is handled in the following order from step 1 to step 4.

1. The GDT issues an image data input request interrupt to the DTC.
2. The DTC transfers the input image data from the memory to the GDTIBUFn registers by chain transfer.
3. The GDT rotates the data stored in the GDTIBUFn registers, stores the rotated data in the GDTIOBUFn registers, and issues an image data output request interrupt as a trigger for the DTC.
4. The DTC transfers the output image data from the GDTIOBUFn registers to the memory.

When rotations of N times are completed, the DTC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the DTC, which, however, does not trigger the DTC to transfer the input data since processing of predetermined times has been completed.

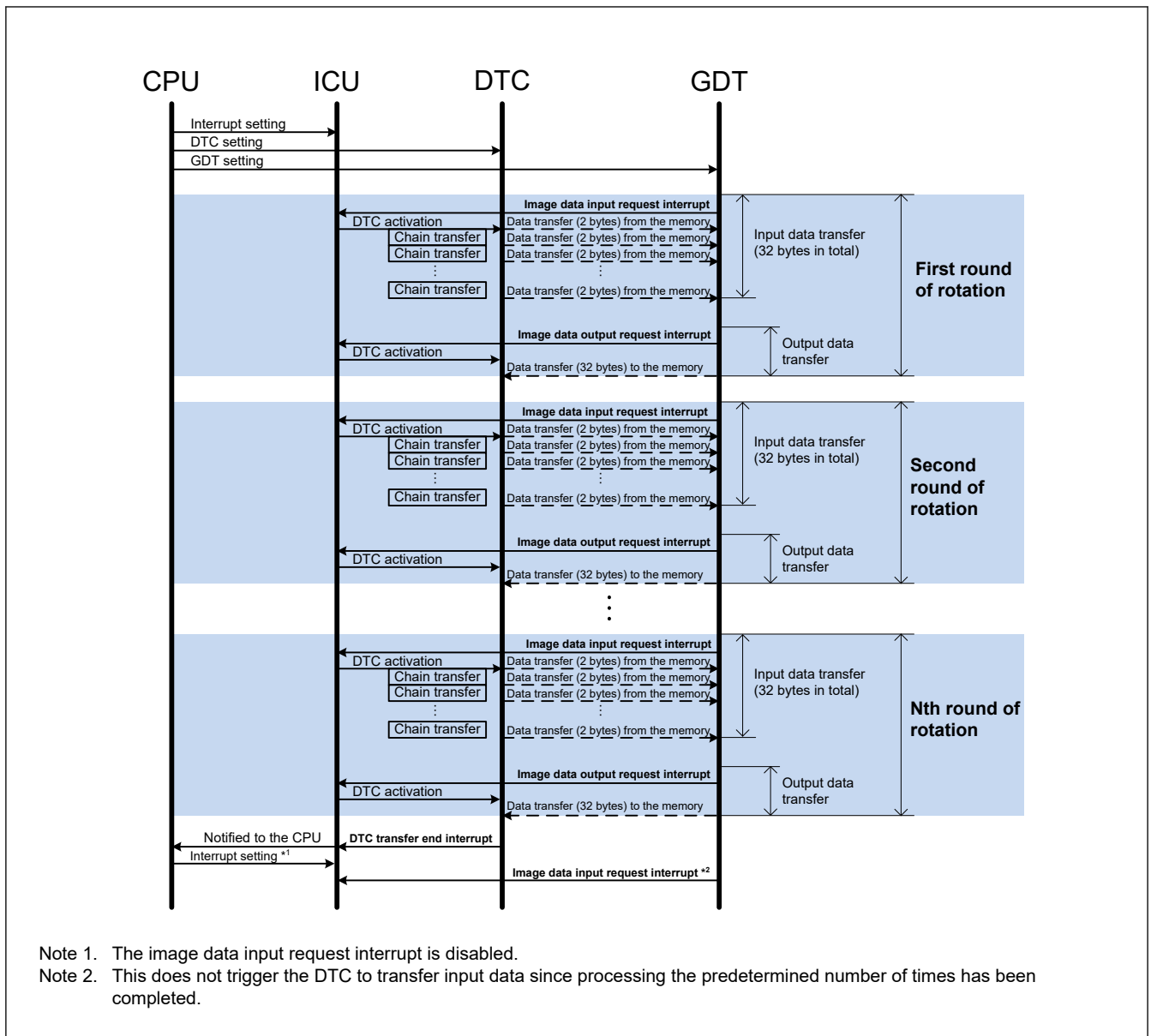


Figure 41.11 Example of a flow of Rotation with the use of the DTC for data transfer

41.3.3 Scaling Down

The GDT supports the scaling down of images. The following two types of scaling down (in both dimensions) are available:

- Pixel averaging method (reduction to 1/8, 2/8, 3/8, 4/8, 5/8, 6/8, and 7/8 is available)
- Inter-bit thinning method

The unit of image processing for this function is 8 bytes. Scaling down an image always produces some unused bits. For example, if an image is scaled down to 1/8, only 1 bit of pixel data from each byte remains in the scaled-down image. Processing eight consecutive 8-byte images produces an 8-bit scaled-down image, which can be transferred to the LCD in bytes. The number of images to be consecutively scaled down can be set within the range from 1 to 8 by using the GDTSCR.SHRKNUM[2:0] bits. The initial setting is for the GDT to process eight consecutive 8-byte images.

41.3.3.1 Pixel Averaging

Pixel averaging is an algorithm in which each pixel is considered as an area, and values obtained by multiplying the original setting values by the respective area occupancies per bit are accumulated to determine the value after scaling down. This function can only be used for monochrome images.

Figure 41.12 shows an example of scaling down to 6/8. In this example, the original image with a size of 8×8 bits is scaled down to 6/8, i.e. eight pixels are divided up for scaling down to produce six pixels. The dotted lines in the image at left below indicate the division to obtain six pixels. The area occupancy per bit in the areas as divided to produce six pixels are calculated. The following is an example of the calculation for pixel (1) in the image at right.

$$(1) = D1 \times 36/64 + D2 \times 12/64 + D9 \times 12/64 + D10 \times 4/64$$

The value of pixel (1) is determined as 0 or 1 from the result of the above equation. If the result is 0.5, it is rounded up to 1. The writing of 0 or 1 to the unused bits after the scaling down is selectable by using the GDTSCR.SBCS bit.

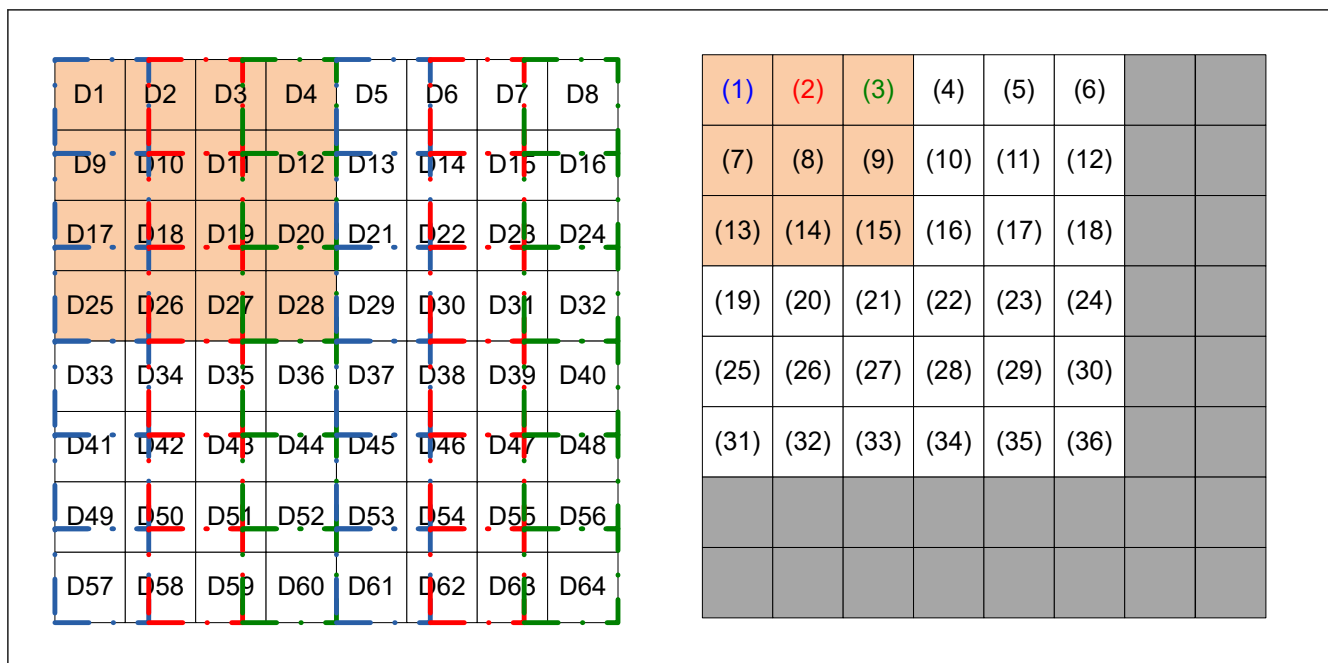


Figure 41.12 Example of scaling down an image to 6/8 by pixel averaging

Table 41.6 lists the GDT image data input registers and GDT image data output registers to be used in pixel averaging. The number of scaling down images setting bits (SHRNKNUM[2:0]) in the GDTSCR register determines the number of image data input request interrupts to be allowed.

Table 41.6 Registers used in scaling down by pixel averaging

Input image size (bit)	Image data input request interrupt				Output image size (bits)	Image data output request interrupt		
	Interrupt mode	Number of interrupts	Transfer for one interrupt (bytes)	GDT Image Data Input Register		Number of interrupts	Transfer for one interrupt (bytes)	GDT Image Data Output Register
8 × 8	— ^{*1}	GDTSCR.S HRNKNUM[2:0]	8	First interrupt: GDTIBUF0 and GDTIBUF1 ^{*2}	Scaled down to 1/8: 1 × 1	1	1	First interrupt: GDTOBUF0
					Scaled down to 2/8: 2 × 2	2	2	First interrupt: GDTOBUF0 2nd: GDTOBUF4
					Scaled down to 3/8: 3 × 3	3	3	First interrupt: GDTOBUF0 2nd: GDTOBUF4 3rd: GDTOBUF8
					Scaled down to 4/8: 4 × 4	4	4	First interrupt: GDTOBUF0 2nd: GDTOBUF4 3rd: GDTOBUF8 4th: GDTOBUF12
					Scaled down to 5/8: 5 × 5	5	5	First interrupt: GDTOBUF0 and GDTOBUF1 2nd: GDTOBUF4 and GDTOBUF5 3rd: GDTOBUF8 and GDTOBUF9 4th: GDTOBUF12 and GDTOBUF13 5th: GDTOBUF16 and GDTOBUF17
					Scaled down to 6/8: 6 × 6	6	6	First interrupt: GDTOBUF0 and GDTOBUF1 2nd: GDTOBUF4 and GDTOBUF5 3rd: GDTOBUF8 and GDTOBUF9 4th: GDTOBUF12 and GDTOBUF13 5th: GDTOBUF16 and GDTOBUF17 6th: GDTOBUF20 and GDTOBUF21
					Scaled down to 7/8: 7 × 7	7	7	First interrupt: GDTOBUF0 and GDTOBUF1 2nd: GDTOBUF4 and GDTOBUF5 3rd: GDTOBUF8 and GDTOBUF9 4th: GDTOBUF12 and GDTOBUF13 5th: GDTOBUF16 and GDTOBUF17 6th: GDTOBUF20 and GDTOBUF21 7th: GDTOBUF24 and GDTOBUF25

Note 1. The number of interrupts, amount transferred per interrupt, and registers used are the same regardless of the interrupt mode setting.

Note 2. The same GDT image data input registers are used each time an interrupt occurs.

Access to the GDT image data output registers in byte units is possible if this is required after scaling down. [Figure 41.13](#) shows an example of data allocation in the memory and registers after scaling down the data for an 8-byte image to 1/8, producing 1 bit of image data. The output data that have been scaled down to be 1 bit can be read as a byte. In such a case, the valid value is that of the lowest-order bit of the given register. Accordingly, read the address 0x4007_0A00 in the GDTOBUF0 register as a byte.

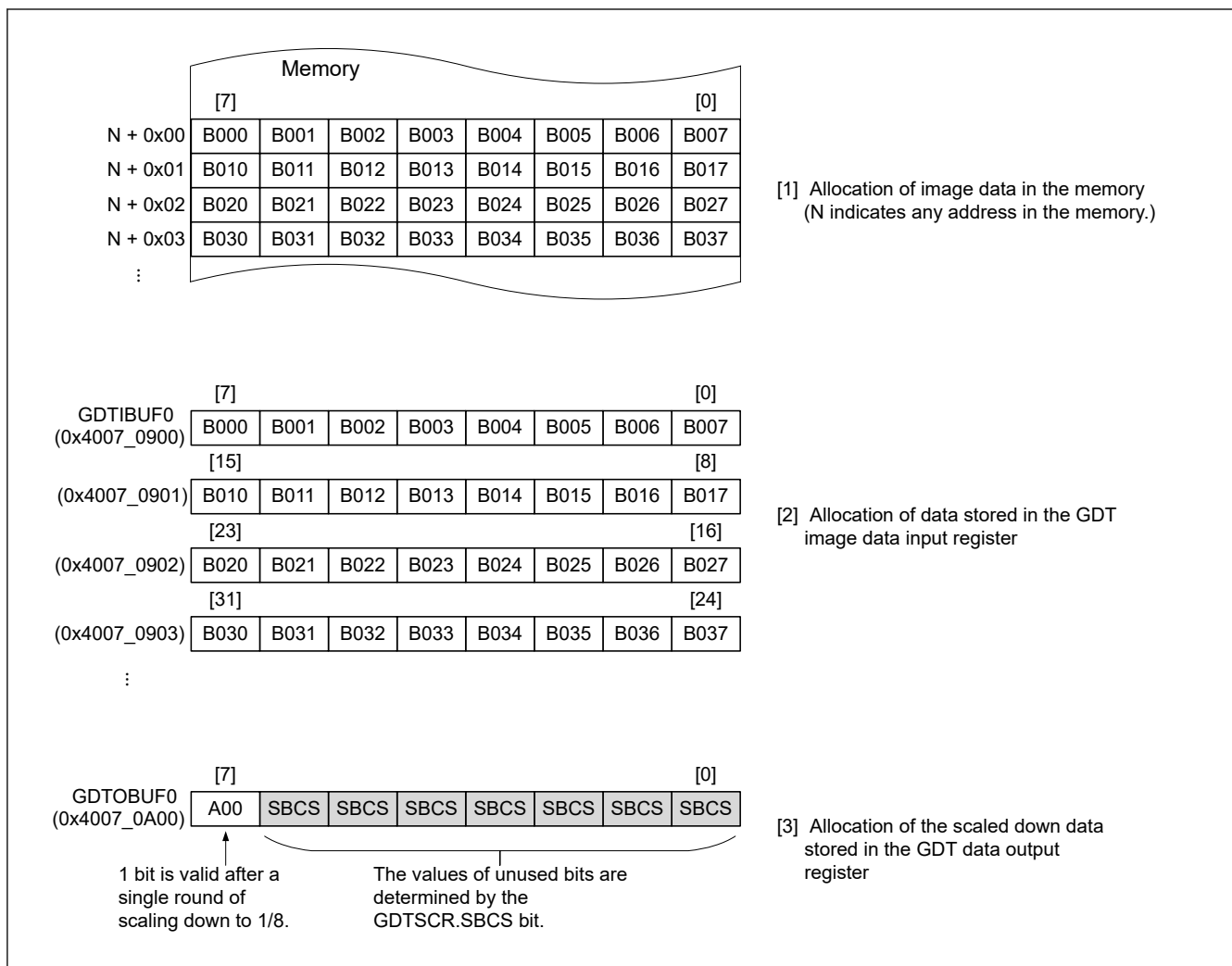


Figure 41.13 Example of the allocation of image data to the memory and registers

Figure 41.14 to Figure 41.16 show examples of scaling down 176-bit horizontal data (22 bytes) eight consecutive times. Figure 41.14 shows an example of the allocation of 22 bytes × 8 lines of data to the memory. Scaling down is applied to the stored data in MSB-first order from the location at upper left. Figure 41.15 shows an example of the allocation of data before scaling down as it is displayed on the LCD. Up to 22 bytes of data can be allocated to a line on the LCD. Figure 41.16 shows an example of the allocation of the data scaled down by 3/8 and stored in the GDT image data output registers.

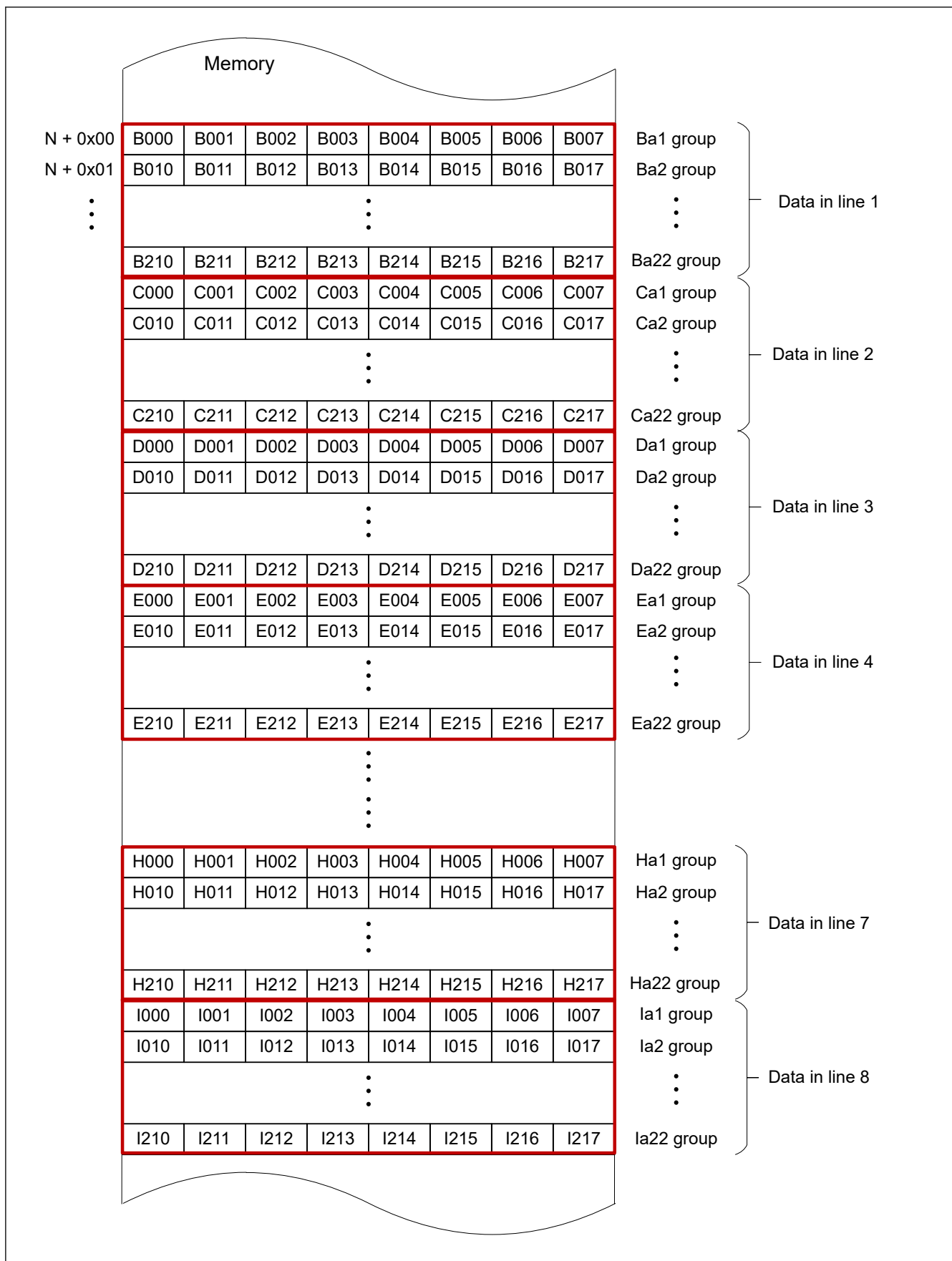


Figure 41.14 Example of the allocation of eight lines of image data to the memory

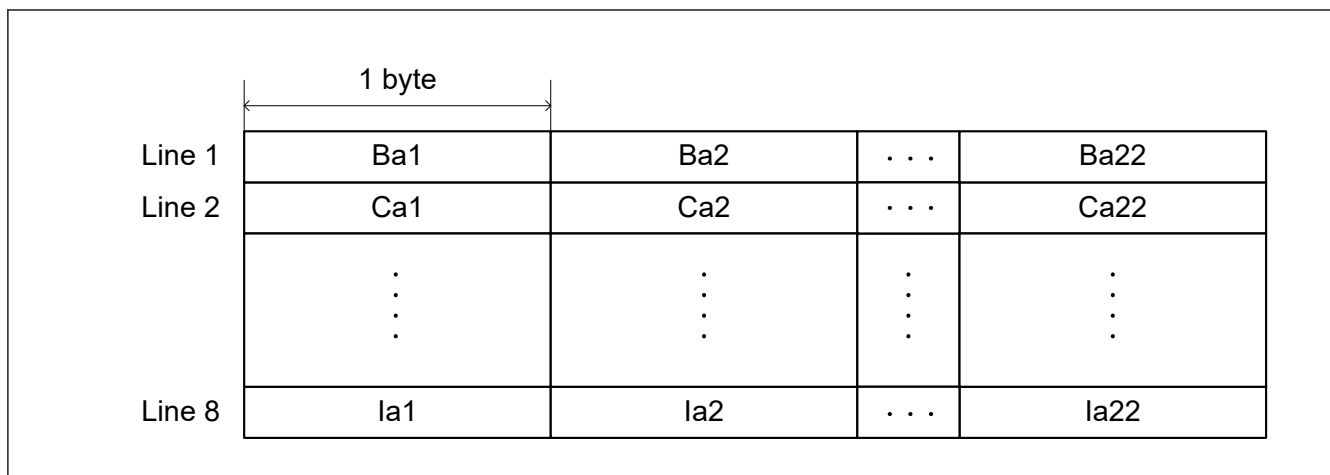


Figure 41.15 Example of the allocation of eight lines of image data to the LCD

The 8×8 -bit image data are scaled down to $3/8$ to produce a 3×3 -bit image. The data within the broken lines in Figure 41.16 are those after scaling down and are stored in the GDT image output registers in the format shown in the figure. The data represented by Ba1 to Ia1 in Figure 41.15 are the first column 8×8 -bit data. These are scaled down and stored in the bits listed below of the GDT image data output registers at the following addresses.

- bits [7:5] (bits [31:29] of GDTBUF0) at address 0x4007_0A00
- bits [7:5] (bits [31:29] of GDTBUF4) at address 0x4007_0A10
- bits [7:5] (bits [31:29] of GDTBUF8) at address 0x4007_0A20

Because up to eight units of input data can be reduced in one GDT processing, eight units of reduced image data are sequentially registered as shown in Figure 41.16. Similarly, data of eight lines of an LCD can be reduced to $3/8$ by reducing images Ba2 to Ia2 to images Ba22 to Ia22.

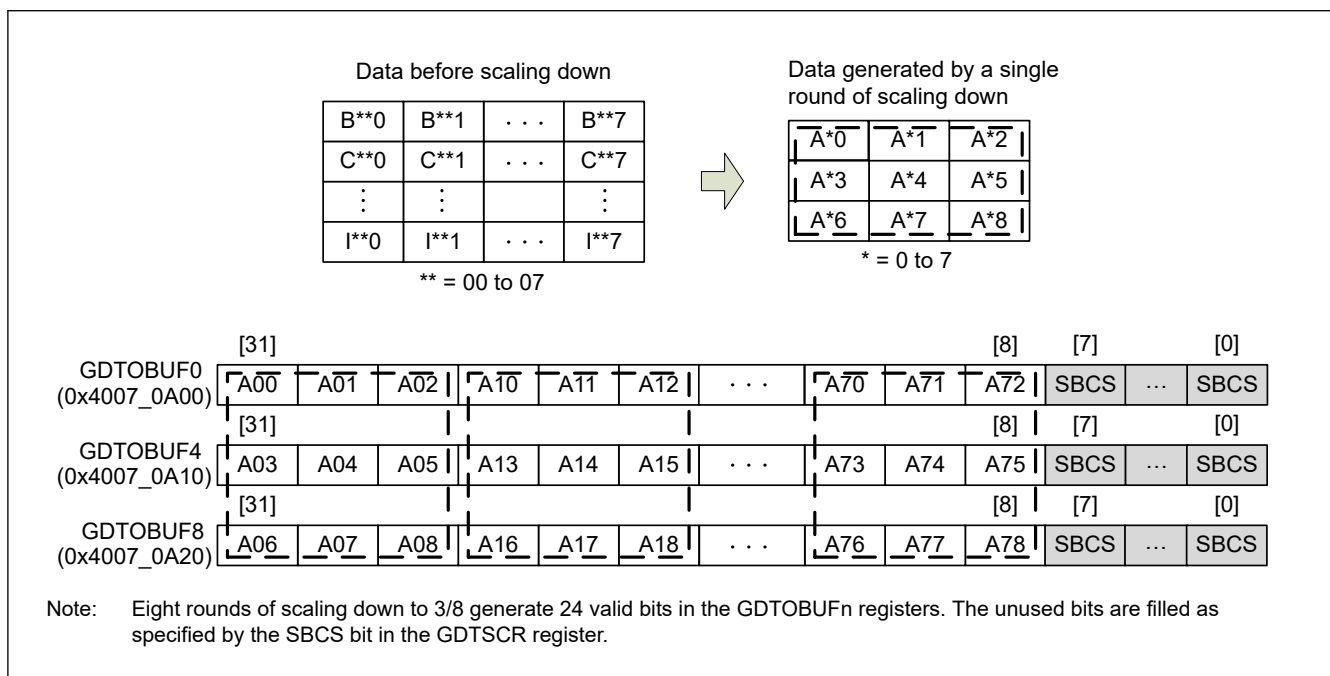
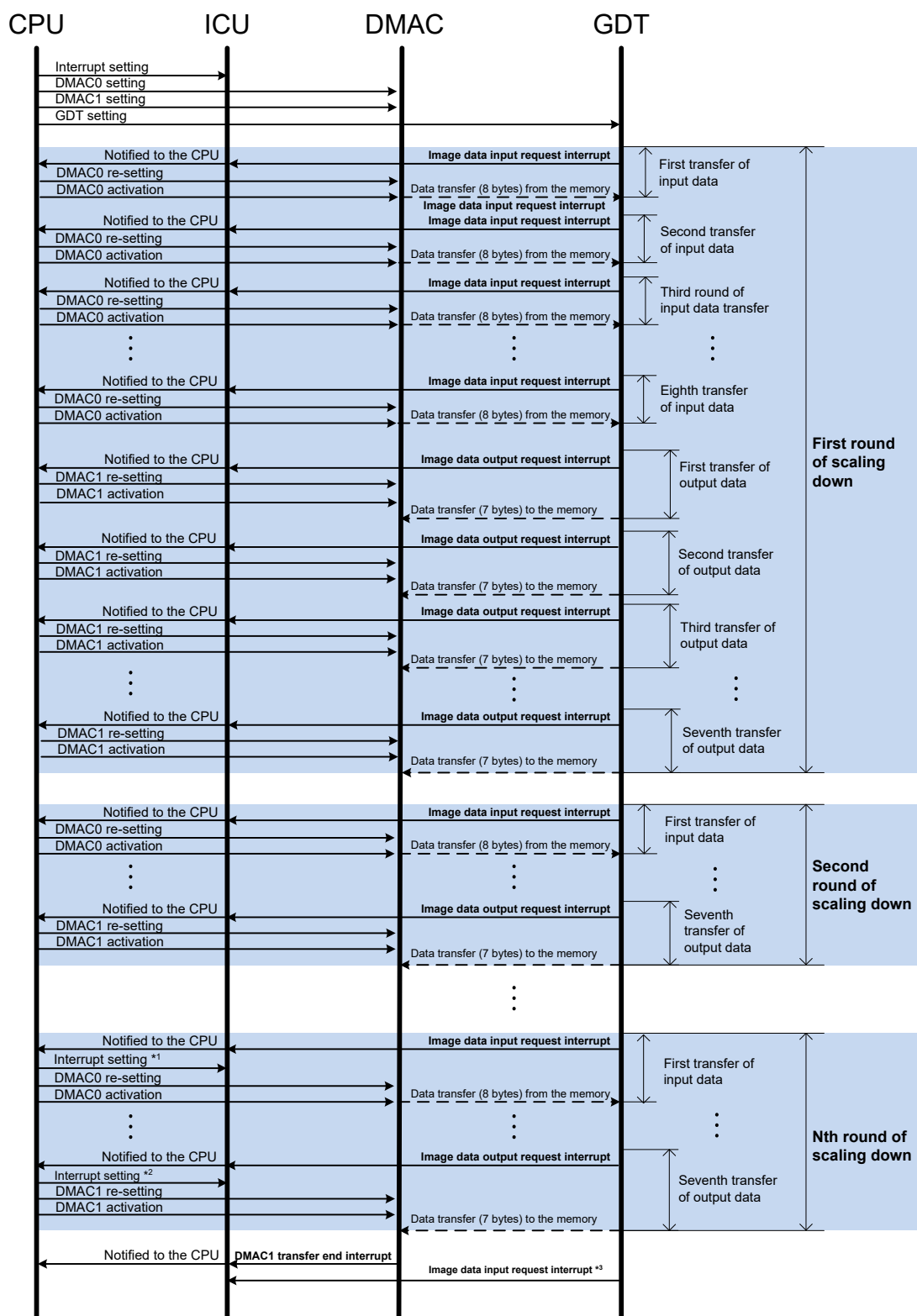


Figure 41.16 Example of the data stored in the GDT image data output registers after scaling down to $3/8$

Figure 41.17 shows an example of a flow of scaling down eight images to $7/8$ in a single round of processing and repeating it N consecutive times when data transfer is handled by the DMAC. In this example, DMAC0 is used for transfer of the input image data from the memory to the GDTIBUF_n registers. DMAC1 is used for transfer of the output image data from the GDTIBUF_n registers to the memory. Intervention by the CPU is required for notification of image data input request interrupts and image data output request interrupts. Each of the shaded parts in the figure shows a flow of transfers for a single round of scaling down. A single round of scaling down is executed in the following order from step 1 to step 9.

1. The GDT issues an image data input request interrupt to the CPU.
2. The CPU re-sets the information required for transfer by the DMAC and then issues a DMA transfer request.
3. The DMAC transfers the input image data from the memory to the GDTIBUFn registers.
4. The GDT scales down the data stored in the GDTIBUFn registers, stores the scaled down data in the GDTIOBUFn registers, and issues an image data output request interrupt as a trigger for the DMAC.
5. Steps 1 to 4 are repeated eight times.
6. The GDT issues an image data output request interrupt to the CPU.
7. The CPU re-sets the information required for transfer by the DMAC and then issues a DMA transfer request.
(For seventh output data transfer of Nth reduction processing, a DMAC transfer end notification is set before DMAC is reset.)
8. The DMAC transfers the output image data from the GDTIOBUFn registers to the memory.
9. Steps 6 to 8 are repeated seven times.

When the scaling down of N times are completed, the DMAC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the CPU, which, however, does not trigger the DMAC to transfer the input data since processing of predetermined number of times has been completed.



Note 1. The image data input request interrupt is disabled.

Note 2. The DMAC1 transfer end interrupt is enabled.

Note 3. This does not trigger the DMAC to transfer input data since processing the predetermined number of times has been completed.

Figure 41.17 Example of a flow of scaling down with the use of the DMAC for data transfer requiring CPU intervention for control

Figure 41.18 shows an example of a flow of scaling down eight images to 7/8 in a single round of processing and repeating it N consecutive times when data transfer is handled by the DTC. Each of the shaded parts in the figure shows a flow of transfers for a single round of scaling down. A single round of scaling down is handled in the following order of step 1 to step 6.

1. The GDT issues an image data input request interrupt to the DTC.
2. The DTC transfers the input image data from the memory to the GDTIBUFn registers by chain transfer.
3. The GDT applies scaling down to the data stored in the GDTIBUFn registers, stores the scaled down data in the GDTOBUFn registers, and issues an image data output request interrupt to the DTC.
4. Steps 1 to 3 are repeated eight times.
5. The GDT issues an image data output request interrupt to the DTC.
6. The DTC transfers the output image data from the GDTOBUFn registers to the memory.

When scaling down of N times is completed, the DTC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the DTC, which, however, does not trigger the DTC to transfer the input data since processing of predetermined times has been completed.

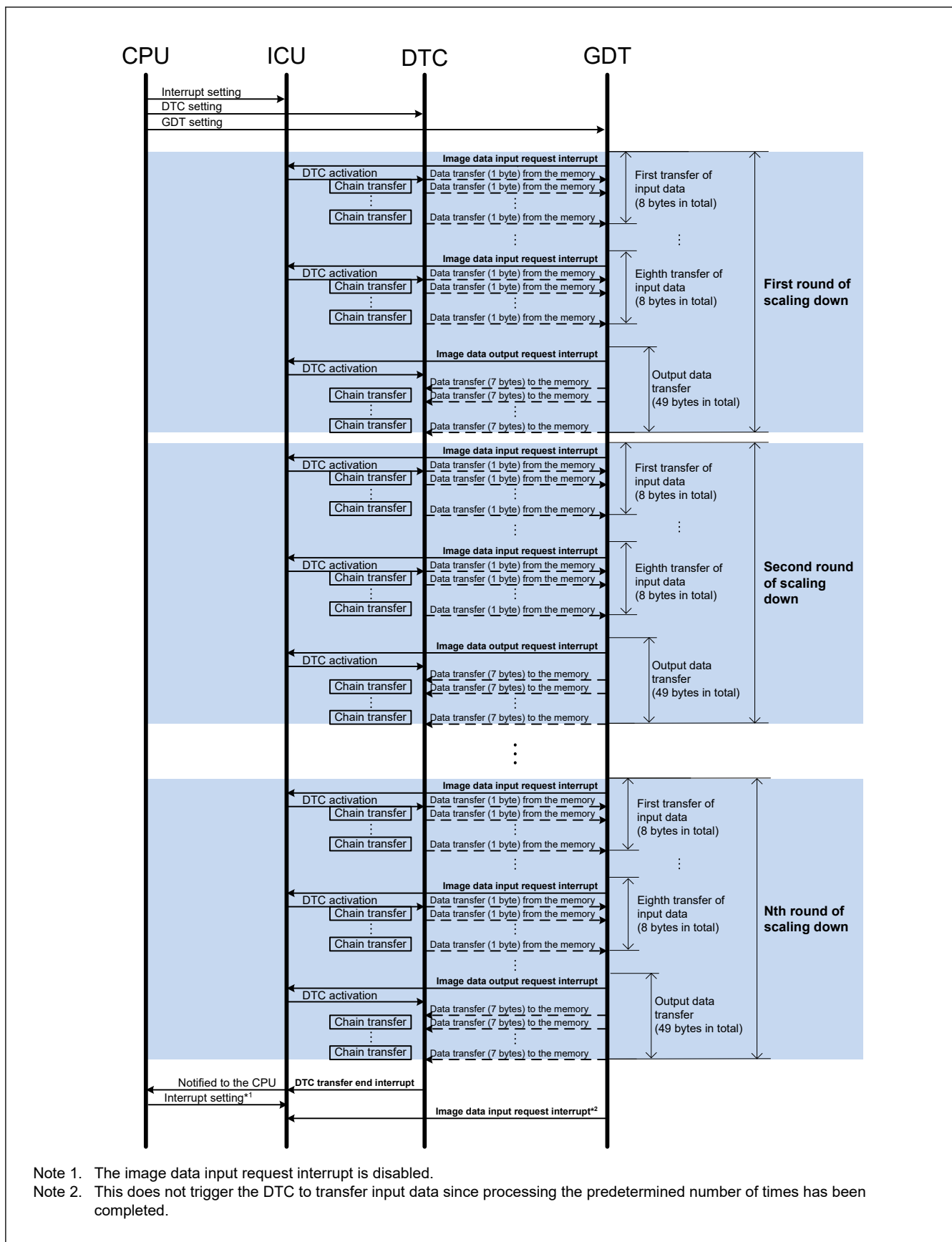


Figure 41.18 Example of a flow of scaling down with the use of the DTC for data transfer

41.3.3.2 Pixel Skipping

With pixel skipping, the scaling-down algorithm can handle monochrome images or color images. The only scaling down ratio is 1/2 (the area is scaled down to 1/4). [Figure 41.19](#) illustrates pixel skipping. In the figure below, (a) represents the original 8-byte image data, and (b) represents the scaled down image. The writing of 0 or 1 to the unused bits after scaling down is selectable by using the GDTSCR.SBCS bit.

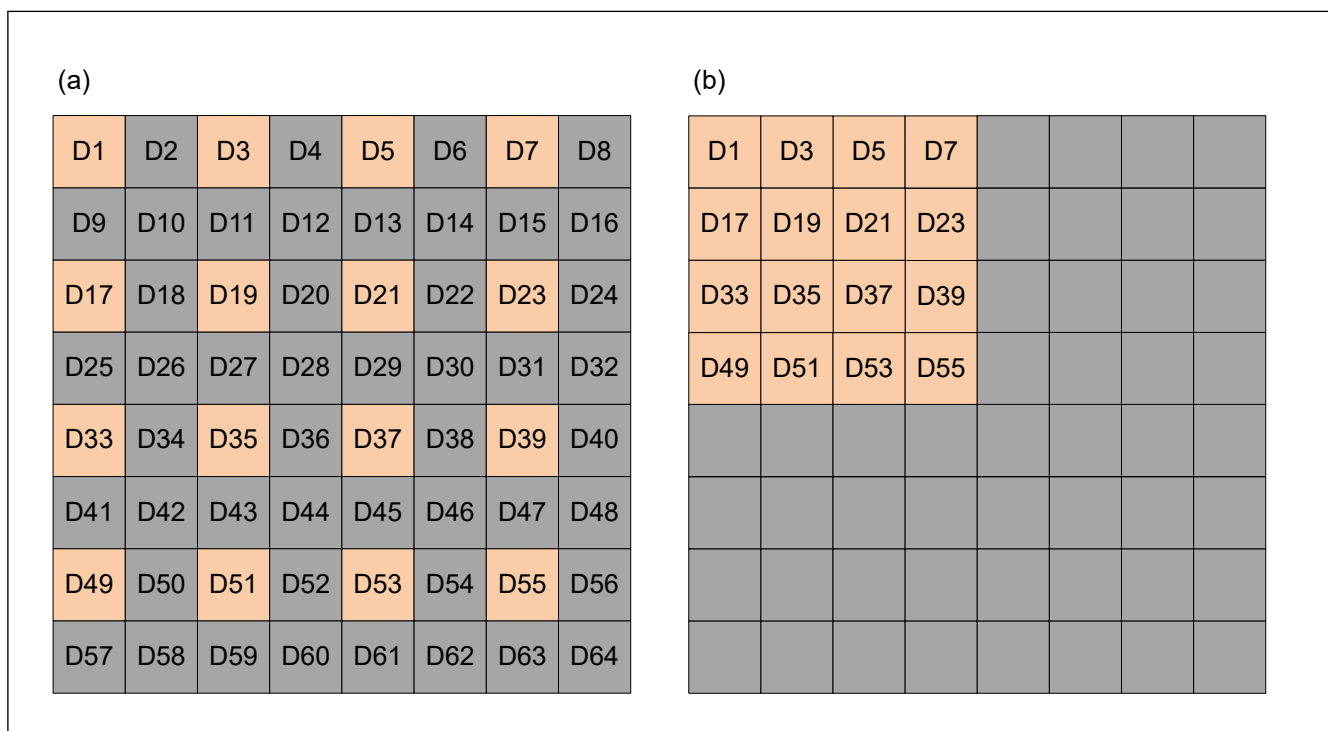


Figure 41.19 Pixel skipping on the LCD

[Table 41.7](#) shows the GDT Image Data Input Register and GDT Image Data Output Register to be used in pixel skipping.

Table 41.7 Registers used in scaling down by pixel skipping

Input image size (bit)	Image data input request interrupt				Output image size (bit)	Image data output request interrupt		
	Interrupt mode	Number of interrupts	Transfer for one interrupt (byte)	GDT Image Data Input Register		Number of interrupts	Transfer for one interrupt (byte)	GDT Image Data Output Register
8 × 8	—*1	GDTSCR.SHRNKNUM[2:0]	8	First interrupt: GDTIBUF0 and GDTIBUF1 *2	4 × 4	4	4	First interrupt: GDTOBUF0 2nd: GDTOBUF4 3rd: GDTOBUF8 4th: GDTOBUF12

Note 1. The number of interrupts, amount transferred per interrupt, and registers used are the same regardless of the interrupt mode setting.

Note 2. The same GDT image data input registers are used each time an interrupt occurs.

41.3.4 Inversion

The GDT supports the bit-wise inversion of images. This function allows the inversion of the bits of pixel data as shown in [Figure 41.20](#). This is the only function that can be executed in combination with other functions. For example, data can be composited, black-and-white inverted, and the result stored in the GDT image data output registers within the same cycle. The function to be used at the same time determines the buffer size to be used.

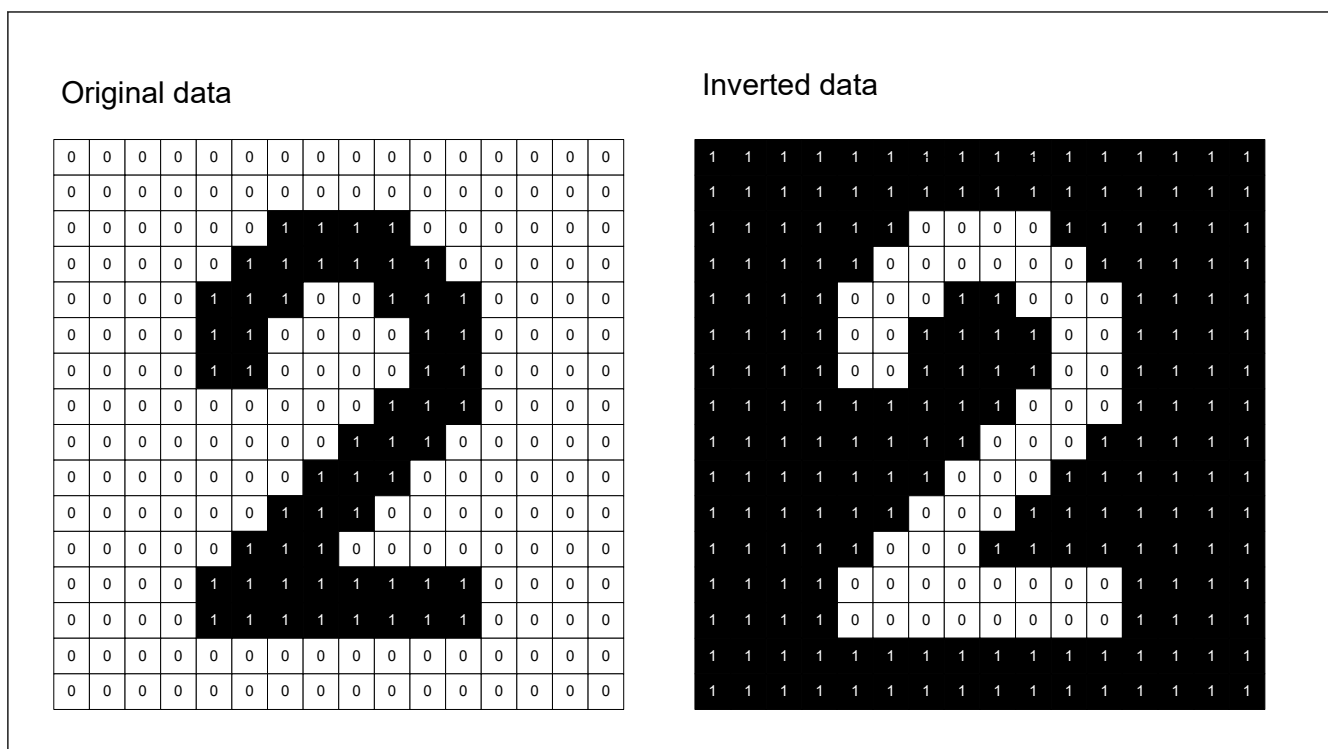


Figure 41.20 Example of inversion

Table 41.8 shows the GDT Image Data Input Register and GDT Image Data Output Register to be used when applying inversion alone.

Table 41.8 Registers used in inversion

Input image size (bit)	Image data input request interrupt				Output image size (bit)	Image data output request interrupt		
	Interrupt mode	Number of interrupts	Transfer for one interrupt (byte)	GDT Image Data Input Register		Number of interrupts	Transfer for one interrupt (byte)	GDT Image Data Output Register
16 × 16	—*1	1	32	First interrupt: GDTIBUF0 to GDTIBUF7	16 × 16	1	32	First interrupt: GDTIBUF0 to GDTIBUF7

Note 1. The number of interrupts, amount transferred per interrupt, and registers used are the same regardless of the interrupt mode setting

Figure 41.21 shows an example of a flow of repeating inversion or endian conversion N consecutive times when data transfer is handled by the DMAC. In this example, DMAC0 is used for transfer of the input image data from the memory to the GDTIBUFn registers. DMAC1 is used for transfer of the output image data from the GDTIBUFn registers to the memory. Each of the shaded parts in the figure shows a flow of transfers for a single round of inversion or endian conversion. A single round of processing of each function is handled in the following order from step 1 to step 4.

1. The GDT issues an image data input request interrupt to the DMAC.
2. The DMAC transfers the input image data from the memory to the GDTIBUFn registers.
3. The GDT applies inversion or endian conversion to the data stored in the GDTIBUFn registers, stores the processed data in the GDTIBUFn registers, and issues an image data output request interrupt as a trigger for the DMAC.
4. The DMAC transfers the output image data from the GDTIBUFn registers to the memory.

When inversion or endian conversion of N times are completed, the DMAC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the DMAC, which, however, does not trigger the DMAC to transfer the input data since processing of predetermined times has been completed.

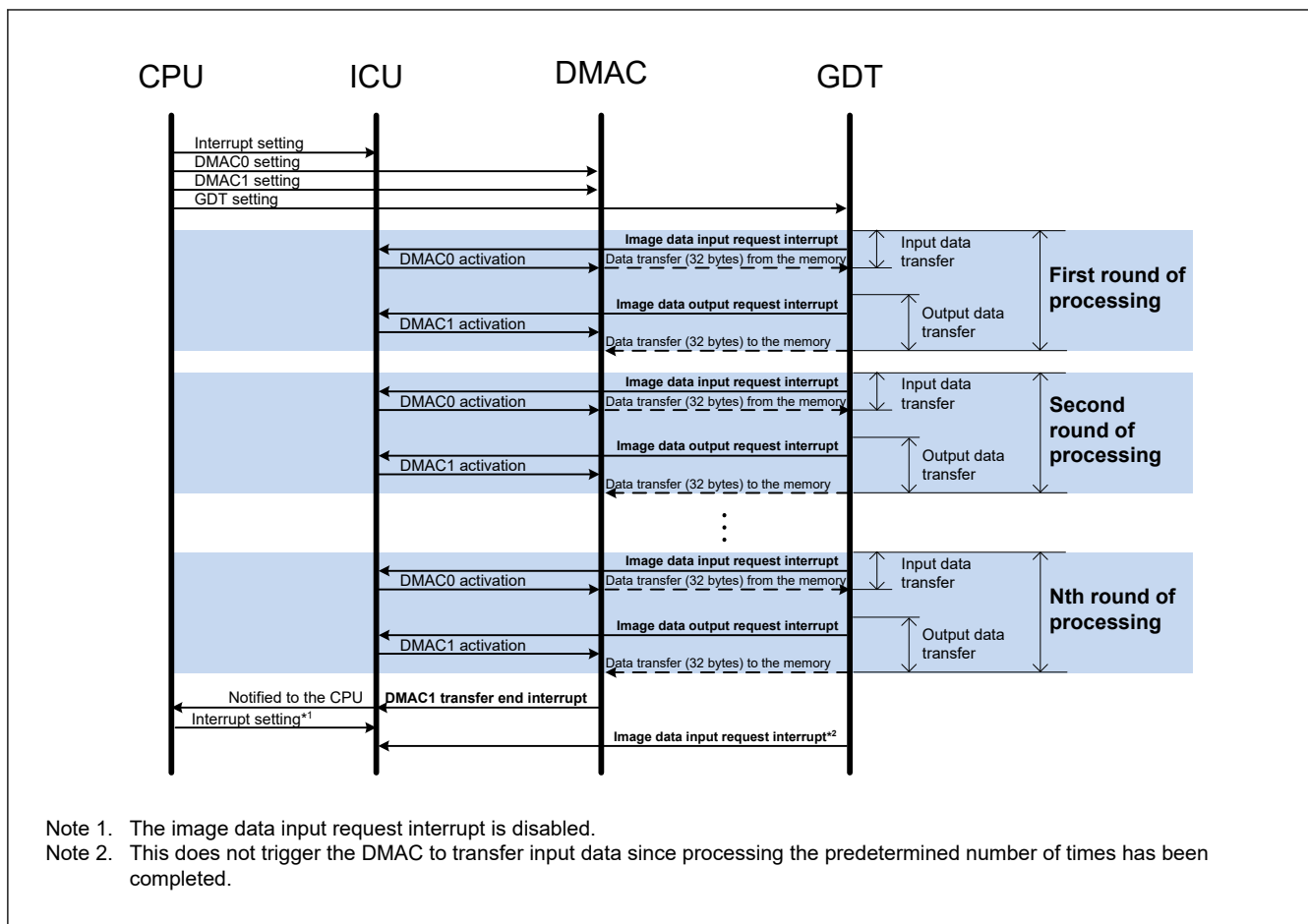


Figure 41.21 Example of a flow with the use of the DMAC for transfer in inversion and endian conversion

Figure 41.22 is an example of a flow of repeating inversion or endian conversion N consecutive times when data transfer is handled by the DTC. Each of the shaded parts in the figure shows a flow of transfers for a single round of inversion or endian conversion. A single round of processing of each function is handled in the following order from step 1 to step 4.

1. The GDT issues an image data input request interrupt to the DTC.
2. The DTC transfers the input image data from the memory to the GDTIBUFn registers.
3. The GDT applies inversion or endian conversion to the data stored in the GDTIBUFn registers, stores the processed data in the GDTIOBUFn registers, and issues an image data output request interrupt as a trigger for the DTC.
4. The DTC transfers the output image data from the GDTIOBUFn registers to the memory.

When inversion or endian conversion of N times are completed, the DTC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the DTC, which, however, does not trigger the DTC to transfer the input data since processing of predetermined times has been completed.

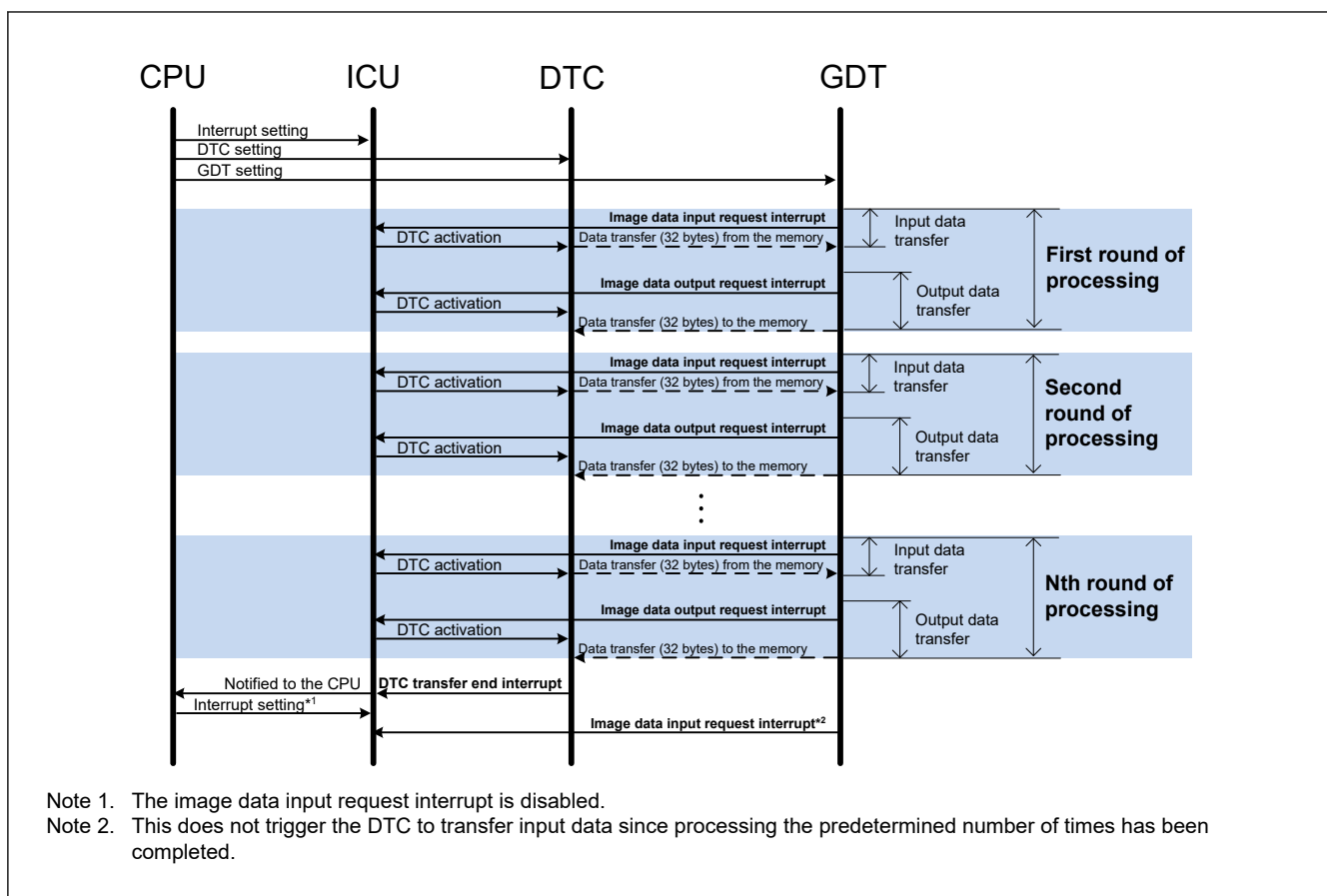


Figure 41.22 Example of a flow with the use of the DTC for data transfer in inversion and endian conversion

41.3.5 Monochrome Compositing

The GDT supports monochrome compositing. A single round of processing allows the input of three types of image data (foreground, background, and border image) and output of the composite image data. Image trimming can be enabled or disabled by the GDTTCR.MBRDEN bit. Only the foreground and background images are input if trimming is disabled. Two image sizes can be handled by this function: 8 bytes (8×8 bits) and 32 bytes (16×16 bits).

Compositing pixels having the value 1

Pixels having the value 1 in the foreground and background images are output as 1. Pixels having the value 1 in the trimming image are output as 0. [Figure 41.23](#) shows an example of compositing pixels having the value 1.

Compositing pixels having the value 0

Pixels having the value 0 in the foreground and background images are output as 0. Pixels having the value 0 in the trimming image are output as 1. [Figure 41.24](#) shows an example of compositing pixels having the value 0.

[Table 41.9](#) and [Table 41.10](#) show the GDT image data input registers and GDT image data output registers to be used in monochrome compositing when image trimming is enabled (GDTTCR.MBRDEN = 1) and disabled (GDTTCR.MBRDEN = 0), respectively.

Table 41.9 Correspondence between registers used in monochrome compositing with image trimming enabled and interrupt mode

Input image size (bit)	Image data input request interrupt				Output image size (bit)	Image data output request interrupt		
	Interrupt mode	Number of interrupts	Transfer for one interrupt (byte)	GDT Image Data Input Register		Number of interrupts	Transfer for one interrupt (byte)	GDT Image Data Output Register
8 × 8	DMAC	3	8	First interrupt: GDTIBUF0 and GDTIBUF1 (foreground image data) 2nd: GDTIBUF8 and GDTIBUF9 (background image data) 3rd: GDTIBUF16 and GDTIBUF17 (trimmed image data)	8 × 8	1	8	First interrupt: GDTIBUF0 and GDTIBUF1
	DTC	1	24	First interrupt: GDTIBUF0 and GDTIBUF1 (foreground image data) 2nd: GDTIBUF8 and GDTIBUF9 (background image data) 3rd: GDTIBUF16 and GDTIBUF17 (trimmed image data)				
16 × 16	DMAC	3	32	First interrupt: GDTIBUF0 to GDTIBUF7 (foreground image data) 2nd: GDTIBUF8 to GDTIBUF15 (background image data) 3rd: GDTIBUF16 to GDTIBUF23 (trimmed image data)	16 × 16	1	32	First interrupt: GDTIBUF0 to GDTIBUF7
	DTC	1	96	First interrupt: GDTIBUF0 to GDTIBUF7 (foreground image data) GDTIBUF8 to GDTIBUF15 (background image data) GDTIBUF16 to GDTIBUF23 (trimmed image data)				

Table 41.10 Correspondence between registers used in monochrome compositing with image trimming disabled and interrupt modes

Input image size (bit)	Image data input request interrupt				Output image size (bit)	Image data output request interrupt		
	Interrupt mode	Number of interrupts	Transfer for one interrupt (bytes)	GDT Image Data Input Register		Number of interrupts	Transfer for one interrupt (bytes)	GDT Image Data Output Register
8 × 8	DMAC	2	8	First interrupt: GDTIBUF0 and GDTIBUF1 (foreground image data) 2nd: GDTIBUF8 and GDTIBUF9 (background image data)	8 × 8	1	8	First interrupt: GDTIBUF0 and GDTIBUF1
	DTC	1	16	First interrupt: GDTIBUF0 and GDTIBUF1 (foreground image data) GDTIBUF8 and GDTIBUF9 (background image data)				
16 × 16	DMAC	2	32	First interrupt: GDTIBUF0 to GDTIBUF7 (foreground image data) 2nd: GDTIBUF8 to GDTIBUF15 (background image data)	16 × 16	1	32	1st: GDTIBUF0 to GDTIBUF7
	DTC	1	64	First interrupt: GDTIBUF0 to GDTIBUF7 (foreground image data) GDTIBUF8 to GDTIBUF15 (background image data)				

Background

[illegible]

Foreground

[illegible]

Trimming

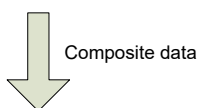
[illegible][illegible]

Figure 41.23 Compositing pixels having the value 1

Compositing pixels having the value 0

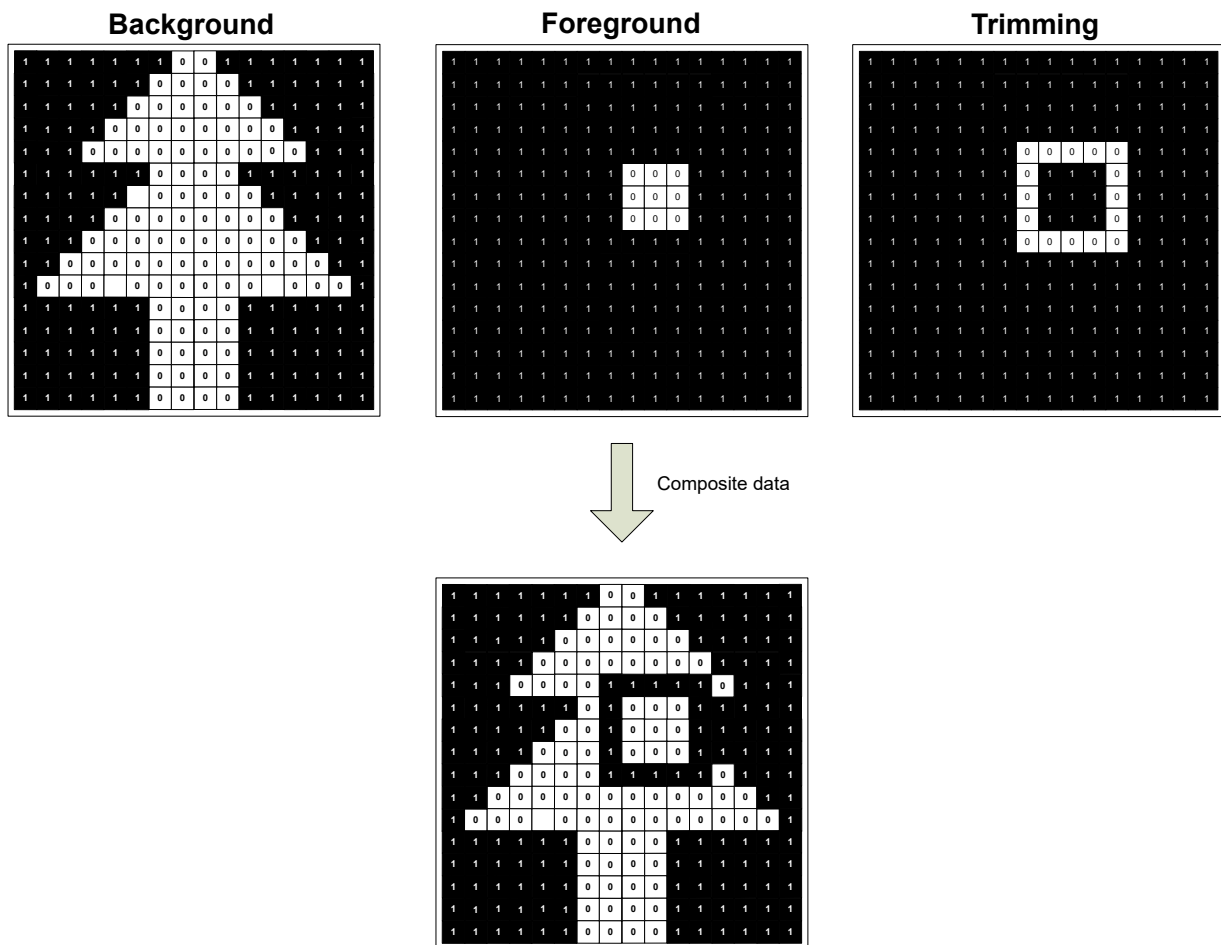


Figure 41.24 Compositing pixels having the value 0

Figure 41.25 shows an example of a flow of monochrome compositing the 16×16 -bit images N consecutive times with trimming is enabled when data transfer is handled by the DMAC. The image size handled in a single round of processing is $16 \text{ bits} \times 16 \text{ bits}$. In this example, DMAC0 is used for transfer of the input image data from the memory to the GDTIBUFn registers. DMAC1 is used for transfer of the output image data from the GDTIOBUFn registers to the memory. Each of the shaded parts in the figure shows a flow of transfers for a single round of monochrome compositing. A single round of monochrome compositing is handled in the following order from step 1 to step 6.

1. The GDT issues an image data input request interrupt to the CPU.
2. The CPU re-sets the information required for transfer by the DMAC and then issues a DMA transfer request.
3. The DMAC transfers the input image data from the memory to the GDTIBUFn registers.
4. Steps 1 to 3 are repeated three times, and the foreground, background, and trimming images are written to the GDT.
5. The GDT applies monochrome compositing to the data stored in the GDTIBUFn registers, stores the composite data in the GDTIOBUFn registers, and issues an image data output request interrupt as a trigger for the DMAC.
6. The DMAC transfers the output image data from the GDTIOBUFn registers to the memory.

When monochrome compositing has been completed N times, the DMAC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the CPU, which, however, does not trigger the DMAC to transfer the input data since processing of predetermined number of times has been completed.

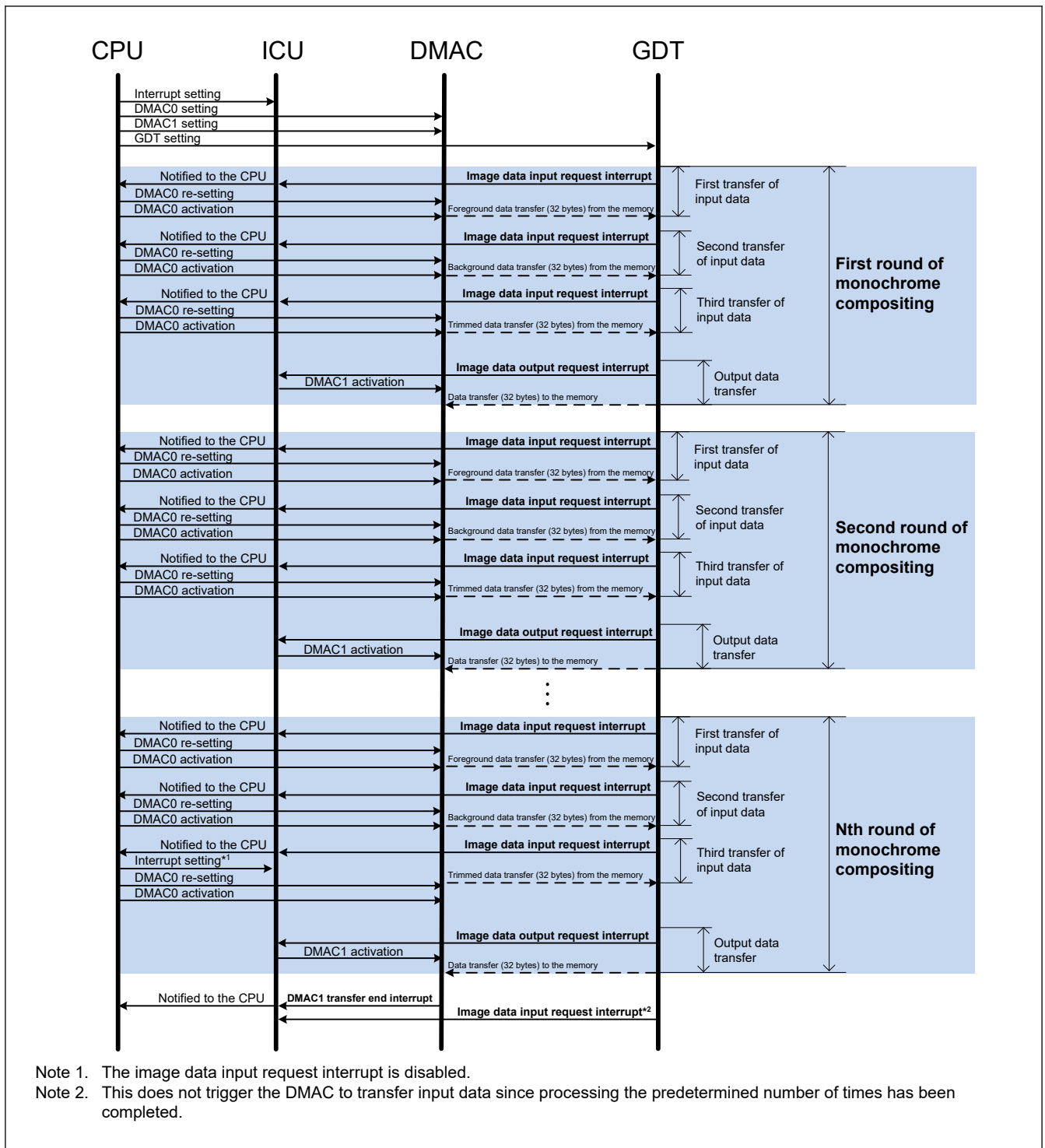


Figure 41.25 Example of a flow of monochrome compositing with the use of the DMAC for data transfer requiring CPU intervention for control

Figure 41.26 shows an example of a flow of monochrome compositing the 16×16 -bit images N consecutive times with trimming is enabled when data transfer is handled by the DTC. The image size handled in a single round of processing is 16 bits $\times$ 16 bits. Each of the shaded parts in the figure shows a flow of transfers for a single round of monochrome compositing. A single round of monochrome compositing is handled in the following order from step 1 to step 4.

1. The GDT issues an image data input request interrupt to the DTC.
2. The DTC transfers the input image data (foreground, background, and trimming images) from the memory to the GDTIBUF $_n$ registers by chain transfer.

3. The GDT applies monochrome compositing to the data stored in the GDTIBUFn registers, stores the composite data in the GDTOBUFn registers, and issues an image data output request interrupt as a trigger for the DTC.
4. The DTC transfers the output image data from the GDTOBUFn registers to the memory.

When monochrome compositing has been completed N times, the DTC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the DTC, which, however, does not trigger the DTC to transfer the input data since processing of predetermined number of times has been completed.

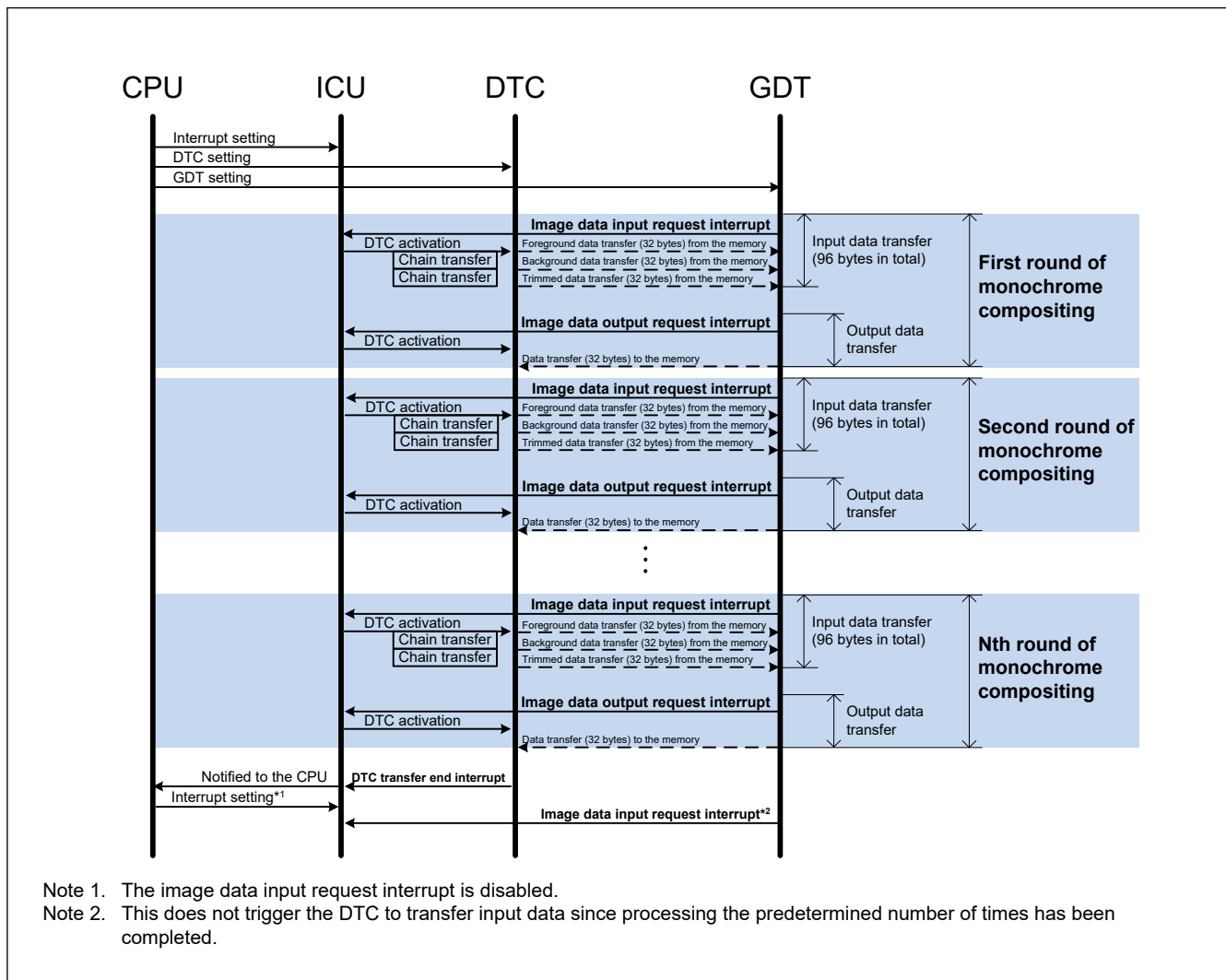


Figure 41.26 Example of a flow of monochrome compositing with the use of the DTC for data transfer

41.3.6 Color Compositing

The GDT also supports color compositing. A single round of processing of this function allows the input of two types of image data (foreground and background) and output of the composed image data. Two image sizes can be handled by this function: 8 bytes and 32 bytes. Either of two types of compositing is selectable. Table 41.11 lists the types of color compositing.

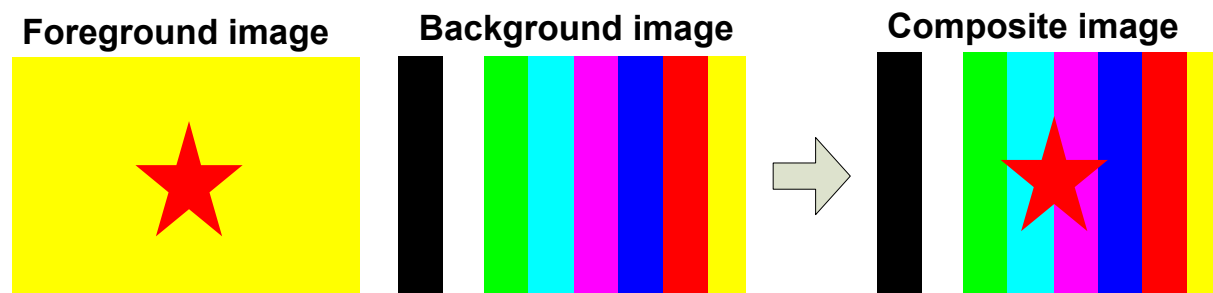
Table 41.11 Types of color compositing (1 of 2)

Display mode	Setting	Description
Priority color mode	GDTCR.CPTS = 0	After compositing, the pixels in the foreground image having the color specified by the GDTCR.CDCS[2:0] bits are displayed as the top layer, and the rest of the display is from the background image (See Figure 41.27).

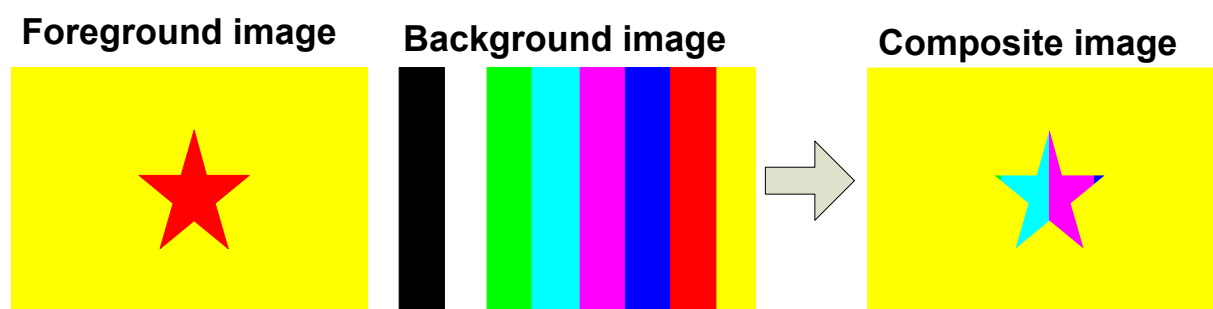
Table 41.11 Types of color compositing (2 of 2)

Display mode	Setting	Description
Transparent color mode	GDTCR.CPTS = 1	After compositing, the pixels in the foreground image having the color specified by the GDTCR.CDCS[2:0] bits are treated as transparent, so the background image is displayed in the corresponding positions. The rest of the display is from the foreground image (See Figure 41.28).

■ Priority color specified as red

**Figure 41.27** Priority color mode

■ Transparent color specified as red

**Figure 41.28** Transparent color mode

[Table 41.12](#) shows the GDT image data input registers and GDT image data output registers to be used in color compositing.

Table 41.12 Correspondence between registers used in color compositing and interrupt modes

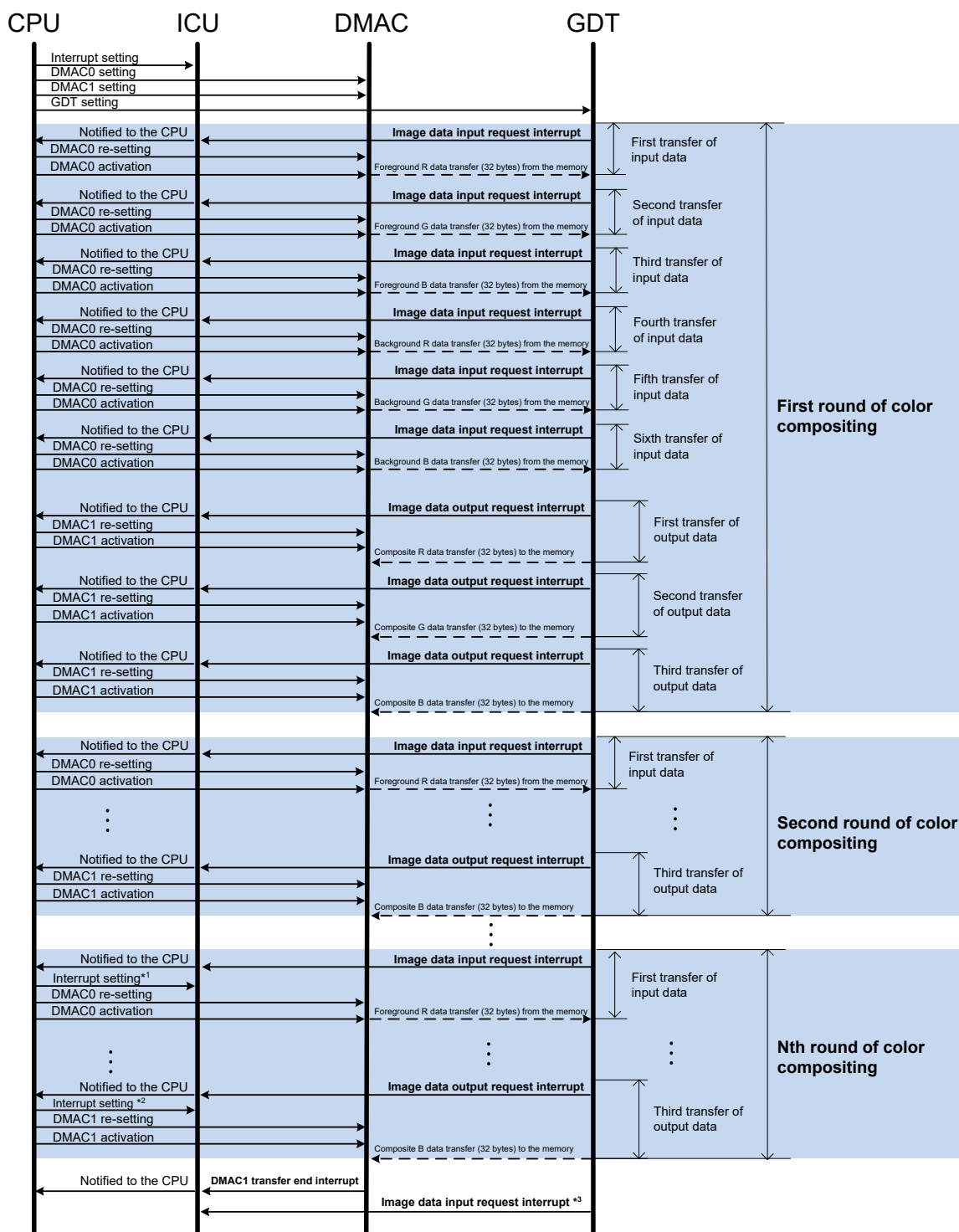
Input image size (bit)	Image data input request interrupt				Output image size (bit)	Image data output request interrupt		
	Interrupt mode	Number of interrupts	Transfer for one interrupt (bytes)	GDT Image Data Input Register		Number of interrupts	Transfer for one interrupt (bytes)	GDT Image Data Output Register
8 × 8	DMAC	6	8	First interrupt: GDTIBUF0 and GDTIBUF1 (foreground R image data) 2nd: GDTIBUF8 and GDTIBUF9 (foreground G image data) 3rd: GDTIBUF16 and GDTIBUF17 (foreground B image data) 4th: GDTIBUF24 and GDTIBUF25 (background R image data) 5th: GDTIBUF32 and GDTIBUF33 (background G image data) 6th: GDTIBUF40 and GDTIBUF41 (background B image data)	8 × 8	3	8	First interrupt: GDTOBUF0 and GDTOBUF1 (R data after compositing) 2nd: GDTOBUF8 and GDTOBUF9 (G data after compositing) 3rd: GDTOBUF16 and GDTOBUF17 (B data after compositing)
	DTC	1	48	First interrupt: GDTIBUF0 and GDTIBUF1 (foreground R image data) GDTIBUF8 and GDTIBUF9 (foreground G image data) GDTIBUF16 and GDTIBUF17 (foreground B image data) GDTIBUF24 and GDTIBUF25 (background R image data) GDTIBUF32 and GDTIBUF33 (background G image data) GDTIBUF40 and GDTIBUF41 (background B image data)	8 × 8	1	24	First interrupt: GDTOBUF0 and GDTOBUF1 (R data after compositing) GDTOBUF8 and GDTOBUF9 (G data after compositing) GDTOBUF16 and GDTOBUF17 (B data after compositing)
16 × 16	DMAC	6	32	First interrupt: GDTIBUF0 to GDTIBUF7 (foreground R image data) 2nd: GDTIBUF8 to GDTIBUF15 (foreground G image data) 3rd: GDTIBUF16 to GDTIBUF23 (foreground B image data) 4th: GDTIBUF24 to GDTIBUF31 (background R image data) 5th: GDTIBUF32 to GDTIBUF39 (background G image data) 6th: GDTIBUF40 to GDTIBUF47 (background B image data)	16 × 16	3	32	First interrupt: GDTOBUF0 to GDTOBUF7 (R data after compositing) 2nd: GDTOBUF8 to GDTOBUF15 (G data after compositing) 3rd: GDTOBUF16 to GDTOBUF23 (B data after compositing)
	DTC	1	192	First interrupt: GDTIBUF0 to GDTIBUF7 (foreground R image data) GDTIBUF8 to GDTIBUF15 (foreground G image data) GDTIBUF16 to GDTIBUF23 (foreground B image data) GDTIBUF24 to GDTIBUF31 (background R image data) GDTIBUF32 to GDTIBUF39 (background G image data) GDTIBUF40 to GDTIBUF47 (background B image data)	16 × 16	1	96	First interrupt: GDTOBUF0 to GDTOBUF7 (R data after compositing) GDTOBUF8 to GDTOBUF15 (G data after compositing) GDTOBUF16 to GDTOBUF23 (B data after compositing)

Figure 41.29 shows an example of a flow of color compositing the 16 × 16-bit images N consecutive times with data transfer handled by the DMAC. In this example, DMAC channel 0 is used to transfer the input image data from the memory

to the GDTIBUFn registers. DMAC channel 1 is used to transfer the output image data from the GDTOBUFn registers to the memory. Each of the shaded parts in the figure shows the flow of transfers for a single round of color compositing. A single round of color compositing is handled in the following order from step 1 to step 9.

1. The GDT issues an image data input request interrupt for the CPU.
2. The CPU re-sets the information required for transfer by the DMAC and then issues a DMA transfer request.
3. The DMAC transfers the input image data from the memory to the GDTIBUFn registers.
4. Steps 1 to 3 are repeated six times, and the R, G, and B images of foreground and background are written to the GDT.
5. The GDT applies color compositing to the data stored in the GDTIBUFn registers, and stores the composite data in the GDTOBUFn registers.
6. The GDT issues an image data output request interrupt for the CPU.
7. The CPU re-sets the information required for transfer by the DMAC and then issues a DMA transfer request. A DMAC transfer end notification should be set before the DMAC is re-set for the third transfer of output data in the Nth color compositing.
8. The DMAC transfers the output image data from the GDTOBUFn registers to the memory.
9. Steps 6 to 8 are repeated three times.

When color compositing has been completed N times, the DMAC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the CPU, which does not, however, trigger the DMAC to transfer input data since processing of predetermined number of times has been completed.



Note 1. The image data input request interrupt is disabled.

Note 2. The DMAC1 transfer end interrupt is enabled.

Note 3. This does not trigger the DMAC to transfer input data since processing the predetermined number of times has been completed.

Figure 41.29 Example of a flow of color compositing with the use of the DMAC for data transfer requiring CPU intervention for control

Figure 41.30 is an example of a flow of color compositing the 16×16 -bit images N consecutive times with data transfer handled by the DTC. Each of the shaded parts in the figure shows the flow of transfers for a single round of color compositing. A single round of color compositing is handled in the following order from step 1 to step 4.

1. The GDT issues an image data input request interrupt to the DTC.
2. The DTC transfers the input image data from the memory to the GDTIBUFn registers by chain transfer.
3. The GDT applies color compositing to the data stored in the GDTIBUFn registers, and stores the composite data in the GDTOBUFn registers. The GDT then issues an image data output request interrupt to the DTC.
4. The DTC transfers the output image data from the GDTOBUFn registers to the memory by chain transfer.

When color compositing has been completed N times, the DTC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the DTC, which, however, does not trigger the DTC to transfer the input data since processing of predetermined number of times has been completed.

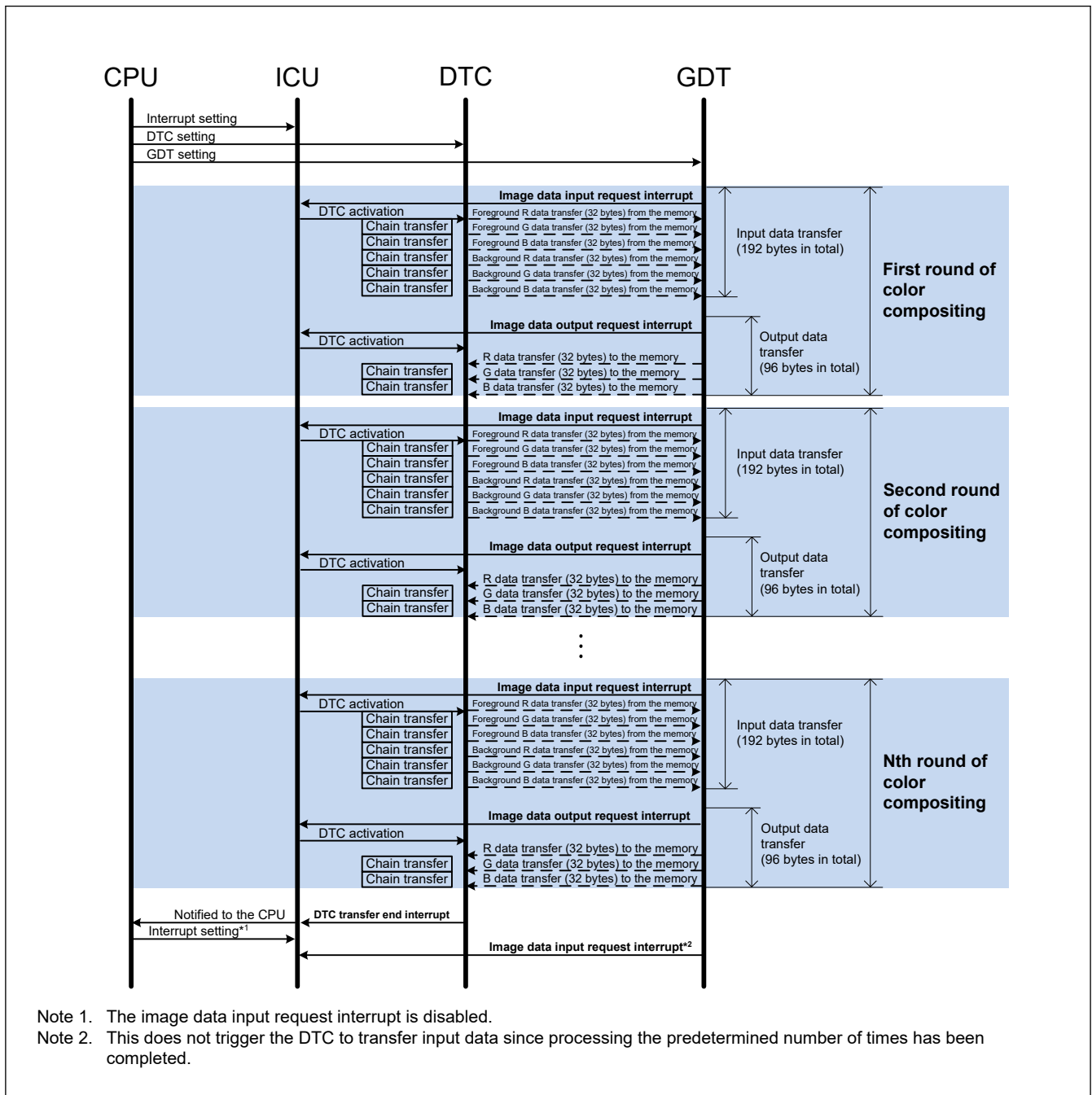


Figure 41.30 Example of a flow of color compositing with the use of the DTC for data transfer

41.3.7 Scrolling

The GDT supports scrolling of images. This function allows the input of data for two consecutive images (8×16 bits per image) and output of an 8×16 -bit image, scrolled by the number of pixels specified by the value of the GDTCCR.ISCREN[2:0] bits.

Figure 41.31 is an example of the image data on the LCD when images are scrolled. In the figure, the part enclosed in the red broken lines, which is shifted to the right from the position of input image 1 (enclosed in the blue broken lines) into image 2 by the amount set in the GDTCCR.ISCREN[2:0] bits (011b in this example), is output as the image after scrolling. Figure 41.32 is an example where the input image data are stored in the GDT image data input registers and the output image data after scrolling are stored in the GDT image data output registers. In this example, the conveying of interrupt signals to the DTC is specified as the GDT interrupt mode.

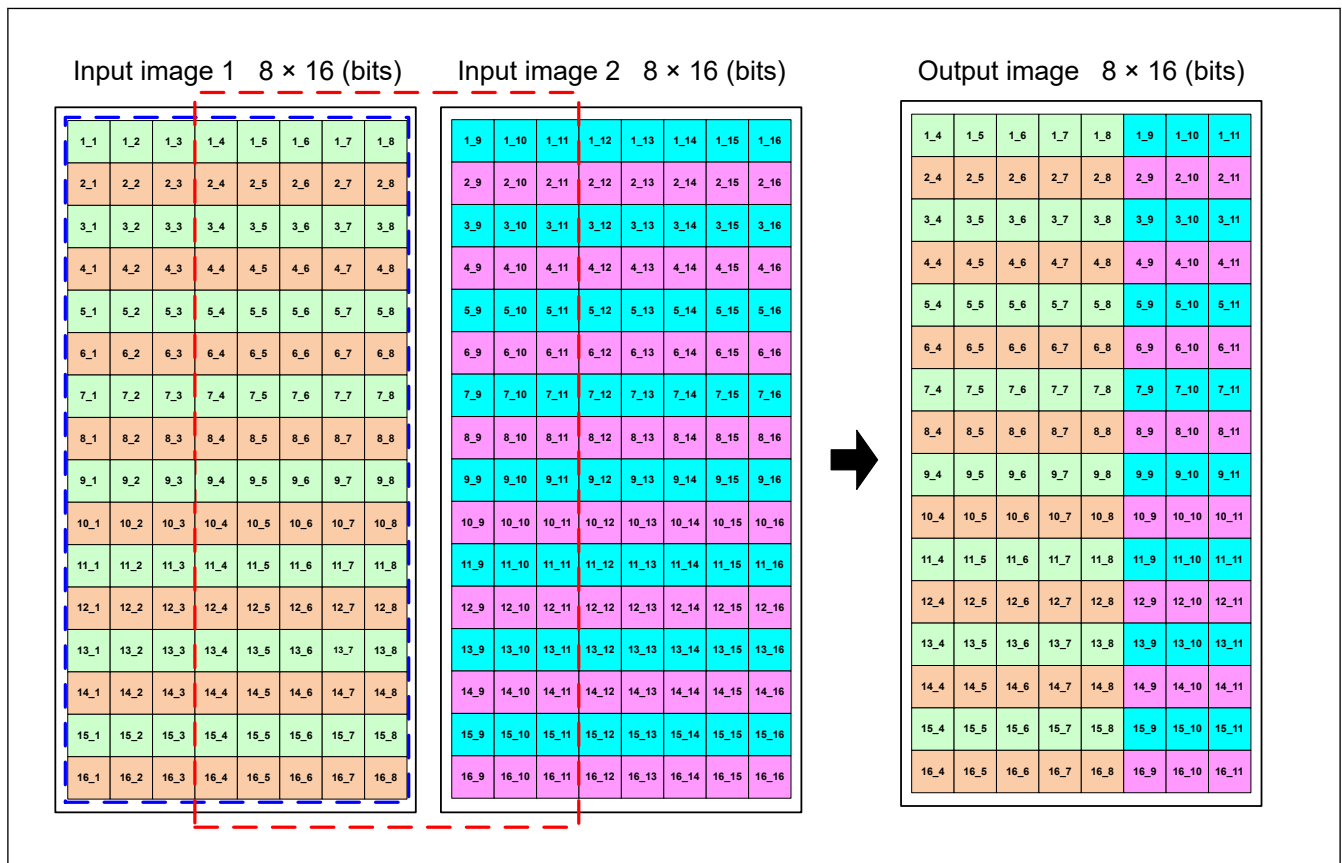


Figure 41.31 Example of image data allocation on the LCD after scrolling

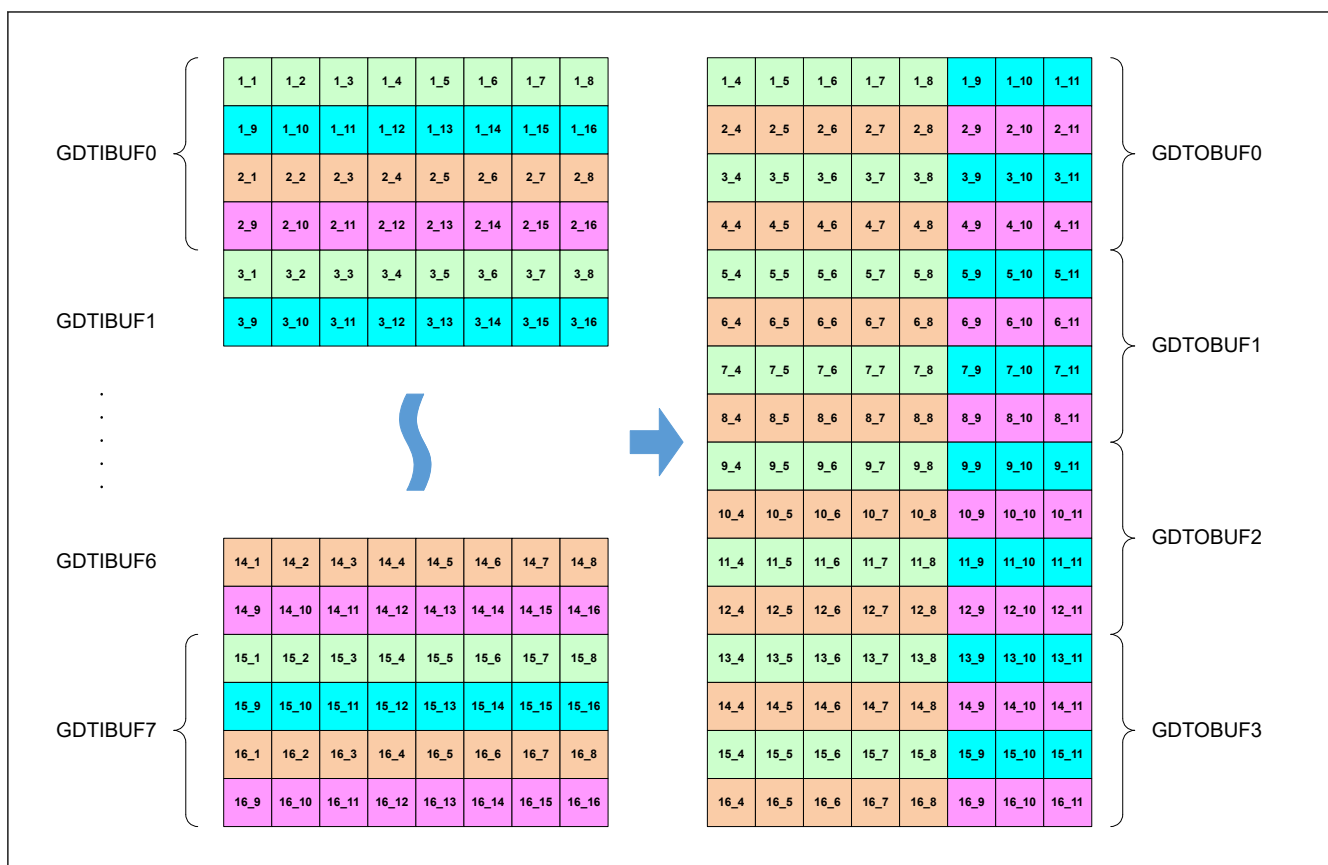


Figure 41.32 Example of image data allocation to the memory after scrolling

Table 41.13 lists the GDT image data input registers and GDT image data output registers to be used in scrolling. Figure 41.33 and Figure 41.34 show examples of the transfer of input data when the DMAC and DTC, respectively, are selected as the GDT interrupt modes. In these examples, the source addresses for transfer by the DMAC are updated by adding offset values. Figure 41.35 is an example of the transfer of output data. In this case, the number of interrupts, amount transferred per interrupt, and registers used are the same regardless of the interrupt mode setting.

Table 41.13 Correspondence between registers used in scrolling and interrupt mode

Input image size (bit)	Image data input request interrupt				Output image size (bits)	Image data output request interrupt		
	Interrupt mode	Number of interrupts	Transfer for one interrupt (byte)	GDT Image Data Input Register		Number of interrupts	Transfer for one interrupt (byte)	GDT Image Data Output Register
8 × 16 × 2	DMAC	2	16	First interrupt: GDTIBUF0 to GDTIBUF3 2nd: GDTIBUF0 to GDTIBUF3	8 × 16	1	16	First interrupt: GDTOBUF0 to GDTOBUF3
	DTC	1	32	First interrupt: GDTIBUF0 to GDTIBUF7	8 × 16	1	16	First interrupt: GDTOBUF0 to GDTOBUF3

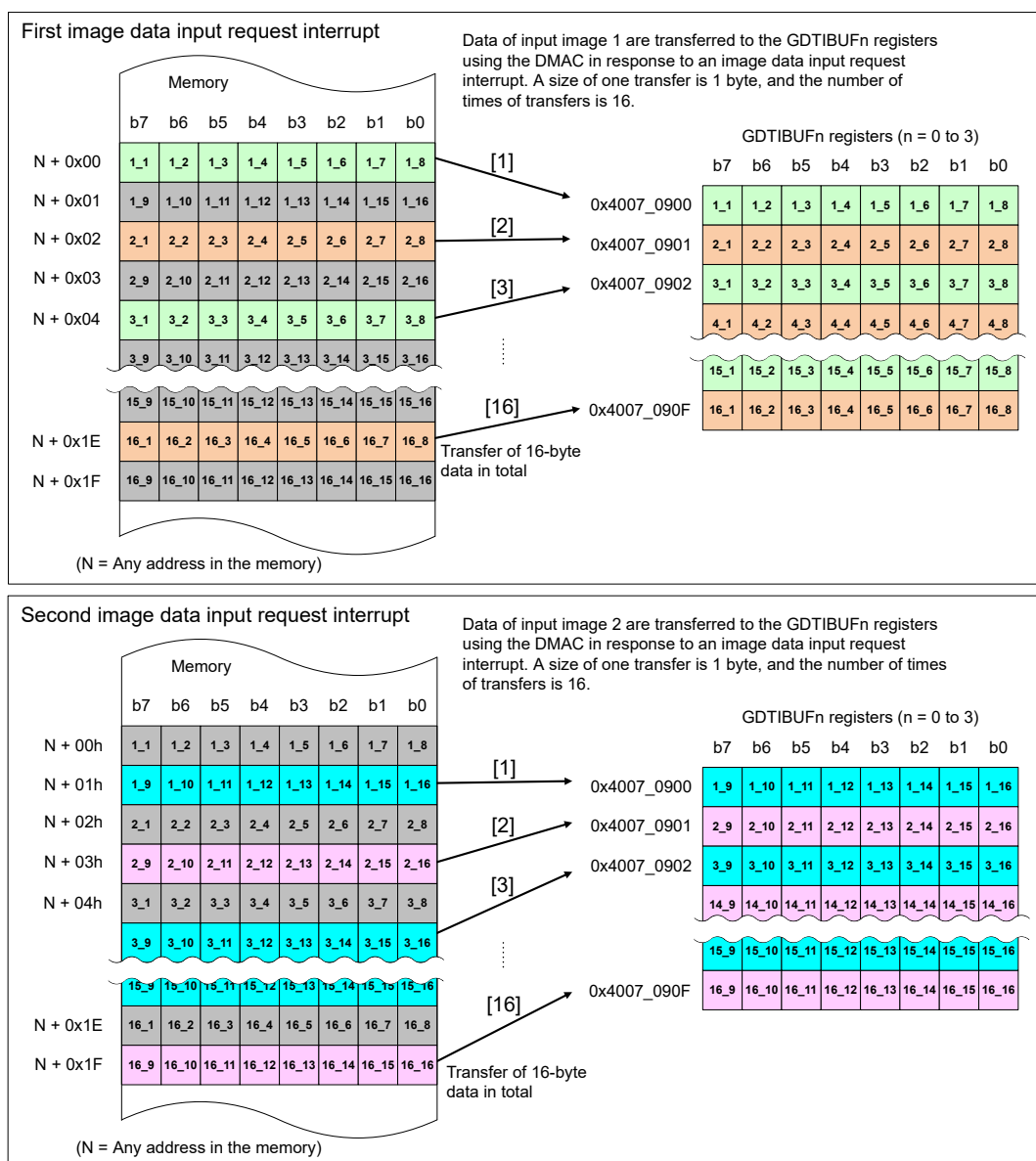


Figure 41.33 Example of transfer of input data when the DMAC is selected for the GDT interrupt mode in scrolling

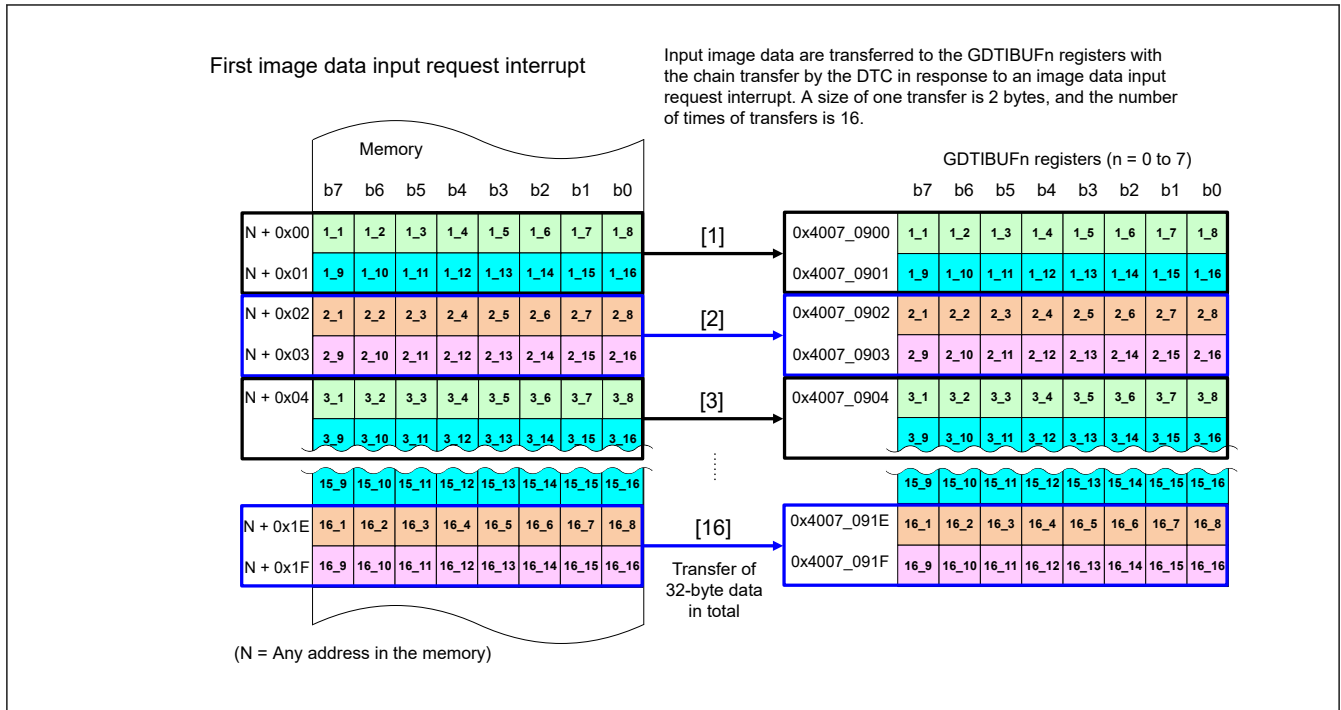


Figure 41.34 Example of transfer of input data when the DTC is selected for the GDT interrupt mode in scrolling

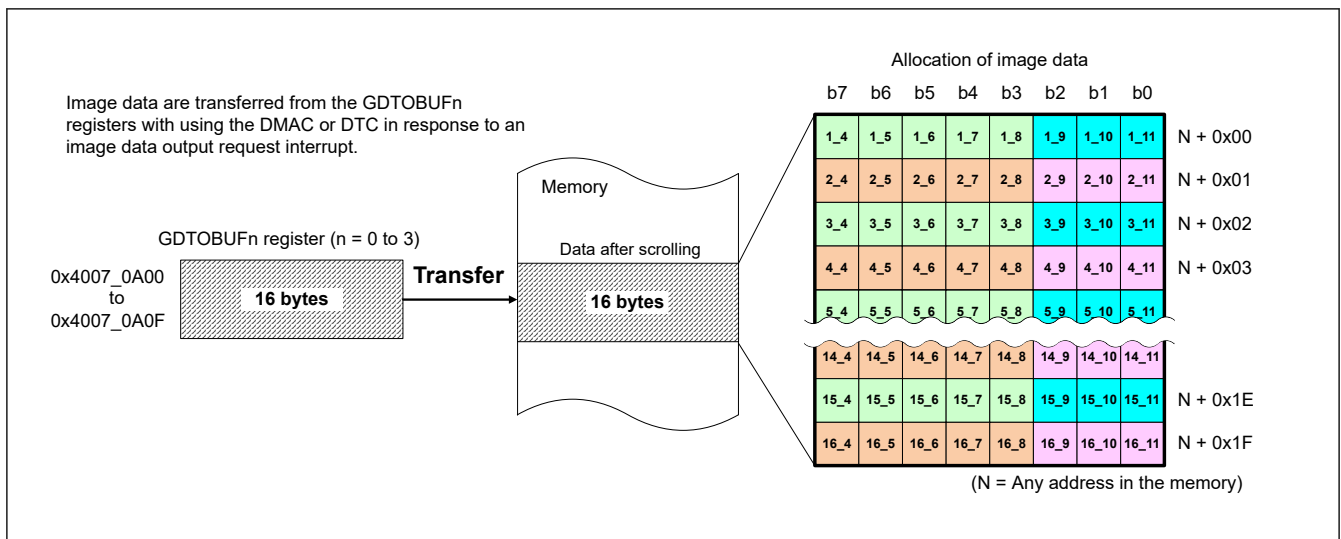


Figure 41.35 Example of transfer of output data in scrolling

Figure 41.36 is an example of a flow of scrolling images N consecutive times with data transfer handled by the DMAC. In this example, DMAC channel 0 is used to transfer the input image data from the memory to the GDTIBUFn registers. DMAC channel 1 is used to transfer the output image data from the GDTIBUFn registers to the memory. Each of the shaded parts in the figure shows the flow of transfers for a single round of scrolling. A single round of scrolling is handled in the following order from step 1 to step 6.

1. The GDT issues an image data input request interrupt to the CPU.
2. The CPU re-sets the information required for transfer by the DMAC and then issues a DMA transfer request.
3. The DMAC transfers the input image data from the memory to the GDTIBUFn registers.
4. Steps 1 to 3 are repeated two times.
5. The GDT applies scrolling to the data stored in the GDTIBUFn registers, and stores the scrolled data in the GDTIBUFn registers. The GDT then issues an image data output request interrupt to the DMAC.
6. The DMAC transfers the output image data from the GDTIBUFn registers to the memory.

When scrolling has been completed N times, the DMAC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the CPU, which, however, does not trigger the DMAC to transfer the input data since processing of predetermined number of times has been completed.

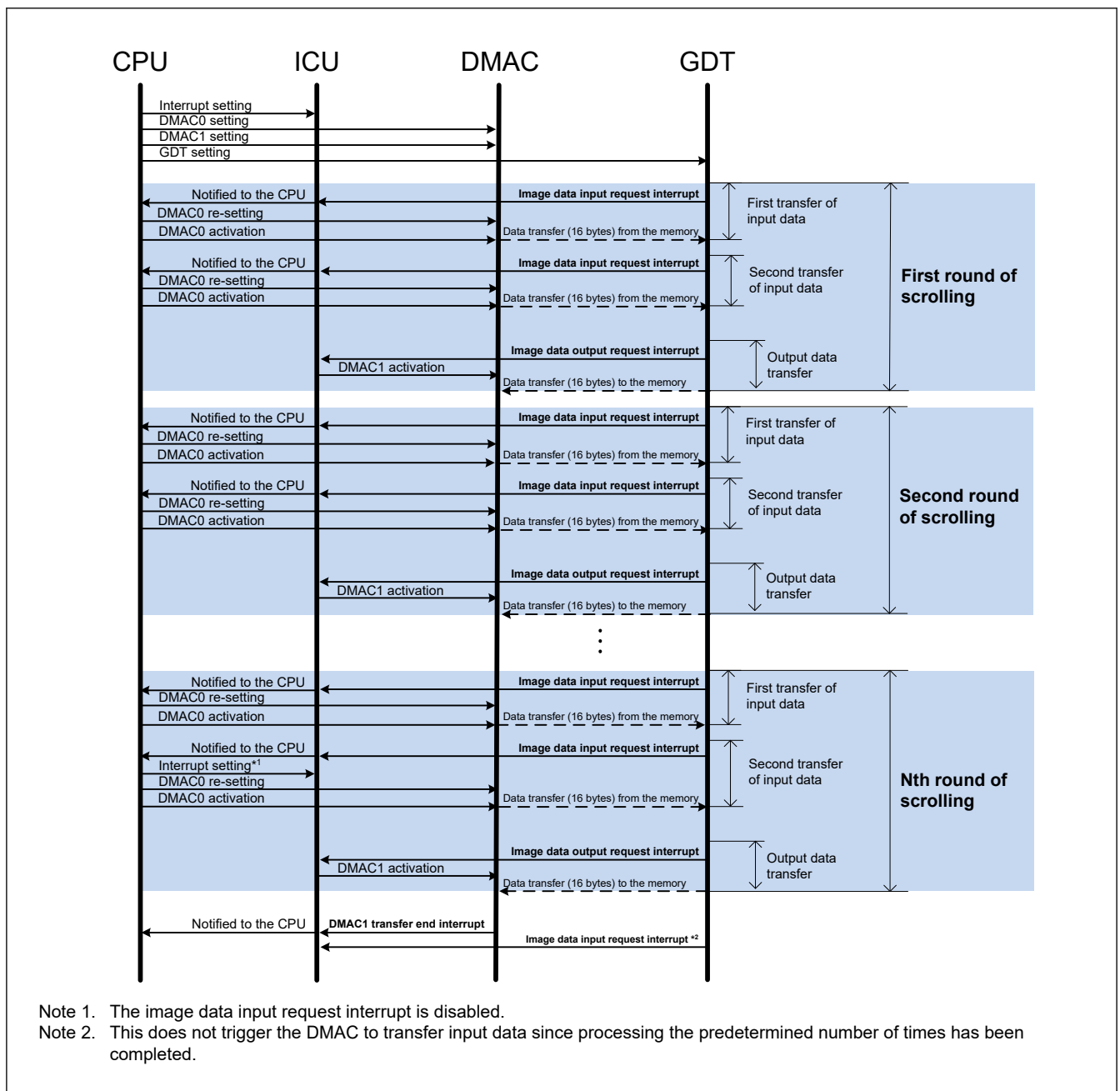


Figure 41.36 Example of a flow of scrolling with the use of the DMAC for data transfer requiring CPU intervention for control

Figure 41.37 shows an example of a flow of scrolling images N consecutive times with data transfer handled by the DTC. Each of the shaded parts in the figure shows the flow of transfers for a single round of color compositing. A single round of color compositing is handled in the following order from step 1 to step 4.

1. The GDT issues an image data input request interrupt to the DTC.
2. The DTC transfers the input image data from the memory to the GDTIBUFn registers by chain transfer.
3. The GDT scrolls the data stored in the GDTIBUFn registers, and stores the scrolled data in the GDTOBUFn registers. The GDT then issues an image data output request interrupt to the DTC.
4. The DTC transfers the output image data from the GDTOBUFn registers to the memory by chain transfer.

When scrolling has been completed N times, the DTC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the DTC, which, however, does not trigger the DTC to transfer the input data since processing of predetermined number of times has been completed.

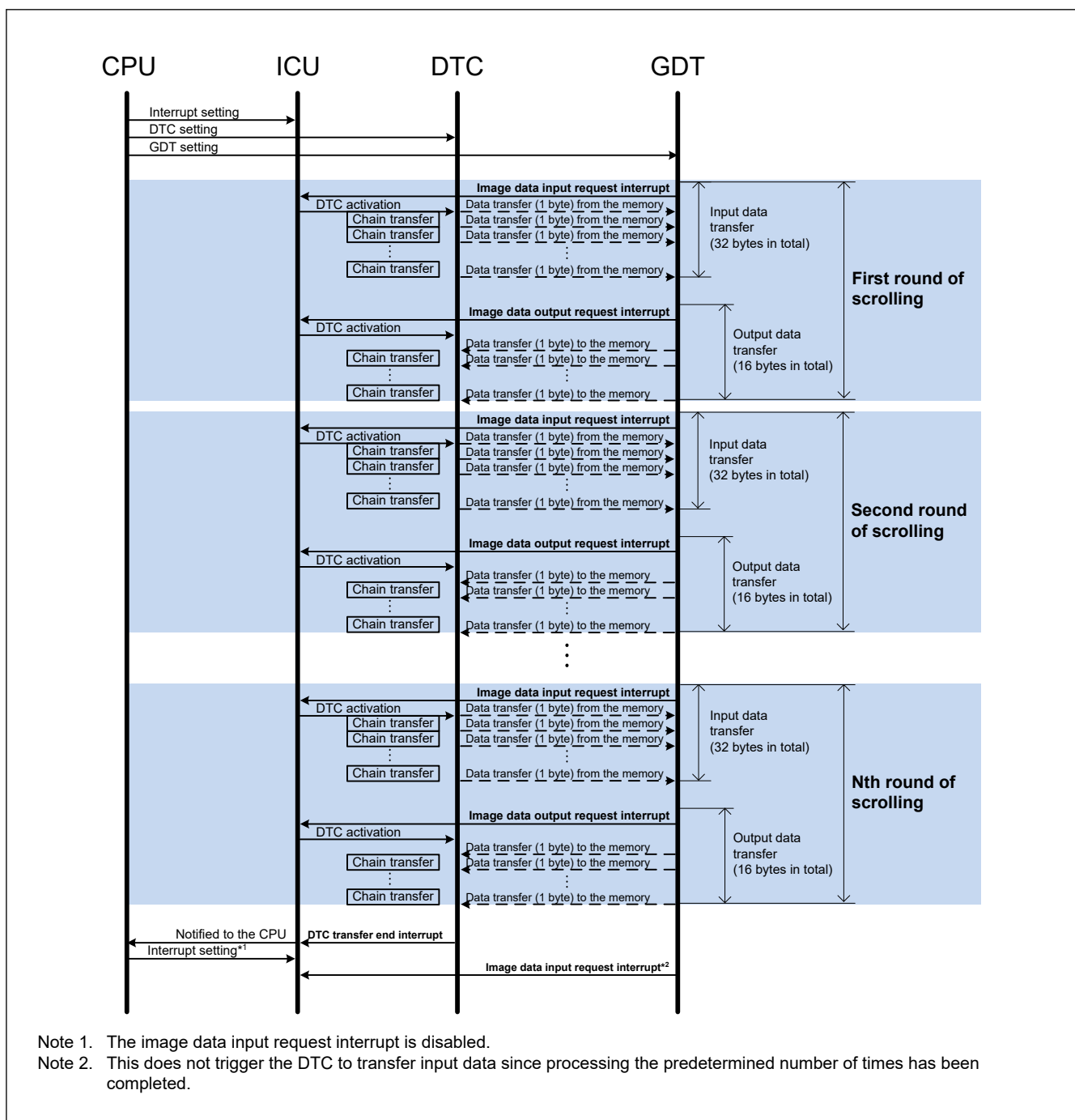


Figure 41.37 Example of a flow of scrolling with the use of the DTC for data transfer

41.3.8 Conversion of Glyph Data into Image Data

The GDT supports conversion of glyph data into image data. This function converts glyph data contiguously stored in the memory to a format that can easily be displayed on the LCD. The glyph sizes that can be converted are from 7 to 63 bits in the horizontal direction and 7 to 64 bits in the vertical direction.*1

Figure 41.38 shows an example of the data allocation when 10 × 10-bit glyph data are converted to an image format and the start of the image after conversion is shifted to the right by 4 bits.

Note 1. Glyphs with a size that is a multiple of 8 do not require conversion because the data sequence stored in the memory is the same as that after conversion.

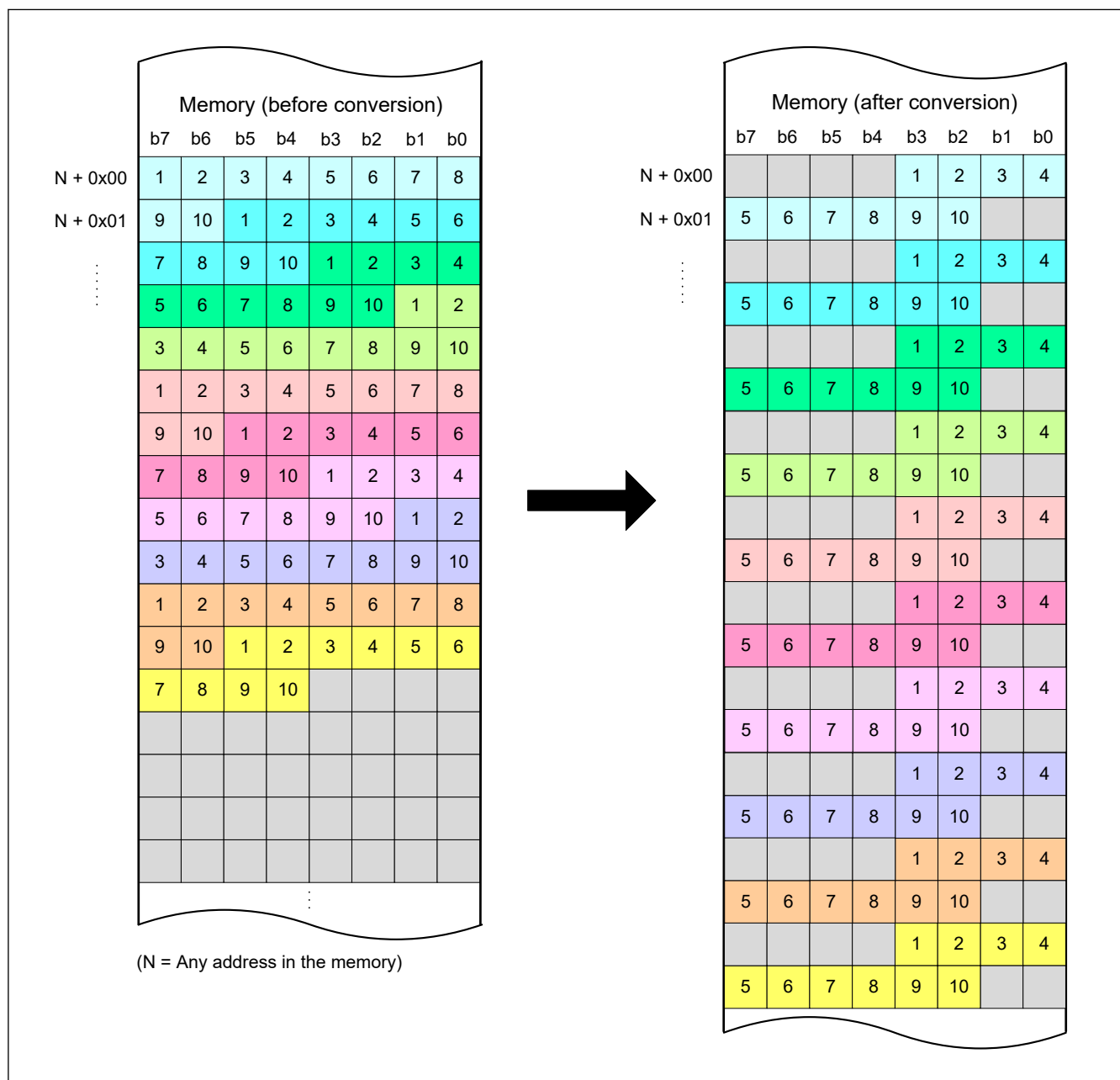


Figure 41.38 Example of data allocation in the memory before and after conversion of glyph data into image data

For glyph data, the actual data can be shifted to the right by the number of bits specified by the GDTFDCS.SAC[2:0] bits at the time of conversion. Figure 41.39 illustrates the case where SAC[2:0] = 100b (right-shifting by 4 bits).

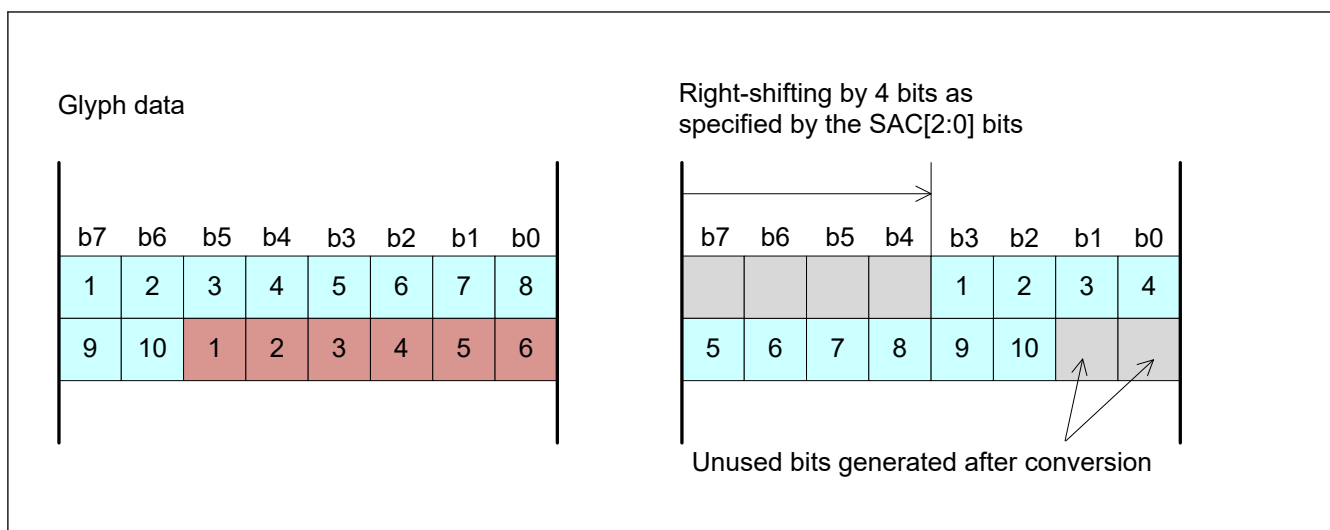


Figure 41.39 Changing the start position in conversion of glyph data into image data

By setting the GDTFDCS.FDHAD bit, 1 or 0 can be written arbitrarily in blank bits (shown as gray bits in Figure 41.39) that occur when the start address is changed. Table 41.14 lists the GDT Image Data Input Register and GDT Image Data Output Register used in the font development function. The table describes values as an example where the 10×10 bits font data is converted to an image format and the start address after image development is shifted to the right by 4 bits. Figure 41.40 and Figure 41.41 show examples of input data transfer and output data transfer, respectively.

Table 41.14 Registers used in conversion of glyph data into image data

Input image size (bit)	Image data input request interrupt				Output image size (bit)	Image data output request interrupt		
	Interrupt mode	Number of interrupts	Transfer for one interrupt (byte)	GDT Image Data Input Register		Number of interrupts	Transfer for one interrupt (byte)	GDT Image Data Output Register
10×10	—*1	2^2	8	First interrupt: GDTIBUF0 and GDTIBUF1*3	10×10	10^4	2^5	First interrupt: GDTIBUF0 to GDTIBUF2*3

Note 1. The number of interrupts, amount transferred per interrupt, and registers used are the same regardless of the interrupt mode setting.

Note 2. The number of times of image data input request interrupts depends on the number of data processing, setting in the GDTFDCS.FDIR[5:0] bits.

Note 3. The same GDT image data input registers and GDT image data output registers are used each time an interrupt occurs.

Note 4. The number of image data output request interrupts depends on the number of vertical bits of the font data, setting in the GDTFDCS.FDLTDSZ[6:0] bits.

Note 5. The size of one image data output request interrupt differs depending on the amount of data processed. Use the following equation to obtain the amount of data:

Font Data Horizontal Size Setting bit (GDTFDCS.FDLTDSZ[5:0]) + Start Address Change bit (GDTFDCS.SAC[2:0]) $\div$ 8

If the remainder is 0, the output image size is the quotient, otherwise it is the value of the quotient + 1.

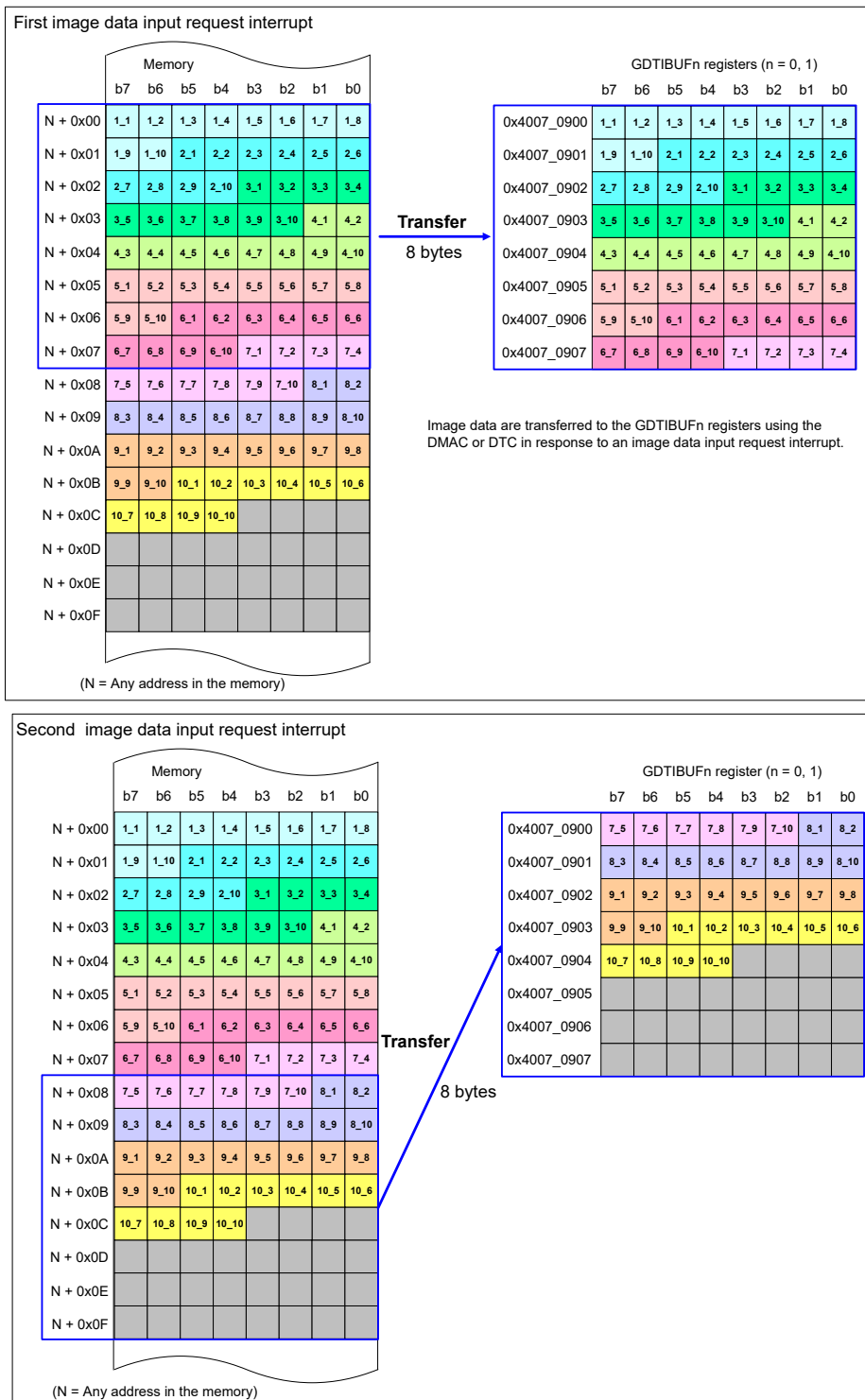


Figure 41.40 Example of input data transfer in glyph conversion

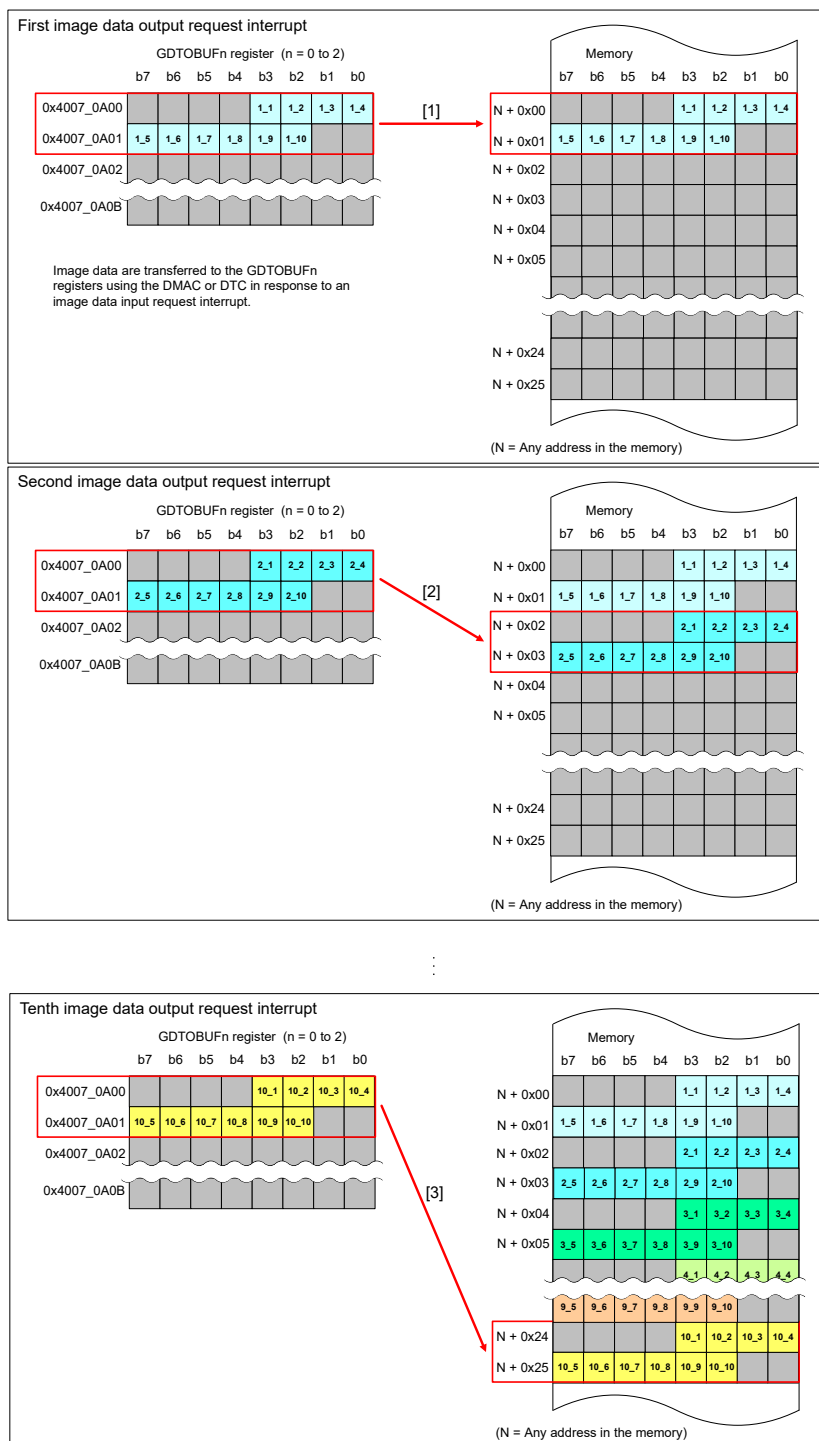


Figure 41.41 Example of output data transfer in glyph conversion

Figure 41.42 is an example of a flow of converting 63×64 -bit glyph images N consecutive times by right-shifting 7 bits with data transfer handled by the DMAC. In this example, DMAC channel 0 is used to transfer the input image data from the memory to the GDTIBUFn registers. DMAC channel 1 is used to transfer the output image data from the GDTIBUFn registers to the memory. A single round of glyph conversion is handled in the following order from step 1 to step 12.

1. The GDT issues an image data input request interrupt to the DMAC.
2. The DMAC transfers the input image data from the memory to the GDTIBUFn registers.

3. The GDT converts data stored in the GDTIBUFn registers (first round), and stores the converted data in the GDTOBUFn registers.
4. The GDT issues an image data input request interrupt to the DMAC.
5. The DMAC transfers the input image data from the memory to the GDTIBUFn registers.
6. The GDT converts data stored in the GDTIBUFn registers (second round).
7. The GDT issues an image data output request interrupt to the DMAC.
8. The DMAC transfers the output image data (first round) from the GDTOBUFn registers to the memory.
9. The converted data are stored in the GDTOBUF registers (second round).
10. The GDT issues an image data output request interrupt to the DMAC.
11. The DMAC transfers the output image data from the GDTOBUFn registers to the memory (second round).
12. Steps 1 to 11 are repeated 32 times (steps 4 to 6 are skipped in the 32nd round of conversion).

When glyph conversions have been completed 64 times, the DMAC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the DMAC, which, however, does not trigger the DMAC to transfer the input data since processing of predetermined number of times has been completed.

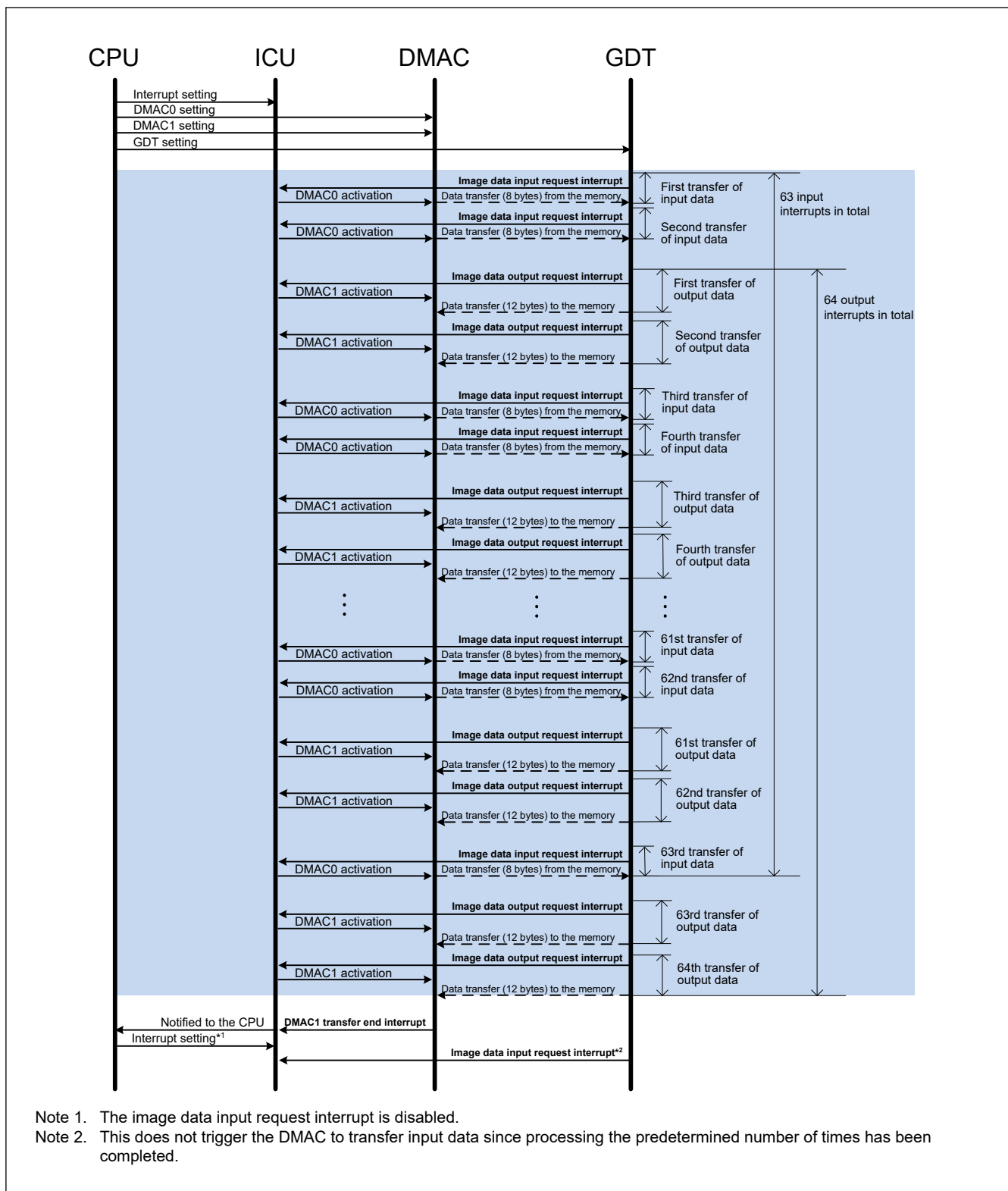


Figure 41.42 Example of a flow with the use of the DMAC for transfer in glyph conversion

Figure 41.43 shows an example of a flow of converting 63×64 -bit glyph images N consecutive times by right-shifting 7 bits with data transfer handled by the DTC. A single round of glyph conversion is handled in the following order from step 1 to step 12.

1. The GDT issues an image data input request interrupt to the DTC.
2. The DTC transfers the input image data from the memory to the GDTIBUF n registers.

3. The GDT converts data stored in the GDTIBUFn registers (first round), and stores the converted data in the GDTIOBUFn registers.
4. The GDT issues an image data input request interrupt to the DTC.
5. The DTC transfers the input image data from the memory to the GDTIBUFn registers.
6. The GDT converts data stored in the GDTIBUFn registers (second round).
7. The GDT issues an image data output request interrupt to the DTC.
8. The DTC transfers the output image data (first round) from the GDTIOBUFn registers to the memory.
9. The converted data are stored in the GDTIOBUF registers (second round).
10. The GDT issues an image data output request interrupt to the DTC.
11. The DTC transfers the output image data from the GDTIOBUFn registers to the memory (second round).
12. Steps 1 to 11 are repeated 32 times (steps 4 to 6 are skipped in the 32nd round of conversion).

When glyph conversions have been completed 64 times, the DTC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the DTC, which, however, does not trigger the DTC to transfer the input data since processing of predetermined number of times has been completed.

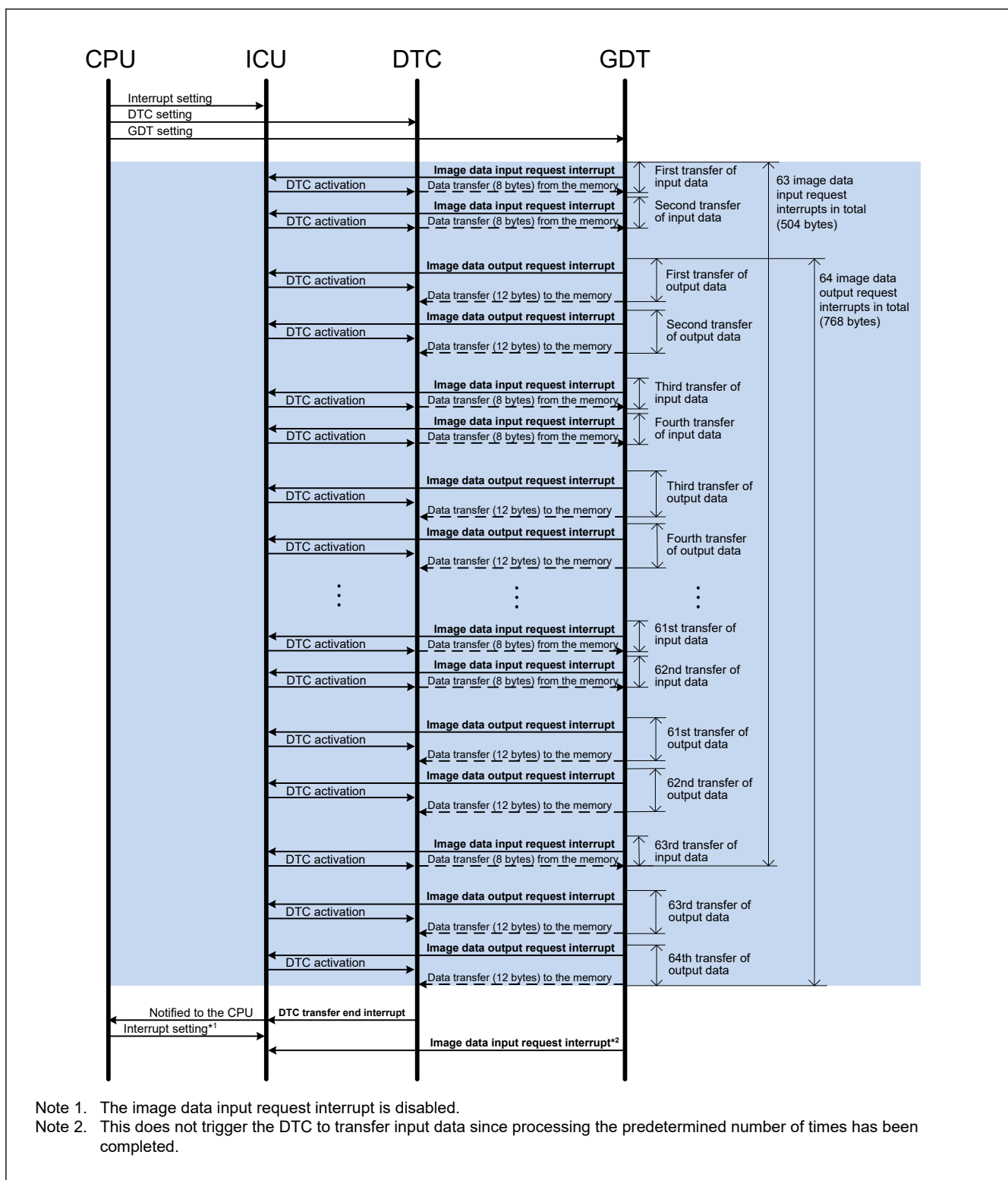


Figure 41.43 Example of a flow with the use of the DTC for data transfer in glyph conversion

41.3.9 Colorization

The GDT supports colorization of monochrome images. The image size that can be handled in colorization is 32 bytes (16×16 bits). This function generates a color image through the specification of colors to be assigned to the black and white parts of a monochrome image.

Figure 41.44 is an example of colorization. The color for the pixels having the value 0 of the monochrome image is specified by using the GDCR.CLRDS0[2:0] bits. Similarly, the color for the pixels having the value 1 of the monochrome

image is specified by using the GDTCR.CLRDS1[2:0] bits. In this example, RGB = 010 (green) is set for the pixels having the value 0, and RGB = 101 (red + blue = magenta) is set for the pixels having the value 1. This function divides the colored image into three images (R, G, and B images) and writes them back to the memory.

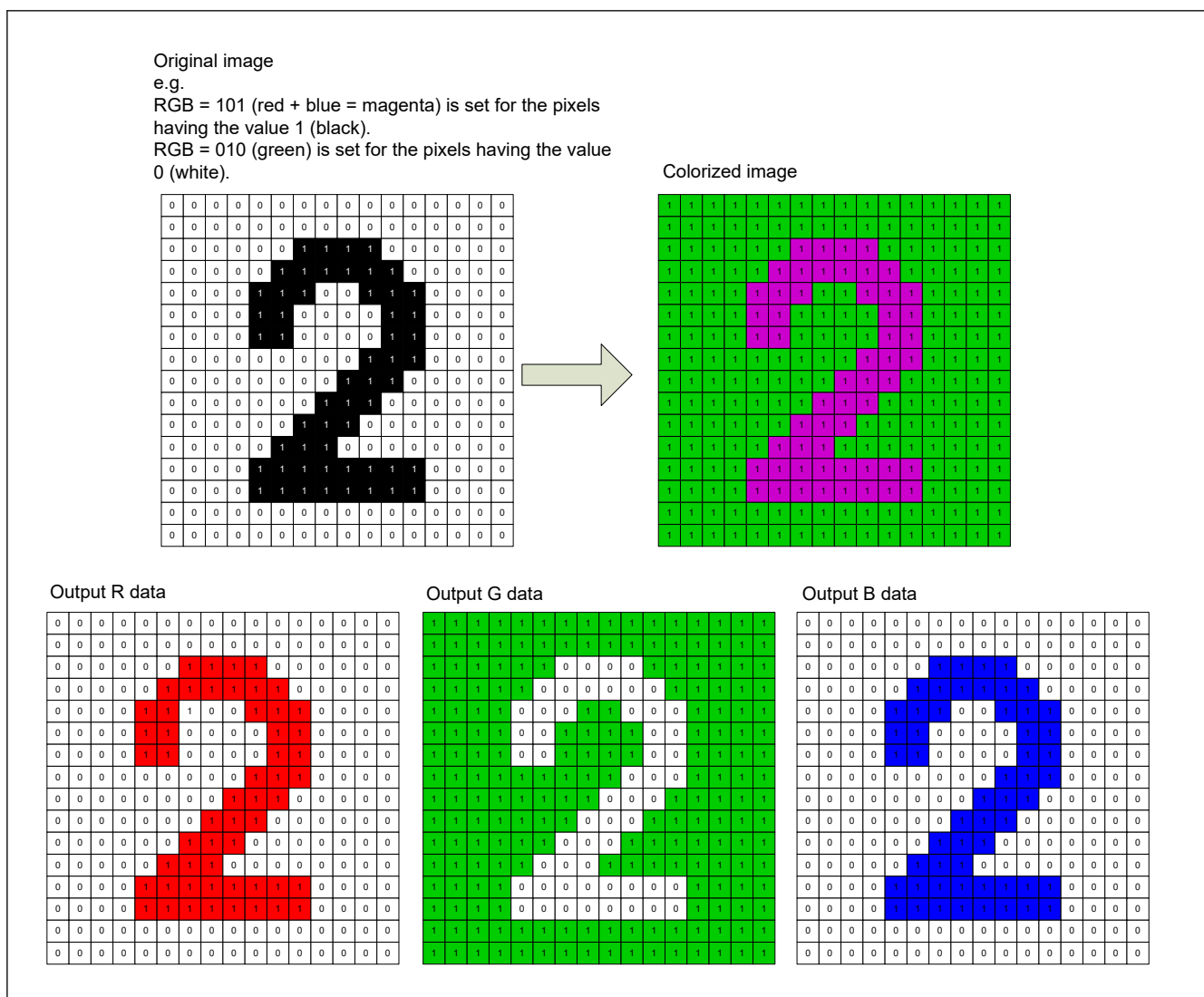


Figure 41.44 Example of colorization on the LCD

Table 41.15 lists the GDT image data input registers and GDT image data output registers used in image colorization. On output, the R, G, and B images are written back to separate memory areas in response to three interrupts.

Table 41.15 Correspondence between registers used in image colorization and interrupt mode

Input image size (bit)	Image data input request interrupt				Output image size (bit)	Image data output request interrupt		
	Interrupt mode	Number of interrupts	Transfer for one interrupt (bytes)	GDT Image Data Input Register		Number of interrupts	Transfer for one interrupt (bytes)	GDT Image Data Output Register
16 × 16	DMAC	1	32	First interrupt: GDTIBUF0 to GDTIBUF7	16 × 16	3	32	First interrupt: GDTIBUF0 to GDTIBUF7 (after colorization, R image) 2nd: GDTIBUF8 to GDTIBUF15 (after colorization, G image) 3rd: GDTIBUF16 to GDTIBUF23 (after colorization, B image)
	DTC	1	32	First interrupt: GDTIBUF0 to GDTIBUF7	16 × 16	1	96	First interrupt: GDTIBUF0 to GDTIBUF7 (after colorization, R image) GDTIBUF8 to GDTIBUF15 (after colorization, G image) GDTIBUF16 to GDTIBUF23 (after colorization, B image)

Figure 41.45 shows an example of a flow of colorizing images N consecutive times with data transfer handled by the DMAC. In this example, DMAC channel 0 is used to transfer the input image data from the memory to the GDTIBUFn registers. DMAC channel 1 is used to transfer the output image data from the GDTIBUFn registers to the memory. Each of the shaded parts in the figure shows the flow of transfers for a single round of colorization. A single round of colorization is handled in the following order from step 1 to step 7.

1. The GDT issues an image data input request interrupt to the DMAC.
2. The DMAC transfers the input image data from the memory to the GDTIBUFn registers.
3. The GDT colorizes the data stored in the GDTIBUFn registers, and stores the colorized data in the GDTIBUFn registers.
4. The GDT issues an image data output request interrupt to the CPU.
5. The CPU re-sets the information required for transfer by the DMAC and then issues a DMA transfer request. A DMAC transfer end notification should be set before the DMAC is re-set for the third transfer of output data in the Nth colorization.
6. The DMAC transfers the output image data from the GDTIBUFn registers to the memory.
7. Steps 4 to 6 are repeated three times.

When colorizations have been completed N times, the DMAC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the CPU, which, however, does not trigger the DMAC to transfer the input data since processing of predetermined number of times has been completed.

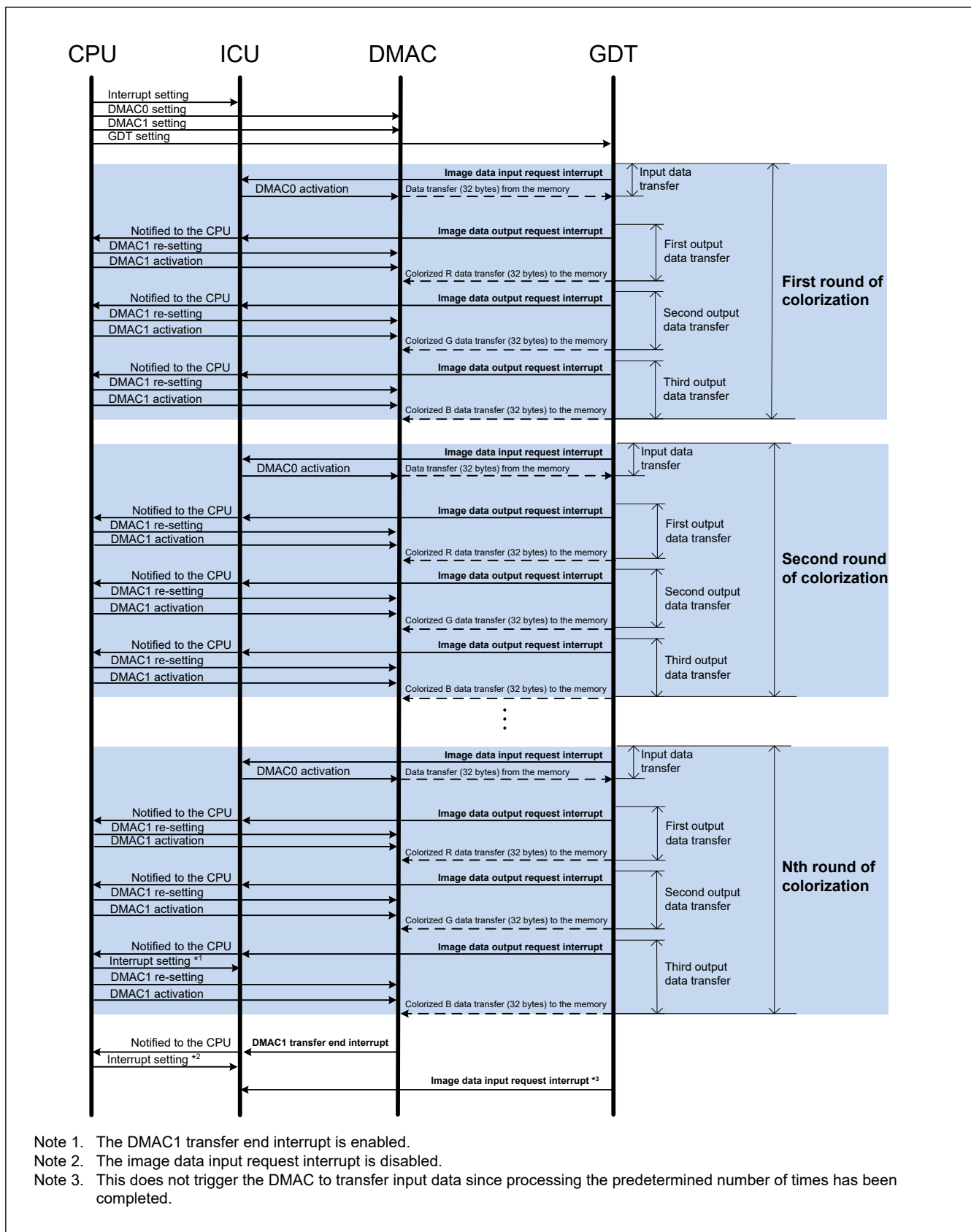


Figure 41.45 Example of a flow of colorization with the use of the DMAC for data transfer requiring CPU intervention for control

Figure 41.46 shows an example of a flow of colorizing images N consecutive times with data transfer handled by the DTC. Each of the shaded parts in the figure shows the flow of transfers for a single round of colorization. A single round of colorization is handled in the following order from step 1 to step 4.

1. The GDT issues an image data input request interrupt to the DTC.
2. The DTC transfers the input image data from the memory to the GDTIBUFn registers by chain transfer.
3. The GDT colorizes the data stored in the GDTIBUFn registers, and stores the colorized data in the GDTOBUFn registers. The GDT then issues an image data output request interrupt to the DTC.
4. The DTC transfers the output image data from the GDTOBUFn registers to the memory by chain transfer.

When colorizations have been completed N times, the DTC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the DTC, which, however, does not trigger the DTC to transfer the input data since processing of predetermined number of times has been completed.

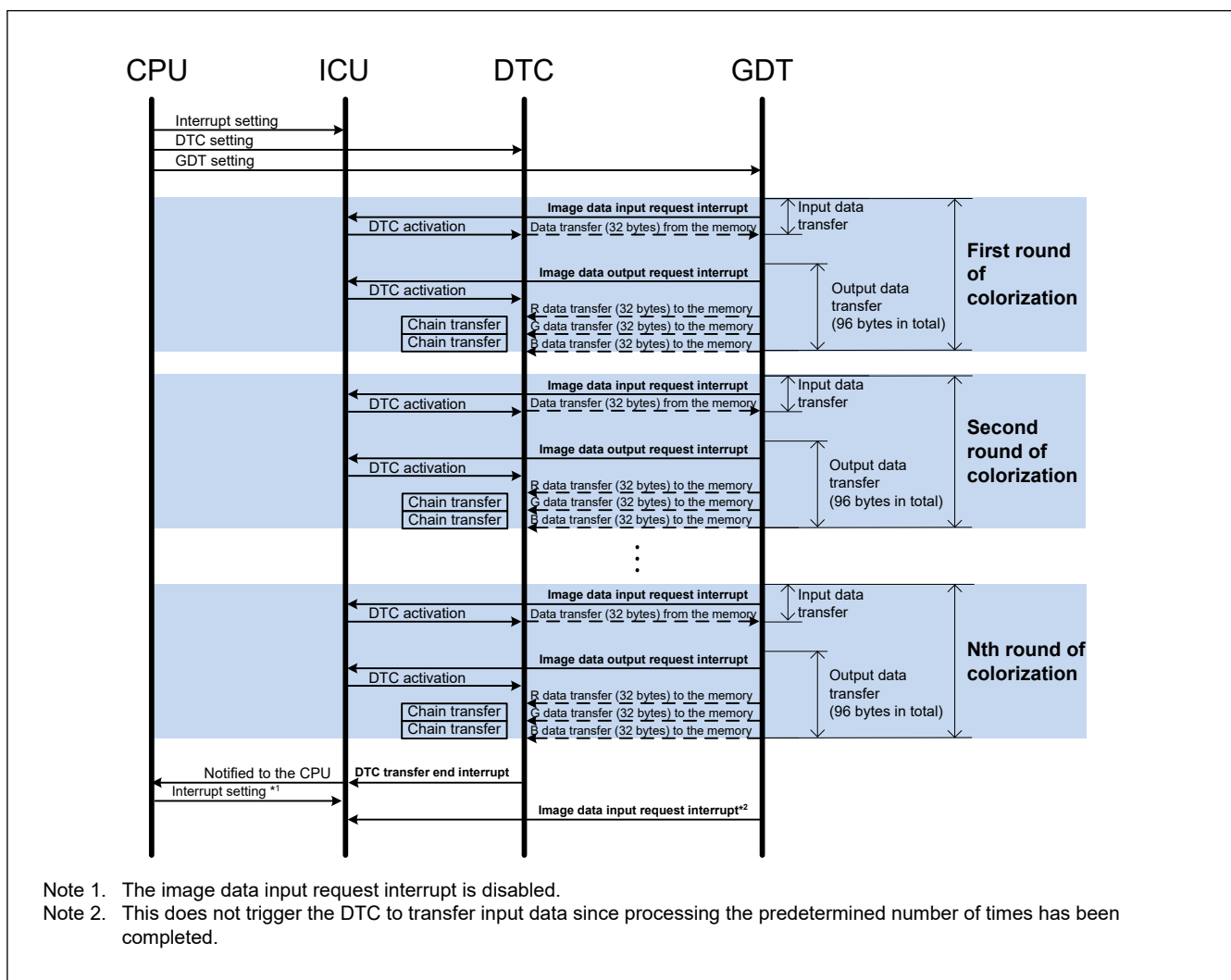


Figure 41.46 Example of a flow of colorization with the use of the DTC

41.3.10 Color Data Sorting

The GDT supports the sorting of color images. This function sorts R, G, and B image data that are stored in separate areas of the memory into contiguous data and writes them back to the memory, or outputs them externally through the communications interface such as an SPI. With the input of three 32-byte images (R, G, and B), sorting them into the contiguous data represented by three bits in order of R, G, and B is possible by setting the GDTCCR.CIALGSL bit to 0. The output image size is 96 bytes. Setting the GDTCCR.CIALGSL bit to 1 allows sorting into contiguous R, G, and B data represented by three bits, with one bit of padding, for a total of four bits. The output image size in this case is 128 bytes.

Figure 41.47 shows an example of color data sorting.

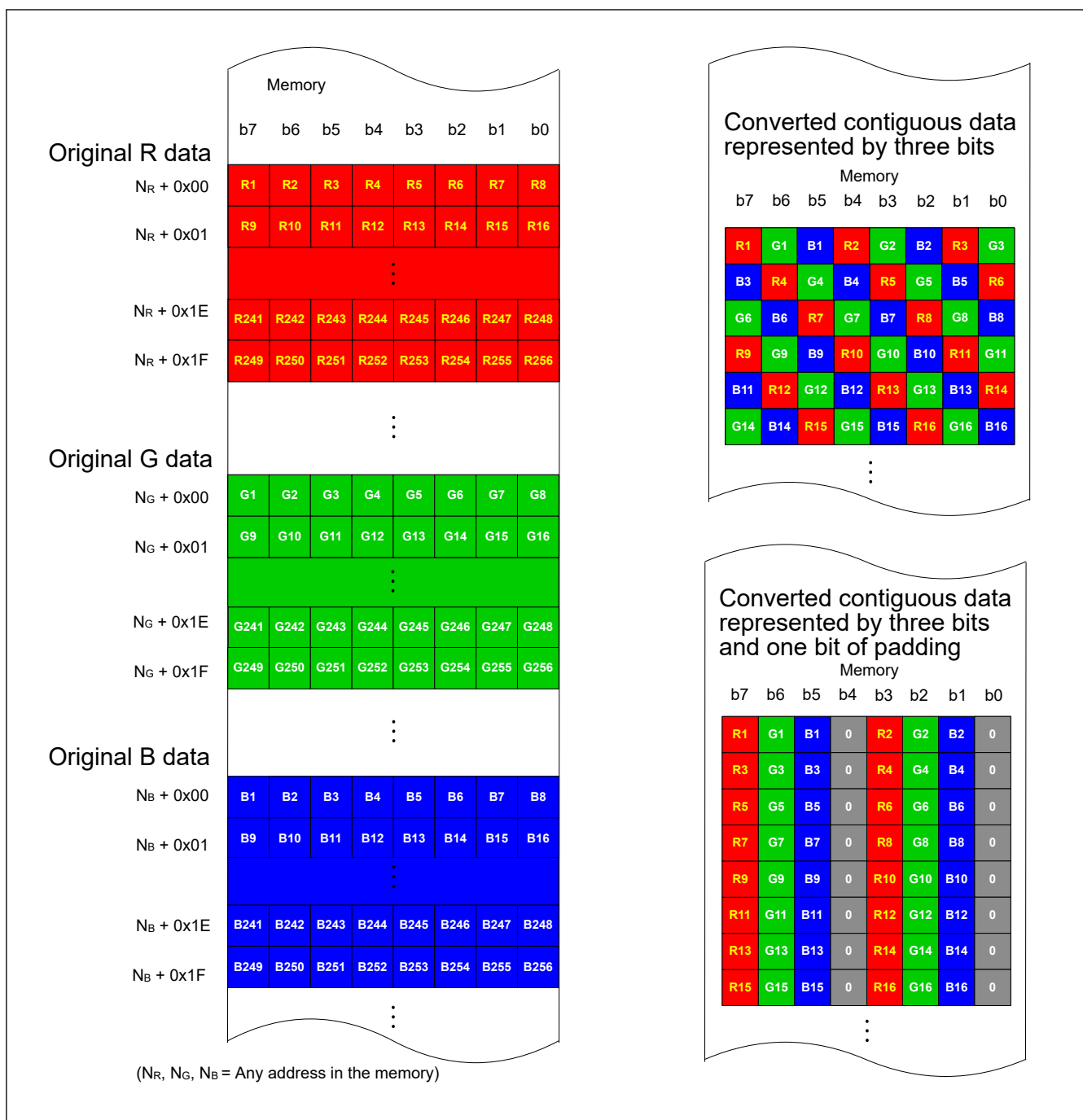


Figure 41.47 Example of color data sorting

Table 41.16 and Table 41.17 list the GDT image data input registers and GDT image data output registers to be used in color sorting when GDTCR.CIALGSL = 0 and when GDTCR.CIALGSL = 1, respectively.

Table 41.16 Correspondence between registers used in color data sorting (3-bit data mode) and interrupt mode

Input image size (bit)	Image data input request interrupt				Output image size (bit)	Image data output request interrupt		
	Interrupt mode	Number of interrupts	Transfer for one interrupt (bytes)	GDT Image Data Input Register		Number of interrupts	Transfer for one interrupt (bytes)	GDT Image Data Output Register
16 × 16	DMAC	3	32	First interrupt: GDTIBUF0 to GDTIBUF7 (R image data) 2nd: GDTIBUF8 to GDTIBUF15 (G image data) 3rd: GDTIBUF16 to GDTIBUF23 (B image data)	16 × 16	1	96	First interrupt: GDTIOBUF0 to GDTIOBUF23
	DTC	1	96	First interrupt: GDTIBUF0 to 7 (R image data) GDTIBUF8 to GDTIBUF15 (G image data) GDTIBUF16 to GDTIBUF23 (B image data)				

Table 41.17 Correspondence between registers used in color data sorting (4-bit data mode) and interrupt mode

Input image size (bit)	Image data input request interrupt				Output image size (bit)	Image data output request interrupt		
	Interrupt mode	Number of interrupts	Transfer for one interrupt (bytes)	GDT Image Data Input Register		Number of interrupts	Transfer for one interrupt (bytes)	GDT Image Data Output Register
16 × 16	DMAC	3	32	First interrupt: GDTIBUF0 to GDTIBUF7 (R image data) 2nd: GDTIBUF8 to GDTIBUF15 (G image data) 3rd: GDTIBUF16 to GDTIBUF23 (B image data)	16 × 16	1	128	First interrupt: GDTIOBUF0 to GDTIOBUF31
	DTC	1	96	First interrupt: GDTIBUF0 to GDTIBUF7 (R image data) GDTIBUF8 to GDTIBUF15 (G image data) GDTIBUF16 to GDTIBUF23 (B image data)				

Figure 41.48 shows an example of a flow of sorting color images N consecutive times to produce contiguous data represented by four bits with data transfer handled by the DMAC. In this example, DMAC channel 0 is used to transfer the input image data from the memory to the GDTIBUFn registers. DMAC channel 1 is used to transfer the output image data from the GDTIOBUFn registers to the memory. Each of the shaded parts in the figure shows the flow of transfers for a single round of sorting. A single round of color data sorting is handled in the following order from step 1 to step 6.

1. The GDT issues an image data input request interrupt to the CPU.
2. The CPU re-sets the information required for transfer by the DMAC and then issues a DMA transfer request.
3. The DMAC transfers the input image data from the memory to the GDTIBUFn registers.
4. Steps 1 to 3 are repeated three times.
5. The GDT sorts the data stored in the GDTIBUFn registers, stores the sorted data in the GDTIOBUFn registers, and issues an image data output request interrupt as a trigger for the DMAC.
6. The DMAC transfers the output image data from the GDTIOBUFn registers to the memory.

When sorting has been completed N times, the DMAC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the CPU, which, however, does not trigger the DMAC to transfer the input data since processing of predetermined number of times has been completed.

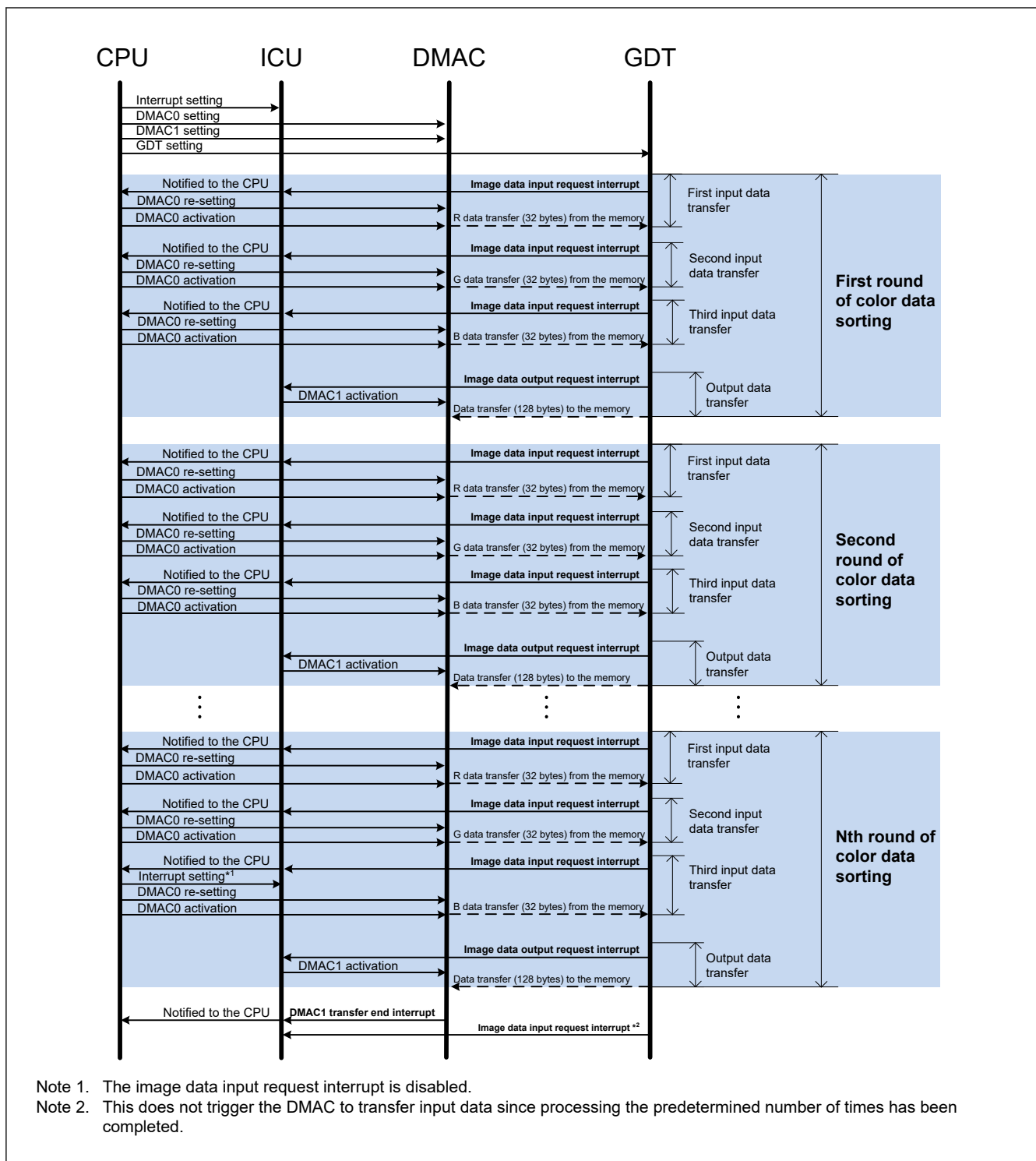


Figure 41.48 Example of a flow of color data sorting with the use of the DMAC for data transfer requiring CPU intervention for control

Figure 41.49 shows an example of a flow of sorting color images N consecutive times to produce the contiguous data represented by four bits with data transfer handled by the DTC. Each of the shaded parts in the figure shows the flow of transfers for a single round of sorting. A single round of sorting is handled in the following order from step 1 to step 4.

1. The GDT issues an image data input request interrupt to the DTC.
2. The DTC transfers the input image data from the memory to the GDTIBUFn registers by chain transfer.
3. The GDT sorts the data stored in the GDTIBUFn registers, stores the sorted data in the GDTTOBUFn registers, and issues an image data output request interrupt as a trigger for the DTC.

4. The DTC transfers the output image data from the GDTBUF_n registers to the memory.

When sorting has been completed N times, the DTC notifies the CPU of the completion of transfer. The GDT issues an image data input request to the DTC, which, however, does not trigger the DTC to transfer the input data since processing of predetermined number of times has been completed.

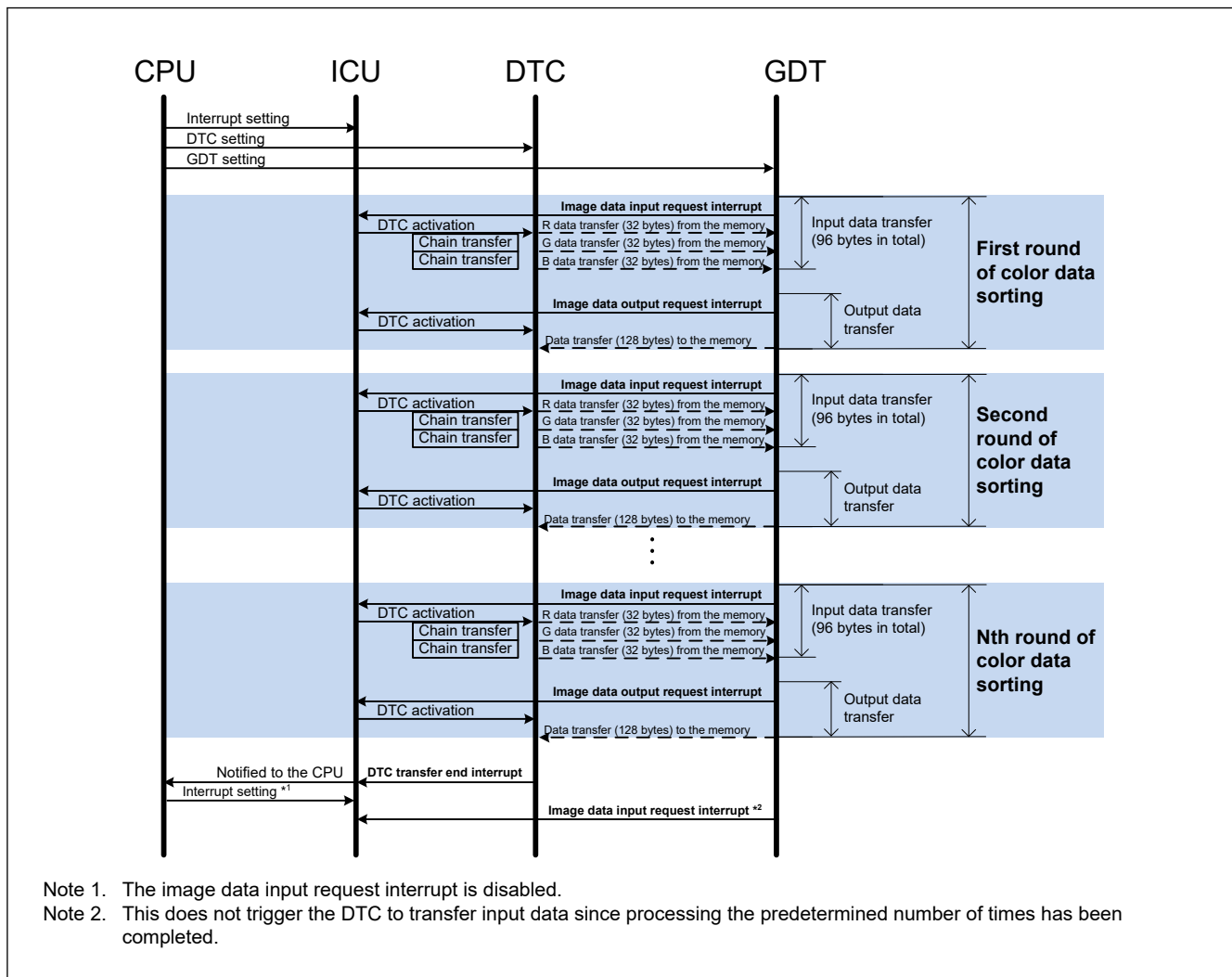


Figure 41.49 Example of a flow of color data sorting with the use of the DTC for data transfer requiring CPU intervention for control

41.3.11 Endian Conversion Function

The GDT supports endian conversion. This function allows swapping of the MSB and LSB in the arrangement of bits in the output data. Figure 41.50 shows the endian conversion.

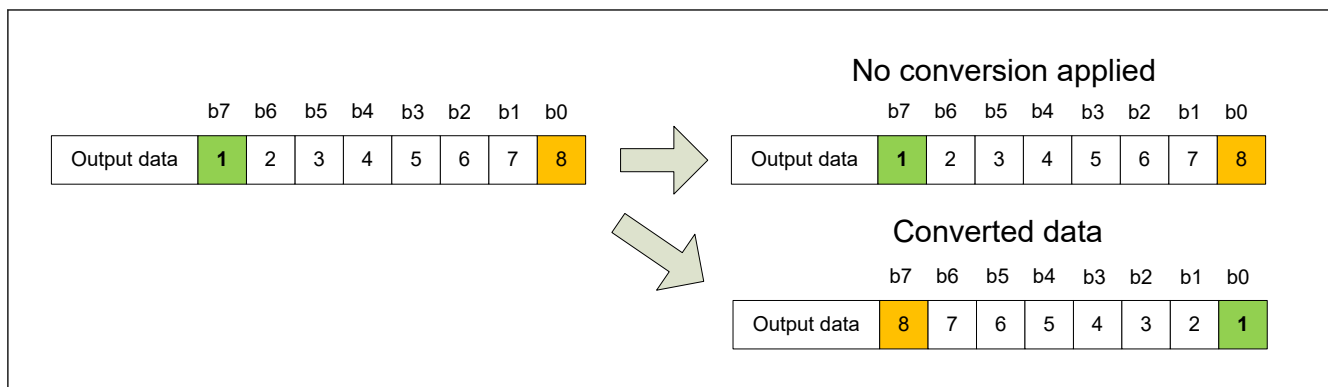
**Figure 41.50 Endian conversion**

Table 41.18 shows the GDT Image Data Input Register and GDT Image Data Output Register used in the endian conversion function.

Table 41.18 Registers used in endian conversion

Input image size (bit)	Image data input request interrupt				Output image size (bit)	Image data output request interrupt		
	Interrupt mode	Number of interrupts	Transfer for one interrupt (byte)	GDT Image Data Input Register		Number of interrupts	Transfer for one interrupt (byte)	GDT Image Data Output Register
16 × 16	—*1	1	32	First interrupt: GDTIBUF0 to GDTIBUF7	16 × 16	1	32	First interrupt: GDTIBUF0 to GDTIBUF7

Note 1. The number of interrupts, amount transferred per interrupt, and registers used are the same regardless of the interrupt mode setting.

For more information on data transfer flows using DMAC and DTC, see Figure 41.21 and Figure 41.22 in section 41.3.4. [Inversion](#)

41.3.12 Time Required for Processing a Single Image

Table 41.19 lists the time required for processing by functions. Inversion of an image can be done at the same time as any function and takes virtually zero cycles to complete the processing. The figure in parentheses in the “Time Required for Processing” column for inversion indicates what it takes when inversion is solely used.

Table 41.19 Time required for processing a single image by functions (1 of 2)

Function	Input image size in bits	Processing time in cycles of PCLKA
Monochrome compositing	8 × 8	6
	16 × 16	12
Color compositing	8 × 8	6
	16 × 16	12
Scrolling	16 × 16	12
Colorization	16 × 16	12
Color data sorting	16 × 16	12
Endian conversion	16 × 16	12
Rotation	8 × 8	2
	16 × 16	3
Scaling down	8 × 8	15 for scaling down to 1/8 13 for scaling down to 2/8 or 3/8 7 for scaling down to 4/8 to 7/8
		7 for scaling down to 1/2 by pixel skipping

Table 41.19 Time required for processing a single image by functions (2 of 2)

Function	Input image size in bits	Processing time in cycles of PCLKA
Conversion of glyph data into image data	8 × 8	9 to 11
Inversion	16 × 16	0 (12)

41.3.13 Interrupt Sources

The GDT has three interrupt sources:

- Image data input request interrupt
- Image data output request interrupt
- Glyph data conversion complete interrupt

Table 41.20 Interrupt sources

Name	Interrupt source	Interrupt Enable Bit	Interrupt to the CPU	Starting DMAC	Starting DTC
GDT_DATII	• The GDT processing starts. • Output data are read from the GDTBUF_n registers.	GDTPIER.DATIIE	Possible	Possible	Possible
GDT_DATOI	The GDT processing starts.	GDTPIER.DATOIE	Possible	Possible	Possible
GDT_FDCENDI	Conversion of glyph data for the size set in the GDTDFCS.FDLTDSZ[5:0] and GDTDFCS.FDLNGSZ[6:0] bits is completed.	GDTPIER.FDCENDIE	Possible	Possible	Possible

To use the DMAC or DTC for data transfer, start by setting the DMAC or DTC to enable operation and make settings for the GDT. For details on setting up the DMAC or DTC, see [section 19, DMA Controller \(DMAC\)](#), or [section 20, Data Transfer Controller \(DTC\)](#).

41.4 Transferring Image Data to a Color LCD

41.4.1 Overview

Using the GDT with the serial peripheral interface (SPI), general PWM timer (GPT), and I/O port functions allows the transfer of images to an LCD that can display images in color. This section describes how to display color images on the memory-in-pixel (MIP) color LCD indicated in [Table 41.21](#) as an example.

Table 41.21 Reference Memory-in-Pixel (MIP) color LCD

Manufacturer	Product name	Number of pixels
Japan Display Inc. (JDI)	LPM013M126A	176 × (RGB) × 176

[Table 41.22](#) lists the pins of the reference MIP color LCD.

Table 41.22 I/O Pins of the reference MIP color LCD (1 of 2)

Pin	Symbol	Description	I/O	Connection
1	SCLK	Serial clock	Input	Connect the SPI output signal (RSPCK) of the MCU
2	SI	Serial data	Input	Connect the SPI output signal (MOSI) of the MCU
3	SCS	Chip select	Input	Connect the SPI output signal (SSL) of the MCU
4	EXTCOMIN	COM reversal signal	Input	Connect the GPT output signal of the MCU
5	DISP	Display ON/OFF control	Input	Connect the I/O port output signal of the MCU
6	VDDA	Power supply (analog)	—	Supply power on the board
7	VDD	Power supply (logic)	—	Supply power on the board
8	EXTMODE	COM reversal mode select	Input	Control mode by EXTCOMIN. Connect this to VDD of the MIP LCD on the board.
9	VSS	Ground (logic)	—	Connect this to the ground on the board

Table 41.22 I/O Pins of the reference MIP color LCD (2 of 2)

Pin	Symbol	Description	I/O	Connection
10	VSSA	Ground (analog)	—	Connect this to the ground on the board

41.4.2 Peripheral Functions Used for Color Display

Table 41.23 shows the control for the MIP color LCD and functions to be used.

Table 41.23 Type of control for MIP color LCD and functions to be used

Control item	Description	Function used
Display control	Control the DISP pin. See section 41.4.2.1. Display Control .	section 22, I/O Ports
COM signal control	Input periodic pulses in the EXTCOMIN pin. See section 41.4.2.2. COM Signal Control	section 24, General PWM Timer (GPT)
Command data transmission	Transfer a control command from the serial interface pin. See section 41.4.2.3. Transfer of Command Data	section 35, Serial Peripheral Interface (SPI)
Image data transmission	Transfer image data from the serial interface pin	section 35, Serial Peripheral Interface (SPI)

Figure 41.51 shows an example of a block diagram with control and linkage for the functions used.

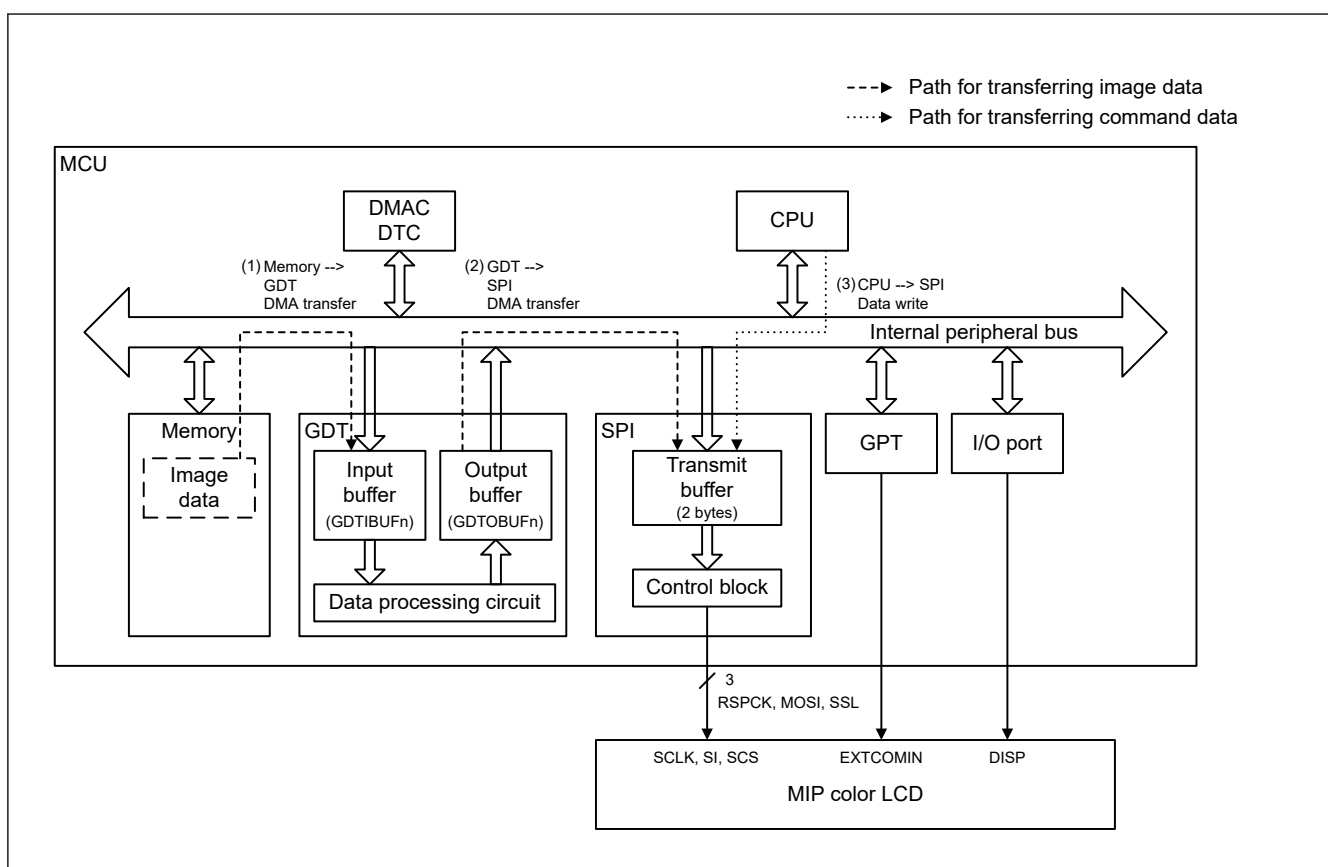
**Figure 41.51 Example of controlling the MIP color LCD**

Image data in the memory are transferred to the GDT image data input register through DMA transfer to be processed. The amount of data transferred is one line^{*1}. One line of data stored in the GDT image data output register is transferred to the SPI transmit buffer through DMA transfer and serially transferred to the color LCD. When transmitting command data or transmitting header data in transmission of image data to the color LCD, the CPU writes 2-byte data to the SPI transmit buffer. For more details on the color LCD specifications such as the start/stop sequence, command data mode setting, and image data format, refer to the manual of the color LCD.

Note 1. For 176 × 176-bit LCD, one line is: 176 bits (R image) + 176 bits (G image) + 176 bits (B image) = 528 bits (66 bytes)

41.4.2.1 Display Control

Driving the I/O port pin connected to the DISP pin of the color LCD high or low to switch the display on and off is possible. For details on the setting, see [section 22, I/O Ports](#).

41.4.2.2 COM Signal Control

Periodic pulse signals are input to the EXTCOMIN pin of the color LCD. Pulse signals can be generated by using the GPT function. [Table 41.24](#) shows a setting example when a pulse signal is generated by the GPT. For details about the registers and setting procedure, see [section 24, General PWM Timer \(GPT\)](#). For details about the input signal specifications of the EXTCOMIN pin, see the manual for the color LCD.

Table 41.24 Example of settings for the GPT

Parameter	Description
Counting mode	Set the GTUDDTYC.UD bit to 1 to select counting up.
Pin function select	Set the GTIOA/GTIOB[4:0] bits and the OAE/OBE bit in the GTIOR register to 01001b and 1, respectively, to produce a waveform that is driven high at the end of the cycle of counting and driven low by matches in comparison with the GTCCRA/GTCCRB register (use the registers and bits that correspond to the output pin of the GPT to be used).
Prescaler	Set the GTCR.TPCS[2:0] bits to 101b to select PCLKA division by 1024 (in general, set a value that suits the cycle of generation).
Cycle setting	Set the counter cycle. Setting the GTPR register to n selects counting from 0 to n.
Compare capture	Set the GTCCRn register to 0x0000 to set low output when the counter is cleared to 0.

[Table 41.24](#) shows the waveform output by the GPT with the settings in [Table 41.24](#).

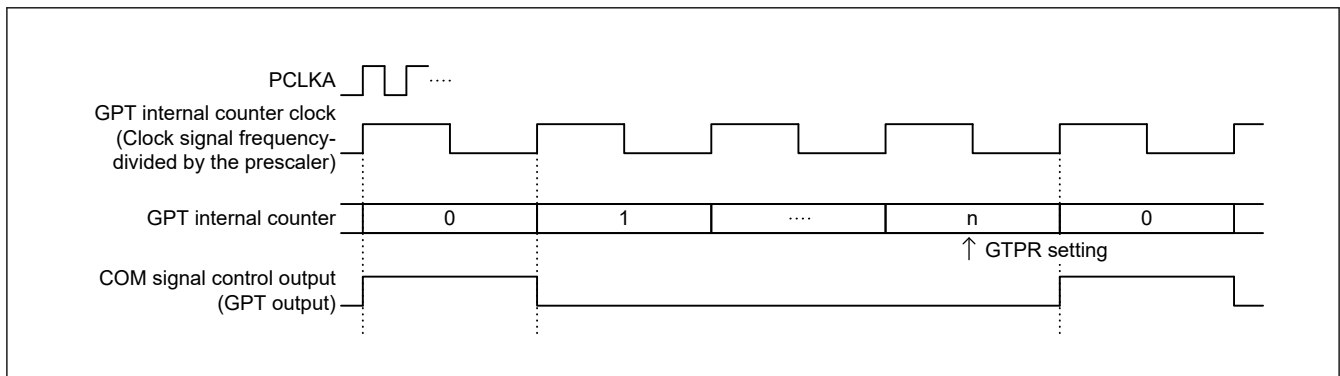


Figure 41.52 Example of waveform output by the GPT

41.4.2.3 Transfer of Command Data

To transfer control command data such as No-Update, All-Clear, Blinking Color, and Color Inversion, to the color LCD, start by specifying the basic settings for the SPI and then write 2-byte data to the SPI transmission buffer. DMA transfer is not used. All transfer is controlled by the CPU. For the settings of control command data, refer to the manual for the color LCD.

[Table 41.25](#) shows an example of the settings for serial transfer through the SPI.

Table 41.25 Example of setting the SPI (1 of 2)

Parameter	Description
Operating mode	Set the MSTR and TXMD bits in the SPCR register to 1 to select transmission in master mode.
Bit rate	The bit rate is determined from the division ratio of PCLKA by setting the SPBR register and SPCMD0.BRDV[1:0] bits. Set these registers so that the bit rate is no higher than 1.33 Mbps, see Figure 41.53 .
Clock	Set the CPHA and CPOL bits in the SPCMD0 register to 0 to specify the clock phase for data sampling as rising edges and driving of the signal low while transfer is not in progress.

Table 41.25 Example of setting the SPI (2 of 2)

Parameter	Description
Slave signal select (SSLn)	Set the bit (SSL0P to SSL3P) of the SSLP register for the slave signal to be used to 1 to select active high. Set the SPCKD.SCKDL[2:0] bits (RSPCK delay) so that t_{LEAD} (SSLn setting time) becomes at least 6 μs, see Figure 41.53. t_{LEAD} is adjusted by the number of cycles of RSPCK specified by this register. When the bit rate is no higher than 1.33 Mbps, this value should be set to at least 6 μs. Set the SSLND.SLNDL[2:0] bits (SSL negation delay) so that t_{LAG} (SSLn hold time) is at least 2 μs. t_{LAG} is adjusted by the number of cycles of RSPCK specified by this register. Note that the delay time set in these bits is for setting the period from the transmission of a final RSPCK edge to the negation of the signal input on the SSLn bit, so add a half cycle of RSPCK and then adjust the value to be at least 2 μs.
Data transfer unit	Set the SPCMD0.SPB[3:0] bits to 0xF so that transfer is handled in 2-byte units.

Figure 41.53 shows an example of waveform in serial transfer with the basic SPI settings in Table 41.25.

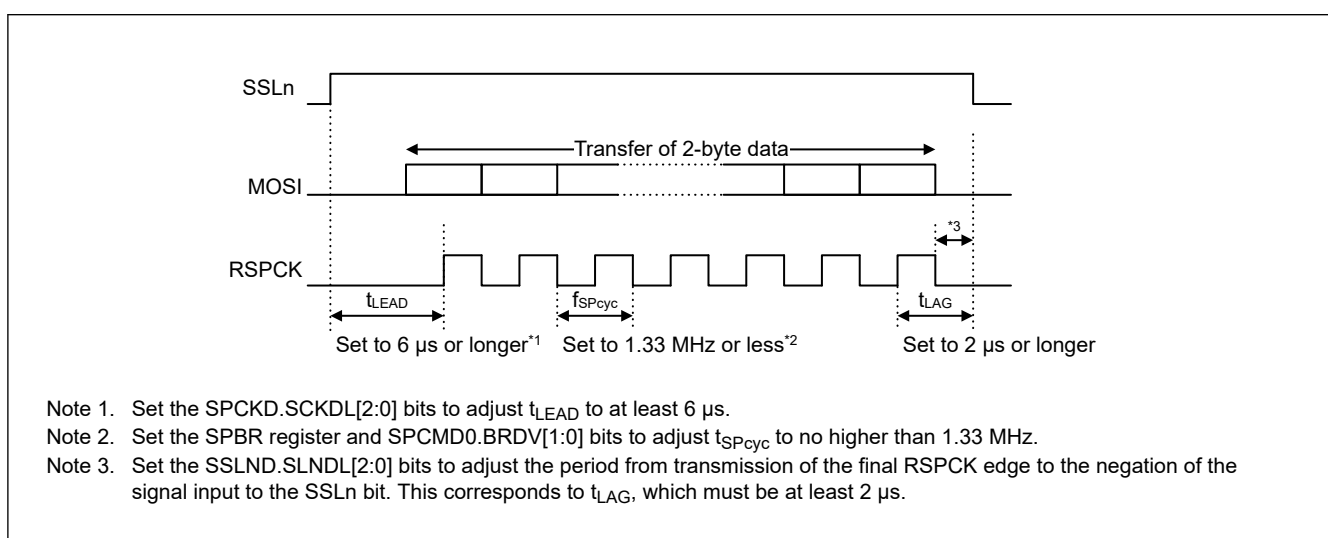
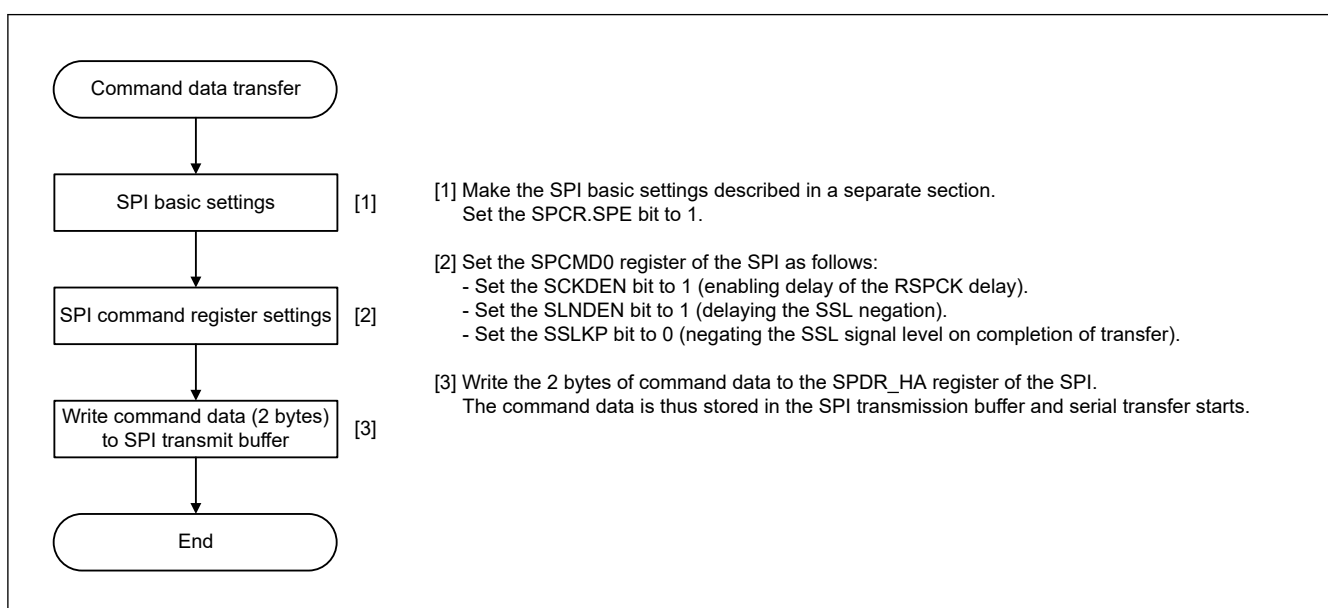
**Figure 41.53 Example of waveforms in serial transfer**

Figure 41.54 shows the procedure for transferring command data through the SPI.

**Figure 41.54 Example of procedure for transferring command data**

41.4.2.4 Transfer of Image Data

One line of image data stored in the GDT image data output register is DMA transferred to the SPI transmission buffer in 2 bytes. The data are then serially transferred from the SPI to the color LCD in line units. Transfer of multiple lines is handled by repeating the transfer in line units.

Transfer of one line of image data is handled while the SSL signal is being asserted. Data are transferred in the order listed in [Table 41.26](#). The SSL signal is negated once transfer is completed.

Table 41.26 Order of transfer and transmission of image data

Order	Transmit data	Description
1	Header data	The header, including the transfer mode and line address settings, is transmitted first. It has a fixed length of 16 bits, and is written to the SPI transmission buffer by the CPU.
2	Image data	One line of image data follows the header. The image data can be in the 1-bit (black or white), 3-bit (RGB), or 4-bit (RGB with a padding bit) format. The length of one line of data depends on the format. The data are DMA transferred to the SPI transmission buffer.
3	Transfer end data	These 16 bits of data follow the image data. The bits can have any value; in this example of a flow, the bits are all 0 and are written to the SPI transmission buffer by the CPU.

For the settings of the header data, see the manual for the color LCD. [Figure 41.55](#) shows an example of the sequence for the transmission of one line of image data.

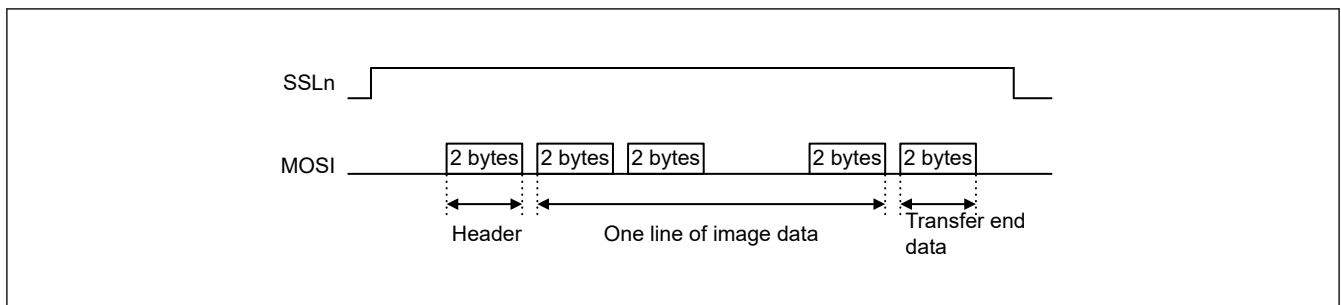


Figure 41.55 Example of transfer and transmission of image data

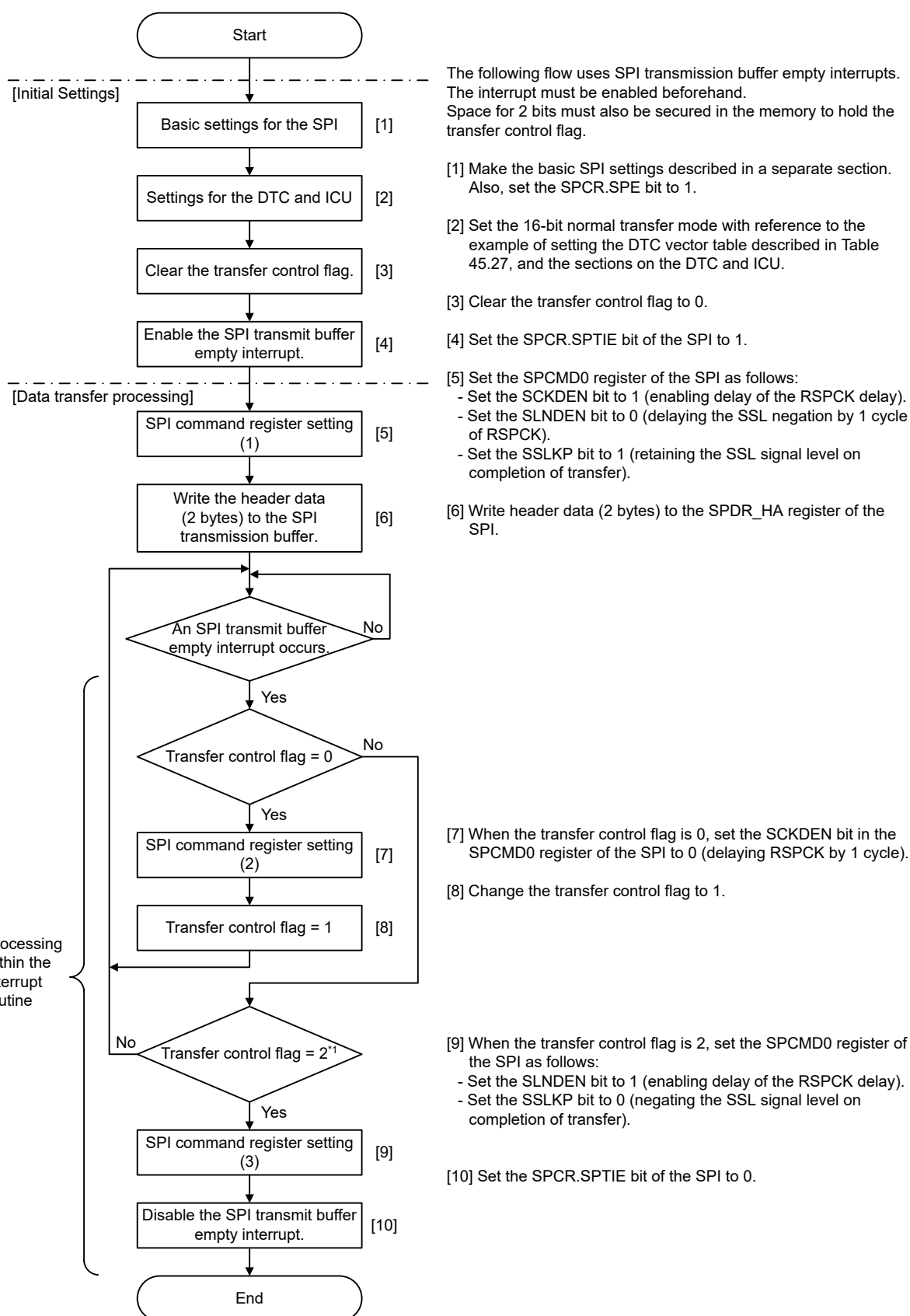
DMA transfer from the GDT to the SPI is handled in 16-bit normal transfer mode by using the DMAC or DTC. [Table 41.27](#) shows an example of settings for the vector table when transfer is to be handled by the DTC. For more details on settings of the GDT registers and related setting procedure, see [section 20, Data Transfer Controller \(DTC\)](#).

Table 41.27 Example of settings in the DTC vector table

Register	Setting
MRA	SM[1:0] 10b: The transfer source (GDT) address is to be incremented.
	SZ[1:0] 01b: The unit of transfer is 16 bits.
	MD[1:0] 00b: Normal transfer mode
MRB	DM[1:0] 00b: The transfer destination (SPI) address is fixed.
SAR[31:0]	Address where the GDT image data output registers (GDTBUF0) start
DAR[31:0]	Address of the SPI transmission buffer (SPDR_HA register)
CRA[15:0]	The number of rounds of image data transfer = one line (bytes) / 2 (bytes).

Activating the DTC requires setting of the required ICU event link setting register n. For details, see [section 16, Interrupt Controller Unit \(ICU\)](#).

[Figure 41.56](#) an example of a flow of the transfer of image data to a color LCD. [Figure 41.57](#) shows an example of operation during the transfer of one line of image data to the color LCD and [Table 41.28](#) shows the operation details.



Note 1. The transfer control flag is set to 2 within the interrupt routine triggered by completion of the DTC transfer.

Figure 41.56 Example of a flow for transferring one line of image data to a color LCD

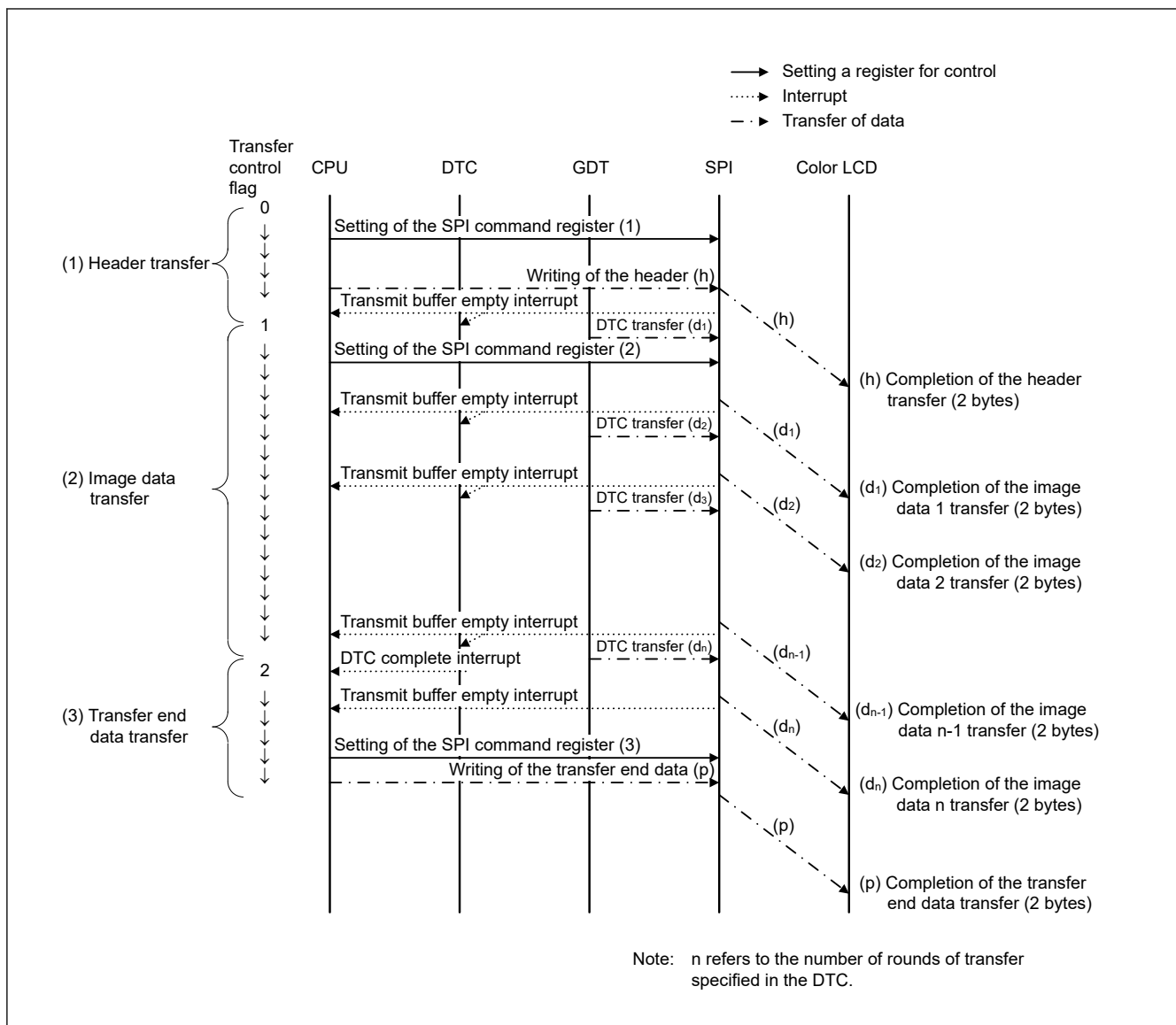


Figure 41.57 Example of operations during the transfer of one line of image data to a color LCD

Table 41.28 Processing details during the transfer of one line of image data to a color LCD

Transfer phase	Details of processing
(1) Header data transfer	- Set the SPI (SPI command register (1)) for transferring the header. - Write the header data (2 bytes) to the SPDR_HA register of the SPI to transfer the header to the SPI transmission buffer. Transfer to the color LCD starts after that.
(2) Image data transfer	- Transfer by the DTC starts in response to a transmit buffer empty interrupt triggered by transfer of the header. - When the first interrupt occurs, set the SPI (SPI command register setting (2)) for transfer of the image data. - The image data are transferred from the GDT buffer to the SPI in 2-byte units the number (n) of times specified in the DTC (2 bytes × n).
(3) Transfer end data transfer	- Transfer of the transfer end data starts in response to a transmit buffer empty interrupt triggered by completion of the DTC transfer. - Set the SPI (SPI command register setting 3) for transfer of the transfer end data. - Write the 2-byte transfer end data (all 0s) to the SPDR_HA register of the SPI to serially transfer the transfer end data, which completes the transfer of one line of data.

41.5 Usage Notes

41.5.1 Handling of Color Images

The handling of color images requires you to store the R, G, and B values in separate areas of the memory. Arranging data in the order R, G, and B may also be required. This depends on the color LCD module that you are using. In such cases, use the CIALGEN bit in the GDTCCR to enable the sorting of color data.

41.5.2 Settings for the Module-Stop Function

The Module Stop Control Register C (MSTPCRC) can enable or disable GDT operation. The GDT module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

42. Boundary Scan

42.1 Overview

The boundary scan function provides a serial I/O interface based on the JTAG (Joint Test Action Group), IEEE Std.1149.1, and IEEE Standard Test Access Port and Boundary Scan Architecture. [Table 42.1](#) lists the boundary scan specifications, [Figure 42.1](#) shows a block diagram, and [Table 42.2](#) lists the I/O pins.

Table 42.1 Boundary scan specifications

Parameter	Specifications
Execution condition	Boundary scan must be executed when the RES# pin is driven low.
Test modes	• BYPASS mode • EXTEST mode • SAMPLE/PRELOAD mode • CLAMP mode • HIGHZ mode • IDCODE mode

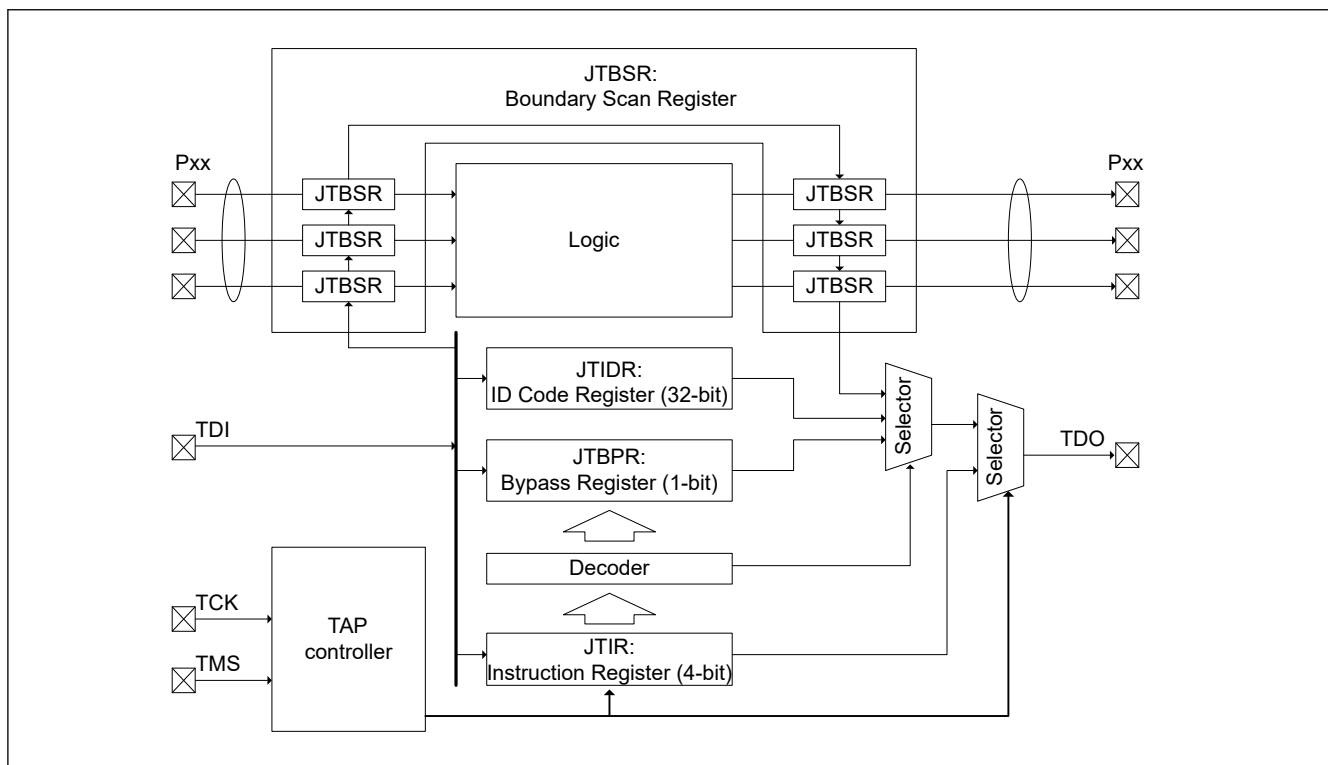


Figure 42.1 Boundary scan function block diagram

Table 42.2 Boundary scan I/O pins

Pin name	I/O	Description
TCK	Input	Test clock input pin Clock signal for boundary scan. The input clock duty cycle is 50% when the boundary scan function is used.
TMS	Input	Test mode select pin
TDI	Input	Test data input pin
TDO	Output	Test data output pin
BSCANP	Input	IOVCC power supply forced input pin When this pin is set to high level with IOVCC power supply, power supply to all I/O ports is enabled.

Note: This device does not support the TRST# pin for the JTAG interface.

42.2 Register Descriptions

Table 42.3 lists the boundary scan registers.

Table 42.3 Boundary scan registers

Register name	Symbol	Value after reset
Instruction Register	JTIR	0xE
ID Code Register	JTIDR	0x083E_4447
Bypass Register	JTBPR	Undefined
Boundary Scan Register	JTBSR	Undefined

Usage notes for the boundary scan registers:

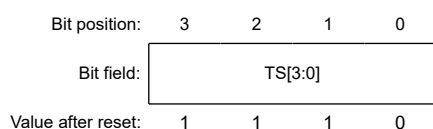
- Instructions can be input to the Instruction Register (JTIR) through the TDI pin by serial transfer.
- The Bypass Register (JTBPR), which is a 1-bit register, is connected between the TDI and TDO pins in BYPASS mode.
- The Boundary Scan Register (JTBSR), which is configured according to the BSDL description, is connected between the TDI and TDO pins when test data is being shifted in.

Table 42.4 shows the availability of serial transfer for the registers.

Table 42.4 Serial transfer for registers

Register name	Serial input	Serial output
Instruction Register (JTIR)	Available	Available
ID Code Register (JTIDR)	Available	Available
Bypass Register (JTBPR)	Available	Available
Boundary Scan Register (JTBSR)	Available	Available

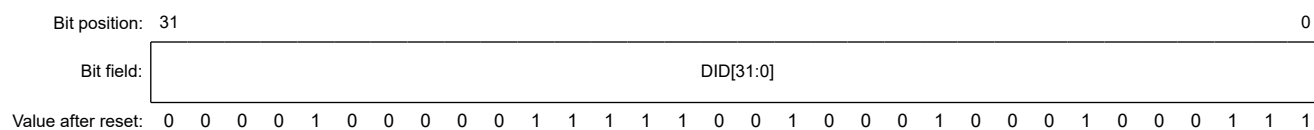
42.2.1 JTIR : Instruction Register



Bit	Symbol	Function	R/W	
3:0	TS[3:0]	Test Bit Set The command configuration for these bits	—	
		TS[3:0]		Instruction
		0x0		EXTEST
		0x1		SAMPLE/PRELOAD
		0x3		IDCODE (Renesas code)
		0x5		CLAMP
		0x6		HIGHZ
		0xF		BYPASS
		Others		Reserved

JTAG instructions can be transferred to the JTIR register by serial input from the TDI pin. The JTIR register is initialized when a power-on reset occurs, or when the TAP controller is in the Test-Logic-Reset state.

42.2.2 JTIDR : ID Code Register



Bit	Symbol	Function	R/W
31:0	DID[31:0]	Device ID These bits store the fixed value that indicates the device IDCODE (0x083E_4447).	—

The JTIDR register data is output from the TDO pin when the IDCODE instruction is executed.

42.2.3 JTBPR : Bypass Register

The JTBPR register is a 1-bit register and is connected between the TDI and TDO pins when the JTIR register is set to BYPASS mode. The JTBPR register cannot be read from or written to by the CPU.

42.2.4 JTBSR : Boundary Scan Register

The JTBSR register is a shift register for controlling the external input and output pins of this device, and is distributed across the pads. To apply the JTBSR register in boundary-scan testing, issue the EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ instructions. The BSDL file describes the associations between the JTBSR register bits and the pins of this device. The value after reset is undefined.

42.3 Operation

During a reset, the JTAG ports, TCK, TMS, TDI, and TDO, are assigned as default pin functions. The TCK, TMS, and TDI pins are pulled up by the pull-up resistors. Boundary scan testing can be executed after the setup time elapses when POR is negated and RES# is driven low.

42.3.1 TAP Controller

Figure 42.2 shows the state transition diagram of the TAP controller. All transitions are controlled by the TMS signal.

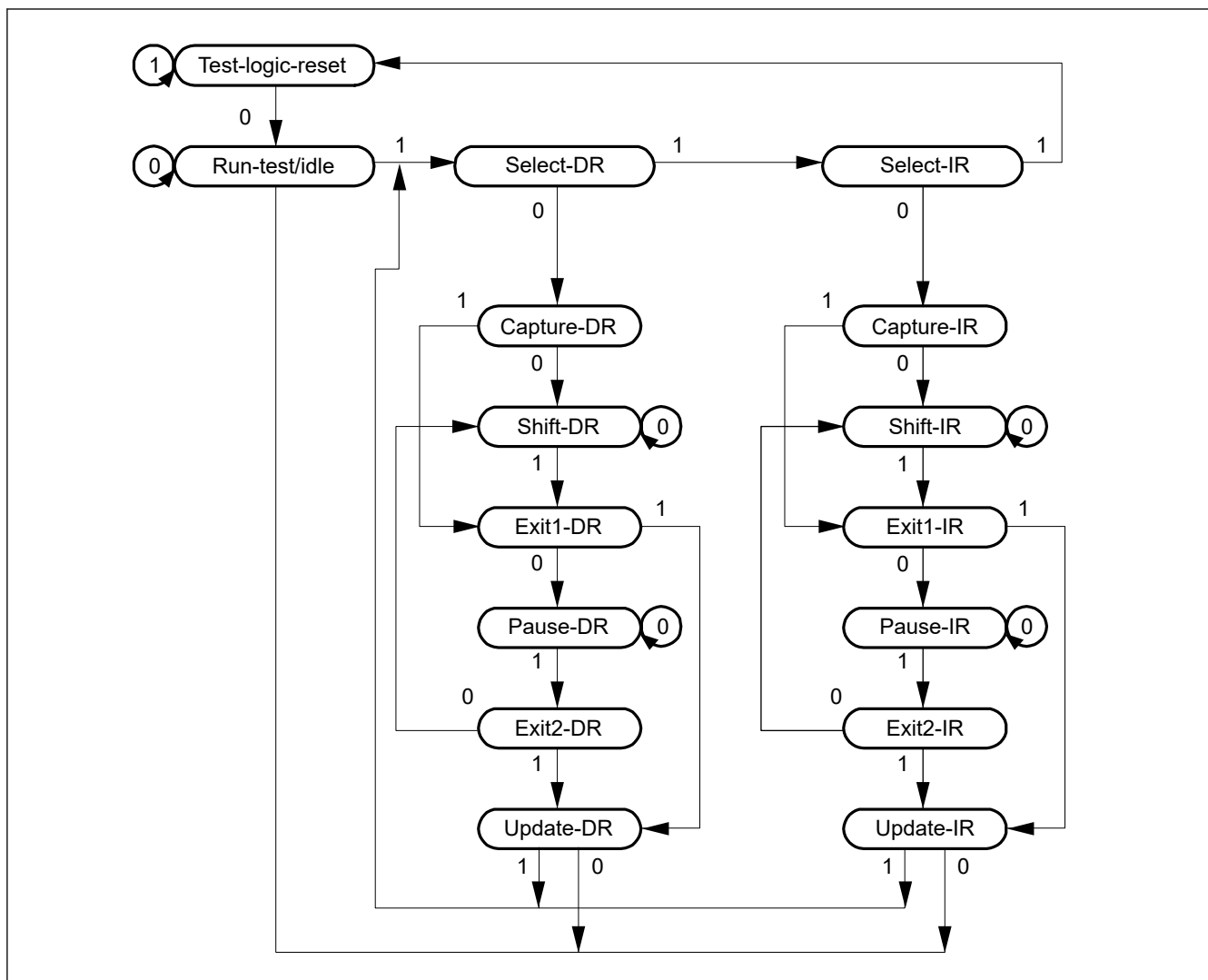


Figure 42.2 State transition diagram of TAP controller

42.3.2 Commands

(1) BYPASS

The BYPASS instruction drives the Bypass Register (JTBPR). This instruction shortens the shift path, facilitating the transfer of serial data to other LSIs on a printed circuit board at higher speeds. While this instruction is being executed, the test circuit has no effect on the system circuits.

The JTBPR register is connected between the TDI and TDO pins. Bypass operation is initiated from the Shift-DR operation. The TDO is low in the first clock cycle in the Shift-DR state. In the subsequent clock cycles, values input to the TDI pin are output from the TDO pin.

(2) EXTEST

The EXTEST instruction is used to test external circuits when this device is installed on the printed circuit board. If this instruction is executed, output pins are used to output test data (specified in the SAMPLE/PRELOAD instruction) from the Boundary Scan Register (JTBSR) to the other devices, and input pins are used to input the test result.

(3) SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction is used to input data from the internal circuits of this device to the JTBSR register, output data from the scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to this device and output signals are also directly output to the external circuits. This device system circuit is not affected by this instruction.

In SAMPLE operation, the JTBSR register latches a snapshot of the data transferred from the input pins to the internal circuit or data transferred from the internal circuit to the output pins. The latched data is read from the scan path. The JTBSR register latches the data snapshot on the rising edge of the TCK pin in the Capture-DR state. The data snapshot is only transferred from the internal circuit to the output pins during a reset.

In PRELOAD operation, the initial value is written from the scan path to the parallel output latch of the JTBSR register prior to the EXTEST instruction execution. If EXTEST is executed without executing this PRELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. (In the EXTEST instruction, output parallel latches are always output to the output pins.)

(4) IDCODE

When the IDCODE instruction is selected, the ID Code Register (JTIDR) value is output to the TDO pin in the Shift-DR state of the TAP controller. In this case, the JTIDR register value is output LSB-first. During this instruction execution, the test circuit does not affect the system circuit.

(5) CLAMP

When the CLAMP instruction is selected, output pins output the JTBSR register value that was specified in the SAMPLE/PRELOAD instruction in advance. While the CLAMP instruction is selected, the status of the JTBSR register is maintained regardless of the TAP controller state.

The JTBPR register is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction is selected.

(6) HIGHZ

When the HIGHZ instruction is selected, all output pins enter high-impedance state. While the HIGHZ instruction is selected, the JTBSR register is maintained regardless of the state of the TAP controller.

The JTBPR register is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction is selected.

42.4 Usage Notes

The boundary scan function is subject to the following constraints:

- The boundary scan must be executed when the RES# pin is driven low
- Serial data input/output is in LSB order, as shown in [Figure 42.3](#)

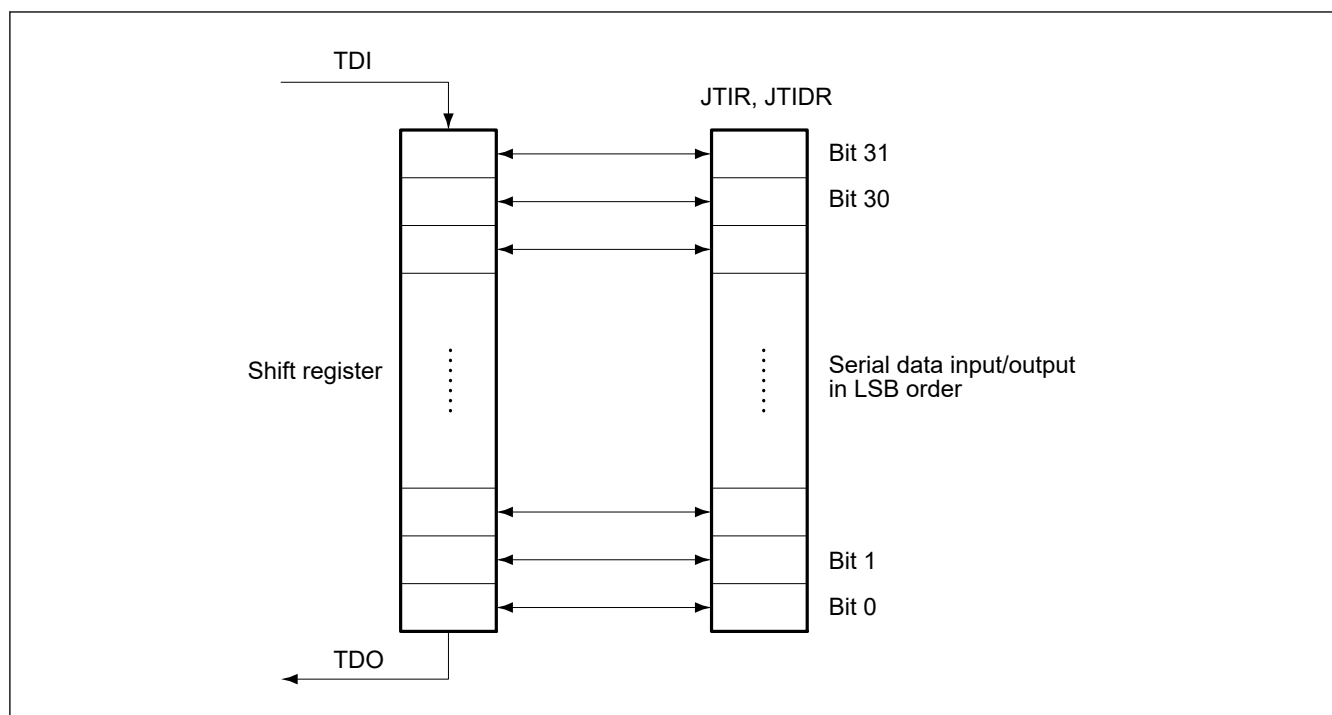


Figure 42.3 Serial data input/output

The following pins cannot be boundary-scanned:

- Power supply pins (VCC, VCL, VCLH, VSS, AVCC0, AVSS0, VSC_VCC, VSC_GND, IOVCC, IOVCC0/1, VBAT_EHC, VCC_SU, VBN, VBP, VREFL0, and VREFH0)
- Clock pins (EXTAL, XTAL, XCIN, and XCOU)
- Reset pin (RES#)
- The boundary-scan pins (TCK, TMS, TDI, and TDO)
- Mode pins (MD, EHMD, and BSCANP).

43. Trusted Secure IP Lite (TSIP-Lite)

43.1 Overview

The Trusted Secure IP Lite (TSIP-Lite) consists of the access management circuit, encryption engine, and random number generation circuit. In combination with the TSIP-Lite library, the TSIP-Lite can prevent eavesdropping (to maintain confidentiality), falsification of information (to ensure integrity), and impersonation (to verify authenticity).

Because key information required for encryption and decryption is stored only in the TSIP-Lite and all accesses from the outside can be blocked, TSIP-Lite enables building a more robust security system.

Table 43.1 lists the TSIP-Lite specifications. Figure 43.1 shows the TSIP-Lite block diagram.

Table 43.1 TSIP-Lite specifications

Parameter	Specifications
Access control	Access management circuit - In case of irregular access to the TSIP-Lite due to a tampered program or CPU runaway, this circuit blocks all subsequent accesses and stops data output from the TSIP-Lite - Privileged access signals are connected to the access management circuit to make it possible to limit the TSIP-Lite control only to privileged accesses.
Encryption engine	AES: Compliant with NIST FIPS PUB 197 - Key length: 128 bits or 256 bits - Data block size: 128 bits - Encryption usage modes ECB, CBC, CTR: Compliant with NIST SP 800-38A CMAC: Compliant with NIST SP 800-38B CCM: Compliant with NIST SP 800-38C GCM: Compliant with NIST SP 800-38D XTS: Compliant with NIST SP 800-38E GCTR - Number of execution cycles*1 ECB, CBC, CTR, CMAC, GCTR, XTS: 128-bit key: 44 PCLKA cycles, 256-bit key: 61 PCLKA cycles CCM: 128-bit key: 88 PCLKA cycles. AES-GCM - AES-GCM is realized by combining AES-GCTR and GHASH. Key management - Keys are only valid within the TSIP-Lite - Only key generation information is output to the outside of TSIP-Lite - Keys can be regenerated by inputting the key generation information to the TSIP-Lite Endian - Big-endian and little-endian are supported.
Random number generation	32-bit true random number generation circuit - The TSIP-Lite library can use 32-bit true random numbers to generate 128-bit and 256-bit true random numbers - Generated 128-bit and 256-bit true random numbers can be used as encryption and decryption keys.
Unique ID	- An ID unique to an MCU (unique ID) is accessible from the access management circuit through the dedicated bus - Combining the unique ID with the key generation information prevents the illicit copying of data to another MCU of the MCU group.
Interrupt source	Nine sources Data transfer by the DTC or DMAC is possible (two types).
Function for reducing power consumption	Transition to the module-stop state is possible.

Note 1. This does not include the overhead of calling TSIP-Lite library functions.

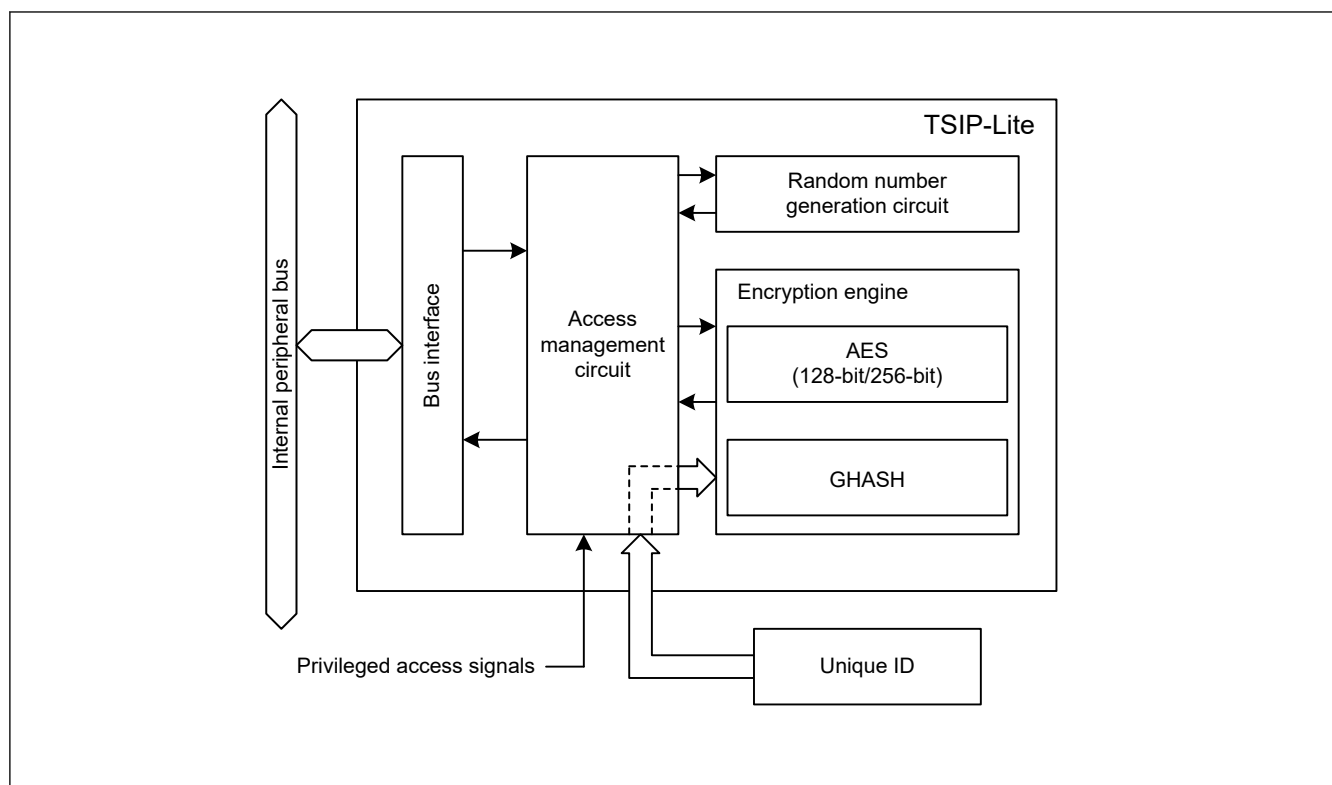


Figure 43.1 TSIP-Lite block diagram

43.2 Operation

43.2.1 Operating Modes and State Transition

Figure 43.2 shows the TSIP-Lite state transitions.

To execute the security function of the TSIP-Lite, use the TSIP-Lite library supplied by Renesas Electronics to perform state transitions as shown in the figure below.

If an irregular access to the TSIP-Lite occurs in breach of procedure due to a tampered program or CPU runaway, the subsequent accesses are blocked and data output from the TSIP-Lite stops.

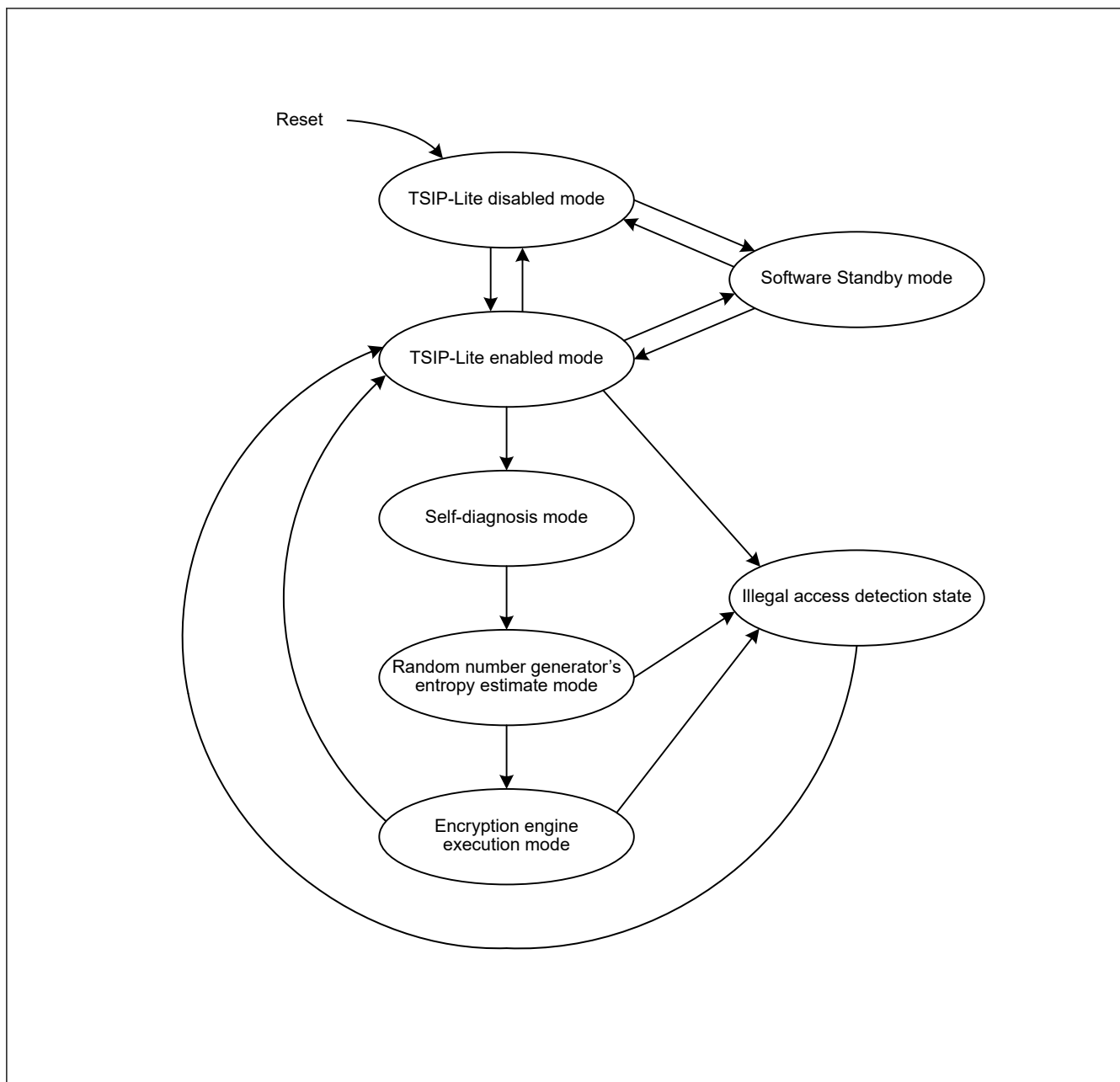


Figure 43.2 TSIP-Lite operating mode transition diagram

Most of the security functions available with the TSIP-Lite are executed in the encryption engine execution mode. In the encryption engine execution mode, the following processes can be executed.

1. Key data installation
2. Encryption/decryption
3. Creation of the key generation information
4. Random number generation

43.2.2 Encryption Engine

Figure 43.3 shows conceptual diagram of the encryption engine installed in the TSIP-Lite.

The encryption engine uses the key generation information, and converts the plaintext data to ciphertext or ciphertext data to plaintext through the hardware.

The encryption/decryption process can be completed without exposing the key data and the process's intermediate data to the outside of the TSIP-Lite.

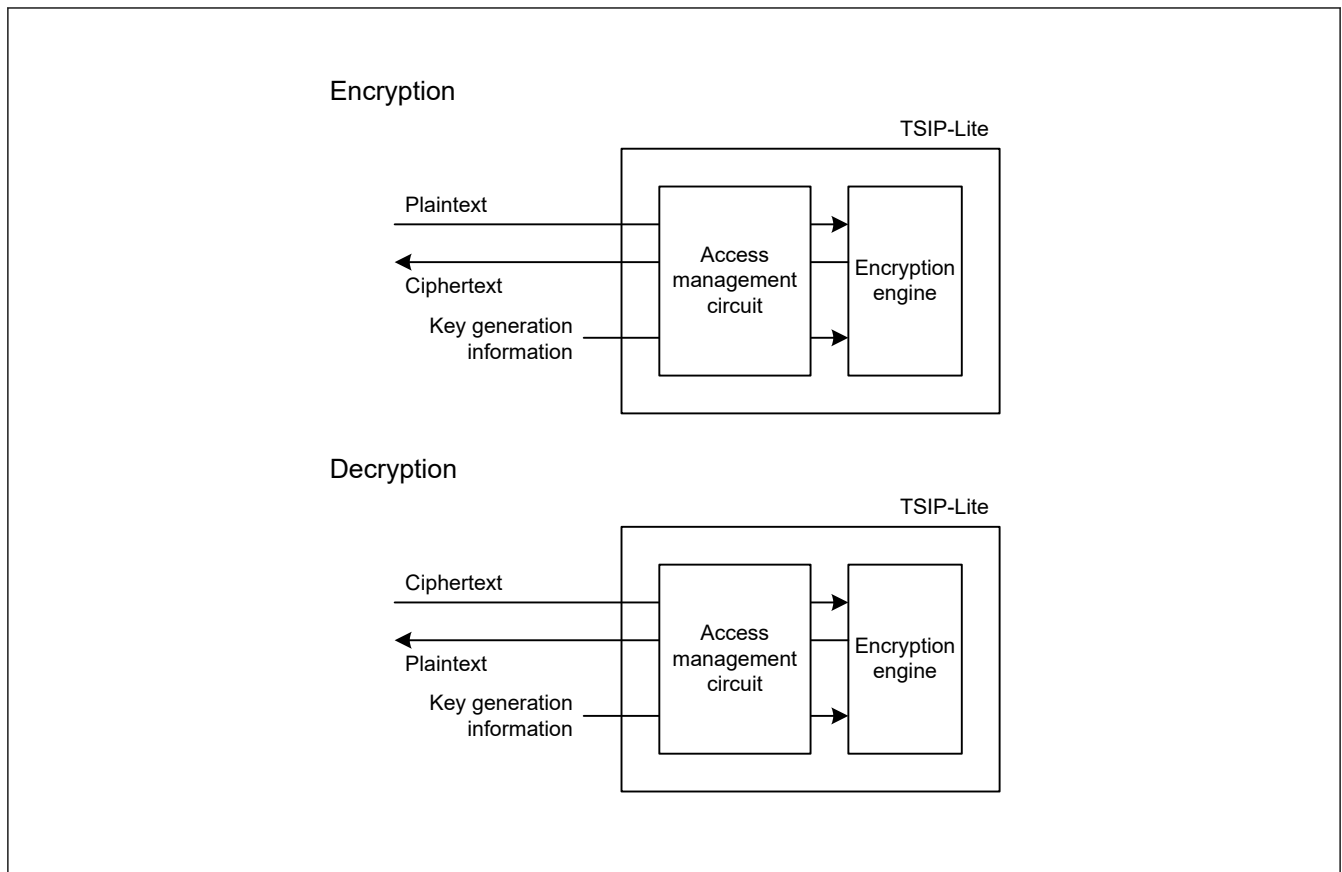


Figure 43.3 Conceptual diagram of the encryption engine

43.2.3 Key Data Installation

Installation of key data is a process that converts a user key safely to the key generation information and stores the converted information in the flash memory. Key data installation is executed through the procedure described below.

1. The user encrypts the user key (Key-1) with the user key encryption key (Key-2).
2. Then the user sends the encrypted user key (eKey-1) to the TSIP-Lite through the serial interface.
3. The TSIP-Lite restores Key-2 using the Key-2 key generation information (Index-2) provided by the TSIP-Lite library, and then uses this key to decrypt the user key (Key-1).
4. The TSIP-Lite converts the decrypted user key (Key-1) to the user key generation information (Index-1) using a unique ID and a random number, and then stores the Index-1 in the flash memory.

Figure 43.4 shows the conceptual diagram of the key installation and Figure 43.5 shows the installation flow.

After the key data is installed, encryption/decryption process can be performed using the user key generation information (Index-1).

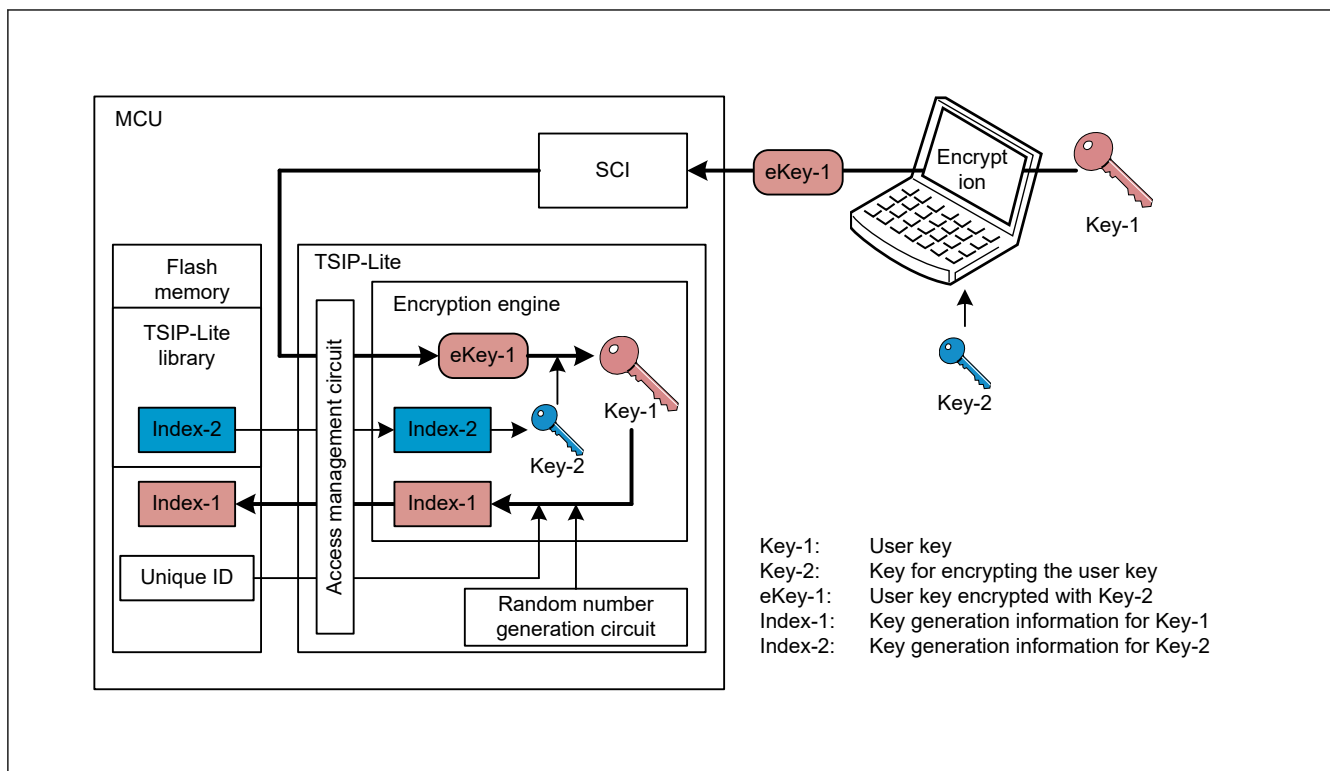


Figure 43.4 Conceptual diagram for key installation

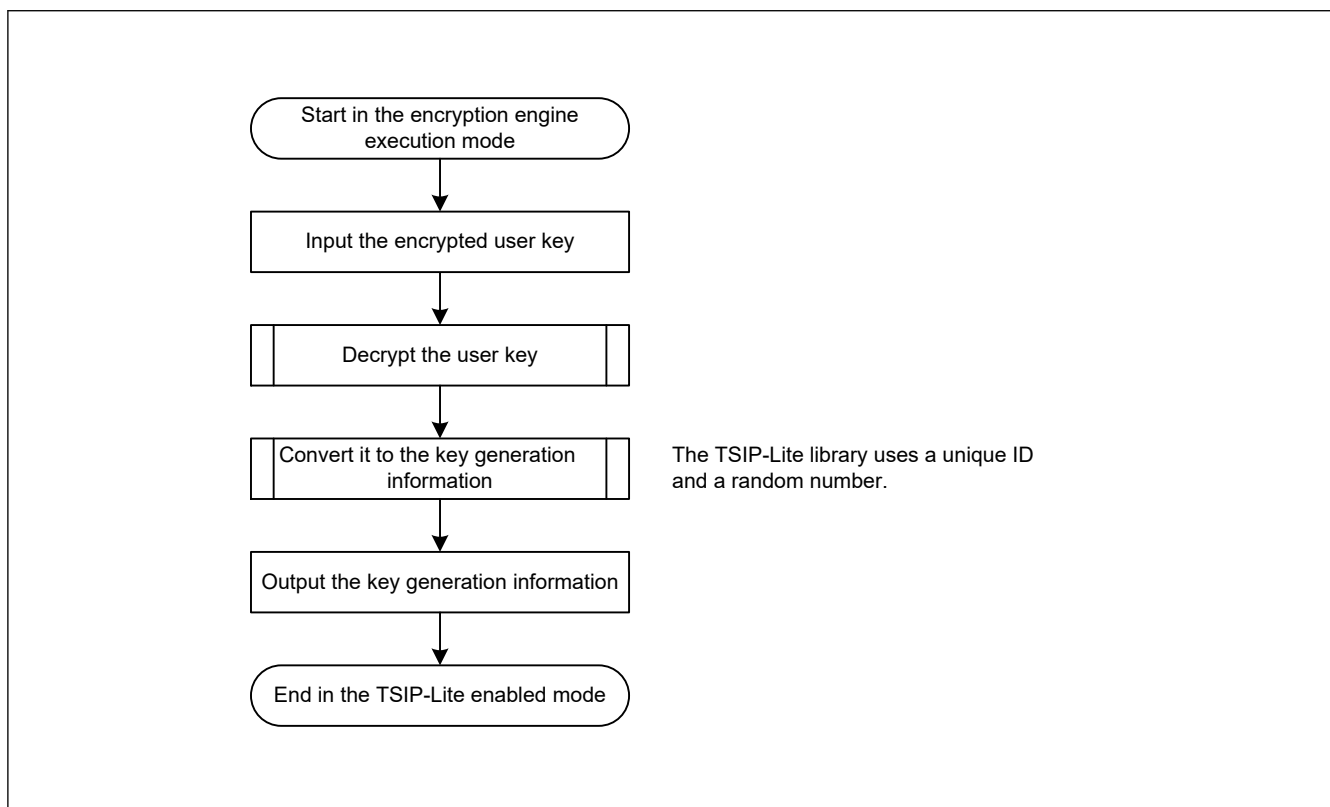


Figure 43.5 Key installation flowchart

43.2.4 Encryption and Decryption

Follow the procedure below to encrypt and decrypt the data:

1. Enter the key generation information to the TSIP-Lite and restore the key data.

2. Enter the target data to the TSIP-Lite. Plaintext data is converted to ciphertext and ciphertext data to plaintext.
3. Read the converted data.

The encryption engine has input and output buffers, and can perform encryption/decryption in parallel with data input/output.

Figure 43.6 shows Encryption and decryption timing, Figure 43.7 shows Encryption flowchart, and Figure 43.8 shows Decryption flowchart.

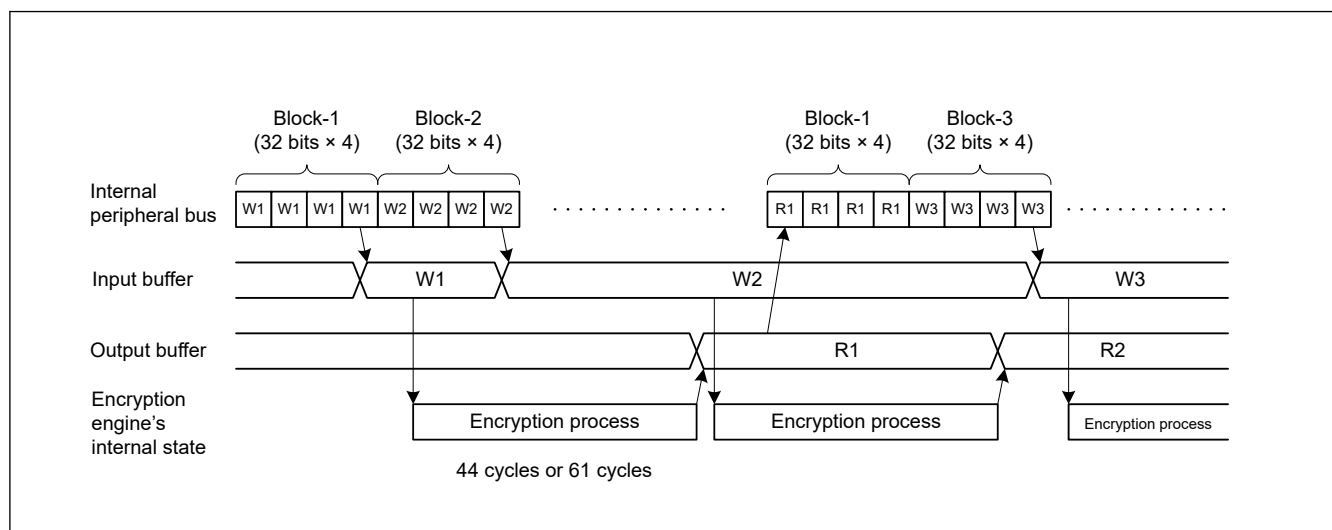
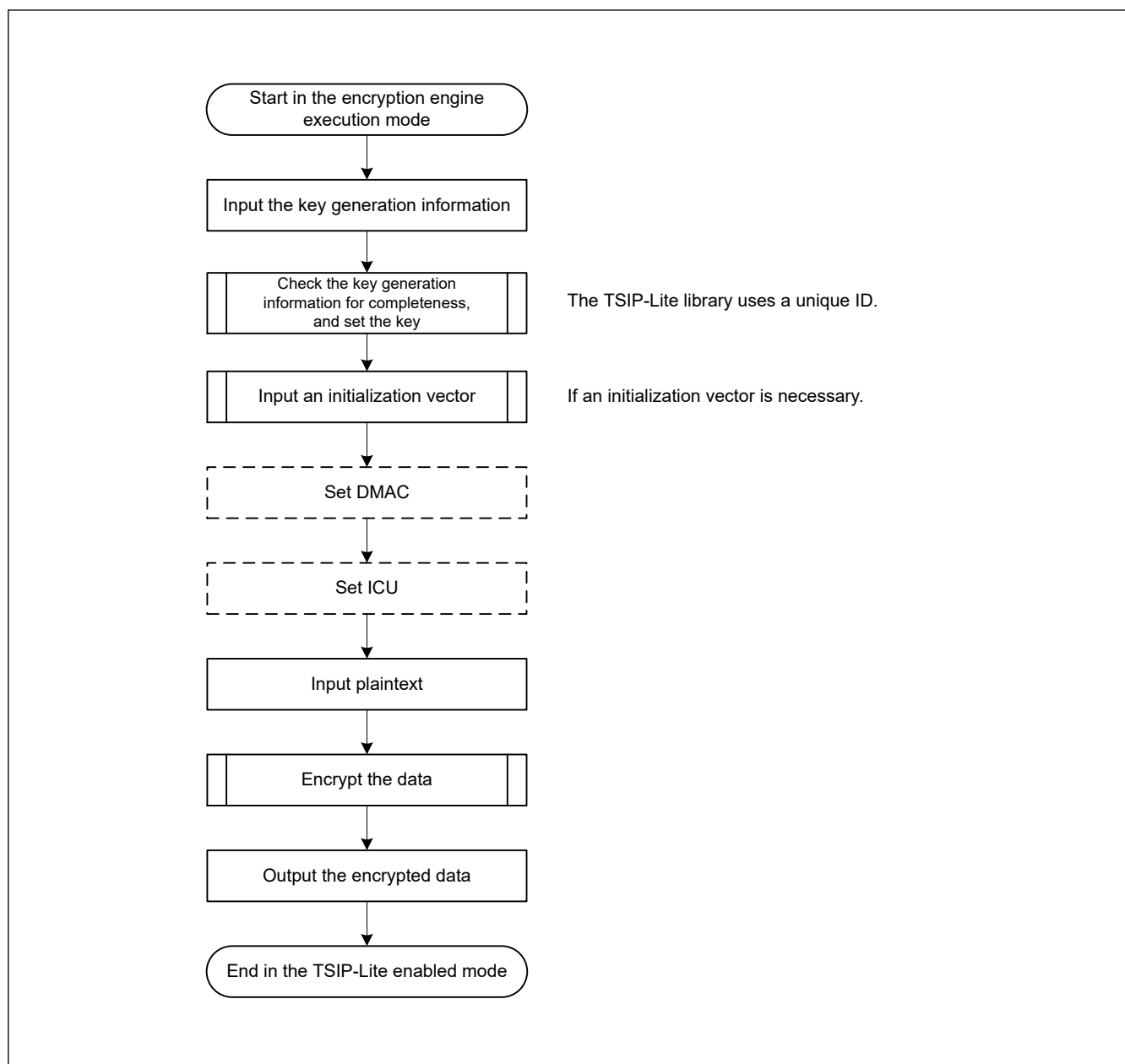


Figure 43.6 Encryption and decryption timing

**Figure 43.7 Encryption flowchart**

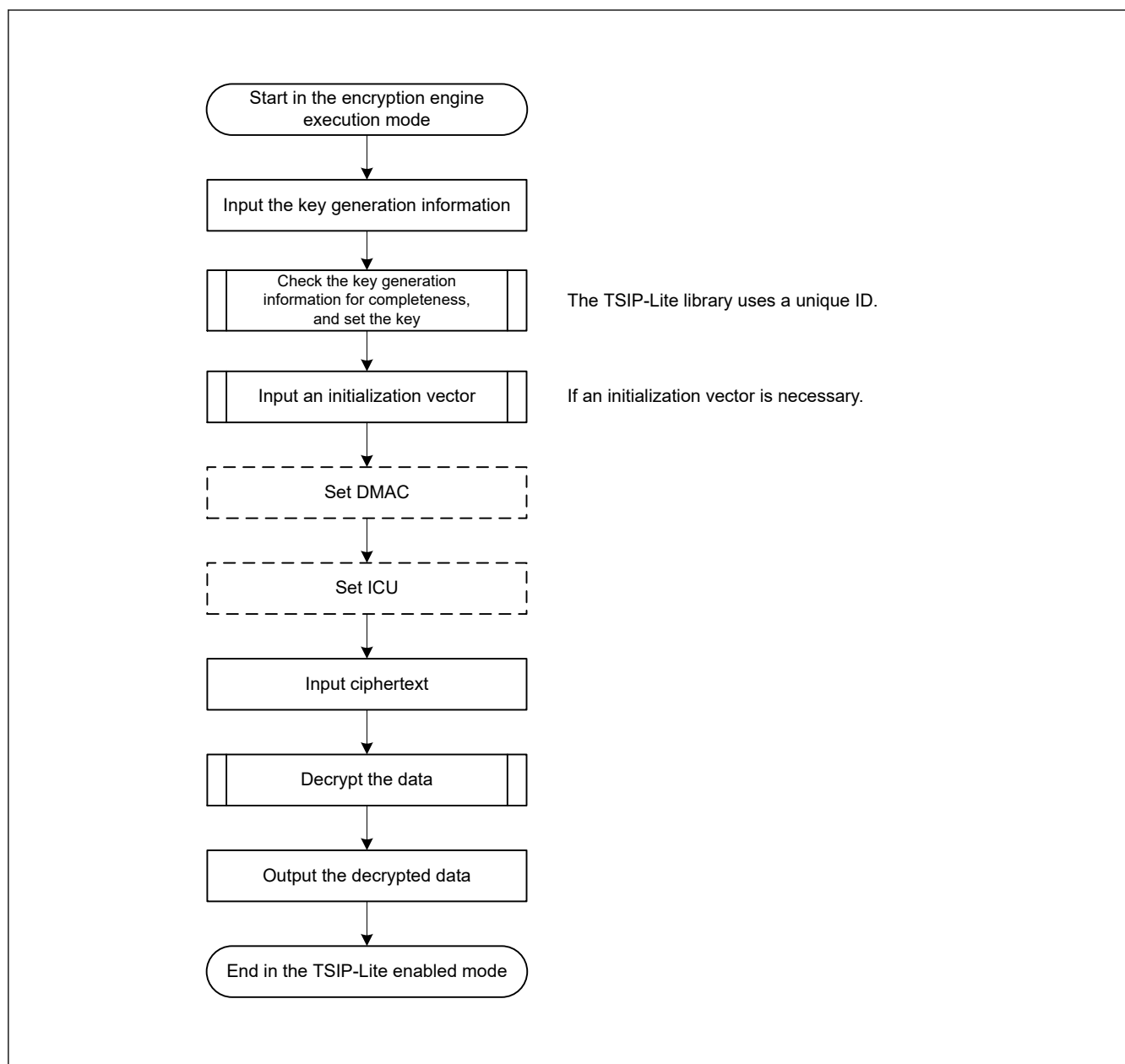


Figure 43.8 Decryption flowchart

43.2.5 Creation of Key Generation Information (Using a Random Number)

Figure 43.9 shows the flowchart of creating the key generation information using a random number.

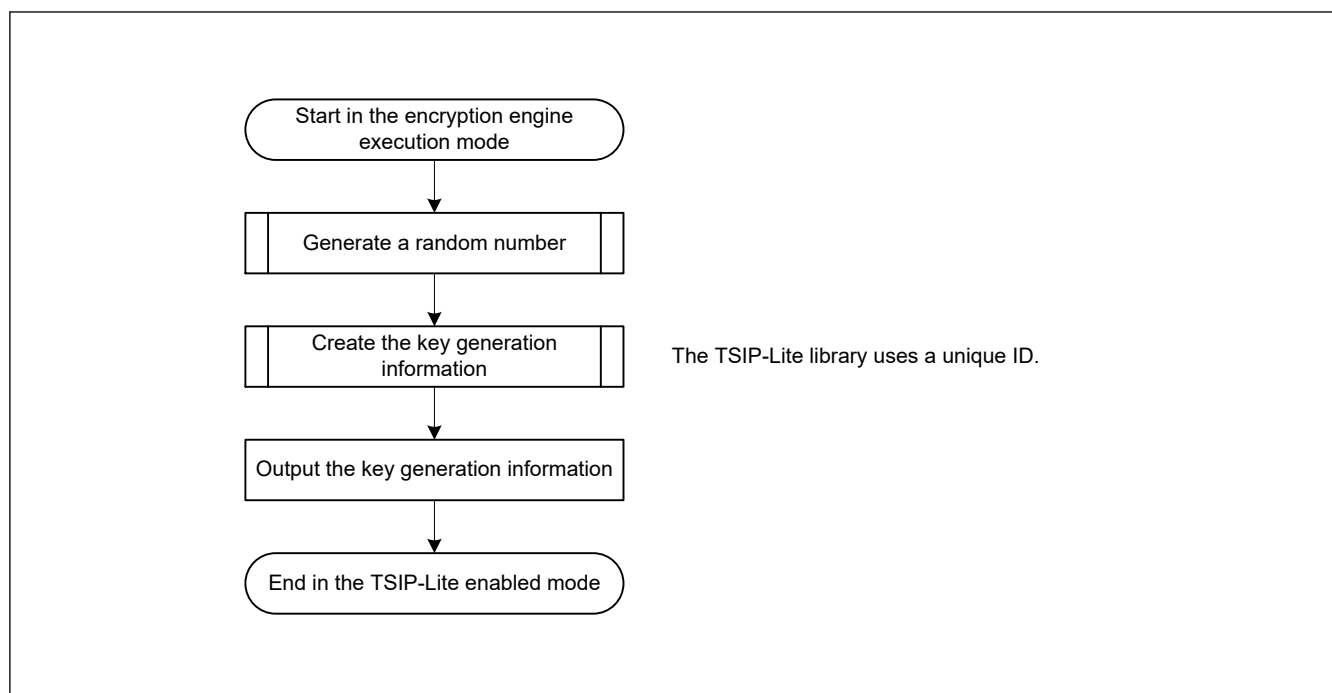


Figure 43.9 Creation flow of key generation information (using a random number)

43.2.6 Random Number Generation

Figure 43.10 shows the random number generation flow.

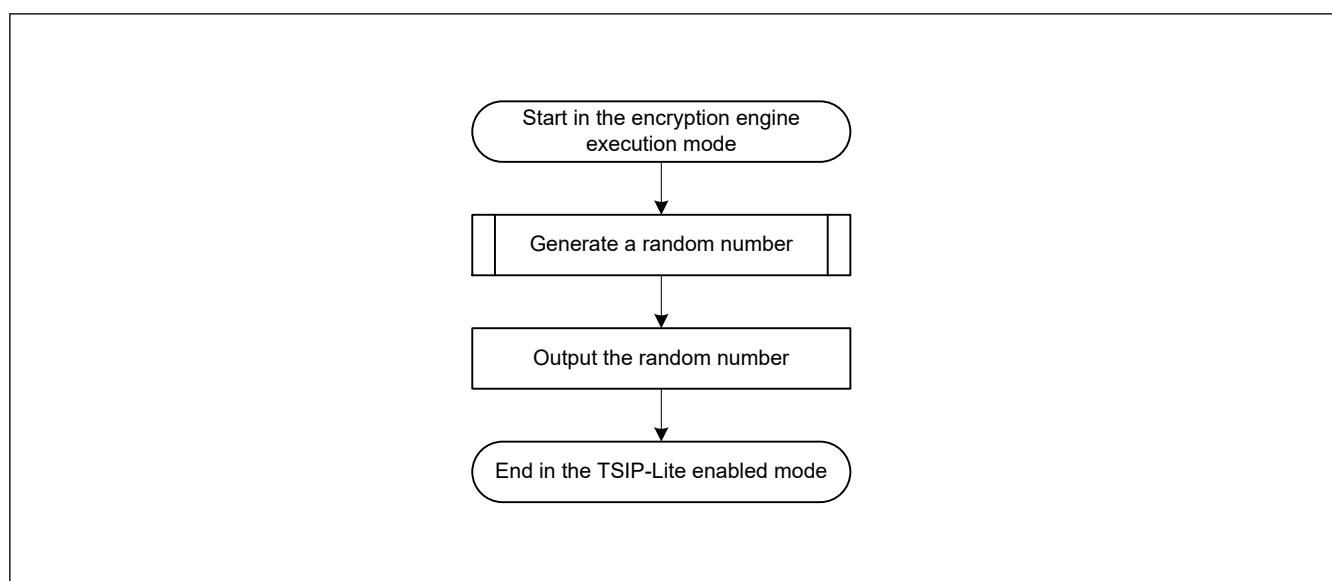


Figure 43.10 Random number generation flow

43.3 Interrupts

Table 43.2 lists the interrupt sources for TSIP-Lite.

In the TSIP-Lite library, the process is executed using the interrupts. Set the IELSRn.IELS[4:0] bits to specify the event signal corresponding to the interrupt source.

Table 43.2 Interrupt sources for TSIP-Lite (1 of 2)

Name	Interrupt source	Interrupt to the CPU	DMAC activation	DTC activation
PROC_BUSY	Procedure-completed interrupt.	Possible	Not possible	Not possible

Table 43.2 Interrupt sources for TSIP-Lite (2 of 2)

Name	Interrupt source	Interrupt to the CPU	DMAC activation	DTC activation
ROMOK	Tampering detection interrupt	Possible	Not possible	Not possible
LONG_PLG	Calculation-completed interrupt	Possible	Not possible	Not possible
TEST_BUSY	Test busy	Possible	Not possible	Not possible
WRRDY0	Write ready 0	Possible	Not possible	Not possible
WRRDY4	Write ready 4	Possible	Not possible	Not possible
RDRDY0	Read ready 0	Possible	Not possible	Not possible
NTEGRATE_WRRDY	Integrated write ready	Possible	Possible	Possible
INTEGRATE_RDRDY	Integrated read ready	Possible	Possible	Possible

43.4 Usage Notes

43.4.1 Software Standby Mode

When the software standby mode is entered while the encryption engine is in process, proper processing cannot be resumed after the software standby mode is exited. The software standby mode should therefore be entered while the encryption engine is not running.

43.4.2 Module-Stop Function Setting

The TSIP-Lite operation can be set enabled or disabled using Module Stop Control Register C (MSTPCRC). The TSIP-Lite is stopped for the value after reset. The TSIP-Lite can be accessed by releasing from the module-stop state. For details, [section 13, Power-Saving Functions](#).

43.4.3 TSIP-Lite Library

When using the TSIP-Lite, the TSIP-Lite library supplied by Renesas Electronics is necessary. For details about the TSIP-Lite library, contact a Renesas Electronics sales office.

44. 14-Bit A/D Converter (ADC14)

44.1 Overview

The MCU includes one 14-bit successive approximation A/D converter (ADC14) unit. Up to 12 analog input channels, temperature sensor output, can be selected for conversion.

The A/D conversion accuracy is selectable from 14-bit, 12-bit conversion, making it possible to optimize the trade-off between speed and resolution in generating a digital value.

The ADC14 supports the following operating modes:

- Single scan mode to convert analog inputs of selected channels in ascending order of setting in the ADSCSn register
- Continuous scan mode to convert analog inputs of selected channels continuously in ascending order of setting in the ADSCSn register
- Group scan mode to divide analog inputs of channels into two groups (group A and B) or three groups (group A, B, and C) and convert the analog inputs of selected channels for each group in ascending order of setting in the ADSCSn register.

In group scan mode, select two groups (group A and B) or three groups (group A, B, and C). You can individually select the scan start conditions for each group (group A, B and C) and start scanning of each group at different times. In addition, the scan start of a priority group is accepted during scanning of a lower-priority group, and then scanning of the lower-priority group is stopped. Then scanning of the priority group is started. The order of priority for group priority operation is group A > group B > group C. In group priority operation, if the scan start of group B is accepted during scanning of group C, the group C scanning is stopped and then scanning of group B is started. If the scan start of group A is accepted scanning of group C, the group C scanning is stopped and then scanning of group A is started. Similarly, if the scan start of group A is accepted during scanning of group B, the group B scanning is stopped and then scanning of group A is started. The stopped scanning of a group can be restarted after scanning of the priority group completes.

In double trigger mode, the analog input of a selected channel is converted in single scan mode or group scan mode (group A), and data converted by the first and second A/D conversion start triggers are stored in different registers, providing duplexing of A/D converted data.

Self-diagnosis is performed once at the beginning of each scan, and one of the three reference voltage values generated in ADC14 is A/D converted.

The ADC14 also provides a compare function (window A and window B). The compare function specifies the upper reference value for window A and lower reference value for window B, and outputs an interrupt when the A/D converted value of the selected channel meets the comparison conditions.

The reference power supply pin (VREFH0), the analog block power supply pin (AVCC0) can selected as the high-potential reference voltage. The reference power supply ground pin (VREFL0) or the analog block power supply ground pin (AVSS0) can be selected as the low-potential reference voltage.

[Table 44.1](#) lists the ADC14 specifications and [Table 44.2](#) list the functions. [Figure 44.1](#) shows a block diagram of ADC14 and [Table 44.3](#) lists the I/O pins.

Table 44.1 ADC14 specifications (1 of 3)

Parameter	Specifications
Number of units	one unit
Input channels	Up to 12 channels (AN000 to AN007, AN016, AN017, AN020 to AN021) Extended
Extended analog function	Temperature sensor output, VSC_VCC pins voltage output
A/D conversion method	Successive approximation method
Resolution	14-bit, 12-bit
Conversion time	1.0 μ s/channel (when 14-bit A/D conversion clock PCLKB (ADCLK) is operating at 32 MHz)
A/D conversion clock	Peripheral module clock PCLKB (1 MHz or higher, or 32.768 kHz)

Table 44.1 ADC14 specifications (2 of 3)

Parameter	Specifications
Data registers	12 registers for analog input - One register for A/D-converted data duplication in double trigger mode - One register for temperature sensor output - One register for VSC_VCC voltage - One register for self-diagnosis - A/D conversion results are stored in A/D data registers 14-bit, 12-bit accuracy for A/D conversion results - A/D-converted value addition mode, in which the sum of all A/D-converted results is stored in the A/D data registers as a value with the conversion accuracy bit count + extended bits^{*1} - 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.
Operating modes	- Single scan mode: - A/D conversion is performed only once on the analog inputs of arbitrarily selected channels, on the temperature sensor output. - Continuous scan mode: - A/D conversion is performed repeatedly on the analog inputs of the selected channels. - Group scan mode: - Analog inputs of selected channels are divided into groups A and B or groups A, B, and C. Then A/D conversion of the analog inputs selected on a group basis is performed once. - The number of groups to be used is selectable from two (groups A and B) and three (groups A, B, and C). (If two groups are to be used, only the combination of group A and group B is selectable.) - The scan start conditions can be independently selected for group A, B, and C, allowing A/D conversion of group A, B, and C to be started independently. - Group scan mode (when group priority operation is selected): - If a priority group trigger is input during scanning of a lower-priority group, the scanning of the lower-priority group is stopped and then scanning of the priority group is started. The order of priority is group A > group B > group C. - It is possible to select whether to restart scanning (rescan) of the lower-priority group upon completion of the priority group scan. It is also possible to specify rescanning to be started from the first channel of the selected channels or from the channel for which A/D conversion has not been completed.
Conditions for A/D conversion start	- Software trigger - Synchronous triggers from the Event Link Controller (ELC), 8-bit timer (TMR) - Asynchronous triggering by the external trigger pins, ADTRG0
Functions	- Variable sampling state count - Self-diagnosis of A/D converter - Selectable A/D-converted value addition mode or average mode - Analog input disconnection detection function (discharge and precharge functions) - Double-trigger mode (duplication of A/D conversion data) - Switching function of 12-bit and 14-bit conversion^{*2} - Automatic clear function for A/D data registers - Digital comparison of values in the comparison register and data register, and comparison between values in the data registers

Table 44.1 ADC14 specifications (3 of 3)

Parameter	Specifications
Interrupt sources	- In single scan mode (double trigger deselected), an A/D scan end interrupt request (ADC140_ADI) and ELC event signal (ADC140_ELC) can be generated on completion of single scan. - A compare interrupt request (ADC140_CMPAI/ADC140_CMPBI) can be generated in response to a match with a digital comparison condition. - A window compare ELC event signal (ADC140_WCMPPM) can be generated in response to a match with a digital comparison condition. - A window compare ELC event signal (ADC140_WCMPUM) can be generated in response to a mismatch with a digital comparison condition. - In single scan mode (double trigger selected), an A/D scan end interrupt request (ADC140_ADI) and ELC event signal (ADC140_ELC) is generated on completion of two scans. - In continuous scan mode, an A/D scan end interrupt request (ADC140_ADI) and ELC event signal (ADC140_ELC) is generated on completion of all the selected channel scans. - In group scan mode (double trigger deselected), an A/D scan end interrupt request (ADC140_ADI) and ELC event signal (ADC140_ELC) is generated on completion of group A scan, and an A/D scan end interrupt request for group B (ADC140_GBADI) can be generated on completion of group B scan, and an A/D scan end interrupt request for group C (ADC140_GCADI) can be generated on completion of group C scan. - In group scan mode (double trigger selected), an A/D scan end interrupt request (ADC140_ADI) and ELC event signal (ADC140_ELC) is generated on completion of two group A scans, and an A/D scan end interrupt request for group B and C (ADC140_GBADI and ADC140_GCADI) can be generated on completion of group B and C scan. - ADC140_ADI, ADC140_GBADI, ADC140_WCMPPM, and ADC140_WCMPUM can activate the Data Transfer Controller (DTC).
ELC interface	- An event is generated upon completion of group A scan in group-scan mode. - An event is generated upon completion of group B scan in group-scan mode. - An event is generated upon completion of group C scan in group-scan mode. - An event is generated when all scans complete. - Scan can be started by a trigger from the ELC. - An event is generated according to conditions of the compare function window in single-scan mode.
Reference voltage	AVCC0 or VREFH0 (external reference voltage or output voltage from reference voltage generation circuit) can be selected as the high-potential reference voltage. VREFL0 or AVSS0 can be selected as the low-potential reference voltage.
Module-stop function	Module-stop state can be set to reduce power consumption.*3

Note 1. The number of extended bits for addition varies with the A/D conversion accuracy and the number of addition times. In details, see [\(2\)When A/D-converted value average mode is selected.](#)

Note 2. When the A/D conversion accuracy is modified, the A/D conversion time is also changed. For details, see [section 44.3.6. Analog Input Sampling and Scan Conversion Time.](#)

Note 3. For details, see [section 13, Power-Saving Functions.](#)

Table 44.2 ADC14 functions (1 of 2)

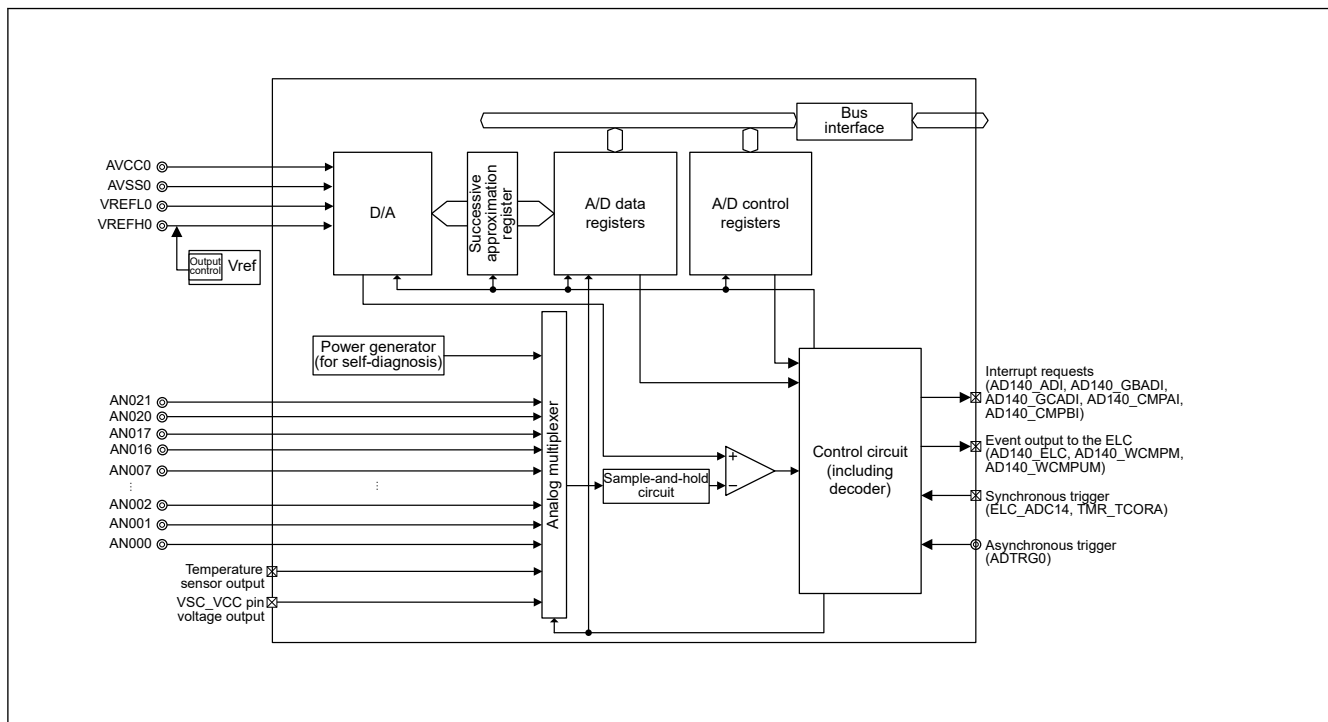
Parameter	function		
Analog input channel	AN000 to AN007, AN016, AN017, AN020 to AN021 Temperature sensor output VSC_VCC pin voltage		
Conditions for A/D conversion start	Software	Software trigger	Enabled
	Asynchronous trigger (external trigger)	Trigger input pin	ADTRG0
	Synchronous trigger (trigger from ELC)	ELC trigger, TMR_TCORA compare match	ELC_ADC14, TMR_TCORA
Interrupt	ADC140_ADI ADC140_GBADI ADC140_GCADI ADC140_CMPAI ADC140_CMPBI		
Output to ELC	ADC140_ELC ADC140_WCMPPM ADC140_WCMPUM		

Table 44.2 ADC14 functions (2 of 2)

Parameter	function
Module-stop function settings ^{*1 *2}	MSTPCRD.MSTPD16 bit

Note 1. For details, see [section 13, Power-Saving Functions](#).

Note 2. Wait 1 μ s or longer to start A/D conversion after release from the module-stop state.

**Figure 44.1 ADC14 block diagram**

[Table 44.3](#) lists the ADC14 I/O pins.

Table 44.3 ADC14 I/O pins

Pin name	I/O	Function
AVCC0	Input	When ADHVREFCNT.HVSEL [1: 0] = 00b: High-potential reference voltage pin and power supply pin of analog block When ADHVREFCNT.HVSEL [1: 0] = 01b: Analog block power supply pin
AVSS0	Input	When ADHVREFCNT.LVSEL [1: 0] = 00b: Low-potential reference voltage ground pin and power supply ground pin of analog block When ADHVREFCNT.LVSEL [1: 0] = 01b: Analog block power supply ground pin
VREFH0	Input	When ADHVREFCNT.HVSEL [1: 0] = 00b: Unused When ADHVREFCNT.HVSEL [1: 0] = 01b: Reference high-potential power supply pin
VREFL0	Input	When ADHVREFCNT.LVSEL [1: 0] = 00b: Unused When ADHVREFCNT.LVSEL [1: 0] = 01b: Reference low-potential power supply ground pin
AN000 to AN007, AN016, AN017, AN020 to AN021	Input	Analog input pins 00 to 07, 16, 17, 20, 21
ADTRG0	Input	External trigger input pin for starting A/D conversion

44.2 Register Descriptions

44.2.1 ADDRn : A/D Data Registers n

Base address: ADC140 = 0x4005_C000

Offset address: $0x020 + 0x2 \times n$ ($n = 00$ to $07, 16, 17, 20, 21$)

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

Bit	Symbol	Function	R/W
15:0	ADDRn [15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 44.4 and Table 44.5 .	R

ADDRn registers are 16-bit read-only registers to store A/D conversion results.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (14-bit, 12-bit is selectable.)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

[Table 44.4](#) shows the example of bit assignment for 12-bit accuracy.

Table 44.4 Example of bit assignment for 12-bit accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, These registers indicate the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 14-bit, 12-bit accuracy, 1, 2, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, these registers indicate the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

[Table 44.5](#) shows example of the bit assignment for 12-bit accuracy.

Table 44.5 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, or 4 conversion times is specified	These bits are read as 0.		Added Value 13 to 0: 14-bit sum of A/D conversion results													
Left-justified data with 12-bit accuracy	When 1, 2, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

44.2.2 ADDBLDR : A/D Data Duplexing Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x018

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

Bit	Symbol	Function	R/W
15:0	ADDBLDR [15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 44.6 and Table 44.7 .	R

ADDBLDR register is a 16-bit read-only register to store A/D conversion results in response to a second trigger in double-trigger mode.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (14-bit, 12-bit is selectable.)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

[Table 44.6](#) shows the example of bit assignment for 12-bit accuracy.

Table 44.6 Example of bit assignment for 12-bit accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, This register indicates the mean of A/D-converted values on a

specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 14-bit, 12-bit accuracy, 1, 2, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, this register indicates the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 44.7 shows example of the bit assignment for 12-bit accuracy.

Table 44.7 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, or 4 conversion times is specified	These bits are read as 0.		Added Value 13 to 0: 14-bit sum of A/D conversion results													
Left-justified data with 12-bit accuracy	When 1, 2, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

44.2.3 ADTSDR : A/D Temperature Sensor Data Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x01A

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field: ADTSDR [15:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	ADTSDR [15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 44.8 and Table 44.9 .	R

ADTSDR register is a 16-bit read-only register to store A/D conversion result of the temperature sensor output.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (14-bit, 12-bit is selectable.)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

Table 44.8 shows the example of bit assignment for 12-bit accuracy.

Table 44.8 Example of bit assignment for 12-bit accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, This register indicates the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 14-bit, 12-bit accuracy, 1, 2, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, this register indicates the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 44.9 shows example of the bit assignment for 12-bit accuracy.

Table 44.9 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, or 4 conversion times is specified	These bits are read as 0.		Added Value 13 to 0: 14-bit sum of A/D conversion results													
Left-justified data with 12-bit accuracy	When 1, 2, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

44.2.4 ADVSCDR : A/D VSC_VCC Voltage Voltage Data Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x05E

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field: ADVSCDR[15:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	ADVSCDR[15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 44.10 and Table 44.11 .	R

ADVSCDR register is a 16-bit read-only register to store A/D conversion results of the VSC_VCC voltage. Available only when EHCCR0.CHGDETEN = 1.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (14-bit, 12-bit is selectable.)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

[Table 44.10](#) shows the example of bit assignment for 12-bit accuracy.

Table 44.10 Example of bit assignment for 12-bit accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 14-bit, 12-bit accuracy, 1, 2, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

[Table 44.11](#) shows example of the bit assignment for 12-bit accuracy.

Table 44.11 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected (1 of 2)

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, or 4 conversion times is specified	These bits are read as 0.				Added Value 13 to 0: 14-bit sum of A/D conversion results											

Table 44.11 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected (2 of 2)

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Left-justified data with 12-bit accuracy	When 1, 2, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

44.2.5 ADDR : A/D Self-Diagnosis Data Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x01E

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

Bit	Symbol	Function	R/W
13:0	AD[13:0]	Converted Value 13 to 0 14-bit A/D-converted value	R
15:14	DIAGST[1:0]	Self-Diagnosis Status 0 0: Self-diagnosis not executed after power-on. 0 1: Self-diagnosis was executed using the 0 V voltage. 1 0: Self-diagnosis was executed using the reference power supply*1 voltage × 1/2. 1 1: Self-diagnosis was executed using the reference power supply*1 voltage. For details on self-diagnosis, see section 44.2.17. ADCER : A/D Control Extended Register .	R

Note: The example of the bit assignment for the right-justified data with 14-bit accuracy is indicated.

Note 1. The reference voltage refers to VREFH0.

ADDR is a 16-bit read-only register that holds the A/D conversion results based on the self-diagnosis of the ADC14. In addition to the AD[13:0] bits indicating the A/D-converted value, it includes the Self-Diagnosis Status bit (DIAGST[1:0]).

The settings of the A/D data register format and the A/D conversion accuracy determines the formats for data in this register.

The A/D-converted value addition and average modes cannot be applied to the A/D self-diagnosis function. For details on self-diagnosis, see [section 44.2.17. ADCER : A/D Control Extended Register](#).

This section describes the data formats for each condition. The register diagram and the register bit table shown in this section indicate example of the bit assignment for the left and right-justified data with 12-bit accuracy.

Table 44.12 Bit assignment for each right-justified accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	DIAGST[1:0]		—		AD[11:0]											

Table 44.13 Bit assignment for each left-justified accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Left-justified data with 12-bit accuracy	AD[11:0]												—		DIAGST[1:0]	

44.2.6 ADCSR : A/D Control Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADST	ADCS[1:0]	ADIE	—	—	TRGE	EXTRG	DBLE	GBADIE	—	DBLANS[4:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	DBLANS[4:0]	Double Trigger Channel Select These bits select one analog input channel for double trigger operation. The setting is only valid in double-trigger mode.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	GBADIE	Group B Scan End Interrupt and ELC Event Enable Group B scan only works in group scan mode. 0: Disable ADC140_GBADI interrupt generation on group B scan completion. 1: Enable ADC140_GBADI interrupt generation on group B scan completion.	R/W
7	DBLE	Double Trigger Mode Select 0: Deselect double-trigger mode. 1: Select double-trigger mode.	R/W
8	EXTRG	Trigger Select ^{*1} 0: Start A/D conversion by the synchronous trigger (TMR, ELC). 1: Start A/D conversion by the asynchronous trigger (ADTRG0).	R/W
9	TRGE	Trigger Start Enable 0: Disable A/D conversion to be started by the synchronous or asynchronous trigger 1: Enable A/D conversion to be started by the synchronous or asynchronous trigger	R/W
10	—	These bits are read as 0. The write value should be 0.	R/W
11	—	These bits are read as 0. The write value should be 0.	R/W
12	ADIE	Scan End Interrupt Enable 0: Disable ADC140_ADI ADC140_ADI interrupt generation upon completion of scanning 1: Enable ADC140_ADI interrupt generation upon completion of scanning.	R/W
14:13	ADCS[1:0]	Scan Mode Select 0 0: Single scan mode 0 1: Group scan mode 1 0: Continuous scan mode 1 1: Setting prohibited	R/W
15	ADST	A/D Conversion Start 0: Stop A/D conversion process. 1: Start A/D conversion process.	R/W

Note 1. To start A/D conversion using an external pin (asynchronous trigger):

After a high-level signal is input to the external pin (ADTRG0), write 1 to both the TRGE and EXTRG bits in the ADCSR register and drive the ADTRG0 pin low. With these settings, the scan conversion process starts on detection of the falling edge of ADTRG0. The pulse width of the low-level input must be at least PCLKB 1.5 clock cycles.

The ADCSR register sets double-trigger mode and A/D conversion start trigger, enables or disables scan end interrupt, selects the scan mode, and starts or stops A/D conversion.

DBLANS[4:0] bits (Double Trigger Channel Select)

The DBLANS[4:0] bits select one channel for A/D conversion data duplication in double trigger mode. This can be selected by setting the binary value of the channel number to be duplicated. The A/D conversion results of the analog input of the channel selected in the DBLANS[4:0] bits are stored in A/D Data Register y when conversion is started by the first trigger, and in the A/D Data Duplexing Register when conversion is started by the second trigger.

In double trigger mode, the channels selected in the ADANSA0 and ADANSA1 registers are invalid, and the channel selected in the DBLANS[4:0] bits is A/D converted instead.

When double trigger mode is used in group scan mode, double trigger control is only applied to group A and not to group B, C. Therefore, multiple channel analog input can be selected for group B, C even in double trigger mode.

Only set the DBLANS[4:0] bits when the ADST bit is 0. Do not set the DBLANS[4:0] bits at the same time that you write 1 to the ADST bit.

To enter A/D-converted value addition/average mode when in double trigger mode, select the channel using the DBLANS[4:0] bits in the ADANSA0 and ADANSA1 registers.

A/D-converted data from the self-diagnosis function, temperature sensor output, VSC_VCC cannot be used in double trigger mode

Table 44.14 Relationship between DBLANS bit Settings and Double-trigger Enabled Channels

DBLANS[4:0]	Duplication Channel	DBLANS[4:0]	Duplication Channel
00000	AN000	10000	AN016
00001	AN001	10001	AN017
00010	AN002	10100	AN020
00011	AN003	10101	AN021
00100	AN004	—	—
00101	AN005	—	—
00110	AN006	—	—
00111	AN007	—	—

GBADIE bit (Group B Scan End Interrupt and ELC Event Enable)

The GBADIE bit enables or disables group B scan end interrupt (ADC140_GBADI) in group scan mode.

DBLE bit (Double Trigger Mode Select)

The DBLE bit selects or deselects double-trigger mode. Double-trigger mode can only be operated by the synchronous trigger (ELC) selected in the ADSTRGR.TRSA[5:0] bits.

Double-trigger operation is as follows:

1. The ADC140_ADI interrupt is not output on completion of the first conversion but on completion of the second conversion.
2. The A/D conversion results from the duplication channel (selected in DBLANS[4:0]) started by the first trigger are stored in A/D Data Register y and those started by the second trigger are stored in the A/D Data Duplexing Register.

When the DBLE bit is set (double trigger mode is selected), the channels specified in the ADANSA0 and ADANSA1 registers are invalid. Double trigger mode is deselected by setting DBLE to 0. Setting DBLE to 1 again enables the same double-trigger operation described in 1. and 2. for first time scanning with the first trigger.

Do not select double trigger mode in continuous scan mode. Software triggering cannot be used in double trigger mode. Always set the ADST bit to 0 before setting the DBLE bit. Do not set the DBLE bit at that same time as writing 1 to the ADST bit.

EXTRG bit (Trigger Select)

The EXTRG bit selects the synchronous or asynchronous trigger as the trigger for starting A/D conversion.

In group-scan mode, the setting of this bit takes effect on the trigger selected for group A. For groups B and C, A/D conversion is started by the selected synchronous trigger regardless of this bit setting.

TRGE bit (Trigger Start Enable)

The TRGE bit enables or disables A/D conversion by the synchronous and asynchronous triggers. In group scan mode, set this bit to 1.

ADIE bits (Scan End Interrupt Enable)

The ADIE bit enables or disables an A/D scan end interrupt in scans except for group B or group C scan in group-scan mode.

With double-trigger mode deselected, if the ADIE bit is set to 1, an A/D scan conversion end interrupt is generated when the first scan completes.

With double-trigger mode selected, if the ADIE bit is set to 1, an A/D scan conversion end interrupt is generated when the second scan completes. This only applies if the scan is started by the synchronous trigger selected in the ADSTRGR.TRSA[5:0] bits.

ADCS[1:0] bits (Scan Mode Select)

The ADCS[1:0] bits select the scan mode.

In single scan mode, A/D conversion is performed for the analog inputs of the channels selected in the ADANSA0 and ADANSA1 registers, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, the scan conversion stops.

In continuous scan mode, when the ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs of the channels selected with the ADANSA0 and ADANSA1 registers, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion repeats from the first channel. If the ADCSR.ADST bit is set to 0 during continuous scan, A/D conversion stops even if scanning is in progress.

In group scan mode:

- Group A scanning is started by the synchronous trigger (TMR, ELC) selected in the TRSA[5:0] bits in the ADSTRGR register. A/D conversion is performed on group A analog inputs of the channels selected in the ADANSA0 and ADANSA1 registers, in user-specified 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 (TMR, ELC) selected in the ADSTRGR.TRSB[5:0] bits. A/D conversion is performed on group B analog inputs of the channels selected in the ADANSB0 and ADANSB1 registers, in user-specified order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion stops.
- Group C scanning is started by the synchronous trigger (TMR, ELC) selected in the ADGCTRGR.TRSC[5:0] bits. A/D conversion is performed on group C analog inputs of the channels selected in the ADANSC0 and ADANSC1 registers, in user-specified 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, C occur at the same time, those conversions cannot be controlled separately. In this case, set group A Priority Control Setting bit (ADGSPCR.PGS) in the A/D Group Scan Priority Control Register (ADGSPCR) to 1 to assign a priority to group A conversion.

In group scan mode, select different channels and triggers for group A and group B and group C.

The temperature sensor output can be selected in single-scan and group-scan mode.

Only set the ADCS[1:0] bits when the ADST bit is 0. Do not set the ADCS[1:0] bits at the same time that you write 1 to the ADST bit.

Table 44.15 Selectable targets for A/D conversion depending on scan and double-trigger mode settings

Scan mode setting	Double-trigger mode setting	Targets for A/D conversion			
		Self-diagnosis	Analog input (group A)	Analog input (group B/C)	Temperature sensor output
Single scan	DBLE = 0	✓	✓	—	✓
	DBLE = 1	—	✓ (1 ch only)	—	—
Continuous scan	DBLE = 0	✓	✓	—	✓
	DBLE = 1	—	—	—	—
Group scan	DBLE = 0	✓	✓	✓	✓
	DBLE = 1	—	✓ (1 ch only)	✓	✓ ^{*1}

Note: ✓: Selectable, —: Not selectable

Note 1. Selectable for group B or C

ADST bit (A/D Conversion Start)

The ADST bit starts or stops the A/D conversion process. Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and the conversion target analog input.

[Setting conditions]

- 1 is written.
- The synchronous trigger (ELC) selected in the ADSTRGR.TRSA[5:0] bits is detected when ADCSR.EXTRG is 0 and ADCSR.TRGE is 1.
- The synchronous trigger (ELC) selected in the ADSTRGR.TRSB[5:0] bits is detected when ADCSR.TRGE is set to 1 in group scan mode.
- The asynchronous trigger is detected when the ADCSR.TRGE and ADCSR.EXTRG bits are set to 1 and the ADSTRGR.TRSA[5:0] bits are set to 0x00.
- When group priority operation mode is enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRP bit is set to 1, and each time A/D conversion on the group with the lowest priority is started.

[Clearing conditions]

- 0 is written.
- The A/D conversion of all the selected channels, the temperature sensor output completes in single scan mode.
- Group A scan completes in group scan mode.
- Group B scan completes in group scan mode.
- Group C scan completes in group scan mode.
- When group priority operation mode is enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1, and A/D conversion on the group with the lowest priority started by trigger completes.
- Offset Calibration ends.

Note: When group priority operation mode is enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), do not set the ADST bit to 1.

Note: When group priority operation mode is enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), do not set the ADST bit to 0. When forcing A/D conversion to terminate, follow the procedure for clearing the ADST bit.

44.2.7 ADANSA0 : A/D Channel Select Register A0

Base address: ADC140 = 0x4005_C000

Offset address: 0x004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ANSA15	ANSA14	ANSA13	ANSA12	ANSA11	ANSA10	ANSA09	ANSA08	ANSA07	ANSA06	ANSA05	ANSA04	ANSA03	ANSA02	ANSA01	ANSA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ANSAn	A/D Conversion Channels Select Bit 15 (ANSA15) is associated with AN015 and bit 0 (ANSA00) is associated with AN000. 0: Do not select associated input channel. 1: Select associated input channel.	R/W

Note: n = 00 to 07

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

ADANSA0 register selects analog input channels for A/D conversion. In group scan mode, this register selects group A channels.

Only set the ADANSA0 register when the ADCSR.ADST bit is 0.

ANSAn bits (A/D Conversion Channels Select)

The ADANSA0 register selects any combination of analog input channels for A/D conversion. The channels and the number of channels can be arbitrarily set.

When the temperature sensor output and analog input channels are selected at the same time, the voltages on the selected analog input channels are A/D-converted before the temperature sensor output. All analog input channels can also be deselected so that only the temperature sensor output is converted.

In double trigger mode, the channels selected in the ADANSA0 register are invalid, and the channel selected in the ADCSR.DBLANS[4:0] bits is selected in group A instead.

When group scan mode is selected, do not select the channels specified in A/D Channel Select Register B0 (ADANSB0) and A/D Channel Select Register B1 (ADANSB1).

44.2.8 ADANSA1 : A/D Channel Select Register A1

Base address: ADC140 = 0x4005_C000

Offset address: 0x006

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ANSA 31	ANSA 30	ANSA 29	ANSA 28	ANSA 27	ANSA 26	ANSA 25	ANSA 24	ANSA 23	ANSA 22	ANSA 21	ANSA 20	ANSA 19	ANSA 18	ANSA 17	ANSA 16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ANSAn	A/D Conversion Channels Select Bit 15 (ANSA31) is associated with VSC_VCC. Bit 14 (ANSA30) is associated with AN030 and bit 0 (ANSA16) is associated with AN016. 0: Do not select associated input channel. 1: Select associated input channel.	R/W

Note: n = 16, 17, 20, 21, 31

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

ADANSA1 register selects analog input channels for A/D conversion. In group scan mode, this register selects group A channels.

Only set the ADANSA1 register when the ADCSR.ADST bit is 0.

ANSAn bits (A/D Conversion Channels Select)

The ADANSA1 register selects any combination of analog input channels for A/D conversion. The channels and the number of channels can be arbitrarily set.

If the EHC is used, the level of power generation by the power generation element can be detected with high accuracy by selecting the ANSA31 bit. For details, see [section 14.3.7. Level of Power Generation Detection](#)

When the temperature sensor output and analog input channels are selected at the same time, the voltages on the selected analog input channels are A/D-converted before the temperature sensor output. All analog input channels can also be deselected so that only the temperature sensor output is converted.

In double trigger mode, the channels selected in the ADANSA1 register are invalid, and the channel selected in the ADCSR.DBLANS[4:0] bits is selected in group A instead.

When group scan mode is selected, do not select the channels specified in A/D Channel Select Register B0 (ADANSB0) and A/D Channel Select Register B1 (ADANSB1).

44.2.9 ADANSB0 : A/D Channel Select Register B0

Base address: ADC140 = 0x4005_C000

Offset address: 0x014

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ANSB 15	ANSB 14	ANSB 13	ANSB 12	ANSB 11	ANSB 10	ANSB 09	ANSB 08	ANSB 07	ANSB 06	ANSB 05	ANSB 04	ANSB 03	ANSB 02	ANSB 01	ANSB 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ANSBn	A/D Conversion Channels Select Bit 15 (ANSB15) is associated with AN015 and bit 0 (ANSB00) is associated with AN000. 0: Do not select associated input channel. 1: Select associated input channel.	R/W

Note: n = 00 to 07

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

ADANSB0 selects analog input channels for A/D conversion in group B when group scan mode is selected. The ADANSB0 register is not used in any scan mode other than group scan mode.

Only set the ADANSB0 register when the ADCSR.ADST bit is 0.

ANSBn bits (A/D Conversion Channels Select)

The ADANSB0 register selects any combination of analog input channels in group B for A/D conversion when group scan mode is selected. The ADANSB0 register is used for group scan mode only and not for any other modes.

Do not select channels specified in group A as selected in the ADANSA0 and ADANSA1 registers or the ADCSR.DBLANS[4:0] bits in double trigger mode.

When the temperature sensor output and analog input channels are selected at the same time, the voltages on the selected analog input channels are A/D-converted before the temperature sensor output.

44.2.10 ADANSB1 : A/D Channel Select Register B1

Base address: ADC140 = 0x4005_C000

Offset address: 0x016

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ANSB 31	ANSB 30	ANSB 29	ANSB 28	ANSB 27	ANSB 26	ANSB 25	ANSB 24	ANSB 23	ANSB 22	ANSB 21	ANSB 20	ANSB 19	ANSB 18	ANSB 17	ANSB 16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ANSBn	A/D Conversion Channels Select Bit 15 (ANSB31) is associated with VSC_VCC. Bit 14 (ANSB30) is associated with AN030 and bit 0 (ANSB16) is associated with AN016. 0: Do not select associated input channel. 1: Select associated input channel.	R/W

Note: n = 16, 17, 20, 21, 31

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

ADANSB1 selects analog input channels for A/D conversion in group B when group scan mode is selected. The ADANSB1 register is not used in any scan mode other than group scan mode.

Only set the ADANSB1 register when the ADCSR.ADST bit is 0.

ANSBn bits (A/D Conversion Channels Select)

The ADANSB1 register selects any combination of analog input channels in group B for A/D conversion when group scan mode is selected. The ADANSB1 register is used for group scan mode only and not for any other modes.

Do not select channels specified in group A as selected in the ADANSA0 and ADANSA1 registers or the ADCSR.DBLANS[4:0] bits in double trigger mode.

When the temperature sensor output and analog input channels are selected at the same time, the voltages on the selected analog input channels are A/D-converted before the temperature sensor output.

If the EHC is used, the level of power generation by the power generation element can be detected with high accuracy by selecting the ANSB31 bit. For details, see [section 14.3.7. Level of Power Generation Detection](#)

44.2.11 ADANSC0 : A/D Channel Select Register C0

Base address: ADC140 = 0x4005_C000

Offset address: 0x0D4

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ANSC 15	ANSC 14	ANSC 13	ANSC 12	ANSC 11	ANSC 10	ANSC 09	ANSC 08	ANSC 07	ANSC 06	ANSC 05	ANSC 04	ANSC 03	ANSC 02	ANSC 01	ANSC 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ANSCn	A/D Conversion Channels Select Bit 15 (ANSC15) is associated with AN015 and bit 0 (ANSC00) is associated with AN000. 0: The associated input channel is not selected 1: The associated input channel is selected.	R/W

Note: n = 00 to 07

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

The ADANSC0 register selects the high-precision analog input channels to be used in A/D conversion in group C when group-scan mode is selected. The ADANSC0 register is only used for group-scan mode, not for any other modes.

Set the ADANSC0 register while the ADCSR.ADST bit is 0.

ANSCn bits (A/D Conversion Channels Select)

The ADANSC0 register selects any combination of the analog input channels to be used in A/D conversion in group C when group-scan mode is selected.

The ADANSC0 register is only used for group-scan mode, not for any other modes.

Do not select channels specified in group A (as selected in the ADANSA0 and ADANSA1 registers or the ADCSR.DBLANS[4:0] bits in double-trigger mode).

When the temperature sensor output and analog input channels are selected at the same time, the voltages on the selected analog input channels are A/D-converted before the temperature sensor output.

44.2.12 ADANSC1 : A/D Channel Select Register C1

Base address: ADC140 = 0x4005_C000

Offset address: 0x0D6

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ANSC 31	ANSC 30	ANSC 29	ANSC 28	ANSC 27	ANSC 26	ANSC 25	ANSC 24	ANSC 23	ANSC 22	ANSC 21	ANSC 20	ANSC 19	ANSC 18	ANSC 17	ANSC 16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ANSCn	A/D Conversion Channels Select Bit 15 (ANSC31) is associated with VSC_VCC. Bit 14 (ANSC30) is associated with AN030 and bit 0 (ANSC16) is associated with AN016. 0: The associated input channel is not selected 1: The associated input channel is selected	R/W

Note: n = 16, 17, 20, 21, 31

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

The ADANSC1 register selects analog input channels to be used in A/D conversion in group C when group-scan mode is selected. The ADANSC1 register is only used for group scan mode, not for any other modes.

Set the ADANSC1 register while the ADCSR.ADST bit is 0.

ANSCn bits (A/D Conversion Channels Select)

The ADANSC1 register selects any combination of the analog input channels to be used in A/D conversion in group C when group-scan mode is selected.

The ADANSC1 register is only used for group-scan mode, not for any other modes.

Do not select channels specified in group A (as selected in the ADANSA0 and ADANSA1 registers or the ADCSR.DBLANS[4:0] bits in double-trigger mode).

When the temperature sensor output and analog input channels are selected at the same time, the voltages on the selected analog input channels are A/D-converted before the temperature sensor output.

If the EHC is used, the level of power generation by the power generation element can be detected with high accuracy by selecting the ANSC31 bit. For details, see [section 14.3.7. Level of Power Generation Detection](#)

44.2.13 ADSCSn : A/D Channel Conversion Order Setting Register n (n = 0 to 7, 16, 17, 20, 21, 31)

Base address: ADC140 = 0x4005_C000

Offset address: 0x1C0 + 0x1 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SCSm[4:0]				

Value after reset: 0 0 0 See [Table 44.16](#).

Bit	Symbol	Function	R/W
4:0	SCSm[4:0] (m: 00 to 07, 16, 17, 20, 21, 31)	Conversion Channel Order Setting Conversion on channels specified in ADSCSn (n = 0 to 7, 16, 17, 20, 21, 31) is performed in ascending order of the value of n.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The ADSCSn register sets the A/D conversion order. The ADSCSn register changes the conversion order for the analog input.

Table 44.16 Relationship between the value of the Conversion Channel Setting bits (SCSm[4:0]) after reset and the conversion order (1 of 2)

Bit name	Value after reset	Conversion order	Specifiable channels*1
ADSCS0.SCS00[4:0]	0x00	The specified channel is the first to be converted.	AN000 to AN007, AN016, AN017, AN020, AN021, AN031 (0x00 to 0x1F)
ADSCS1.SCS01[4:0]	0x01	The specified channel is the second to be converted.	AN000 to AN007, AN016, AN017, AN020, AN021, AN031 (0x00 to 0x1F)
ADSCS2.SCS02[4:0]	0x02	The specified channel is the third to be converted.	AN000 to AN007, AN016, AN017, AN020, AN021, AN031 (0x00 to 0x1F)
ADSCS3.SCS03[4:0]	0x03	The specified channel is the 4th to be converted.	AN000 to AN007, AN016, AN017, AN020, AN021, AN031 (0x00 to 0x1F)
ADSCS4.SCS04[4:0]	0x04	The specified channel is the 5th to be converted.	AN000 to AN007, AN016, AN017, AN020, AN021, AN031 (0x00 to 0x1F)
ADSCS5.SCS05[4:0]	0x05	The specified channel is the 6th to be converted.	AN000 to AN007, AN016, AN017, AN020, AN021, AN031 (0x00 to 0x1F)

Table 44.16 Relationship between the value of the Conversion Channel Setting bits (SCSm[4:0]) after reset and the conversion order (2 of 2)

Bit name	Value after reset	Conversion order	Specifiable channels*1
ADSCS6.SCS06[4:0]	0x06	The specified channel is the 7th to be converted.	AN000 to AN007, AN016, AN017, AN020, AN021, AN031 (0x00 to 0x1F)
ADSCS7.SCS07[4:0]	0x07	The specified channel is the 8th to be converted.	AN000 to AN007, AN016, AN017, AN020, AN021, AN031 (0x00 to 0x1F)
ADSCS16.SCS16[4:0]	0x10	The specified channel is the 9th to be converted.	AN000 to AN007, AN016, AN017, AN020, AN021, AN031 (0x00 to 0x1F)
ADSCS17.SCS17[4:0]	0x11	The specified channel is the 10th to be converted.	AN000 to AN007, AN016, AN017, AN020, AN021, AN031 (0x00 to 0x1F)
ADSCS20.SCS20[4:0]	0x14	The specified channel is the 11th to be converted.	AN000 to AN007, AN016, AN017, AN020, AN021, AN031 (0x00 to 0x1F)
ADSCS21.SCS21[4:0]	0x15	The specified channel is the 12th to be converted.	AN000 to AN007, AN016, AN017, AN020, AN021, AN031 (0x00 to 0x1F)
ADSCS31.SCS31[4:0]	0x1F	The specified channel is the 13th to be converted.	AN000 to AN007, AN016, AN017, AN020, AN021, AN031 (0x00 to 0x1F)

Note 1. The values set in AN000 to AN007 are 0x00 to 0x07, the values set in AN016 and AN017 are 0x10 and 0x11, the values set in AN020 to AN021 are 0x14 to 0x15, and the values set in VSC_VCC is 0x1F. Setting of values from 0x08 to 0x0F, 0x12 and 0x13, and 0x16 to 0x1E is prohibited.

If the specified channel is excluded from the conversion target in the A/D Channel Select Register, A/D conversion is not performed on that channel.

If the value after reset is used, a maximum of 12 arbitrarily-selected analog input channels are converted in ascending order of channel number.

SCSm bits (m = 00 to 07, 16, 17, 20, 21, 31) (Conversion Channel Order Setting)

The SCSm bits set the A/D conversion order for the analog input channels (AN000 to AN007, AN016, AN017, AN020, AN021, and AN031). Set the conversion order for all the existing channels. Values to be set in ADSCSn (n = 0 to 7, 16, 17, 20, 21, and 31) correspond to channels AN000 to AN007, AN016, AN017, AN020, AN021, and AN031. A/D conversion is performed on the specified channels in order from ADSCS0 with the highest priority to ADSCS31 with the lowest priority. If a channel is set to 0 (disabled) in an A/D Channel Select Register (such as ADANSA0), A/D conversion is not performed on that channel during scanning of the relevant group irrespective of the priority setting in ADSCSn (n = 0 to 7, 16, 17, 20, 21 and 31). Note that the conversion order for temperature sensor output cannot be changed.

44.2.14 ADADS0 : A/D-Converted Value Addition/Average Channel Select Register 0

Base address: ADC140 = 0x4005_C000

Offset address: 0x008

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADS15	ADS14	ADS13	ADS12	ADS11	ADS10	ADS09	ADS08	ADS07	ADS06	ADS05	ADS04	ADS03	ADS02	ADS01	ADS00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADSn	A/D-Converted Value Addition/Average Channel Select Bit 15 (ADS15) is associated with AN015 and bit 0 (ADS00) is associated with AN000. 0: Do not select associated input channel. 1: Select associated input channel.	R/W

Note: n = 00 to 07

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

ADSn bits (A/D-Converted Value Addition/Average Channel Select)

The ADSn bits determine which A/D-converted channels are subject to A/D-converted value addition/averaging. When an ADSn bit associated with a channel selected for A/D conversion is set to 1, A/D conversion of the analog input of the respective channel is performed successively 1, 2, 4, or 16 times, as specified in the ADC[2:0] bits in the ADADC register.

When the ADADC.AVEE bit is 0, the value obtained by addition is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition is stored in the A/D data register.

The ADSn bits apply only to channels that are selected for A/D conversion in:

- The ANSAn bits in the ADANSA0 register or the DBLANS[4:0] bits in the ADCSR register
- The ANSBn bits in the ADANSB0 register
- The ANSCn bits in the ADANSC0 register

For channels on which the A/D conversion is performed and for which addition/average mode is not selected, a normal 1-time conversion is executed and the conversion result is stored in the A/D data register.

Only set ADADS0 register bits when the ADCSR.ADST bit is 0.

Figure 44.2 shows a scanning operation sequence in which both the ADADS0.ADS02 and ADADS0.ADS06 bits are set to 1. In this figure:

- Addition mode is selected (ADADS.AVEE = 0)
- The number of conversions is set to 4 (ADADC.ADC[1:0] = 11b)
- Channels AN000 to AN007 are selected (ADANSA0.ANSA0[15:0] = 0x00FF) in continuous scan mode (ADCSR.ADCS[1:0] = 10b).

The conversion process begins with AN000. The AN002 conversion is performed successively 4 times and the added value is returned to A/D Data Register 2 (ADDR2). Next, the AN003 conversion process is started. The AN006 conversion is performed successively 4 times and the added value is returned to A/D Data Register 6 (ADDR6). After conversion of AN007, the conversion operation repeats in the same sequence starting with AN000.

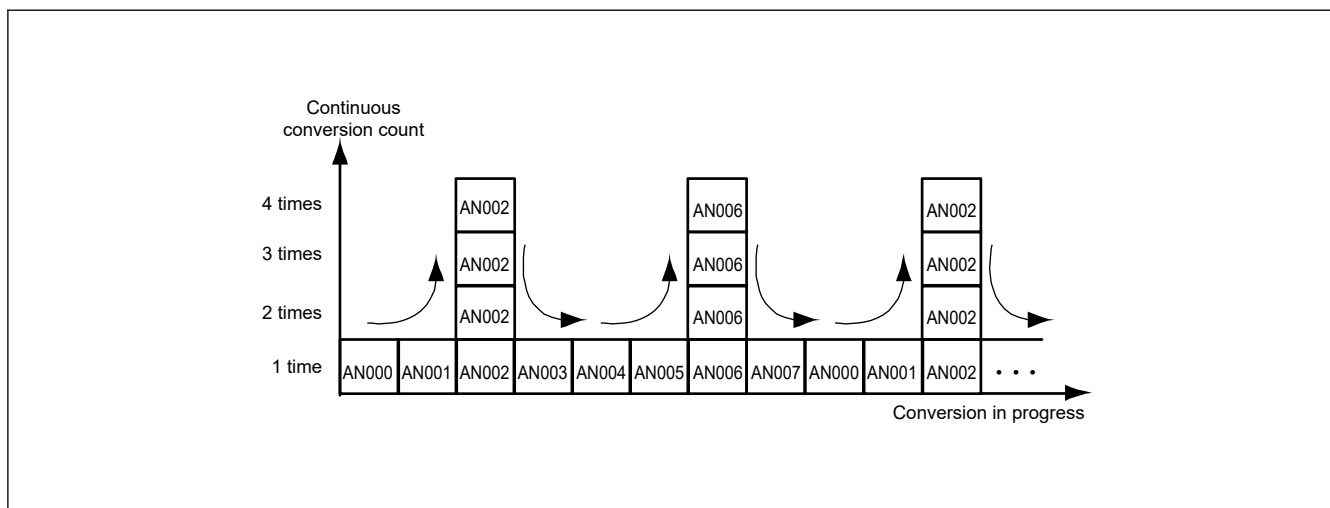


Figure 44.2 Scan conversion sequence with ADADC.ADC[2:0] = 011b, ADADS0.ADS02 = 1, ADADS0.ADS06 = 1

44.2.15 ADADS1 : A/D-Converted Value Addition/Average Channel Select Register 1

Base address: ADC140 = 0x4005_C000

Offset address: 0x00A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADS31	ADS30	ADS29	ADS28	ADS27	ADS26	ADS25	ADS24	ADS23	ADS22	ADS21	ADS20	ADS19	ADS18	ADS17	ADS16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADSn	A/D-Converted Value Addition/Average Channel Select Bit 15 (ADS31) is associated with VSC_VCC. Bit 14 (ADS30) is associated with AN030 and bit 0 (ADS16) is associated with AN016. 0: Do not select associated input channel. 1: Select associated input channel.	R/W

Note: n = 16, 17, 20, 21, 31

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

ADSn bits (A/D-Converted Value Addition/Average Channel Select)

The ADSn bits determine which A/D-converted channels are subject to A/D-converted value addition/averaging. When an ADSn bit associated with a channel selected for A/D conversion is set to 1, A/D conversion of the analog input of the respective channel is performed successively 1, 2, 4, or 16 times, as specified in the ADC[2:0] bits in the ADADC register.

When the ADADC.AVEE bit is 0, the value obtained by addition is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition is stored in the A/D data register.

The ADSn bits apply only to channels that are selected for A/D conversion in:

- The ANSAn bits in the ADANSA1 register or the DBLANS[4:0] bits in the ADCSR register
- The ANSBn bits in the ADANSB1 register.
- The ANSCn bits in the ADANSC0 register

For channels on which the A/D conversion is performed and for which addition/average mode is not selected, a normal 1-time conversion is executed and the conversion result is stored in the A/D data register.

Only set ADADS1 register when the ADCSR.ADST bit is 0.

44.2.16 ADADC : A/D-Converted Value Addition/Average Count Select Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x00C

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

Bit	Symbol	Function	R/W
2:0	ADC[2:0]	Addition/Average Count Select 0 0 0: 1-time conversion (no addition, same as normal conversion) 0 0 1: 2-time conversion (one addition) 0 1 0: Setting prohibited 0 1 1: 4-time conversion (three additions) 1 0 1: 16-time conversion (15 additions) Others: Setting prohibited	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	AVEE	Average Mode Select 0: Enable addition mode 1: Enable average mode	R/W

ADADC sets the addition or average mode and addition count for A/D conversion. Table 44.17 lists the settable combinations of ADADC register.

Table 44.17 Settable combinations of ADADC register

Average mode select (AVEE)	A/D Conversion Accuracy (ADCER.ADPRC[1:0])	Conversion time			
		1-time	2-times	4-times	16-times
1'b0	12-bit (ADPRC[1:0] = 00b)	✓	✓	✓	✓
	14-bit (ADPRC[1:0] = 11b)	✓	✓	✓	—
1'b1	—	✓	✓	✓	✓

Note: ✓: Selectable, —: Not selectable

ADC[2:0] bits (Addition/Average Count Select)

The ADC[2:0] bits set the addition count in all channels for which A/D conversion and addition/average mode are selected, including the channel selected in double trigger mode with the ADCSR.DBLANS[4:0] bits. The count also applies to A/D conversion of the temperature sensor output.

When self-diagnosis is executed (ADCER.DIAGM = 1), do not set the ADC[2:0] bits to any value other than 000b.

AVEE bit (Average Mode Select)

The AVEE bit selects addition or average mode in all channels for which A/D conversion and addition/average mode are selected, including the channels selected in double-trigger mode in the ADCSR.DBLANS[4:0] bits, temperature sensor output,.

44.2.17 ADCER : A/D Control Extended Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x00E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADRF MT	—	—	—	DIAG M	DIAGL D	DIAGVAL[1:0]	—	—	ACE	—	—	ADPRC[1:0]	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
0	—	These bits are read as 0. The write value should be 0.	R/W
2:1	ADPRC[1:0]	A/D Conversion Accuracy Specify 0 0: 12-bit accuracy 0 1: Setting prohibited 1 0: Setting prohibited 1 1: 14-bit accuracy	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	ACE	A/D Data Register Automatic Clearing Enable 0: Disable automatic clearing 1: Enable automatic clearing	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	DIAGVAL[1:0]	Self-Diagnosis Conversion Voltage Select 0 0: Setting prohibited when self-diagnosis is enabled 0 1: 0 volts 1 0: Reference power supply*1 voltage × 1/2 1 1: Reference power supply*1 voltage	R/W

Bit	Symbol	Function	R/W
10	DIAGLD	Self-Diagnosis Mode Select 0: Select rotation mode for self-diagnosis voltage 1: Select mixed mode for self-diagnosis voltage	R/W
11	DIAGM	Self-Diagnosis Enable 0: Disable ADC14 self-diagnosis 1: Enable ADC14 self-diagnosis	R/W
14:12	—	These bits are read as 0. The write value should be 0.	R/W
15	ADRFMT	A/D Data Register Format Select 0: Select right-justified for the A/D data register format 1: Select left-justified for the A/D data register format	R/W

Note 1. The reference voltage refers to VREFH0.

ADPRC[1:0] bit (A/D Conversion Accuracy Specify)

The ADPRC[1:0] bits set the A/D conversion accuracy. Changing the A/D conversion accuracy also changes the bit width of valid data stored in the result register and the A/D conversion time. For details, see [section 44.3.6. Analog Input Sampling and Scan Conversion Time](#) section 45.3.6, Analog Input Sampling and Scan Conversion Time. Only set the ADPRC[1:0] bits while the ADCSR.ADST bit is 0.

ACE bit (A/D Data Register Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all 0) of the ADDRy, ADDR, ADDBLDR, ADTSDR 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 44.3.7. Usage Example of A/D Data Register Automatic Clearing Function](#).

DIAGVAL[1:0] bits (Self-Diagnosis Conversion Voltage Select)

The DIAGVAL[1:0] bits select the voltage value used in self-diagnosis fixed voltage mode. For details, see the DIAGLD bit description.

Do not execute self-diagnosis by setting the DIAGLD bit to 1 when the DIAGVAL[1:0] bits are set to 00b.

DIAGLD bit (Self-Diagnosis Mode Select)

The DIAGLD bit selects whether the three voltage values are rotated or the fixed voltage is used in self-diagnosis.

Setting the DIAGLD bit to 0 selects conversion of the voltages in rotation mode, where 0 V, the reference power supply voltage $\times 1/2$, and the reference power supply voltage are converted, in that order. After reset and when self-diagnosis voltage rotation mode is selected, self-diagnosis is executed from 0 V. The self-diagnosis voltage value does not return to 0 V when scan conversion completes. When scan conversion is restarted, rotation starts at the voltage value following the previous value.

Setting the DIAGLD bit to 1 selects fixed voltage, in which the fixed voltage specified in the ADCER.DIAGVAL[1:0] bits is converted. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value.

Only set the DIAGLD bit when the ADCSR.ADST bit is 0.

DIAGM bit (Self-Diagnosis Enable)

The DIAGM bit enables or disables self-diagnosis.

Self-diagnosis is used to detect a failure of the ADC14. In self-diagnosis mode, one of the three voltage values (0 V, the reference power supply voltage $\times 1/2$, or the reference power supply voltage) is converted. When conversion completes, information on the converted voltage and the conversion result is stored into the A/D Self-Diagnosis Data Register (ADDR). The ADDR register can be read to determine whether the conversion result falls within the normal or abnormal range.

Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. In double trigger mode (ADCSR.DBLE = 1), self-diagnosis (DIAGM = 0) is deselected. When self-diagnosis is selected in group scan mode, self-diagnosis is executed separately for group A, group B and group C.

Only set the DIAGM bit when the ADCSR.ADST bit is 0.

ADRFMT bit (A/D Data Register Format Select)

The ADRFMT bit specifies flush-right or flush-left for data to be stored in the ADDRy, ADDBLDR, ADTSDR, ADCMPDR0/1, ADWINLLB, ADWINULB, or ADRD register.

Only set the ADRFMT bit when the ADCSR.ADST bit is 0.

44.2.18 ADSTRGR : A/D Conversion Start Trigger Select Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x010

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

Bit	Symbol	Function	R/W
5:0	TRSB[5:0]	A/D Conversion Start Trigger Select for Group B Select the A/D conversion start trigger for group B in group scan mode.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
13:8	TRSA[5:0]	A/D Conversion Start Trigger Select Select the A/D conversion start trigger in single scan mode and continuous scan mode. In group scan mode, the A/D conversion start trigger for group A is selected.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

TRSB[5:0] bits (A/D Conversion Start Trigger Select for Group B)

The TRSB[5:0] bits select the trigger to start scanning of the analog input selected in group B. The TRSB[5:0] bits must only be set in group scan mode and are not used in any other scan mode. For the scan conversion start trigger for group B, setting a software trigger or an asynchronous trigger is prohibited. In group scan mode, set the TRSB[5:0] bits to a value other than 0x00 and set the ADCSR.TRGE bit to 1.

When group A is given priority in group scan mode, setting the ADGSPCR.GBRP bit to 1 allows group B to continuously operate in single scan mode. When setting the ADGSPCR.GBRP bit to 1, set the TRSB[5:0] bits to 0x3F. The issuance period for a conversion trigger must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN} , A/D conversion by the trigger might have no effect.

When the GPT module is selected as an A/D conversion start trigger, a delay for synchronization processing occurs. For details, see [section 44.3.6. Analog Input Sampling and Scan Conversion Time](#).

[section 44.2.18. ADSTRGR : A/D Conversion Start Trigger Select Register](#) lists the A/D conversion startup sources selected in the TRSB[5:0] bits.

Table 44.18 Selection of A/D conversion start sources in the TRSB[5:0] bits

Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
Trigger source deselected state	—	1	1	1	1	1	1
Compare match between the TCORA register and the TCNT counter	TMR	0	1	1	1	0	1
ELC_ADC14	ELC	1	1	0	0	0	0

TRSA[5:0] bits (A/D Conversion Start Trigger Select)

The TRSA[5:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode, or the trigger to start scanning of group A analog inputs in group scan mode. When scanning is executed in group scan mode or double trigger mode, software trigger or asynchronous trigger is prohibited.

- When using a synchronous trigger (ELC), set the TRGE bit in the ADCSR register to 1 and set the EXTRG bit in the ADCSR register to 0.
- When using the asynchronous trigger (ADTRG0), set the TRGE bit in the ADCSR register to 1 and set the EXTRG bit in the ADCSR register to 1.
- Software trigger (ADCSR.ADST) is enabled regardless of the settings of the ADCSR.TRGE bit, the ADCSR.EXTRG bit, or the TRSA[5:0] bits.

The issuance period for a conversion trigger must be more than or equal to the actual scan conversion time (tSCAN). If the issuance period is less than tSCAN, A/D conversion by a trigger might have no effect.

[section 44.2.18. ADSTRGR : A/D Conversion Start Trigger Select Register](#) lists the A/D conversion start sources selected in the TRSA[5:0] bits.

Table 44.19 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 deselected state	—	1	1	1	1	1	1
ADTRGn	Input pin for the trigger	0	0	0	0	0	0
Compare match between the TCORA register and the TCNT counter	TMR	0	1	1	1	0	1
ELC_ADC14	ELC	1	1	0	0	0	0

44.2.19 ADEXICR : A/D Conversion Extended Input Control Registers

Base address: ADC140 = 0x4005_C000

Offset address: 0x012

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

Bit	Symbol	Function	R/W
0	TSSAD	Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select 0: Do not select addition/average mode for temperature sensor output. 1: Select addition/average mode for temperature sensor output.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	TSSA	Temperature Sensor Output A/D Conversion Select 0: Disable A/D conversion of temperature sensor output 1: Enable A/D conversion of temperature sensor output	R/W
9	—	These bits are read as 0. The write value should be 0.	R/W
10	TSSB	Temperature Sensor Output A/D Conversion Select for Group B 0: Disable A/D conversion of temperature sensor output 1: Enable A/D conversion of temperature sensor output	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

TSSAD bit (Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select)

When the TSSAD bit is set to 1, A/D conversion of the temperature sensor output is selected and performed successively the number of times specified in the ADC[2:0] bits in ADADC. The maximum addition count differs depending on the conversion accuracy (see [section 44.2.5. ADRD : A/D Self-Diagnosis Data Register](#)). When the ADADC.AVEE bit is 0, the

value obtained by addition (integration) is returned to the A/D Temperature sensor Data Register (ADTSDR). When the ADADC.AVEE bit is 1, the mean value is returned to ADTSDR.

Only set the TSSAD bit while the ADCSR.ADST bit is 0.

TSSA bit (Temperature Sensor Output A/D Conversion Select)

The TSSA bit selects A/D conversion of the temperature sensor output. When A/D conversion of the temperature sensor output is selected and performed, set the ADCSR.DBLE bit to 0. Only set the TSSA bit when the ADCSR.ADST bit is 0. This bit can be selected for single-scan mode, continuous-scan mode, and group A in group-scan mode.

TSSB bit (Temperature Sensor Output A/D Conversion Select for Group B)

The TSSB bit selects A/D conversion of the temperature sensor output for group B in group scan mode. Only set the TSSB bit while the ADCSR.ADST bit is 0. Do not set the TSSB bit to 1 while the TSSA bit is 1.

44.2.20 ADGCEXCR : A/D Group C Extended Input Control Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x0D8

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

Bit	Symbol	Function	R/W
0	TSSC	Temperature Sensor Output A/D Conversion Select for Group C Select whether to enable A/D conversion of the temperature sensor output for group C in group-scan mode. 0: Disable A/D conversion of temperature sensor output 1: Enable A/D conversion of temperature sensor output.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

ADGCEXCR is an extended input setting register for group C. Set the ADGCEXCR register while the ADCSR.ADST bit is 0.

TSSC bit (Temperature Sensor Output A/D Conversion Select for Group C)

This bit selects A/D conversion of the temperature sensor output for group C in group-scan mode.

Only set the TSSC bit when the ADCSR.ADST bit is 0. Do not set the TSSC bit to 1 while the TSSA or TSSB bit is 1.

44.2.21 ADGCTRGR : A/D Group C Trigger Select Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x0D9

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

Bit	Symbol	Function	R/W
5:0	TRSC[5:0]	A/D Conversion Start Trigger Select for Group C Select the A/D conversion start trigger for group C in group-scan mode.	R/W
6	GCADIE	Group C Scan End Interrupt Enable 0: Disable ADC140_GCADI interrupt generation upon completion of group C scan 1: Enable ADC140_GCADI interrupt generation upon completion of group C scan.	R/W

Bit	Symbol	Function	R/W
7	GRCE	A/D Conversion Operation Enable for Group C Enable or disable A/D conversion operation on group C. 0: Do not use group C 1: Use group C.	R/W

TRSC[5:0] bits (A/D Conversion Start Trigger Select for Group C)

The TRSC[5:0] bits select the trigger to start scanning of the analog input selected in group C. The TRSC[5:0] bits are only used in group-scan mode, and are not used in any other scan mode. Do not use a software trigger or an asynchronous trigger as the scan conversion start trigger for group C. In group-scan mode, set the TRSC[5:0] bits to a value other than 0x00, and set the ADCSR.TRGE bit to 1.

When group priority control is enabled in group-scan mode, setting the ADGSPCR.GBRP bit to 1 allows group C to continuously operate in single-scan mode. When setting the ADGSPCR.GBRP bit to 1, set the TRSC[5:0] bits to 0x3F.

The issuance interval of the trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance interval is less than t_{SCAN} , A/D conversion by the trigger might have no effect.

Table 44.20 lists the A/D conversion start sources selected by the TRSC[5:0] bits.

Table 44.20 Selection of A/D activation sources by TRSC[5:0] bits

Source	Remarks	TRSC[5]	TRSC[4]	TRSC[3]	TRSC[2]	TRSC[1]	TRSC[0]
Trigger source deselected state		1	1	1	1	1	1
Compare match between the TCORA register and the TCNT counter	TMR	0	1	1	1	0	1
ELC_ADC14	ELC	1	1	0	0	0	0

GCADIE bit (Group C Scan End Interrupt Enable)

The GCADIE bit enables or disables the group C scan end interrupt (ADC140_GCADI) in group-scan mode.

GRCE bit (A/D Conversion Operation Enable for Group C)

To use group C in group-scan mode, set the GRCE bit to 1. If the GRCE bit is 0, input of a trigger for group C is disabled.

When group priority operation using group C is enabled (ADGSPCR.PGS bit is 1) and ADGSPCR.GBRP is set to 1, single scan of group C is performed continuously. (If the GRCE bit is set to 1, single scan of group B is not performed continuously.)

Set the GRCE bit while the ADCSR.ADST bit is 0.

44.2.22 ADSSTRn/ADSSTRL/ADSSTRT : A/D Sampling State Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x0E0 + 0x1 × n (n = 0 to 7)
0x0DD (ADSSTRL)
0x0DE (ADSSTRT)

Bit position: 7 6 5 4 3 2 1 0

Bit field: SST[7:0]

Value after reset: 0 0 0 0 1 1 1 1

Bit	Symbol	Function	R/W
7:0	SST[7:0]	Sampling Time Setting These bits set the sampling time in the range from 5 to 255 states.	R/W

The ADSSTRn register sets the sampling time for analog input.

The sampling time can be adjusted if the impedance of the analog input signal source is too high to secure sufficient sampling time, or if the ADCLK clock is slow. The set value indicates the time for one ADCLK cycle, and the required sampling time is specified by the voltage conditions. about detail, see [section 51.4. A/D Conversion Characteristics](#)

In sub clock mode (ADSCCLKCR.SCLKEN = 1), set a value between 2 to 8. For other modes, set a value between 5 to 255.

Table 44.21 shows the relationship between the A/D Sampling State Register and the associated channels. For details, see section 44.3.6. Analog Input Sampling and Scan Conversion Time.

Only set the SST[7:0] bits when the ADCSR.ADST bit is 0.

Table 44.21 Relationship between A/D sampling state register and associated channels

Bit name	Associated channels
ADSSTRn.SST[7:0] bits (n = 00 to 07)*1	AN0n (n = 0 to 7)
ADSSTRL.SST[7:0] bits	AN0n (n = 16, 17, 20, 21), VSC_VCC
ADSSTRT.SST[7:0] bits	Temperature sensor output

Note 1. When the self-diagnosis function is selected, the sampling time set in the ADSSTR0.SST[7:0] bits is applied.

44.2.23 ADDISCR : A/D Disconnection Detection Control Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x07A

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

Bit	Symbol	Function	R/W
3:0	ADNDIS[3:0]	Disconnection Detection Assist Setting 0x00: The disconnection detection assist function is disabled 0x01: Setting prohibited Others: The number of states for the discharge or precharge period.	R/W
4	PCHG	Precharge/discharge select 0: Discharge 1: Precharge	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The ADDISCR register selects either precharge or discharge, and the period of precharge or discharge for the A/D disconnection detection assist function. Only set the ADDISCR register when the ADCSR.ADST bit is 0. When the temperature sensor output is converted, the A/D converter executes discharge automatically.

Disable the disconnection detection assist function if any of the following functions are used:

- The temperature sensor
- A/D self-diagnosis

ADNDIS[3:0] bits (Disconnection Detection Assist Setting)

The ADNDIS[3:0] bits specify the period of precharge or discharge. When ADNDIS[3:0] = 0000b, the disconnection detection assist function is disabled. Setting the ADNDIS[3:0] bits to 0001b is prohibited. Except when ADNDIS[3:0] = 0000b or 0001b, the specified value indicates the number of states for the period of precharge or discharge. When the ADNDIS[3:0] bits are set to any values other than 0000b or 0001b, the disconnection detection assistance function is enabled.

PCHG bit (Precharge/discharge select)

The PCHG bit selects either precharge or discharge.

44.2.24 ADELCCR : A/D Event Link Control Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x07D

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

Bit	Symbol	Function	R/W
1:0	ELCC[1:0]	Event Link Control 0 0: An event is generated upon completion of scanning other than group B and group C scans in group-scan mode. 0 1: An event is generated when group B scan completes in group-scan mode. 1 x: An event is generated when all scans complete.	R/W
2	GCELC	Group C Event Link Control 0: No event is generated when group C scan completes in groupscan mode 1: An event is generated when group C scan completes in groupscan mode.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: x: Don't care

ELCC[1:0] bit (Event Link Control)

The ELCC[1:0] bits select the generating condition of a scan end event (AD140_ELC).

To enable the setting of ELCC[1:0], set GCELC to 0.

GCELC bit (Group C Event Link Control)

The GCELC bit controls scan end events for group C in group-scan mode. If GCELC is set to 1, an event is generated only when group C scan completes in group-scan mode, irrespective of the setting of ELCC[1:0].

To enable the setting of ELCC[1:0], set GCELC to 0.

44.2.25 ADGSPCR : A/D Group Scan Priority Control Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x080

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

Bit	Symbol	Function	R/W
0	PGS	Group Priority Operation Setting*1 0: Operate without group priority control. 1: Operate with group priority control.	R/W
1	GBRSCN	Lower-Priority Group Restart Setting (enabled only when PGS = 1 and reserved when PGS = 0.) 0: Disable rescanning of the group that was stopped in group priority operation 1: Enable rescanning of the group that was stopped in group priority operation.	R/W
13:2	—	These bits are read as 0. The write value should be 0.	R/W
14	LGRRS	Restart Channel Select Enabled only when PGS = 1 and GBRSCN = 1. 0: Start rescanning from the first channel for scanning 1: Start rescanning from the channel for which A/D conversion is not completed.	R/W

Bit	Symbol	Function	R/W
15	GBRP	Single Scan Continuous Start* ² (enabled only when PGS = 1 and reserved when PGS = 0.) 0: Single scan is not continuously activated. 1: Single scan for the group with the lowest-priority is continuously activated.	R/W

Note 1. The ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode) before setting PGS to 1. Operation is not guaranteed if these bits are set to any other value.

Note 2. When the GBRP bit is set to 1, single scan is performed continuously for the group with the lowest-priority regardless of the setting in the GBRSCN bit.

PGS bit (Group Priority Operation Setting)

Set the PGS bit to 1 to enable group priority operation.

The ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode) before setting the PGS bit to 1. Operation is not guaranteed if the bits are set to any other value.

When the PGS bit is set to 0, a clear operation must be performed by software as described in [section 44.7.3. Constraints on Stopping A/D Conversion](#). When the PGS bit is set to 1, use the settings described in [section 44.3.4.3. Group Priority Operation](#).

GBRSCN bit (Lower-Priority Group Restart Setting)

The GBRSCN bit controls the restarting of scan operation in group priority operation.

When the GBRSCN bit is set to 1, if the scan operation of a lower-priority group is stopped by a trigger input of a priority group, the lower-priority group scanning is restarted on completion of the priority group scanning. If a trigger of a lower-priority group is input during scanning of the priority group, the lower-priority group scanning is started on completion of the priority group scanning.

When the GBRSCN bit is set to 0, triggers input during scanning are ignored. Set the GBRSCN bit while the ADCSR.ADST bit is 0.

LGRRS bit (Restart Channel Select)

This bit sets the channel from which rescanning is to be started in group priority operation. The setting of the LGRRS bit is valid when the PGS and GBRSCN bits are 1.

If the LGRRS bit is 0, scanning of a lower-priority group that was stopped in group priority operation is restarted from the first channel after scanning of the priority group completes.

If the LGRRS bit is 1, scanning of a lower-priority group that was stopped in group priority operation is restarted (upon completion of scanning of the priority group) from the channel for which A/D conversion is not complete. If A/D conversion of the addition setting channel was not completed the specified number of times when scanning stopped, A/D conversion of the addition setting channel is performed again the specified number of times when scanning restarts.

Set the LGRRS bit while the ADCSR.ADST bit is 0.

GBRP bit (Single Scan Continuous Start)

The GBRP bit is set when a single scan operation is to be performed continuously on the group with the lowest-priority.

When groups A, B, and C are used, group C has the lowest-priority. When groups A and B are used, group B has the lowest-priority.

Setting the GBRP bit to 1 starts a single scan of the group with the lowest-priority. On completion of the scan, another single scan of the group with the lowest-priority is started automatically. If scanning has been stopped during group priority operation, single scan of the group with the lowest-priority is automatically restarted on completion of the A/D conversion of the priority group.

Before setting the GBRP bit to 1, disable input of a trigger for the lowest-priority group. If the GBRP bit is set to 1, rescanning is performed only on the group with the lowest-priority even if the GBRSCN bit is set to 0.

44.2.26 ADCMPCR : A/D Compare Function Control Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x090

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPAIE	WCMP E	CMPBIE	—	CMPAE	—	CMPBE	—	—	—	—	—	—	—	—	CMPAB[1:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CMPAB[1:0]	Window A/B Composite Conditions Setting These bits are valid when both window A and window B are enabled (CMPAE = 1 and CMPBE = 1). 0 0: Output ADC140_WCMPPM when window A OR window B comparison conditions are met. Otherwise, output ADC140_WCMPUM. 0 1: Output ADC140_WCMPPM when window A EXOR window B comparison conditions are met. Otherwise, output ADC140_WCMPUM. 1 0: Output ADC140_WCMPPM when window A AND window B comparison conditions are met. Otherwise, output ADC140_WCMPUM. 1 1: Setting prohibited.	R/W
8:2	—	These bits are read as 0. The write value should be 0.	R/W
9	CMPBE	Compare Window B Operation Enable 0: Disable compare window B operation. Disable ADC140_WCMPPM and ADC140_WCMPUM outputs. 1: Enable compare window B operation.	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	CMPAE	Compare Window A Operation Enable 0: Disable compare window A operation. Disable ADC140_WCMPPM and ADC140_WCMPUM outputs. 1: Enable compare window A operation.	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	CMPBIE	Compare B Interrupt Enable 0: Disable ADC140_CMPBI interrupt when comparison conditions (window B) are met. 1: Enable ADC140_CMPBI interrupt when comparison conditions (window B) are met.	R/W
14	WCMP E	Window Function Setting 0: Disable window function Window A and window B operate as a comparator to compare the single value on the lower side with the A/D conversion result. 1: Enable window function Window A and window B operate as a comparator to compare the two values on the upper and lower sides with the A/D conversion result.	R/W
15	CMPAIE	Compare A Interrupt Enable 0: Disable ADC140_CMPAI interrupt when comparison conditions (window A) are met. 1: Enable ADC140_CMPAI interrupt when comparison conditions (window A) are met.	R/W

CMPAB[1:0] bits (Window A/B Composite Conditions Setting)

The CMPAB[1:0] bits are valid when both window A and window B are enabled (CMPAE = 1 and CMPBE = 1) in single scan mode. These bits specify the compare function match/mismatch event output conditions and monitoring conditions of ADWINMON.MONCONB. Only set the CMPAB[1:0] bits while the ADCSR.ADST bit is 0.

CMPBE bit (Compare Window B Operation Enable)

The CMPBE bit enables or disables the compare window B operation. Set the CMPBE bit while the ADCSR.ADST bit is 0. Set this bit to 0 before setting the following registers:

- A/D Channel Select Registers A0, A1, B0, B1, C0, C1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1, ADANSC0, ADANSC1)
- TSSB or TSSA bits in the A/D Conversion Extended Input Control Register (ADEXICR)
- A/D Group C Extended Input Control Register (ADGCEXCR.TSSC bit)
- CMPCHB[5:0] bits in the Window B Channel Select Register (ADCMPBNSR)

CMPAE bit (Compare Window A Operation Enable)

The CMPAE bit enables or disables the compare window A operation. Set the CMPAE bit while the ADCSR.ADST bit is 0. Set this bit to 0 before setting the following registers:

- A/D Channel Select Registers A0, A1, B0, B1, C0, C1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1, ADANSC0, ADANSC1)
- TSSB or TSSA bits in the A/D Conversion Extended Input Control Register (ADEXICR)
- A/D Group C Extended Input Control Register (ADGCEXCR.TSSC bit)
- Window A Channel Select Registers 0 and 1 (ADCMPANSR0 and ADCMPANSR1)
- Window A Extended Input Select Register (ADCMPANSER)

CMPBIE bit (Compare B Interrupt Enable)

The CMPBIE bit enables or disables the ADC140_CMPBI interrupt output when the comparison conditions (window B) are met.

WCMPE bit (Window Function Setting)

The WCMPE bit enables or disables the window function. Set the WCMPE bit while the ADCSR.ADST bit is 0.

CMPAIE bit (Compare A Interrupt Enable)

The CMPAIE bit enables or disables the ADC140_CMPAI interrupt output when the comparison conditions (window A) are met.

44.2.27 ADCMPANSR0 : A/D Compare Function Window A Channel Select Register 0

Base address: ADC140 = 0x4005_C000

Offset address: 0x094

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPC HA15	CMPC HA14	CMPC HA13	CMPC HA12	CMPC HA11	CMPC HA10	CMPC HA09	CMPC HA08	CMPC HA07	CMPC HA06	CMPC HA05	CMPC HA04	CMPC HA03	CMPC HA02	CMPC HA01	CMPC HA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPCHAN	Compare Window A Channel Select Bit 15 (CMPCHA15) is associated with AN015 and bit 0 (CMPCHA00) is associated with AN000. 0: Disable compare function for associated input channel 1: Enable compare function for associated input channel	R/W

Note: n = 00 to 07

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

CMPCHAN bits (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 and the ADANSB0.ANSBn bits.

Set the CMPCHAN bits while the ADCSR.ADST bit is 0.

44.2.28 ADCMPANSR1 : A/D Compare Function Window A Channel Select Register 1

Base address: ADC140 = 0x4005_C000

Offset address: 0x096

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPC HA31	CMPC HA30	CMPC HA29	CMPC HA28	CMPC HA27	CMPC HA26	CMPC HA25	CMPC HA24	CMPC HA23	CMPC HA22	CMPC HA21	CMPC HA20	CMPC HA19	CMPC HA18	CMPC HA17	CMPC HA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPCHAN	Compare Window A Channel Select Bit 15 (CMPCHA31) is associated with VSC_VCC. Bit 14 (CMPCHA30) is associated with AN030 and bit 0 (CMPCHA16) is associated with AN016. 0: Disable compare function for associated input channel 1: Enable compare function for associated input channel	R/W

Note: n = 16, 17, 20, 21, 31

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

CMPCHAN bits (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 and the ADANSB1.ANSBn bits.

Set the CMPCHAN bits while the ADCSR.ADST bit is 0.

44.2.29 ADCMPANSER : A/D Compare Function Window A Extended Input Select Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x092

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

Bit	Symbol	Function	R/W
0	CMPTSA	Temperature Sensor Output Compare Select 0: Exclude the temperature sensor output from the compare Window A target range. 1: Include the temperature sensor output in the compare Window A target range.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

CMPTSA bit (Temperature Sensor Output Compare Select)

The window A comparison conditions are applied to the temperature sensor output by setting the CMPTSA bit to 1 while the ADEXICR.TSSA, ADEXICR.TSSB, or ADGCXCR.TSSC bit is set to 1. Set the CMPTSA bit while the ADCSR.ADST bit is 0.

44.2.30 ADCMPLR0 : A/D Compare Function Window A Comparison Condition Setting Register 0

Base address: ADC140 = 0x4005_C000

Offset address: 0x098

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPLCHA15	CMPLCHA14	CMPLCHA13	CMPLCHA12	CMPLCHA11	CMPLCHA10	CMPLCHA09	CMPLCHA08	CMPLCHA07	CMPLCHA06	CMPLCHA05	CMPLCHA04	CMPLCHA03	CMPLCHA02	CMPLCHA01	CMPLCHA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPLCHAN	Compare Window A Comparison Condition Select These bits set comparison conditions for channels to which Window A comparison conditions are applied. Bit 15 (CMPLCHA15) is associated with AN015 and bit 0 (CMPLCHA00) is associated with AN000. Comparison conditions are shown in Figure 44.3. 0: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value > A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADCMPDR0 value, or ADCMPDR1 value < A/D-converted value 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): ADCMPDR0 value < A/D-converted value < ADCMPDR1 value	R/W

Note: n = 00 to 07

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

CMPLCHAN bits (Compare Window A Comparison Condition Select)

The CMPLCHAN bits specify the comparison conditions for channels to which Window A comparison conditions are applied. These bits can be set for each analog input to be compared. When the comparison result of each analog input meets the set condition, the ADCMPDR0.CMPSTCHAN flag sets to 1 and a compare interrupt (ADC140_CMPAI) is generated.

Comparison conditions when the window function is disabled			
CMPLCHAN = 0		CMPLCHAN = 1	
ADCMPDR0 value $\leq$ A/D converted value	Not met	ADCMPDR0 value $<$ A/D converted value	Met
ADCMPDR0 value $>$ A/D converted value	Met	ADCMPDR0 value $\geq$ 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 $\leq$ A/D converted value $\leq$ ADCMPDR1 value	Not met		
A/D converted value $<$ ADCMPDR0 value	Met		
CMPLCHAN = 1			
ADCMPDR1 value $\leq$ A/D converted value	Not met		
ADCMPDR0 value $<$ A/D converted value $<$ ADCMPDR1 value	Met		
A/D converted value $\leq$ ADCMPDR0 value	Not met		

Figure 44.3 Explanation of comparison conditions for compare function Window A

44.2.31 ADCMPLR1 : A/D Compare Function Window A Comparison Condition Setting Register 1

Base address: ADC140 = 0x4005_C000

Offset address: 0x09A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPL CHA3 1	CMPL CHA3 0	CMPL CHA2 9	CMPL CHA2 8	CMPL CHA2 7	CMPL CHA2 6	CMPL CHA2 5	CMPL CHA2 4	CMPL CHA2 3	CMPL CHA2 2	CMPL CHA2 1	CMPL CHA2 0	CMPL CHA1 9	CMPL CHA1 8	CMPL CHA1 7	CMPL CHA1 6
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPLCHAN	Compare Window A Comparison Condition Select These bits set comparison conditions for channels to which Window A comparison conditions are applied. Bit 15 (CMPLCHA31) is associated with VSC_VCC. Bit 14 (CMPLCHA30) is associated with AN030 and bit 0 (CMPLCHA16) is associated with AN016. Comparison conditions are shown in Figure 44.3. 0: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value > A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADCMPDR0 value, or ADCMPDR1 value < A/D-converted value 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): ADCMPDR0 value < A/D-converted value < ADCMPDR1 value	R/W

Note: n = 16, 17, 20, 21, 31

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

CMPLCHAN bits (Compare Window A Comparison Condition Select)

The CMPLCHAN bits specify the comparison conditions for analog channels to which window A comparison conditions are applied. These bits can be set for each analog input to be compared. When the comparison result of each analog input meets the set condition, the ADCMPSR1.CMPSTCHAN bit is set to 1 and a compare interrupt (ADC140_CMPAI) is generated.

44.2.32 ADCMPLER : A/D Compare Function Window A Extended Input Comparison Condition Setting Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x093

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

Bit	Symbol	Function	R/W
0	CMPLTSA	Compare Window A Temperature Sensor Output Comparison Condition Select Comparison conditions are shown in Figure 44.3. 0: When window function is disabled (ADCMPCR.WCMPE = 0) : ADCMPDR0 value > A/D-converted valueCompare Window A Temperature Sensor Output Comparison Condition Select When window function is enabled (ADCMPCR.WCMPE = 1) : Compare Window A Temperature Sensor Output Comparison ConditionA/D-converted value < ADCMPDR0 value, or A/D-converted value > ADCMPDR1 value 1: When window function is disabled (ADCMPCR.WCMPE = 0) : ADCMPDR0 value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1) : ADCMPDR0 value < A/D-converted value < ADCMPDR1 value	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

CMPLTSA bit (Compare Window A Temperature Sensor Output Comparison Condition Select)

The CMPLTSA bit specifies comparison conditions when the temperature sensor output is the target for the Window A comparison condition. When the temperature sensor output comparison result meets the set condition, the ADCMPSER.CMPSTTSA flag sets to 1 and a compare interrupt (ADC140_CMPAI) is generated.

44.2.33 ADCMPDRn : A/D Compare Function Window A Lower-Side/Upper-Side Level Setting Register (n = 0, 1)

Base address: ADC140 = 0x4005_C000

Offset address: 0x09C + (0x2 × n)

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

The ADCMPDRy (y = 0, 1) register specifies the reference data when the compare window A function is used. ADCMPDR0 sets the lower reference for window A, and ADCMPDR1 sets the upper reference for window A.

ADCMPDRy are read/write registers.

ADCMPDRy are writable even during A/D conversion. The reference data can be dynamically changed by rewriting register values during A/D conversion^{*1}.

Set these registers so that the upper reference is not less than the lower reference ($\text{ADCMPDR1} \geq \text{ADCMPDR0}$ and). ADCMPDR1 and are not used when the window function is disabled.

Note 1. The lower and the upper references are changed when each register is written. For example, when the upper reference value is changed and the lower reference value is being changed, the MCU compares the upper reference (after rewrite), and the lower reference (before rewrite) with the A/D conversion result. See [Figure 44.4](#). If the comparison during the rewriting of these two references is erroneous, then rewrite these reference values when both ADCSR.ADST and the target Compare Window Operation Enable bit (ADCMPCR.CMPAE or ADCMPCR.CMPBE) are 0.

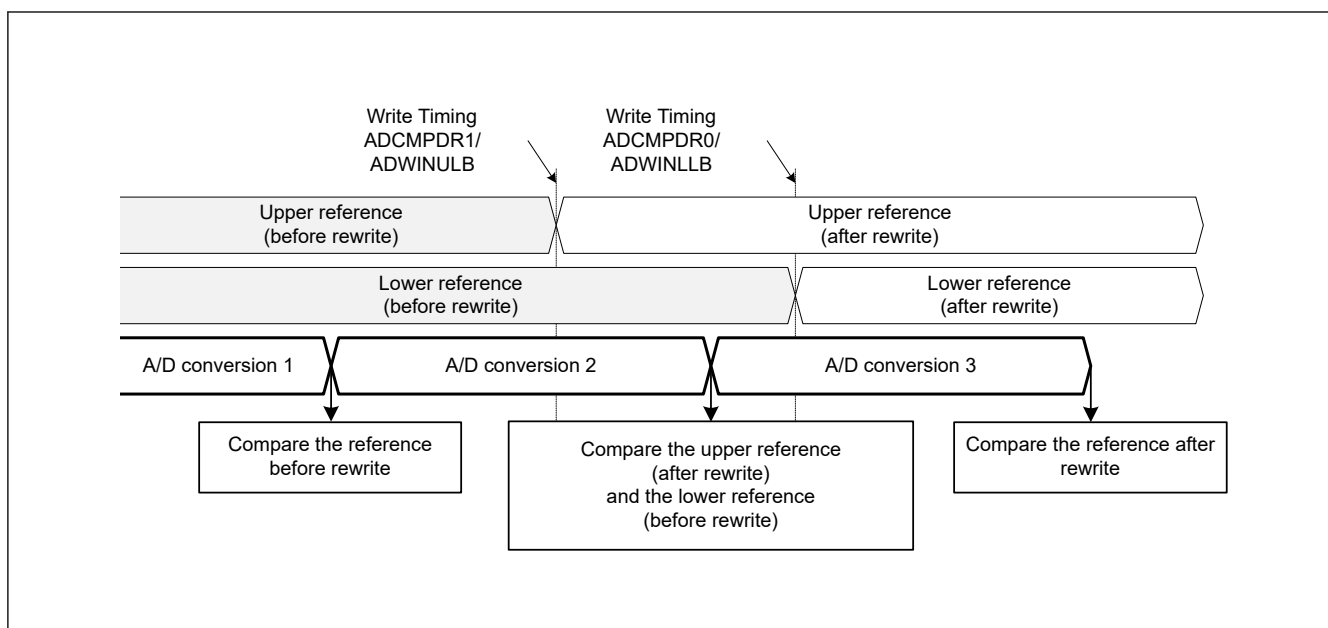


Figure 44.4 Comparison between upper and lower references before and after a rewrite

The ADCMPDRy registers use different formats depending on the following conditions:

- The value of A/D Data Register Format Select bit (flush-right or flush-left)
- The value of the A/D Conversion Accuracy Select bit (14-bit, 12-bit)
- The value of A/D-Converted Value Addition/Average Channel Select bits (A/D-converted value addition mode selected or not selected).

The data formats for each condition are shown as follows:

1. When A/D-converted value addition mode is not selected
 - Flush-right data with 14-bit accuracy — Lower 14 bits ([13:0]) are valid

- Flush-right data with 12-bit accuracy — Lower 12 bits ([11:0]) are valid
- Flush-left data with 14-bit accuracy — Upper 14 bits ([15:2]) are valid
- Flush-left data with 12-bit accuracy — Upper 12 bits ([15:4]) are valid

2. When A/D-converted value addition mode is selected

- Flush-right data with 14-bit accuracy — All bits ([15:0]) are valid
- Flush-right data with 12-bit accuracy — Lower 14 bits ([13:0]) are valid
- Flush-left data with 14-bit accuracy — All bits ([15:0]) are valid
- Flush-left data with 12-bit accuracy — Upper 14 bits ([15:2]) are valid

44.2.34 ADWINnLB : A/D Compare Function Window B Lower-Side/Upper-Side Level Setting Register (n = L, U)

Base address: ADC140 = 0x4005_C000

Offset address: 0x0A8 (n = L)
0x0AA (n = U)

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

The ADWINULB and ADWINLLB registers specify the reference data when the compare window B function is used. ADWINLLB sets the lower reference for window B, and ADWINULB sets the upper reference for window B.

ADWINnLB are read/write registers.

ADWINnLB are writable even during A/D conversion. The reference data can be dynamically changed by rewriting register values during A/D conversion^{*1}.

Set these registers so that the upper reference is not less than the lower reference (and $ADWINULB \geq ADWINLLB$), 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 is changed and the lower reference value is being changed, the MCU compares the upper reference (after rewrite), and the lower reference (before rewrite) with the A/D conversion result. See [Figure 44.5](#). If the comparison during the rewriting of these two references is erroneous, then rewrite these reference values when both ADCSR.ADST and the target Compare Window Operation Enable bit (ADCMPCR.CMPAE or ADCMPCR.CMPBE) are 0.

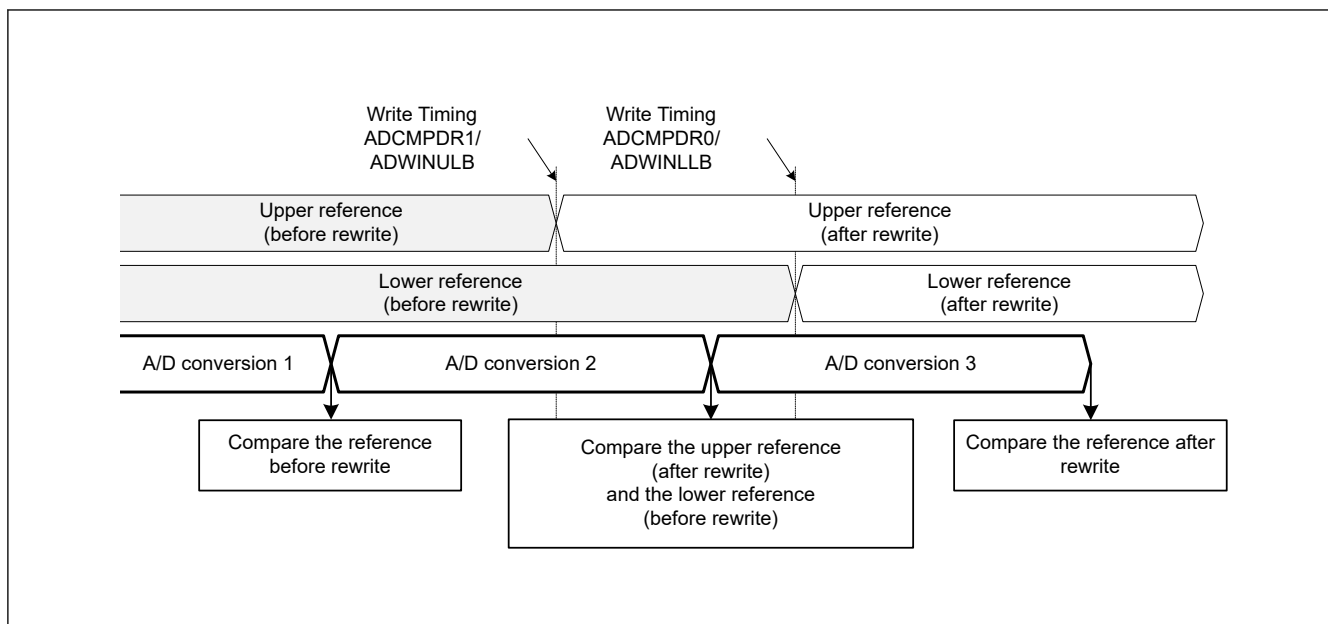


Figure 44.5 Comparison between upper and lower references before and after a rewrite

The ADWINnLB registers use different formats depending on the following conditions:

- The value of A/D Data Register Format Select bit (flush-right or flush-left)
- The value of the A/D Conversion Accuracy Select bit (14-bit, 12-bit)
- The value of A/D-Converted Value Addition/Average Channel Select bits (A/D-converted value addition mode selected or not selected).

The data formats for each condition are shown as follows:

1. When A/D-converted value addition mode is not selected
 - Flush-right data with 14-bit accuracy — Lower 14 bits ([13:0]) are valid
 - Flush-right data with 12-bit accuracy — Lower 12 bits ([11:0]) are valid
 - Flush-left data with 14-bit accuracy — Upper 14 bits ([15:2]) are valid
 - Flush-left data with 12-bit accuracy — Upper 12 bits ([15:4]) are valid
2. When A/D-converted value addition mode is selected
 - Flush-right data with 14-bit accuracy — All bits ([15:0]) are valid
 - Flush-right data with 12-bit accuracy — Lower 14 bits ([13:0]) are valid
 - Flush-left data with 14-bit accuracy — All bits ([15:0]) are valid
 - Flush-left data with 12-bit accuracy — Upper 14 bits ([15:2]) are valid

44.2.35 ADCMPSR0 : A/D Compare Function Window A Channel Status Register 0

Base address: ADC140 = 0x4005_C000

Offset address: 0x0A0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPS TCHA 15	CMPS TCHA 14	CMPS TCHA 13	CMPS TCHA 12	CMPS TCHA 11	CMPS TCHA 10	CMPS TCHA 09	CMPS TCHA 08	CMPS TCHA 07	CMPS TCHA 06	CMPS TCHA 05	CMPS TCHA 04	CMPS TCHA 03	CMPS TCHA 02	CMPS TCHA 01	CMPS TCHA 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPSTCHAN	Compare Window A Flag When Window A operation is enabled (ADCMPCR.CMPAE = 1b), these bits indicate the comparison result of channels to which Window A comparison conditions are applied. Bit 15 (CMPSTCHA15) is associated with AN015 and bit 0 (CMPSTCHA00) is associated with AN000. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W

Note: n = 00 to 07

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

CMPSTCHAN flags (Compare Window A Flag)

The CMPSTCHAN flags indicate the comparison results for channels to which Window A comparison conditions are applied. When a comparison condition set in ADCMPLR0.CMPLCHAN is met at the end of A/D conversion, the associated CMPSTCHAN flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC140_CMPAI) is generated when this flag sets to 1.

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.

44.2.36 ADCMPSR1 : A/D Compare Function Window A Channel Status Register1

Base address: ADC140 = 0x4005_C000

Offset address: 0x0A2

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPSTCHA 31	CMPSTCHA 30	CMPSTCHA 29	CMPSTCHA 28	CMPSTCHA 27	CMPSTCHA 26	CMPSTCHA 25	CMPSTCHA 24	CMPSTCHA 23	CMPSTCHA 22	CMPSTCHA 21	CMPSTCHA 20	CMPSTCHA 19	CMPSTCHA 18	CMPSTCHA 17	CMPSTCHA 16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPSTCHAN	Compare Window A Flag When Window A operation is enabled (ADCMPCR.CMPAE = 1), these bits indicate the comparison result of channels to which Window A comparison conditions are applied. Bit 15 (CMPSTCHA31) is associated with AN031 and bit 0 (CMPSTCHA16) is associated with AN016. Bit 15 (CMPSTCHA31) is associated with VSC_VCC. Bit 14 (CMPSTCHA30) is associated with AN030. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W

Note: n = 16, 17, 20, 21, 31

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

CMPSTCHAN flags (Compare Window A Flag)

The CMPSTCHAN flags indicate the comparison results for channels to which Window A comparison conditions are applied. When the comparison condition set in ADCMPLR1.CMPLCHAN is met at the end of A/D conversion, the associated CMPSTCHAN flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC140_CMPAI) is generated when this flag sets to 1.

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.

44.2.37 ADCMPSER : A/D Compare Function Window A Extended Input Channel Status Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x0A4

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

Bit	Symbol	Function	R/W
0	CMPSTTSA	Compare Window A Temperature Sensor Output Compare Flag When Window A operation is enabled (ADCMPCR.CMPAE = 1), this bit indicates the temperature sensor output comparison result. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

The ADCMPSER register stores compare results of compare function window A.

CMPSTTSA flag (Compare Window A Temperature Sensor Output Compare Flag)

The CMPSTTSA flag indicates the temperature sensor output comparison result. When the comparison condition set in ADCMPLER.CMPLTSA is met at the end of A/D conversion, this flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC140_CMPAI) is generated when this flag sets to 1.

Writing 1 to the CMPSTTSA flag is invalid.

[Setting condition]

- The condition set in ADCMPLER.CMPLTSA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

44.2.38 ADCMPBNSR : A/D Compare Function Window B Channel Select Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x0A6

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

Bit	Symbol	Function	R/W																												
5:0	CMPCHB[5:0]	Compare Window B Channel Select These bits select channels to be compared with the compare Window B conditions.	R/W																												
		<table><tr><th>CMPCHB[5:0]</th><th>Channel</th></tr><tr><td>0x00</td><td>AN000</td></tr><tr><td>0x01</td><td>AN001</td></tr><tr><td>0x02</td><td>AN002</td></tr><tr><td>⋮</td><td>⋮</td></tr><tr><td>0x07</td><td>AN007</td></tr><tr><td>0x10</td><td>AN016</td></tr><tr><td>0x11</td><td>AN017</td></tr><tr><td>0x14</td><td>AN020</td></tr><tr><td>0x15</td><td>AN021</td></tr><tr><td>0x1F</td><td>VSC_VCC</td></tr><tr><td>0x20</td><td>Temparature sensor</td></tr><tr><td>0x3F</td><td>No selection</td></tr><tr><td>Others</td><td>Setting prohibited</td></tr></table>		CMPCHB[5:0]	Channel	0x00	AN000	0x01	AN001	0x02	AN002	⋮	⋮	0x07	AN007	0x10	AN016	0x11	AN017	0x14	AN020	0x15	AN021	0x1F	VSC_VCC	0x20	Temparature sensor	0x3F	No selection	Others	Setting prohibited
		CMPCHB[5:0]		Channel																											
		0x00		AN000																											
		0x01		AN001																											
		0x02		AN002																											
		⋮		⋮																											
		0x07		AN007																											
		0x10		AN016																											
		0x11		AN017																											
		0x14		AN020																											
		0x15		AN021																											
		0x1F		VSC_VCC																											
		0x20		Temparature sensor																											
		0x3F		No selection																											
Others	Setting prohibited																														
6	—	This bit is read as 0. The write value should be 0.	R/W																												
7	CMPLB	Compare Window B Comparison Condition Setting This bit sets comparison conditions for channels for Window B. The comparison conditions are shown in Figure 44.6 . 0: When window function is disabled (ADCMPCR.WCMPE = 0): ADWINLLB value > A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADWINLLB value, or ADWINULB value < A/D-converted value 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADWINLLB value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): ADWINLLB value < A/D-converted value < ADWINULB value	R/W																												

CMPCHB[5:0] bits (Compare Window B Channel Select)

The CMPCHB[5:0] bits specify the channels to be compared with the compare Window B conditions from AN000 to AN007, AN016, AN017, AN020 to AN021, the temperature sensor, VSC_VCC pin voltage output. The compare Window B function is enabled by specifying the hexadecimal number of the A/D conversion channel selected in the ADANSA0, ADANSA1, ADANSB0, ADANSB1 registers.

Set the CMPCHB[5:0] bits while the ADCSR.ADST bit is 0.

CMPLB bit (Compare Window B Comparison Condition Setting)

The CMPLB bit specifies the comparison conditions for channels for Window B. When the comparison result of an analog input meets the set condition, the associated ADCMPBSR.CMPSTB flag sets to 1 and a compare interrupt request (ADC140_CMPBI) is generated.

Compare conditions when the window function is disabled			
CMPLB = 0		CMPLB = 1	
ADWINLLB value $\leq$ A/D converted value	Not met	ADWINLLB value $<$ A/D converted value	Met
ADWINLLB value $>$ A/D converted value	Met	ADWINLLB value $\geq$ 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 $\leq$ A/D converted value $\leq$ ADWINULB value		Not met	
A/D converted value $<$ ADWINLLB value		Met	
CMPLB = 1			
A/D converted value $\geq$ ADWINULB value		Not met	
ADWINLLB value $<$ A/D converted value $<$ ADWINULB value		Met	
A/D converted value $\leq$ ADWINLLB value		Not met	

Figure 44.6 Explanation of compare conditions for compare function Window B

44.2.39 ADCMPBSR : A/D Compare Function Window B Status Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x0AC

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

Bit	Symbol	Function	R/W
0	CMPSTB	Compare Window B Flag When Window B operation is enabled (ADCMPCR.CMPBE = 1), this bit indicates the comparison result of channels to which Window B comparison conditions are applied, temperature sensor output, VSC_VCC pin voltage output. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

CMPSTB flag (Compare Window B Flag)

The CMPSTB flag indicates the comparison result of channels to which Window B comparison conditions are applied, the temperature sensor output, VSC_VCC pin voltage output. When the comparison condition set in ADCMPBNSR.CMPLB is

met at the end of A/D conversion, this flag sets to 1. When the ADCMPCR.CMPBIE bit is 1, a compare interrupt request (ADC140_CMPBI) is generated when this flag sets to 1.

Writing 1 to the CMPSTB flag is invalid.

[Setting condition]

- The condition set in ADCMPBNSR.CMPLB is met when ADCMPCR.CMPBE = 1.

[Clearing condition]

- Writing 0 after reading 1.

44.2.40 ADWINMON : A/D Compare Function Window A/B Status Monitor Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x08C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MONC MPB	MONC MPA	—	—	—	MONC OMB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MONCOMB	Combination Result Monitor This bit indicates the combination result. This bit is valid when both Window A and Window B operations are enabled. 0: Window A/B composite conditions are not met. 1: Window A/B composite conditions are met.	R
3:1	—	These bits are read as 0.	R
4	MONCMPA	Comparison Result Monitor A 0: Window A comparison conditions are not met. 1: Window A comparison conditions are met.	R
5	MONCMPB	Comparison Result Monitor B 0: Window B comparison conditions are not met. 1: Window B comparison conditions are met.	R
7:6	—	These bits are read as 0.	R

MONCOMB bit (Combination Result Monitor)

The read-only MONCOMB bit indicates the combined result of comparison condition results A and B based on the combination condition set in the ADCMPCR.CMPAB[1:0] bits.

[Setting condition]

- The combined result meets the combination condition set in the ADCMPCR.CMPAB[1:0] bits when ADCMPCR.CMPAE = 1 and ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The combined result does not meet the combination condition set in the ADCMPCR.CMPAB[1:0] bits.
- ADCMPCR.CMPAE = 0 or ADCMPCR.CMPBE = 0.

MONCMPA bit (Comparison Result Monitor A)

The read-only MONCMPA bit is read as 1 when the A/D-converted value of the Window A target channel meets the condition set in ADCMPLR0/ADCMPLR1 and ADCMPLE. Otherwise, it is read as 0.

[Setting condition]

- The A/D-converted value meets the condition set in the ADCMPLR0/ADCMPLR1 and ADCMPLE 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).

44.2.41 ADHVREFCNT : A/D High-Potential/Low-Potential Reference Voltage Control Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x08A

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

Bit	Symbol	Function	R/W
1:0	HVSEL[1:0]	High-Potential Reference Voltage Select 0 0: AVCC0 is selected as the high-potential reference voltage 0 1: VREFH0 is selected as the high-potential reference voltage 1 0: No reference voltage pin is selected 1 1: No reference voltage pin is selected (internal node discharge)	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	LVSEL	Low-Potential Reference Voltage Select 0: AVSS0 is selected as the low-potential reference voltage. 1: VREFL0 is selected as the low-potential reference voltage.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

HVSEL[1:0] bits (High-Potential Reference Voltage Select)

The HVSEL[1:0] bits specify the high-potential reference voltage as AVCC0, VREFH0.

When setting the register, make sure that HVSEL[1:0] = 11b is set.

When VREFH0 is selected, setting the ADCER.DIAGVAL[1:0] bits to 10b or 11b for self-diagnosis is prohibited.

LVSEL bit (Low-Potential Reference Voltage Select)

The LVSEL bit specifies the low-potential reference voltage as AVSS0 or VREFL0.

44.2.42 ADEDCRm : A/D Emulator Debug Function Control Register m (m = 0, 1, 4, 5)

Base address: ADC140 = 0x4005_C000

Offset address: 0x170

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	EDANC3[1:0]	—	—	EDANC2[1:0]	—	—	EDANC1[1:0]	—	—	EDANC0[1:0]	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Base address: ADC140 = 0x4005_C000

Offset address: 0x172

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	EDANC7[1:0]	—	—	EDANC6[1:0]	—	—	EDANC5[1:0]	—	—	EDANC4[1:0]	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Base address: ADC140 = 0x4005_C000

Offset address: 0x178

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	EDANC17[1:0]	—	—	EDANC16[1:0]	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Base address: ADC140 = 0x4005_C000

Offset address: 0x17A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	EDANC21[1:0]	—	—	EDANC20[1:0]	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Base address: ADC140 = 0x4005_C000

Offset address: 0x17E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	EDANC31[1:0]	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13, 12, 9, 8, 5, 4, 1, 0	EDANCn[1:0]	Emulator Debug Function Control 0 0: Disable the emulator debug function 0 1: Replace the value of the ADEDDMY0.EDDMY0 dummy data register with the A/D conversion result 1 0: Replace the value of the ADEDDMY1.EDDMY1 dummy data register with the A/D conversion result 1 1: Setting prohibited.	R/W
15, 14*1, 11, 10, 7, 6, 3, 2	—	These bits are read as 0. The write value should be 0.	R/W

Note: n = 0 to 7, 16, 17, 20, 21, 31

Note 1. In the ADEDCR5 register, bits 13, 12, 9, 8 are also reserved bits. In the ADEDCR7 register, bits 9, 8, 5, 4, 1, 0 are also reserved bits.

The ADEDCRm register sets the emulator debug function for the high-precision analog input channels and standard-precision analog input channels.

EDANCn[1:0] bit (Emulator Debug Function Control)

These bits are used to control the emulator debug function. Setting the EDANCn[1:0] bits to 01b or 10b enables emulator debug mode. The EDANCn[1:0] bits correspond to analog input channels, and can be set for each channel.

When emulator debug mode is enabled, the A/D conversion result is replaced with the selected dummy data register value. The replaced data is stored in the data register according to the specified data format, in the same way as for normal A/D conversion.

For channels for which the addition/average function takes effect, the values of these bits are added and the mean value is obtained.

If the digital comparison function is enabled, comparison with the values of these bits is performed.

44.2.43 ADEDEXCR : A/D Emulator Debug Function Extension Control Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x180

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

Bit	Symbol	Function	R/W
1:0	EDDIAGC[1:0]	Emulator Debug Function Self-Diagnosis Control These bits set the emulator debug function for A/D conversion performed in self-diagnosis. 0 0: Disable the emulator debug function 0 1: Replace the EDDMY0 value of the Dummy Data Register (ADEDDMY0) with the A/D conversion result 1 0: Replace the EDDMY1 value of the Dummy Data Register (ADEDDMY1) with the A/D conversion result 1 1: Setting prohibited.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
5:4	EDTSC[1:0]	Emulator Debug Function Temperature Sensor Output Control These bits set the emulator debug function for A/D conversion of the temperature sensor output. 0 0: Disable the emulator debug function 0 1: Replace the EDDMY0 value of the Dummy Data Register (ADEDDMY0) with the A/D conversion result 1 0: Replace the EDDMY1 value of the Dummy Data Register (ADEDDMY1) with the A/D conversion result 1 1: Setting prohibited.	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W

The ADEDEXCR register sets the emulator debug function for self-diagnosis and temperature sensor output.

EDDIAGC[1:0] bit (Emulator Debug Function Self-Diagnosis Control)

These bits are used to control the emulator debug function for A/D conversion performed in self-diagnosis. Setting the EDDIAGC[1:0] bits to 01b or 10b enables emulator debug mode.

For details about the operation, see the explanation of the ADEDCRm.EDANCn[1:0] bits.

EDTSC[1:0] bit (Emulator Debug Function Temperature Sensor Output Control)

These bits are used to control the emulator debug function for A/D conversion of the temperature sensor output. Setting the EDDIAGC[1:0] bits to 01b or 10b enables emulator debug mode.

For details about the operation, see the explanation of the ADEDCRm.EDANCn[1:0] bits.

44.2.44 AEEDDMY0 : A/D Emulator Debug Function Dummy Data Setting Register 0

Base address: ADC140 = 0x4005_C000

Offset address: 0x182

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	EDDMY0[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	EDDMY0[13:0]	Dummy Data Setting 0 These bits set the data value of A/D conversion result that is to be replaced when the emulator debug function is enabled.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

The AEEDDMY0 register sets the A/D conversion result value that is to be replaced when the emulator debug function is enabled.

EDDMY0[13:0] bit (Dummy Data Setting 0)

When the emulator debug function is enabled, the A/D conversion result is replaced with the dummy data value set in EDDMY0[13:0].

The bits to be used for replacement with the conversion result vary depending on the setting in ADCER.ADPRC[1:0] (A/D Conversion Accuracy Select bits). The following shows the relationship between the values set in ADCER.ADPRC[1:0] and the range of data used for replacement.

■ When 14-bit accuracy is selected (ADCER.ADPRC[1:0] = 11b)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	EDDMY0[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

■ When 12-bit accuracy is selected (ADCER.ADPRC[1:0] = 00b)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	EDDMY0[13:2]												—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

When 12-bit accuracy is selected, values set for unused bits are disabled.

44.2.45 AEEDDMY1 : A/D Emulator Debug Function Dummy Data Setting Register 1

Base address: ADC140 = 0x4005_C000

Offset address: 0x184

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	EDDMY1[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	EDDMY1[13:0]	Dummy Data Setting 1 These bits set the data value of A/D conversion result that is to be replaced when the emulator debug function is enabled.	R/W

Bit	Symbol	Function	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

The ADEDDMY1 register sets the A/D conversion result value that is to be replaced when the emulator debug function is enabled.

EDDMY1[13:0] bit (Dummy Data Setting 1)

When the emulator debug function is enabled, the A/D conversion result is replaced with the dummy data value set in EDDMY1[13:0].

For details about the operation, see the explanation of the bits of the ADEDDMY0 register.

When 12-bit resolution is selected, values set for unused bits are disabled.

44.2.46 ADCLKCR : A/D Sub-Clock Mode Control Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x064

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

Bit	Symbol	Function	R/W
0	SCLKEN	Sub-Clock Mode Setting 0: When the ADCLK frequency is set to 1 to 32 MHz 1: When the ADCLK frequency is set to 32.768 kHz	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

The ADCLKCR register sets the operation at low-speed.

SCLKEN bit (Sub-Clock Mode Setting)

Set the SCLKEN bit when using a low-speed clock as the A/D conversion clock. The target frequency is 32.768 kHz, and this applies when the A/D conversion clock is set to the SOSC clock, LOCO clock, MOCO clock divided by 64. [Table 44.22](#) shows the relationship between the operating frequency during A/D conversion and the set value of this bit. Note that any frequency within the guaranteed range indicated by the clock timing in [section 51, Electrical Characteristics](#) can be used as the A/D conversion clock.

Table 44.22 ADCLK frequency and SCLKEN setting value

ADCLK frequency*1	setting value
1 to 32 MHz	SCLKEN = 0
32.768 kHz	SCLKEN = 1

Note 1. A/D conversion other than the specified frequency is prohibited.

44.2.47 ADCALC : A/D Calibration Control Register

Base address: ADC140 = 0x4005_C000

Offset address: 0x140

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

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	—	This bit is read as 1. The write value should be 1.	R/W
7	CALST	Offset Calibration This bit specifies whether to perform offset calibration. 0: Disable offset calibration when A/D conversion starts the next time 1: Enable offset calibration when A/D conversion starts the next time.	R/W

CALST bit (Offset Calibration)

This bit specifies whether to perform offset calibration. When 1 is written to the ADCSR.ADST bit with the CALST bit set to 1, offset calibration is performed but A/D conversion is not. When offset calibration completes, the CALST bit and ADCSR.ADST bit are set to 0. If offset calibration is performed with the ADCSR.ADIE (Scan End Interrupt Enable) bit set to 1, a scan end interrupt (ADC140_ADI) is generated upon completion of offset calibration.

Set the ADCALC register while the ADCSR.ADST bit is 0.

To perform offset calibration, make sure that the ADADC.ADC[2:0] bits are set to 101b and the ADSSTR0.SST[7:0] bits are set to FFh. If a offset calibration end interrupt is required, set the ADCSR.ADIE bit to 1. Set any other registers to the default values. To perform offset calibration, follow the procedure in [Figure 44.39](#).

To stop offset calibration in the way, set the ADCSR.ADST bit to 0 according to the procedure in [Figure 44.40](#). If offset calibration is stopped in the way, the ADCALC.CALST bit remains 1 and the offset correction value is not updated.

44.3 Operation

44.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.

Scan conversion is performed in any of the three operating modes:

- 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 ADCSR.ADST bit to 0. In group scan mode, the selected channels in group A, B, and C are scanned once after scan starts in response to the respective synchronous trigger or asynchronous trigger (group A only).

In single scan mode and continuous scan mode, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel set in the ADSCSn register. In group scan mode, A/D conversion is performed for the ANn channels in group A selected in the ADANSA0 and ADANSA1 registers, and for the ANn channels in group B selected in the ADANSB0 and ADANSB1 registers and for the ANn channels in group C selected in the ADANSC0 and ADANSC1 registers, starting from the channel set in the ADSCSn register.

When self-diagnosis is selected, it is executed once at the beginning of each scan and one of the three reference voltages is converted.

Double trigger mode can be used with single scan mode or group scan mode. With double trigger mode enabled (ADCSR.DBLE = 1), A/D conversion data of a channel selected in the ADCSR.DBLANS[4:0] bits is duplicated only if the conversion is started by the synchronous trigger (TMR, ELC) selected in the ADSTRGR.TRSA[5:0] bits. In group scan mode, only group A can use double trigger mode.

44.3.2 Single Scan Mode

44.3.2.1 Basic Operation

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (TMR, ELC), or an asynchronous trigger input, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel set in the ADSCSn register.
2. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D data register (ADDRy).
3. When A/D conversion of all the selected channels completes, an ADC140_ADI interrupt request is generated.
4. The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically set to 0 when A/D conversion of all the selected channels completes. The ADC14 then enters a wait state.

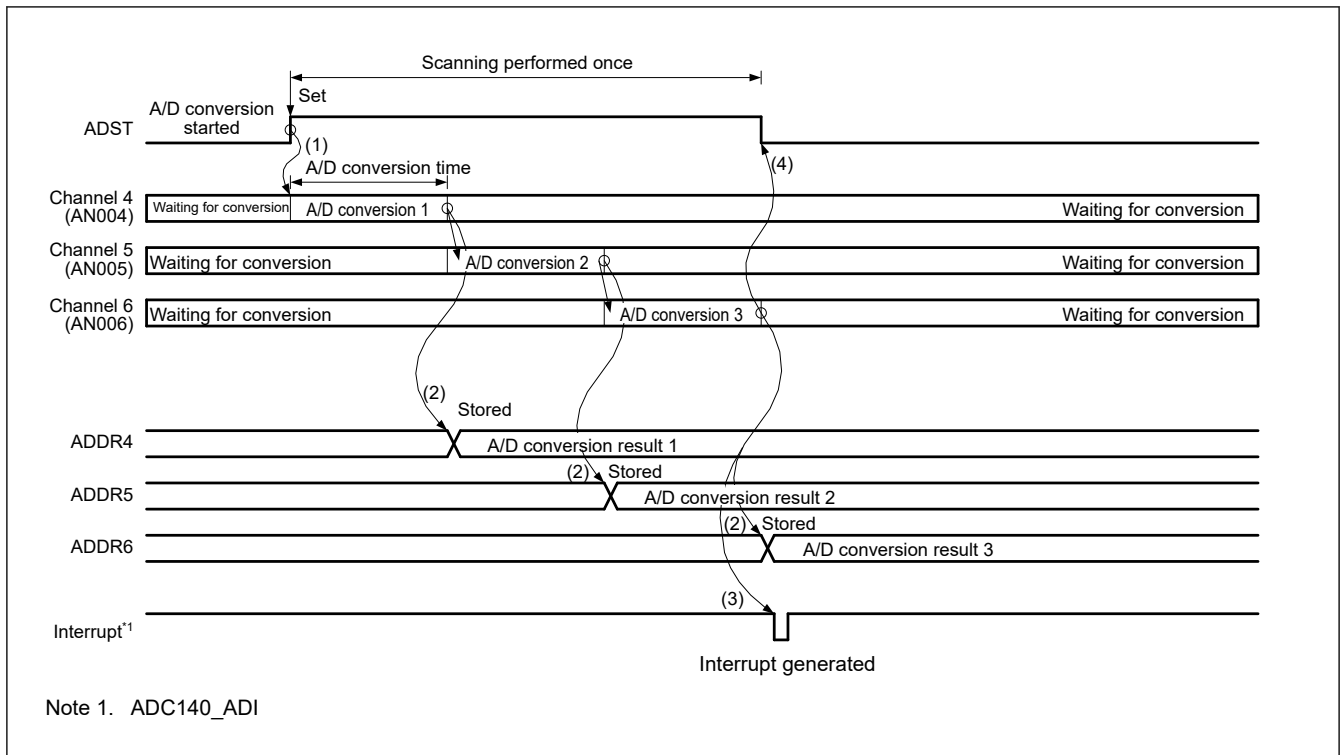


Figure 44.7 Example basic operation in single scan mode when AN004 to AN006 are selected

44.3.2.2 Channel Selection and Self-Diagnosis

When channels and self-diagnosis are selected, A/D conversion is first performed for the reference voltage VREFH0 (x0, x1/2, or x1), then A/D conversion is performed once on the analog input of the selected channels as follows:

1. A/D conversion for self-diagnosis is first started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (TMR, ELC), or an asynchronous trigger input.
2. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel set in the ADSCSn register.
3. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D data register (ADDRy).
4. When A/D conversion of all the selected channels completes, an ADC140_ADI interrupt request is generated.
5. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion of all the selected channels completes. The ADC14 then enters a wait state.

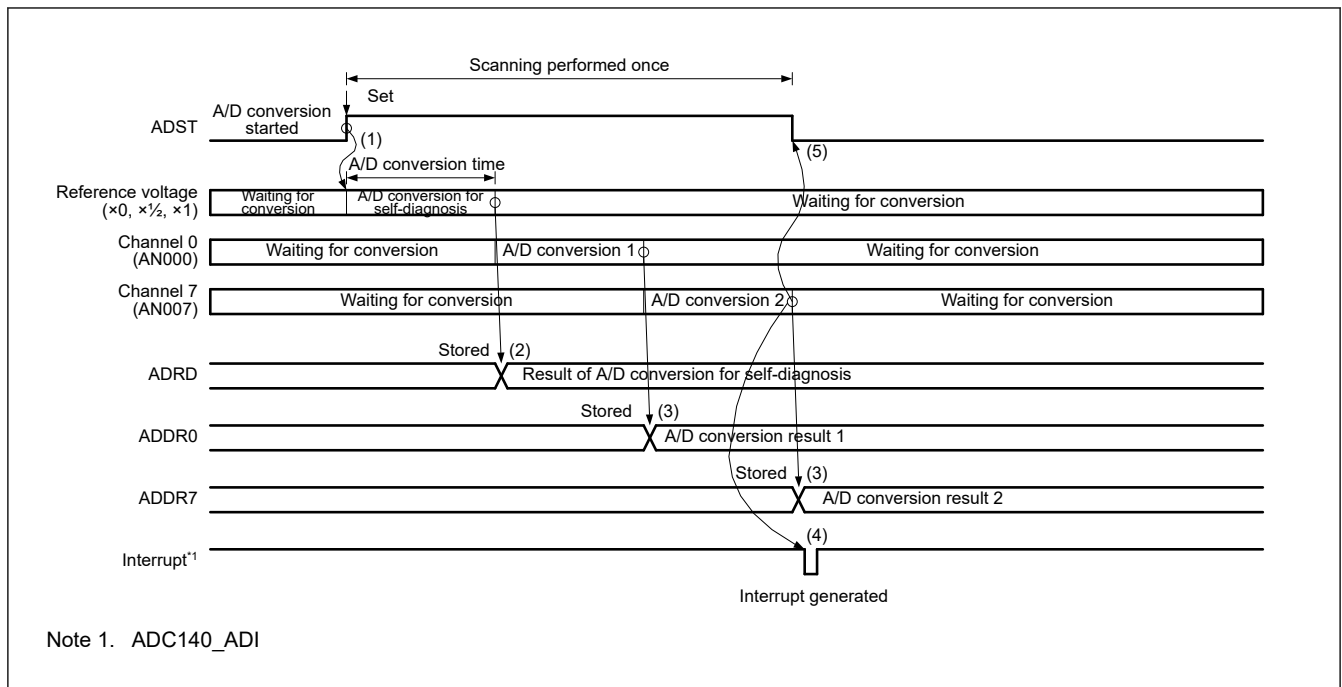


Figure 44.8 Example basic operation in single scan mode when AN000 and AN007 are selected with self-diagnosis

44.3.2.3 A/D Conversion of Temperature Sensor Output

When the channels and temperature sensor output are selected at the same time, A/D conversion is first performed on the analog inputs of the selected channels, and then A/D conversion is performed once on the temperature sensor output as shown below.

With the channels deselected, selecting only the temperature sensor output is also possible.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (starting A/D conversion) by software, a synchronous trigger (TMR or ELC), or an asynchronous trigger, A/D conversion for the analog input channels selected in the ADANSA0 or ADANSA1 register starts according to the conversion order set in the ADSCSn register.
2. On completion of A/D conversion on the channels, the result is stored in the corresponding A/D Data Register y (ADDRy), and then A/D conversion of temperature sensor output starts.
3. On completion of A/D conversion of temperature sensor output, the result is stored in the corresponding A/D Temperature Sensor Data Register (ADTSDR). If the ADCSR.ADIE bit is set to 1 (enabling ADC140_ADI interrupt generation upon completion of scanning), an ADC140_ADI interrupt request is generated.
4. The ADCSR.ADST bit remains 1 (starting A/D conversion) during A/D conversion, and is automatically cleared to 0 upon completion of A/D conversion. Then the 14-bit A/D converter enters a waiting state.

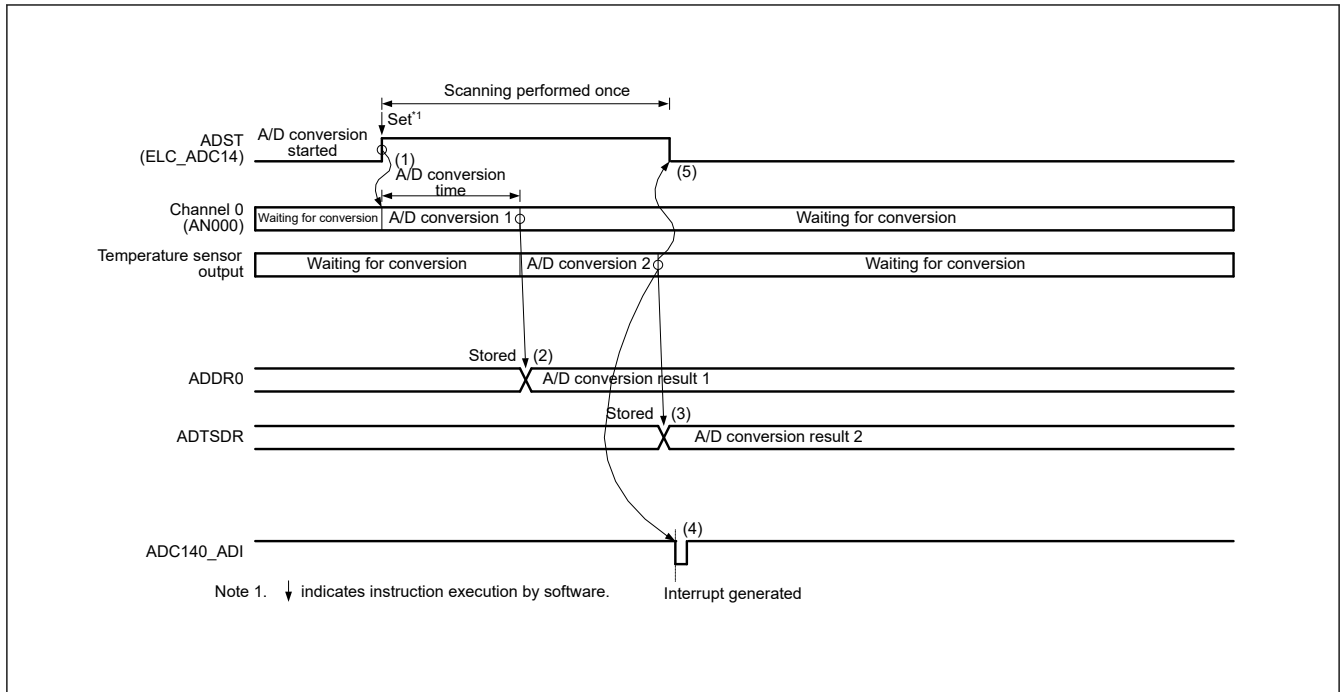


Figure 44.9 Example of operation in single-scan mode (basic operation: AN000 (high precision) and temperature)

44.3.2.4 A/D conversion in double-trigger mode

When double trigger mode is selected in single scan mode, two rounds of single scan operation started by a synchronous trigger (TMR, ELC) are performed in sequence.

Deselect self-diagnosis and set the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) to 0.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated in the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid.

In double trigger mode, select a synchronous trigger (TMR, ELC) with the ADSTRGR.TRSA[5:0] bits. Additionally, set the ADCSR.EXTRG bit to 0 and the ADCSR.TRGE bit to 1. Do not use a software trigger.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a synchronous trigger input (TMR, ELC), A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
2. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy and ADCTDR).
3. The ADCSR.ADST bit is automatically set to 0 and the ADC14 enters a wait state. An ADC140_ADI interrupt request is not generated.
4. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger input, A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
5. When A/D conversion completes, the result is stored in the A/D Data Duplexing Register (ADDBLDR), which is exclusively used in double-trigger mode.
6. An ADC140_ADI interrupt request is generated if the setting of the ADCSR.ADIE bit
7. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically set to 0 when A/D conversion completes. Then the ADC14 enters a wait state.

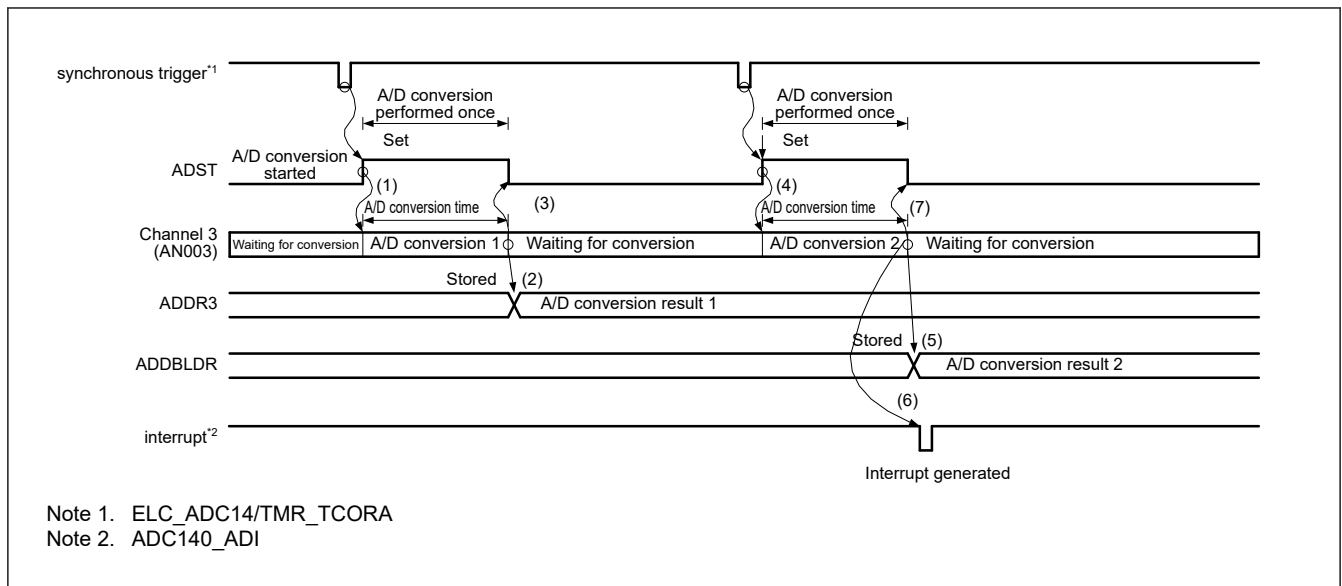


Figure 44.10 Example operation in single scan mode when double-trigger mode is selected and AN003 is duplicated

44.3.3 Continuous Scan Mode

44.3.3.1 Basic Operation

In continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (TMR, ELC), or an asynchronous trigger input, A/D conversion is performed for ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel set in the ADSCSn register.
2. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register (ADDRy).
3. When A/D conversion of all the selected channels completes, an ADC140_ADI interrupt request is generated. The ADC14 sequentially starts A/D conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel set in the ADSCSn register.
4. The ADCSR.ADST bit is not automatically cleared, and steps 2. and 3. are repeated as long as ADCSR.ADST remains 1 (A/D conversion start). When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC14 enters a wait state.
5. When the ADCSR.ADST bit is later set to 1 (A/D conversion start), A/D conversion starts again for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel set in the ADSCSn register.

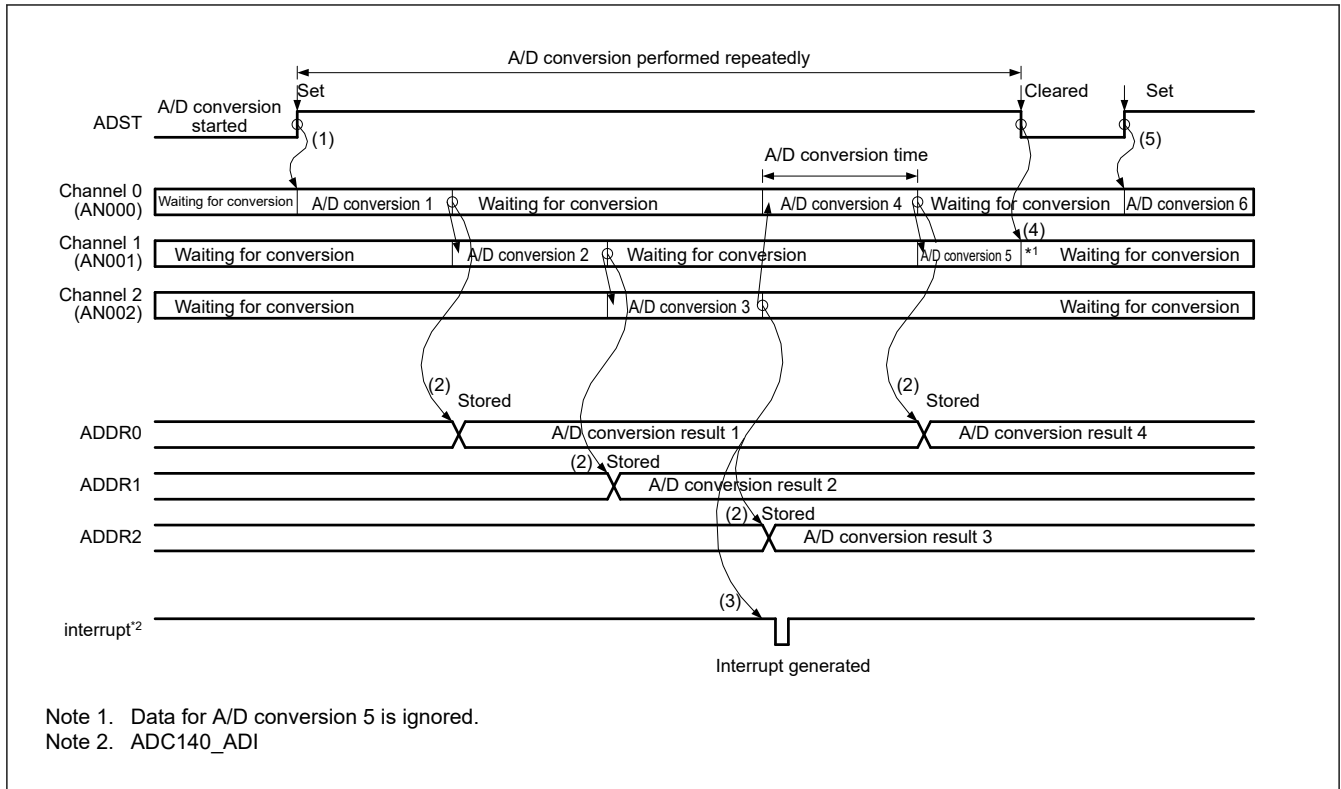


Figure 44.11 Example basic operation in continuous scan mode when AN000 to AN002 are selected

44.3.3.2 Channel Selection and Self-Diagnosis

When channels and self-diagnosis are selected at the same time, A/D conversion is first performed for the reference voltage VREFH0 ($\times 0$, $\times 1/2$, or $\times 1$) supplied to the ADC14, and A/D conversion is performed on the analog input of the selected channels. This sequence is repeated as described in the section that follows.

The operation is as follows:

1. A/D conversion for self-diagnosis is first started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (TMR, ELC), or an asynchronous trigger input.
2. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADDRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel set in the ADSCSn register.
3. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the corresponding A/D Data Register (ADDRy).
4. When A/D conversion of all the selected channels completes, an ADC140_ADI interrupt request is generated. At the same time, the ADC14 starts A/D conversion for self-diagnosis and then on the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel set in the ADSCSn register.
5. The ADCSR.ADST bit is not automatically cleared, and steps 2. to 4. are repeated as long as the ADCSR.ADST bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC14 enters a wait state.
6. When the ADST bit is later set to 1 (A/D conversion start), the A/D conversion for self-diagnosis is started again.

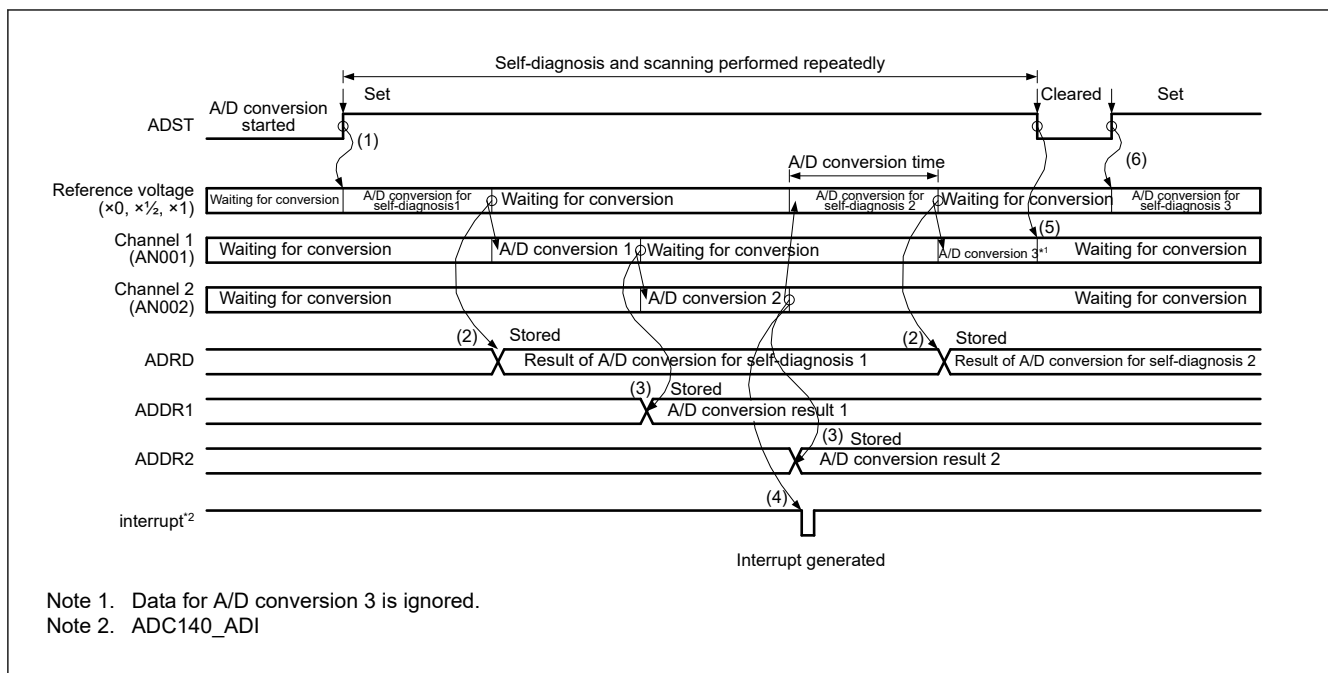


Figure 44.12 Example basic operation in continuous scan mode when AN001 and AN002 selected with self-diagnosis

44.3.3.3 A/D conversion of temperature sensor output

When the channels and temperature sensor output are selected at the same time, A/D conversion is first performed on the analog input of the selected channels, and then the A/D conversion of the temperature sensor output is repeated.

With the channels deselected, selecting only the temperature sensor output is also possible.

The operation is as follows:

1. When a software trigger, synchronous trigger (TMR, ELC), or asynchronous trigger sets the ADCSR.ADST bit to 1 (A/D conversion start), A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
2. On completion of A/D conversion on the channels, the result is stored in the associated A/D Data Register y (ADDRy), and then A/D conversion of temperature sensor output starts.
3. On completion of A/D conversion of the temperature sensor output, the result is stored in the associated A/D Temperature Sensor Data Register (ADTSDR).
4. On completion of A/D conversion and an ADC140_ADI interrupt request is generated. In addition, the ADC14 continuously starts A/D conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the lowest number n.
5. The ADCSR.ADST bit is not cleared automatically, and steps 2 to 4 are repeated as long as this bit remains set to 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC14 enters a wait state.
6. When the ADCSR.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 lowest number n.

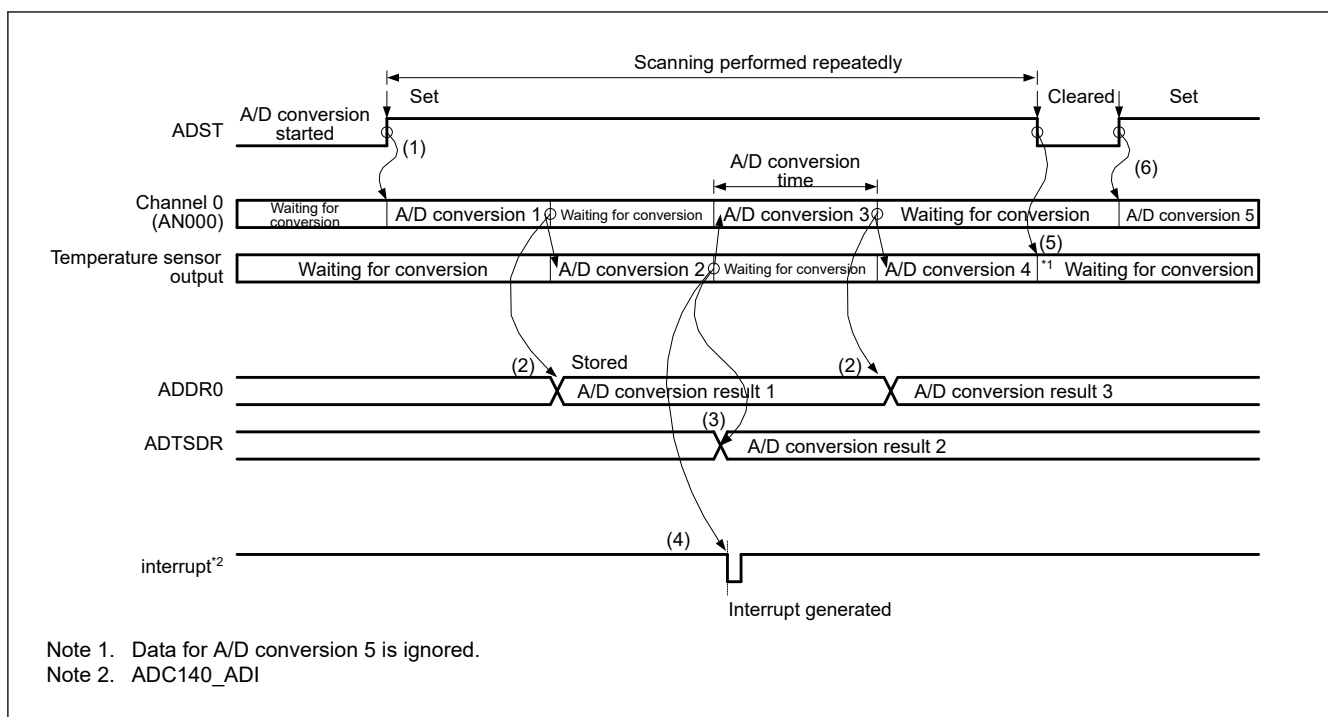


Figure 44.13 Example basic operation in continuous scan mode when AN000 and temperature sensor output are selected

44.3.4 Group Scan Mode

44.3.4.1 Basic Operation

The number of groups to be used in group-scan mode is selected from two (groups A and B) and three (groups A, B, and C). As the basic operation in group-scan mode, A/D conversion is performed once on the analog inputs of all the specified channels selected in groups A and B or groups A, B, and C after scanning is started by the respective synchronous trigger. The scan operation of each group is similar to the scan operation in single-scan mode.

In group-scan mode, the synchronous triggers of groups A, B, and C can be selected by using the ADSTRGR.TRSA[5:0], ADSTRGR.TRSB[5:0], and ADGCTRGR.TRSC[5:0] bits, respectively.

The channels to be scanned must be selected in the registers shown in [Table 44.23](#).

When self-diagnosis is selected in group-scan mode, self-diagnosis is separately performed for each of groups A and B or groups A, B, and C.

Table 44.23 Registers for selecting channels to be scanned

Scan group	Targets for A/D conversion	
	Analog input	Temperature sensor output
Group A	ADANSA0/1 register	ADEXICR.TSSA bit
Group B	ADANSB0/1 register	ADEXICR.TSSB bit
Group C	ADANSC0/1 register	ADGCXCR.TSSC bit

The following describes an example operation in group-scan mode. Specifically, conversion of group A is to be started by an asynchronous trigger, conversion of group B is to be started by an event signal (ELC_ADC14) from the ELC, and conversion of group C is to be started by the TMR_TCORA trigger from the TMR. (In the example operation, the event signal set in ELC.ELSR8 is TMR compare match B.)

1. Scanning of group A is started by an asynchronous trigger.
2. When group A scanning completes, an ADC140_ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (enabling ADC140_ADI interrupt generation upon completion of scanning).
3. Scanning of group B is started by input of the event trigger set in the ELC.ELSR8 register.

4. When group B scanning completes, an ADC140_GBADI interrupt is output if the setting of the ADCSR.GBADIE bit is 1 (enabling ADC140_GBADI interrupt generation).
5. Scanning of group C is started by input of the TCORA compare match trigger from the TMR.
6. When group C scanning completes, an ADC140_GCADI interrupt is output if the setting of the ADGCTRGR.GCADIE bit is 1 (enabling ADC140_GCADI interrupt generation).

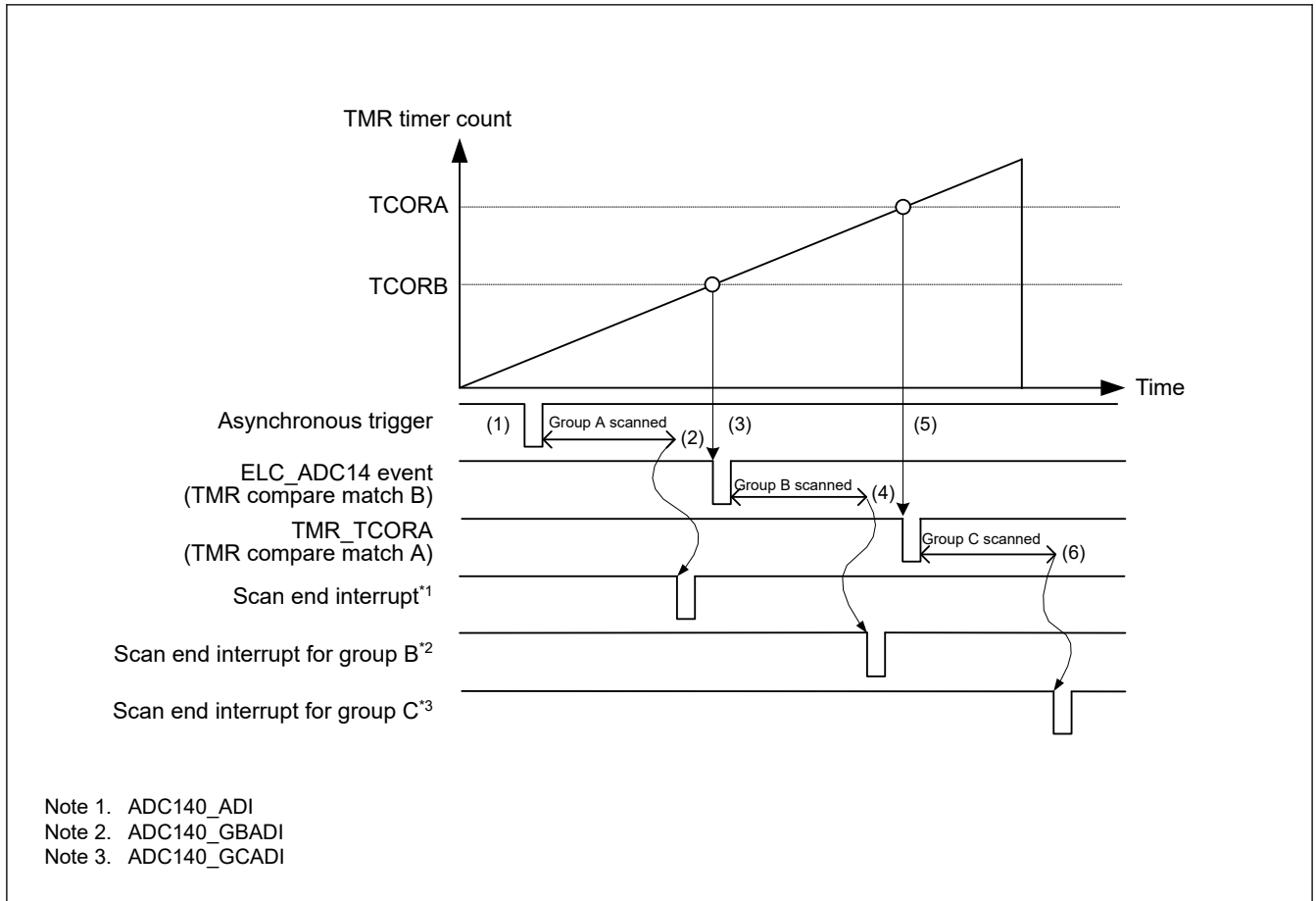


Figure 44.14 Example of operation in group-scan mode

44.3.4.2 A/D Conversion in Double-Trigger Mode

When double-trigger mode is selected in group-scan mode, two rounds of single scan operation started by an asynchronous trigger or a synchronous trigger (TMR or ELC) are performed in sequence for group A. Scan operations in groups B and C are the same as operations in single-scan mode started by a synchronous trigger (TMR or ELC).

In group-scan mode, the asynchronous or synchronous trigger of group A is selected by using the ADSTRGR.TRSA[5:0] bit, and the synchronous triggers of groups B and C are selected by using the ADSTRGR.TRSB[5:0] and ADGCTRGR.TRSC[5:0] bits, respectively. The different triggers should be used for groups A, B, and C. A software trigger should not be used.

The group A channels to be A/D-converted are selected by using the ADCSR.DBLANS[4:0] bits. The group B channels to be A/D-converted are selected by using the ADANSB0 and ADANSB1 registers. The group C channels to be A/D converted are selected by using the ADANSC0 and ADANSC1 registers.

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 number to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1.

The following describes an example operation when group-scan mode and double-trigger mode are set. Specifically, conversion of group A is to be started by an asynchronous trigger, conversion of group B is to be started by an event signal

(ELC_ADC14) trigger from the ELC, and conversion of group C is to be started by the TMR_TCORA trigger from the TMR. (In the example operation, the event signal set in ELC.ELSR8 is TMR compare match B.)

1. Scanning of group B is started by input of the event trigger set in the ELC.ELSR8 register.
2. When group B scanning completes, an ADC140_GBADI interrupt is output if the setting of the ADCSR.GBADI bit is 1 (enabling ADC140_GBADI interrupt generation).
3. Scanning of group C is started by input of the TCORA compare match trigger from the TMR.
4. When group C scanning completes, an ADC140_GCADI interrupt is output if the setting of the ADGCTRGR.GCADIE bit is 1 (enabling ADC140_GCADI interrupt generation).
5. The first scanning of group A is started by the first asynchronous trigger.
6. When the first scanning of group A completes, the converted data is stored in ADDRy. An ADC140_ADI interrupt request is not generated irrespective of the ADIE bit setting in ADCSR.
7. The second scanning of group A is started by the second asynchronous trigger.
8. When the second scanning of group A completes, the conversion result is stored in ADDBLDR. An ADC140_ADI interrupt is generated if the ADIE bit is 1 (enabling ADC140_ADI interrupt generation upon completion of scanning).
9. Scanning of group B is started by the second input of the event trigger set in the ELC.ELSR8 register.
10. When the second scanning of group B completes, an ADC140_GBADI interrupt is output if the ADCSR.GBADI bit is 1 (enabling ADC140_GBADI interrupt generation).
11. The second scanning of group C is started by the second input of the TCORA compare match trigger from the TMR.
12. When the second scanning of group C completes, an ADC140_GCADI interrupt is output if the ADGCTRGR.GCADIE bit is 1 (enabling ADC140_GCADI interrupt generation).

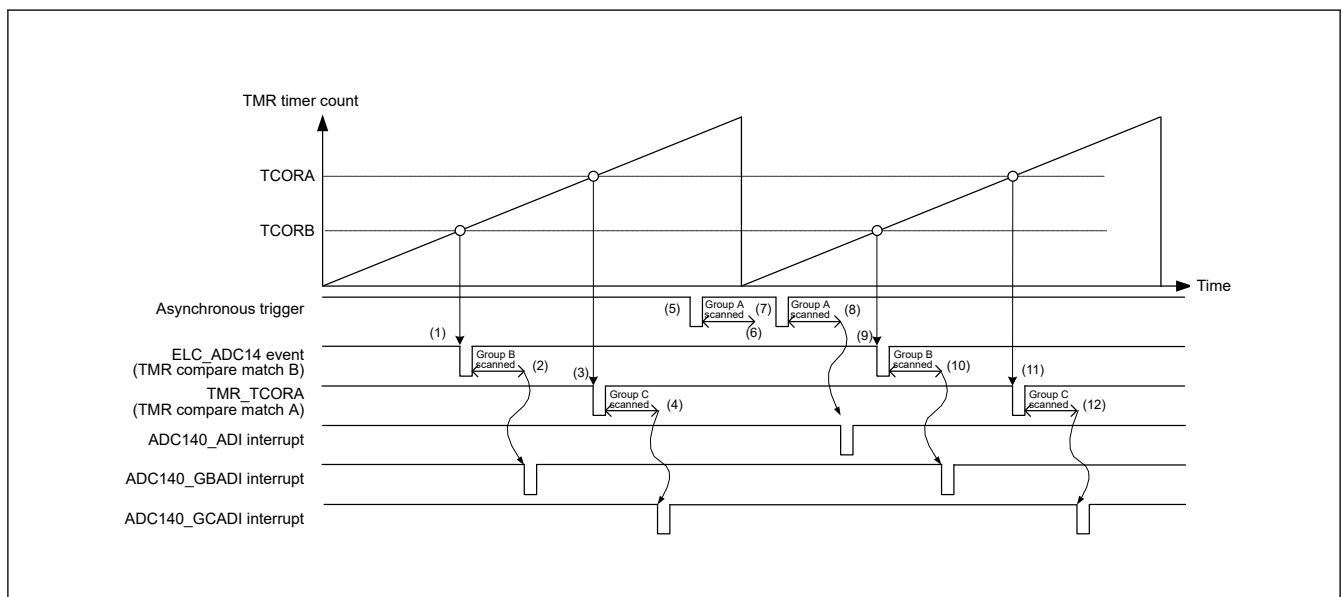


Figure 44.15 Example operation in group-scan mode with double-trigger mode

44.3.4.3 Group Priority Operation

Group priority operation is performed by setting the ADGSPCR.PGS bit to 1 in group-scan mode. The priority of groups is group A > group B > group C. The number of groups to be used in group-scan mode is selected from two (groups A and B) and three (groups A, B, and C) by setting ADGCTRGR.GRCE.

When setting the PGS bit in the ADGSPCR register to 1, follow the procedure described in [Figure 44.16](#). If the procedure is not followed, A/D conversion operation and stored data are not guaranteed.

As the basic operation in group-scan mode, a trigger input generated during A/D conversion of group A, B, or C is ignored, and the A/D conversion operation of each group is similar to the operation in single-scan mode.

In group priority operation, if a trigger for a priority group is input during scanning of a lower-priority group, A/D conversion for the lower-priority group is stopped and A/D conversion for the priority group is performed.

If the setting of the ADGSPCR.GBRSCN bit is 0, the lower-priority group enters a wait state when A/D conversion for the priority group completes. A trigger input of the lower-priority group generated during A/D conversion is ignored.

If the setting of the ADGSPCR.GBRSCN bit is 1, A/D conversion for the lower-priority group automatically restarts upon completion of A/D conversion for the priority group. A trigger input of the lower-priority group generated during A/D conversion on the priority group takes effect, and A/D conversion for the lower-priority group is automatically performed upon completion of A/D conversion on the priority group.

If the ADGSPCR.GBRSCN bit is 1 and the ADGSPCR.LGRRS bit is 0, A/D conversion for the lower-priority group is restarted from the first channel. If the setting of the ADGSPCR.LGRRS bit is 1, A/D conversion for the lower-priority group is restarted from the channel for which the conversion stopped. However, if the self-diagnosis function is used, the A/D conversion is restarted from the channel for which the conversion stopped after self-diagnosis completed.

[section 44.3.4.3. Group Priority Operation](#) summarizes operations in response to the input of a trigger during A/D conversion with the settings of the ADGSPCR.GBRSCN bit.

If the setting of the ADGSPCR.GBRP bit is 1, A/D conversion operation for the lowest-priority group is to continuously perform single scans.

For the trigger settings in group-scan mode, select a synchronous trigger for group A by using the ADSTRGR.TRSA[5:0] bits, a synchronous trigger for group B by using the ADSTRGR.TRSB[5:0] bits, and a synchronous trigger for group C by using the ADGCTRGR.TRSC[5:0] bits. Each trigger must be different from each other.

When using two groups in group-scan mode (setting the ADGCTRGR.GRCE bit to 0) with the ADGSPCR.GBRP bit set to 1, set the ADSTRGR.TRSB[5:0] bits to 3Fh.

When using three groups in group-scan mode (setting the ADGCTRGR.GRCE bit to 1) with the ADGSPCR.GBRP bit set to 1, set the ADGCTRGR.TRSC[5:0] bits to 3Fh.

The channels to be scanned must be selected in the registers shown in [section 44.3.4. Group Scan Mode](#).

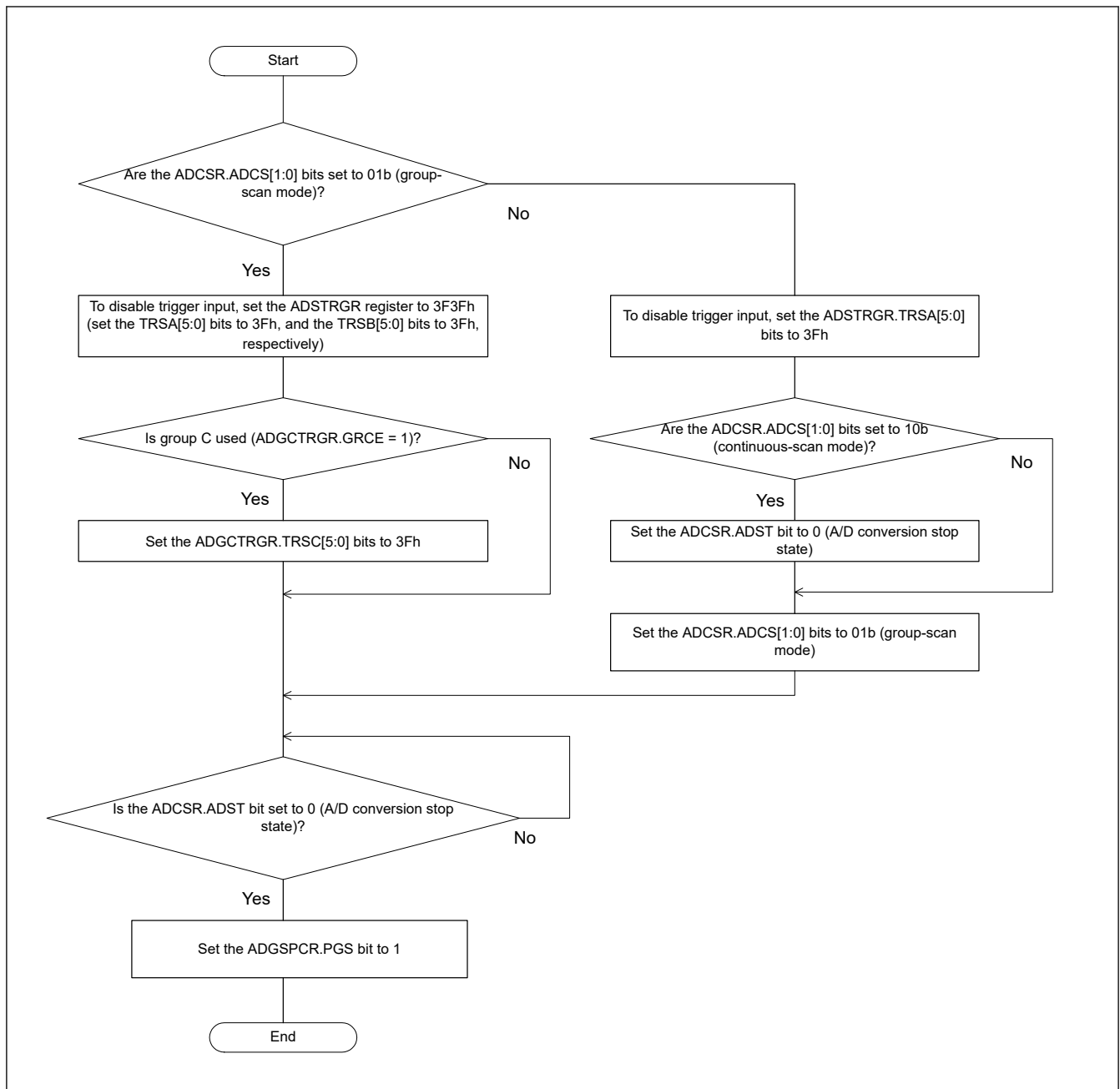


Figure 44.16 Flowchart for ADGSPCR.PGS bit setting

Table 44.24 Control of A/D conversion operations according to ADGSPCR.GBRSCN bit setting (1 of 2)

A/D conversion operation	Trigger input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group A is in progress	Input of trigger for group A	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group B	Trigger input is ineffective.	A/D conversion for group B is performed after A/D conversion for group A completes.
	Input of trigger for group C	Trigger input is ineffective.	A/D conversion for group C is performed after A/D conversion for group A completes.

Table 44.24 Control of A/D conversion operations according to ADGSPCR.GBRSCN bit setting (2 of 2)

A/D conversion operation	Trigger input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group B is in progress	Input of trigger for group A	A/D conversion for group B is discontinued and A/D conversion for group A starts.	- A/D conversion for group B is discontinued and A/D conversion for group A starts. - A/D conversion for group B starts after A/D conversion for group A completes.
	Input of trigger for group B	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group C	Trigger input is ineffective.	A/D conversion for group C is performed after A/D conversion for group B completes.
When A/D conversion for group C is in progress	Input of trigger for group A	A/D conversion for group C is discontinued and A/D conversion for group A starts.	- A/D conversion for group C is discontinued and A/D conversion for group A starts. - A/D conversion for group C starts after A/D conversion for group A completes.
	Input of trigger for group B	A/D conversion for group C is discontinued and A/D conversion for group B starts.	- A/D conversion for group C is discontinued and A/D conversion for group B starts. - A/D conversion for group C starts after A/D conversion for group B completes.
	Input of trigger for group C	Trigger input is ineffective.	Trigger input is ineffective.

To use group priority operation mode, select the operation mode to be implemented and set the registers according to the following table.

Table 44.25 Group priority operation setting and operation mode for two groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 0)

ADGSPCR			Operation category
GBRSCN	LGRRS	GBRP	
0	x	0	Group priority operation for two groups (groups A and B) When a trigger of group A is input, A/D conversion for group B is terminated (and will not be restarted).
1	0	0	Group priority operation for two groups (groups A and B) After A/D conversion for group B stopped, when A/D conversion for group A completes, A/D conversion for the group B channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order set in the ADSCSn register.
1	1	0	Group priority operation for two groups (groups A and B) After A/D conversion for group B stopped, when A/D conversion for group A completes, A/D conversion for the group B channels selected in the ADANSB0/1 register restarts according to the conversion order set in the ADSCSn register, beginning from the channel for which A/D conversion stopped.*1
x	0	1	Group priority operation for two groups (groups A and B) Single scanning for group B is continuously performed without a start trigger input. After A/D conversion for group B stopped, when A/D conversion for group A completes, single scanning for the channels selected in the ADANSB0/1 register restarts according to the conversion order set in the ADSCSn register.
1	1	1	Group priority operation for two groups (groups A and B) Single scanning for group B is continuously performed without a start trigger input. After A/D conversion for group B stopped, when A/D conversion for group A completes, single scanning for the channels selected in the ADANSB0/1 register restarts according to the conversion order set in the ADSCSn register, beginning from the channel for which A/D conversion stopped.*1

Note: x: Don't care.

Note 1. When the self-diagnosis function is enabled (ADCER.DIAGM = 1), A/D conversion for the channel that has been stopped is started after self-diagnosis is performed.

Table 44.26 Group priority operation setting and operation mode for three groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 1)

ADGSPCR			Operation category
GBRSCN	LGRRS	GBRP	
0	x	0	Group priority operation for three groups (groups A, B, and C) - When a trigger of group A is input, A/D conversion for group B completes (and will not be restarted). - When a trigger of group A or B is input, A/D conversion for group C completes (and will not be restarted).
0	x	1	Group priority operation for three groups (groups A, B, and C) - When a trigger of group A is input, A/D conversion for group B completes (and will not be restarted). - Single scanning for group C is continuously performed without a start trigger input. After A/D conversion for group C stopped, when scanning of groups A and B completes, A/D conversion for the channels selected in the ADANSC0 and ADANSC1 registers restarts according to the conversion order set in the ADSCSn register.
1	0	0	Group priority operation for three groups (groups A, B, and C) - After A/D conversion for group B stopped, when A/D conversion for group A completes, A/D conversion for the channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order set in the ADSCSn register. - After A/D conversion for group C stopped, when A/D conversion for groups A and B completes, A/D conversion for the channels selected in the ADANSC0 and ADANSC1 registers restarts according to the conversion order set in the ADSCSn register.
1	1	0	Group priority operation for three groups (groups A, B, and C) - After A/D conversion for group B stopped, when A/D conversion for group A completes, A/D conversion for the channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order set in the ADSCSn register, beginning from the channel for which A/D conversion stopped.*1 - After A/D conversion for group C stopped, when A/D conversion for groups A and B completes, A/D conversion for the channels selected in the ADANSC0 and ADANSC1 registers restarts according to the conversion order set in the ADSCSn register, beginning from the channel for which A/D conversion stopped.*1
1	0	1	Group priority operation for three groups (groups A, B, and C) Single scanning for group C is continuously performed without a start trigger input. After A/D conversion for group C stopped, when A/D conversion for groups A and B completes, single scanning for the channels selected in the ADANSC0 and ADANSC1 registers restarts according to the conversion order set in the ADSCSn register. After A/D conversion for group B stopped, when scanning of group A completes, A/D conversion for the channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order set in the ADSCSn register.
1	1	1	Group priority operation for three groups (groups A, B, and C) - After A/D conversion for group B stopped, when A/D conversion for group A completes, A/D conversion for the channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order set in the ADSCSn register, beginning from the channel for which A/D conversion stopped.*1 - Single scanning for group C is continuously performed without a start trigger input. After scanning of group C stopped, when A/D conversion for groups A and B completes, single scanning for the channels selected in the ADANSC0 and ADANSC1 registers restarts according to the conversion order set in the ADSCSn register, beginning from the channel for which A/D conversion stopped.*1

Note: x: Don't care.

Note 1. When the self-diagnosis function is enabled (ADCER.DIAGM = 1), A/D conversion for the channel that has been stopped is started after self-diagnosis is performed..

(1) Group priority operation for two groups (when ADGSPCR.PGS = 1 and ADGCTRGR.GRCE = 0)

Operation examples 1-1 to 1-5 show group priority operations in group-scan mode (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

Operation example 1-1: “Group A trigger input during group B scan” when rescanning is enabled

1. After A/D conversion for group B stopped, when scanning of group A When input of a trigger for group B sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order set in the ADSCSn register.
2. On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D Data Register y (ADDRy).
3. When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1. Then A/D conversion for the group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order set in the ADSCSn register. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).
4. On completion of A/D conversion on the channels, the result is stored in the corresponding A/D Data Register y (ADDRy).
5. If the setting of the ADCSR.ADIE bit is 1 (enabling interrupt generation upon completion of scanning), a scan end interrupt request is generated.
6. If the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning of the group that was stopped in group priority operation), A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order set in the ADSCSn register while the ADCSR.ADST remains 1.
7. On completion of A/D conversion on the channels, the result is stored in the corresponding A/D Data Register y (ADDRy).
8. If the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan), a group B scan end interrupt request is generated.
9. When A/D conversion for all the channels completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state.

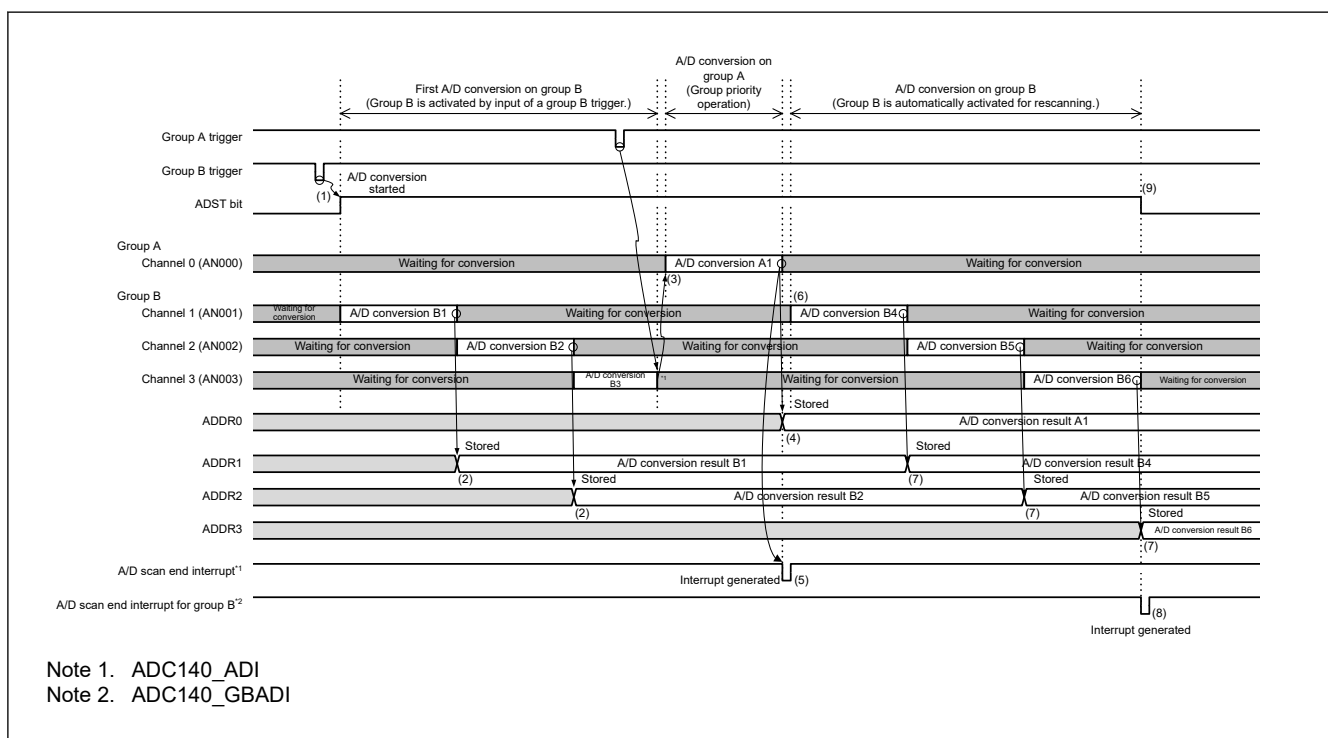


Figure 44.17 Example of group priority operation 1-1: Group A trigger input during group B scanning when rescanning is enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)

Operation example 1-2: “Group A trigger input during rescanning of group B” when rescanning is enabled

Figure 44.18 shows the operation when a group A trigger is input during rescanning operation for group B.

Even during rescanning operation, when a trigger for group A is input, A/D conversion on group B stops and A/D conversion for group A starts. A/D conversion for group B starts after A/D conversion for group A completes.

Operations for setting the ADCSR.ADST bit, storing the A/D conversion result in the corresponding A/D Data Register y (ADDRy), and generating interrupt requests are the same as those in operation example 1-1.

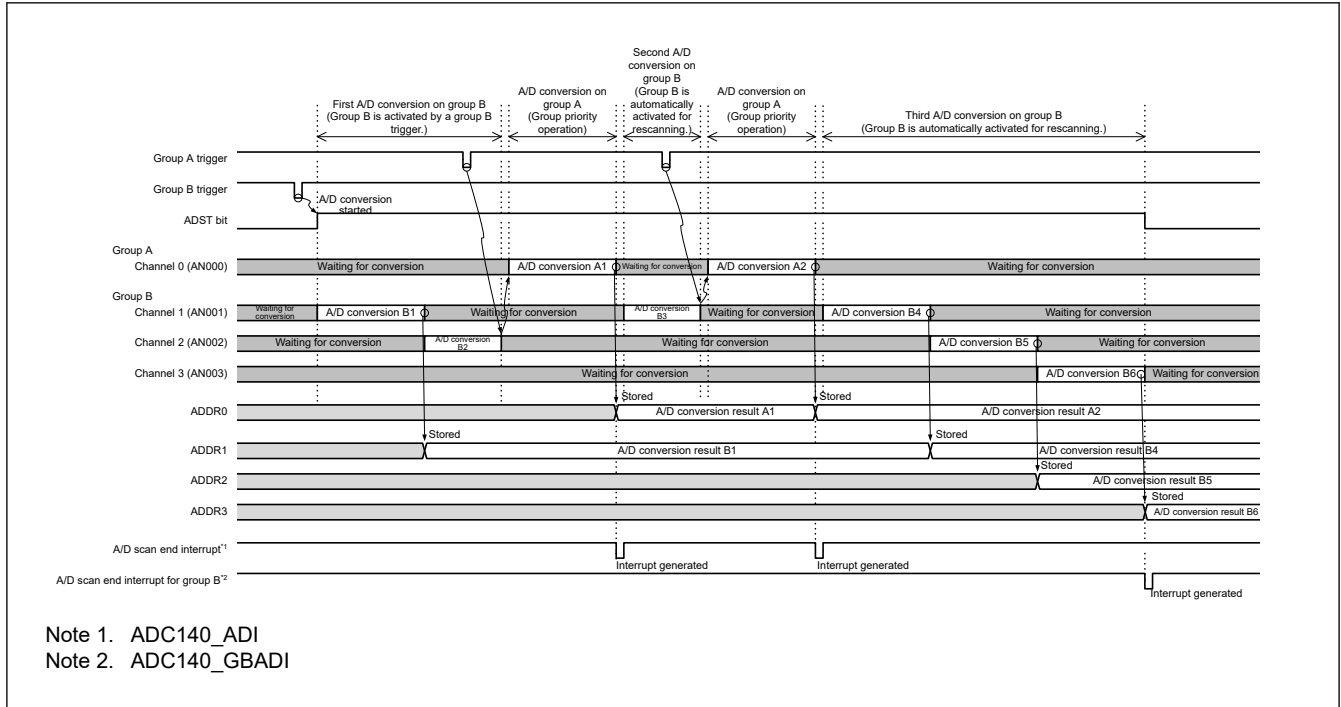


Figure 44.18 Example of group priority operation 1-2: Group A trigger input during rescanning of group B when rescanning is enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)

Operation example 1-3: “Group B trigger input during group A scan” when rescanning is enabled

The following describes the operation when the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning of the group that was stopped in group priority operation) and a trigger for group B is input during scanning operation for group A. If the setting of the ADGSPCR.GBRSCN bit is 0, any trigger for group B that is input during scanning operation for group A is invalid.

- When input of a trigger for group A sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order set in the ADSCSn register.
- When a trigger for group B is input during A/D conversion for group A, group B is ready for A/D conversion.
- On completion of A/D conversion for each channel in group A, the result is stored in the corresponding A/D Data Register y (ADDRy).
- If the setting of the ADCSR.ADIE bit is 1 (enabling interrupt generation upon completion of scanning), a scan end interrupt request is generated.
- When A/D conversion for group A completes, while the ADCSR.ADST bit remains 1, A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order set in the ADSCSn register.
(As with the case of operation example 1-1, if a trigger for group A is input during A/D conversion for group B, A/D conversion for group A starts. Then A/D conversion for group B starts upon completion of A/D conversion for group A.)
- On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
- Upon completion of A/D conversion for group B, a group B scan end interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan).

8. When A/D conversion for all the channels completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state.

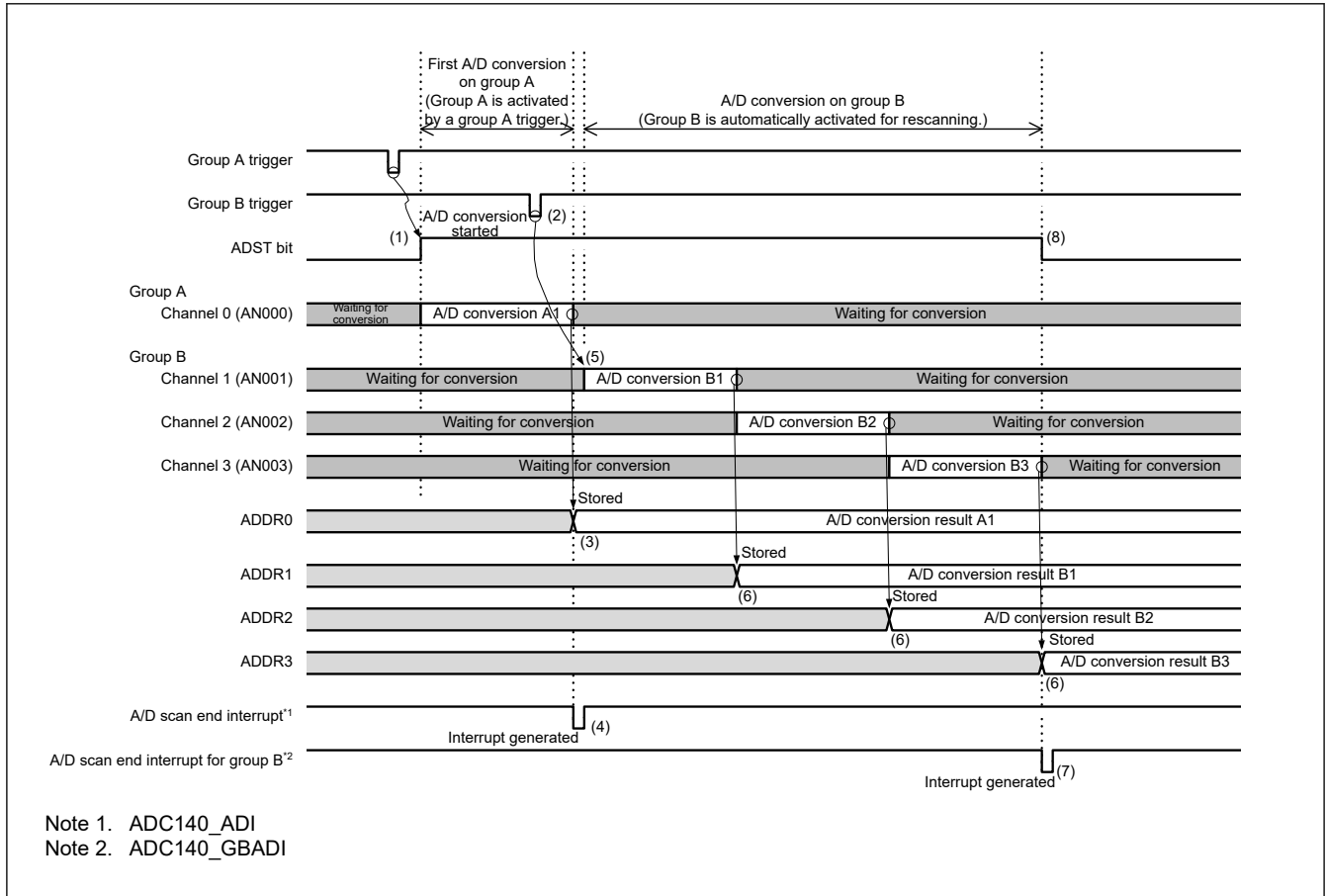


Figure 44.19 Example of group priority operation 1-3: Group B trigger input during group A scan when rescanning is enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)

Operation example 1-4 shows the group priority operation in group-scan mode (when ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

Operation example 1-4: “Group A trigger input during group B scan” when rescanning is disabled

- When input of a trigger for group B sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order set in the ADSCSn register.
- On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D Data Register y (ADDRy).
- When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1, and then A/D conversion for the group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order set in the ADSCSn register. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).
- On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
- On completion of A/D conversion for group A, a scan end interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (enabling interrupt generation upon completion of scanning).
- When A/D conversion for group A completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state. A/D conversion for group B is not performed until a trigger for group B is input the next time.

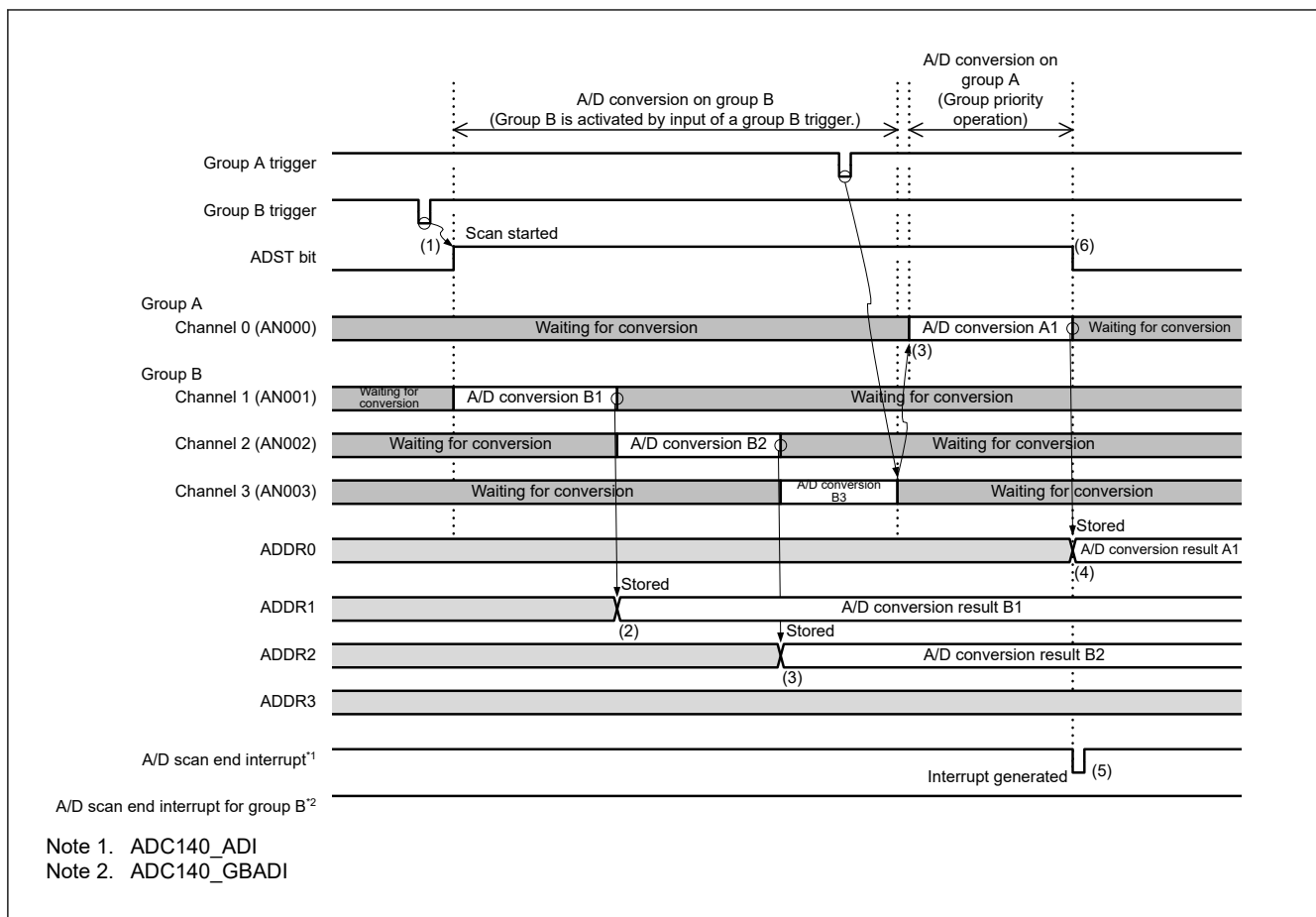


Figure 44.20 Group priority operation example 1-4: “Group A trigger is input during group B scan” when rescanning is disabled (when ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)

Operation example 1-5 shows the group priority operation in group-scan mode (when ADGSPCR.GBRP = 1, and ADGSPCR.LGRRS = 0) when channel 0 is selected for group A and channels 1 and 2 are selected for group B.

If the setting of the ADGCTRGR.GRCE bit is 1, single scan is performed continuously for group C, and group B scan is started by input of a trigger.

Operation example 1-5: Continuously activating single-scan operation for group B

1. When ADGSPCR.GBRP = 1 is set, the ADCSR.ADST bit is set to 1 (starting A/D conversion) and A/D conversion for the analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order set in the ADSCSn register.
2. On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D Data Register y (ADDRy).
3. When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1, and then A/D conversion for group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order zset in the ADSCSn register. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).
4. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
5. On completion of A/D conversion for group A, a scan end interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (enabling interrupt generation upon completion of scanning).
6. If ADGSPCR.GBRP = 1 is set (performing single scan continuously), A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order set in the ADSCSn register while the ADCSR.ADST remains 1 (starting A/D conversion).

7. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
8. If the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan), a group B scan end interrupt request is generated.
9. If ADGSPCR.GBRP = 1 is set (performing single scan continuously), A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order set in the ADSCSn register while the ADCSR.ADST remains 1 (starting A/D conversion).

Steps 6 to 9 are repeated as long as the ADGSPCR.GBRP bit remains 1. Do not clear the ADCSR.ADST bit as long as the ADGSPCR.GBRP bit is 1. To forcibly stop A/D conversion while ADGSPCR.GBRP = 1, follow the procedure shown in Figure 44.40.

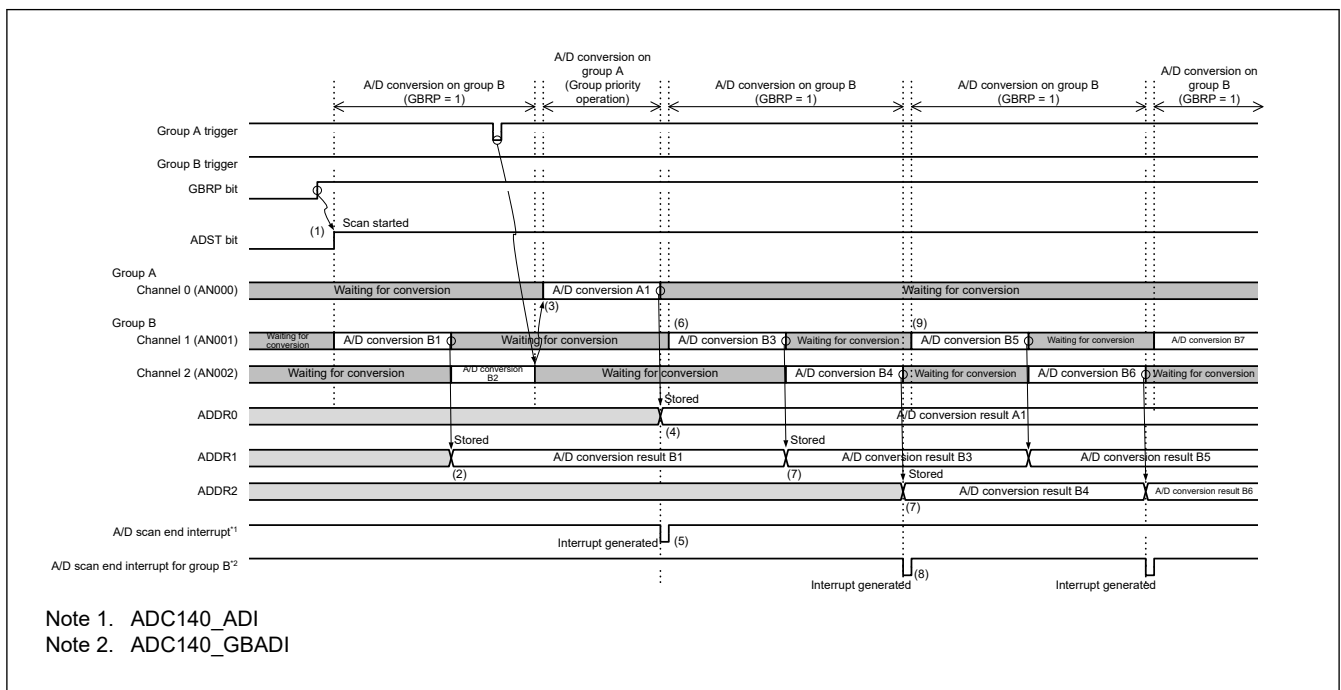


Figure 44.21 Group priority operation example 1-5: Continuously activating single scan for group B (when ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 0, ADGCTRGR.GRCE = 0)

Note: To continuously activate single-scan operation for group B, disable group B trigger input.

(2) Group priority operation for three groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 1)

Operation examples 2-1 to 2-3 show the group priority operations in group-scan mode (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 1) when channel 0 is selected for group A, channels 1 and 2 are selected for group B, and channels 3 and 4 are selected for group C. A priority group means groups A and B for group C, and group A for group B.

Operation example 2-1: "Priority group trigger input during scanning of a lower-priority group" when rescanning is enabled

1. When input of a trigger for group C sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order set in the ADSCSn register.
2. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
3. When a trigger for group B is input during A/D conversion for group C, A/D conversion for group C stops while the ADCSR.ADST bit remains 1, and then A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order set in the ADSCSn register. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).

4. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
5. When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1, and then A/D conversion for the group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order set in the ADSCSn register. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).
6. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
7. If the setting of the ADCSR.ADIE bit is 1 (enabling interrupt generation upon completion of scanning), a scan end interrupt request is generated.
8. If the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning of the group that was stopped in group priority operation), A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order set in the ADSCSn register while the ADCSR.ADST remains 1. At this time, if the setting of the ADGSPCR.LGRRS bit is 1, group B scan starts from the channel for which A/D conversion was stopped.
9. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
10. If the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan), a group B scan end interrupt request is generated.
11. If the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning of the group that was stopped in group priority operation), A/D conversion for the group C analog input channels selected in the ADANSC0 and ADANSC1 registers restarts according to the conversion order set in the ADSCSn register while the ADCSR.ADST remains 1. At this time, if the setting of the ADGSPCR.LGRRS bit is 1, group C scan starts from the channel for which A/D conversion was stopped.
12. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
13. If the setting of the ADCSR.GCADIE bit is 1 (enabling interrupt generation on completion of group C scan), a group C scan end interrupt request is generated.
14. When A/D conversion for all the channels completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state.

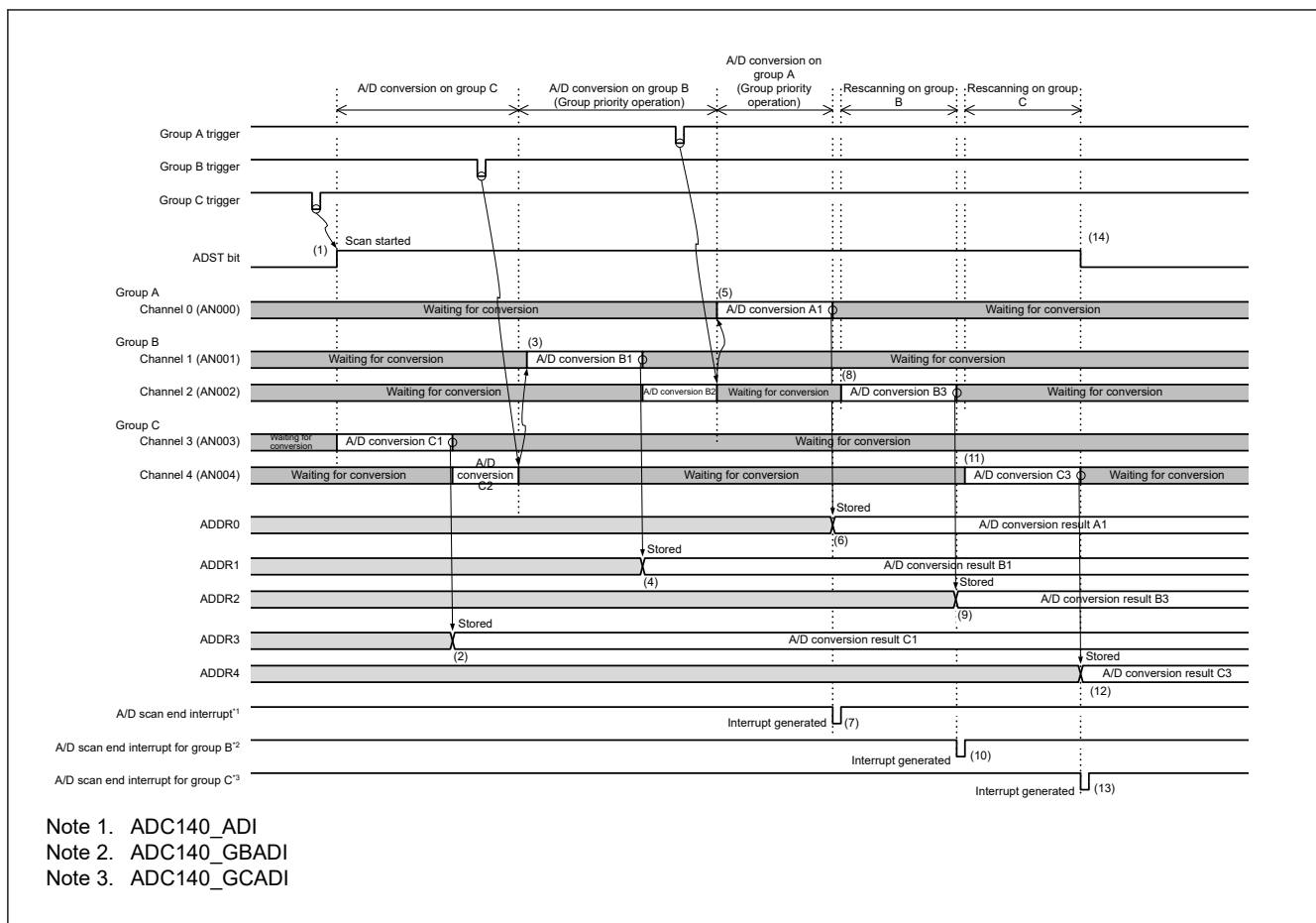


Figure 44.22 Example of group priority operation 2-1: Priority group trigger input during scanning of a lower-priority group when rescanning is enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 1)

Operation example 2-2: “Priority group trigger input during rescanning of a lower-priority group”when rescanning is enabled

Figure 44.23 shows the operation when a group A trigger is input during rescanning operation for group B.

Even while rescanning operation for a lower-priority group is being performed, when a trigger for the priority group (that is, group A or B for group C, or group A for group B) is input, A/D conversion on the priority group starts. When A/D conversion on the priority group completes, the discontinued A/D conversion for the lower-priority group starts.

Operations for setting the ADCSR.ADST bit, storing the A/D conversion result in the corresponding A/D Data Register y (ADDRy), and generating interrupt requests are the same as those in operation example 2-1.

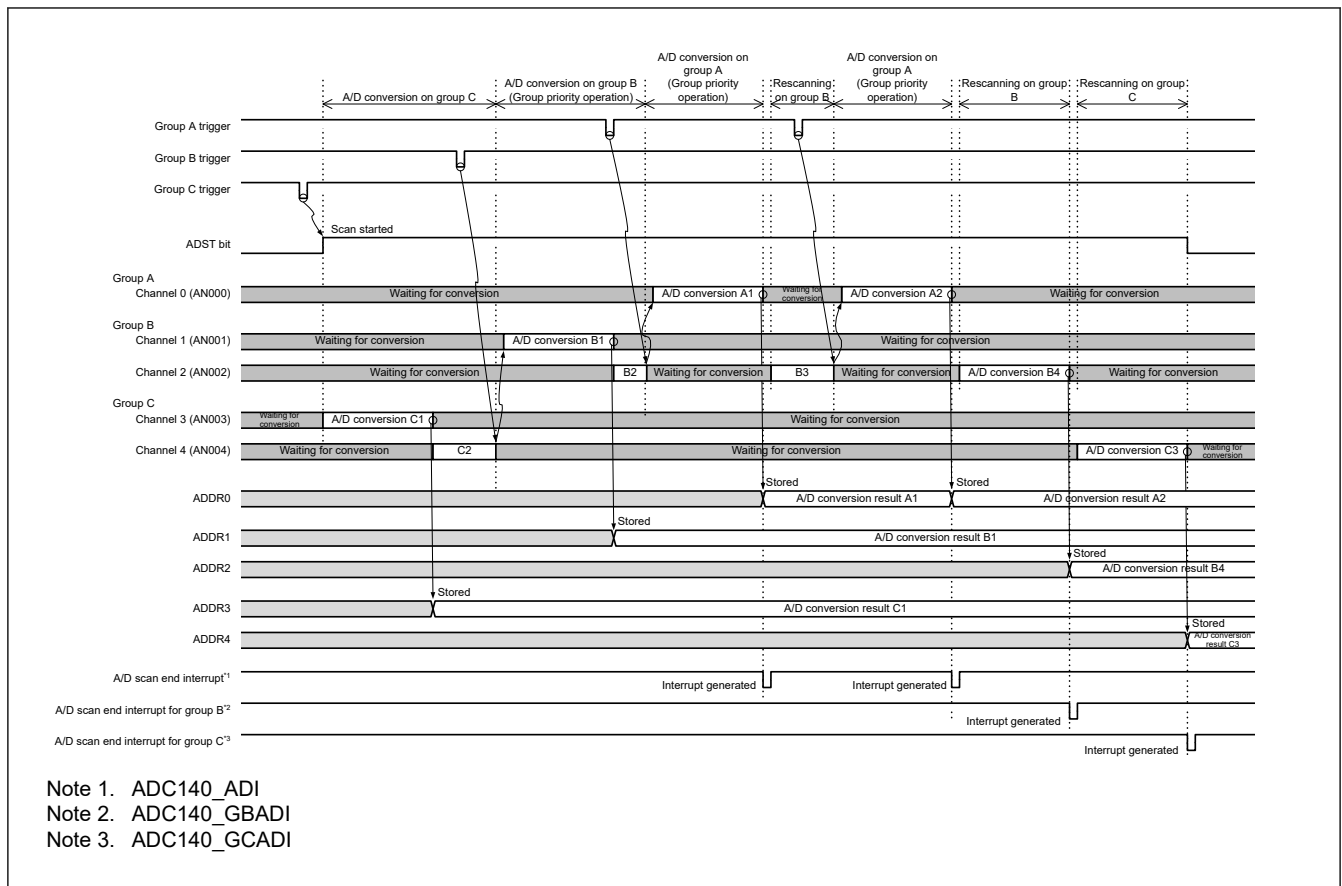


Figure 44.23 Example of group priority operation 2-2: “Priority group trigger input during rescanning of a lower-priority group” when rescanning is enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 1)

Operation example 2-3: “Priority group trigger input during rescanning of a lower-priority group” when rescanning is enabled

The following describes the operation when the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning of the group that was stopped in group priority operation) and a trigger for a lower-priority group is input during A/D conversion for a priority group.

If the setting of the ADGSPCR.GBRSCN bit is 0, any lower-priority group trigger that is input during A/D conversion for the priority group is invalid.

- When input of a trigger for group A sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order set in the ADSCSn register.
- When a trigger for group B is input during A/D conversion for group A, group B is ready for A/D conversion.
- On completion of A/D conversion for each channel in group A, the result is stored in the corresponding A/D Data Register y (ADDRy).
- A scan end interrupt request is generated if the setting of the ADCR.ADIE bit is 1 (enabling interrupt generation upon completion of scanning).
- Upon completion of A/D conversion for group A, while the ADCSR.ADST bit remains 1, A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order set in the ADSCSn register. At this time, if the setting of the ADGSPCR.LGRRS bit is 1, group B scan starts from the channel for which A/D conversion was stopped.
(If a trigger for group A is input during scanning of group B, A/D conversion for group A starts, as with the case of operation example 2-1. Then scanning of group B starts upon completion of A/D conversion for group A.)
- When a trigger for group C is input during A/D conversion for group B, group C is ready for A/D conversion.

7. On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D Data Register y (ADDRy).
8. Upon completion of group B scan, a group B scan interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan).
9. Upon completion of A/D conversion for group B, while the ADCSR.ADST bit remains 1, A/D conversion for the group C analog input channels selected in the ADANSC0 and ADANSC1 registers starts according to the conversion order set in the ADSCSn register. At this time, if the setting of the ADGSPCR.LGRRS bit is 1, group C scan starts from the channel for which A/D conversion was stopped.
(If a trigger for group A or B is input during scanning of group C, A/D conversion for group A or B starts, as with the case of operation example 2-1. Then scanning of group C starts upon completion of A/D conversion for the priority group.)
10. On completion of A/D conversion for each channel in group C, the result is stored in the corresponding A/D Data Register y (ADDRy).
11. Upon completion of group C scan, a group C scan interrupt request is generated if the setting of the ADCSR.GCADIE bit is 1 (enabling interrupt generation on completion of group C scan).
12. When A/D conversion for all the channels completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state.

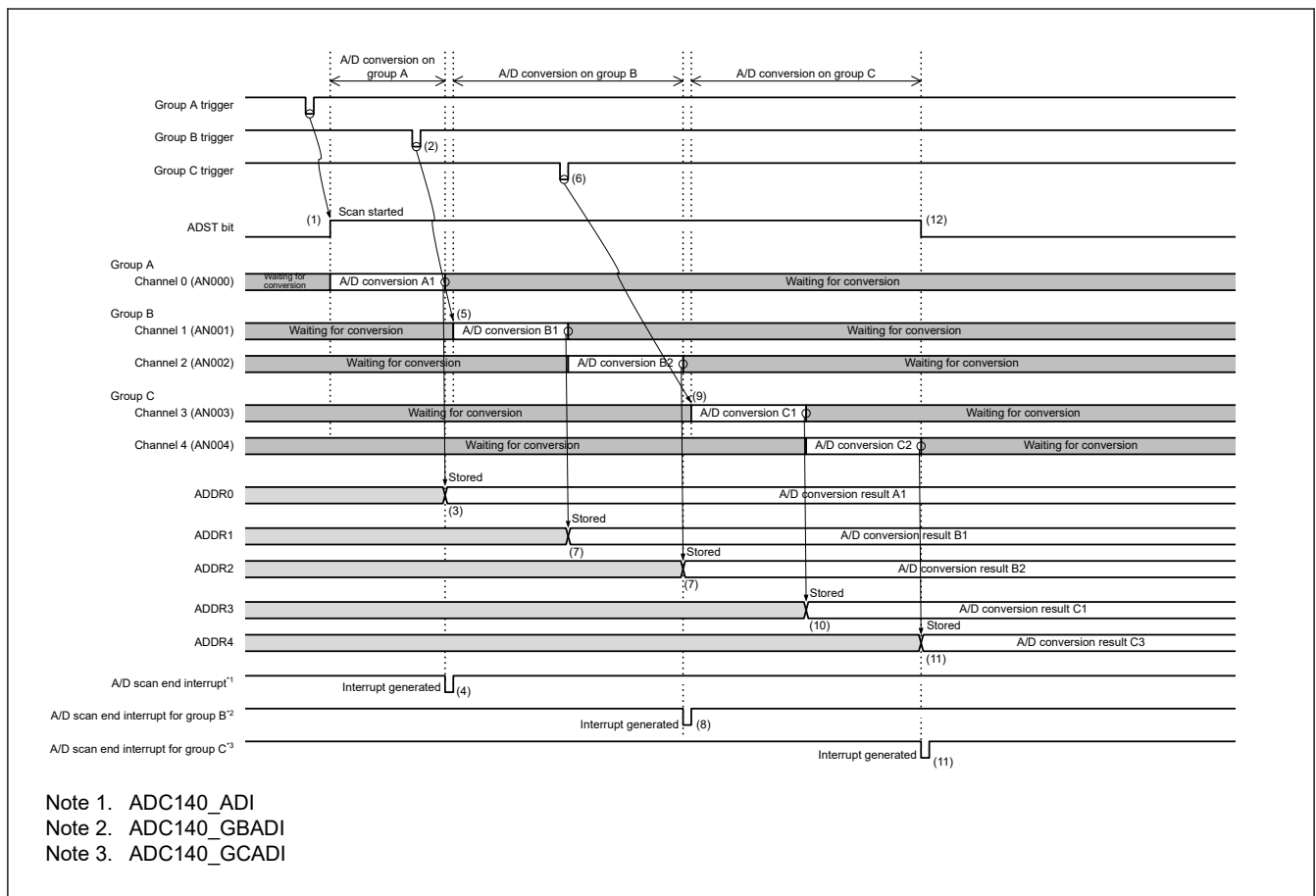


Figure 44.24 Example of group priority operation 2-3: “Priority group trigger input during rescanning of a lower-priority group” when rescanning is enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1, and ADGCTRGR.GRCE = 1)

Operation example 2-4 shows the group priority operation in group-scan mode (when ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0) when channel 0 is selected for group A, channels 1 and 2 are selected for group B, and channels 3 and 4 are selected for group C.

Operation example 2-4: “Priority group trigger input during scanning of a lower-priority group” when rescanning is disabled

1. When input of a trigger for group C sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the analog input channels selected in the ADANSC0 and ADANSC1 registers starts according to the conversion order set in the ADSCSn register.
2. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
3. When a trigger for group B is input during A/D conversion for group C, A/D conversion for group C stops while the ADCSR.ADST bit remains 1, and then A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order set in the ADSCSn register. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).
4. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
5. When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1, and then A/D conversion for the group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order set in the ADSCSn register. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).
6. On completion of A/D conversion for group A, a scan end interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (enabling interrupt generation upon completion of scanning).
7. When A/D conversion for group A completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state. A/D conversion for groups B and C is not performed until a trigger corresponding to a subsequent group is input.

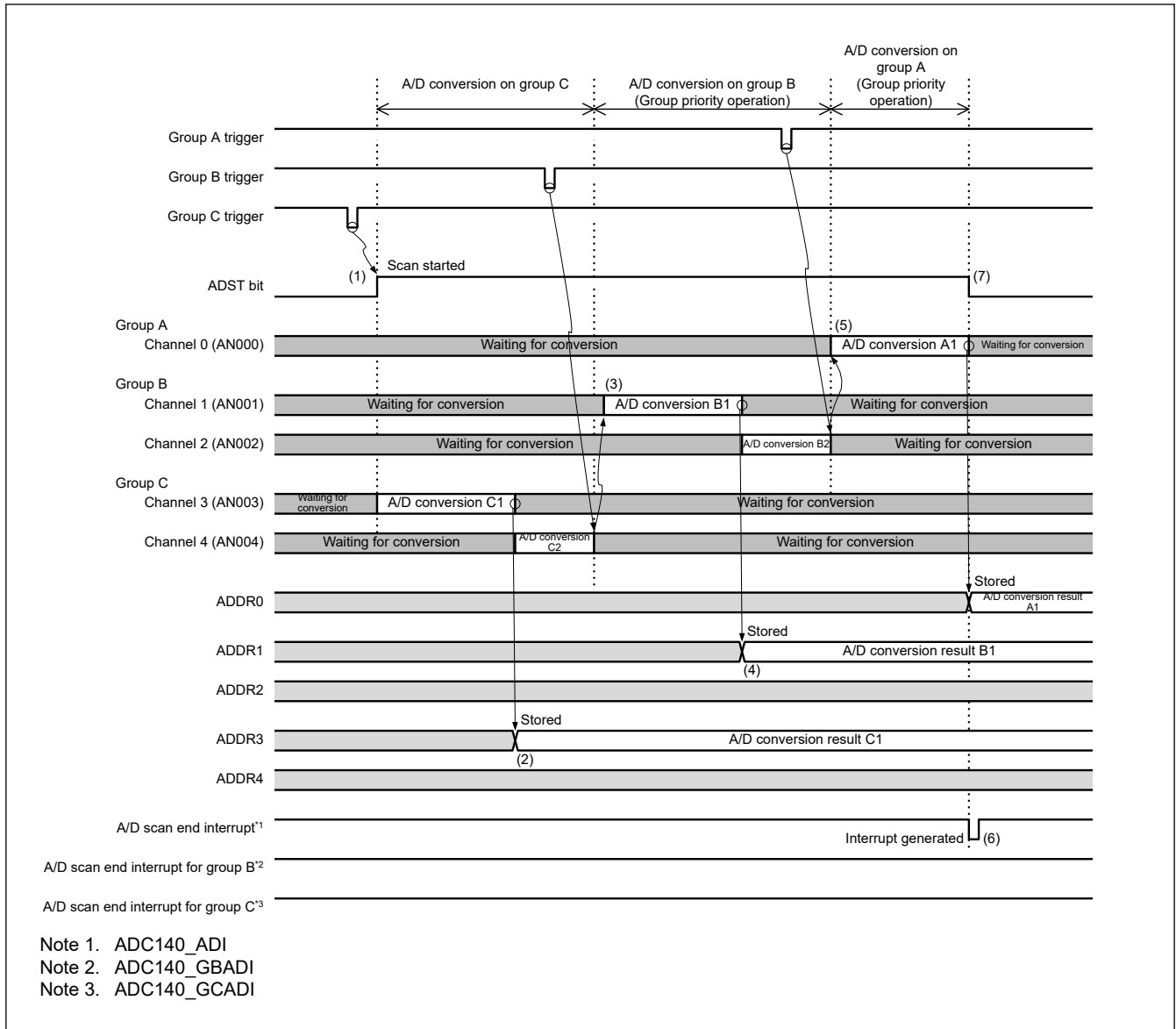


Figure 44.25 Example of group priority operation 2-4: “Priority group trigger input during scanning of a lower-priority group” when rescanning is disabled (when ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)

Operation example 2-5 shows the group priority operation in group-scan mode (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 1, and ADGSPCR.LGRRS = 1) when channel 0 is selected for group A, channel 1 is selected for group B, and channels 2 and 3 are selected for group C.

If the setting of the ADGCTRGR.GRCE bit is 0, single scan is performed continuously for group B, and input of the trigger for group C is disabled.

Operation example 2-5: Continuously activating single scan for group C

1. When ADGSPCR.GBRP = 1 is set, the ADCSR.ADST bit is set to 1 (starting A/D conversion) and A/D conversion for the analog input channels selected in the ADANSC0 and ADANSC1 registers starts according to the conversion order set in the ADSCSn register.
2. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
3. When a trigger for group B is input during A/D conversion for group C, A/D conversion for group C stops while the ADCSR.ADST bit remains 1, and then A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order set in the ADSCSn register. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).

4. When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1, and then A/D conversion for the group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order set in the ADSCSn register. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).
5. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
6. If the setting of the ADCSR.ADIE bit is 1 (enabling interrupt generation upon completion of scanning), a scan end interrupt request is generated.
7. If the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning of the group that was stopped in group priority operation), A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order set in the ADSCSn register while the ADCSR.ADST remains 1 (starting A/D conversion). At this time, if the setting of the ADGSPCR.LGRRS bit is 1, A/D conversion for group B starts from the channel for which A/D conversion was stopped.
8. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
9. If the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan), a group B scan end interrupt request is generated.
10. If the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning of the group that was stopped in group priority operation), A/D conversion for the group C analog input channels selected in the ADANSC0 and ADANSC1 registers restarts according to the conversion order set in the ADSCSn register while the ADCSR.ADST remains 1. At this time, if the setting of the ADGSPCR.LGRRS bit is 1, group C scan starts from the channel for which A/D conversion was stopped.
11. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
12. If the setting of the ADCSR.GCADIE bit is 1 (enabling interrupt generation on completion of group C scan), a group C scan end interrupt request is generated.
13. If ADGSPCR.GBRP = 1 is set (performing single scan continuously), A/D conversion for the group C analog input channels selected in the ADANSC0 and ADANSC1 registers restarts according to the conversion order set in the ADSCSn register while the ADCSR.ADST remains 1 (starting A/D conversion).

Steps (11) to (13) are repeated as long as the ADGSPCR.GBRP bit remains 1.

Do not clear the ADCSR.ADST bit as long as the ADGSPCR.GBRP bit is 1.

To forcibly stop A/D conversion while ADGSPCR.GBRP = 1, follow the procedure shown in [Figure 44.40](#).

Note: To continuously activate single-scan operation for group C, disable group C trigger input.

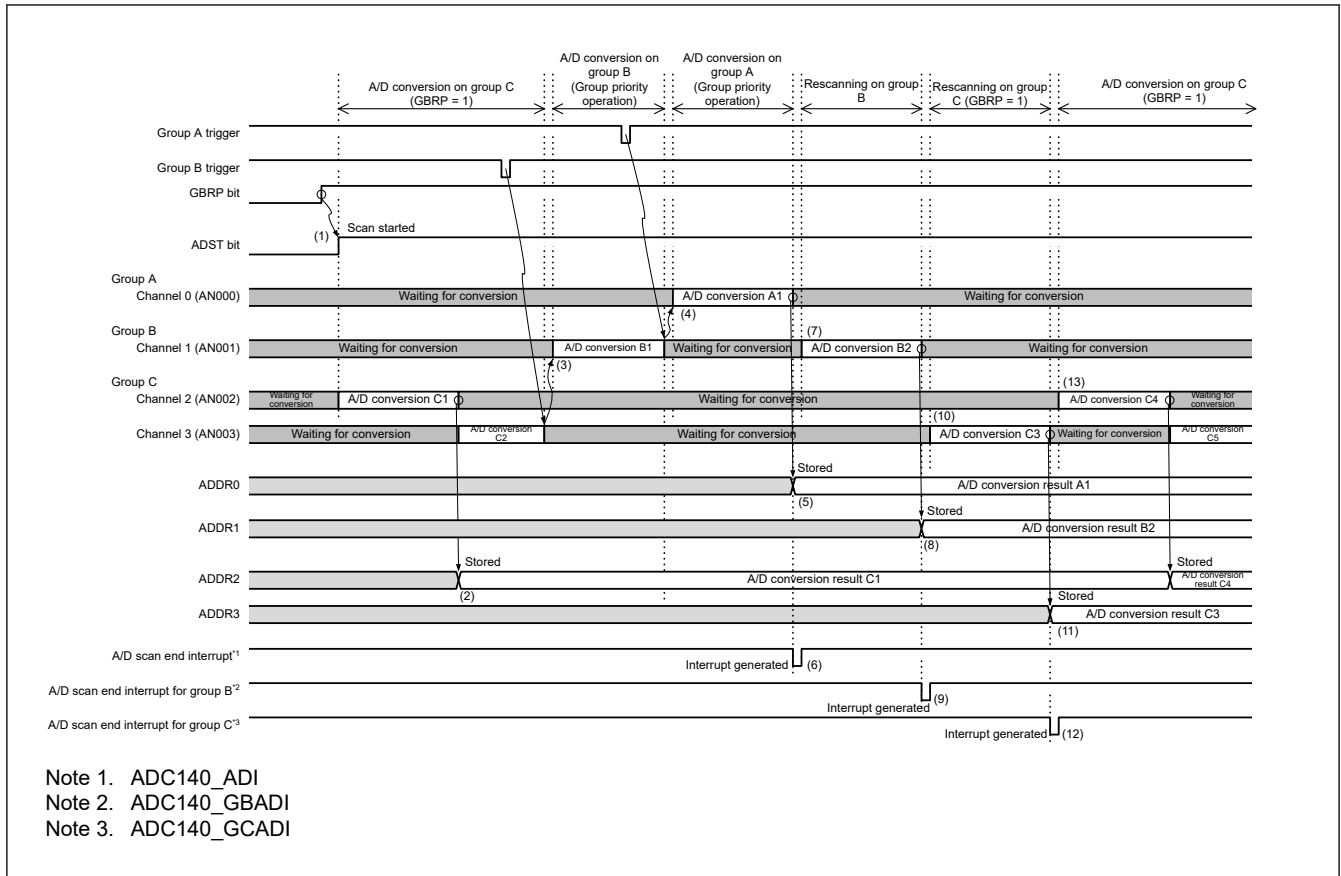


Figure 44.26 Group priority operation example 2-5: Continuously activating single scan for group C (when ADGSPCR.GBRP = 1 and ADGSPCR.LGRRS = 1)

44.3.5 Compare Function for Windows A and B

44.3.5.1 Compare function windows A and B

The compare function compares a reference value with the A/D conversion result. The reference value can be set for Window A and Window B independently. When the compare function is in use, the self-diagnosis function and double trigger mode cannot be used. The main differences between Window A and Window B are their different interrupt output signals and the constraint on Window B of only one selectable channel.

This section provides an example operation that combines continuous scan mode and the compare function.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, a synchronous trigger (TMR, ELC) or an asynchronous trigger, A/D conversion starts.
2. On completion of A/D conversion, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy, ADTSDR). When ADCMPR.CMPAE = 1, if bits in the ADCMPANSRy register or the ADCMPANSER register are set for Window A, the A/D conversion result is compared with the set ADCMPDR0/1 register value. When ADCMPR.CMPBE = 1, if bits in the ADCMPBNSR register are set for Window B, the A/D conversion result is compared with the ADWINULB/ADWINLLB register setting.
3. As a result of the comparison, when Window A meets the condition set in ADCMPLR0/1 or ADCMPLER, the Compare Window A Flag (ADCMPSR0.CMPSTCHAN, ADCMPSR1.CMPSTCHAN, ADCMPSER.CMPSTTSA) sets 1. At this time, if the ADCMPR.CMPAIE bit is 1, an ADC140_CMPAI interrupt request is generated. In the same way, when Window B meets the condition set in ADCMPBNSR.CMPLB, the Compare Window B Flag (ADCMPBSR.CMPSTB) sets to 1. At this time, if the ADCMPR.CMPBIE bit is 1, an ADC140_CMPBI interrupt request is generated.
4. On completion of all selected A/D conversions and comparisons, scan restarts.
5. After the ADC140_CMPAI and ADC140_CMPBI interrupts are accepted, the ADCSR.ADST bit is set to 0 (A/D conversion stop) and processing is performed for channels for which the compare flag is set to 1.

6. When all compare flags of Window A are cleared, the ADC140_CMPAI interrupt request is canceled. In the same way, when all compare flags of Window B are cleared, the ADC140_CMPBI interrupt request is reset. To perform comparison again, restart the A/D conversion.

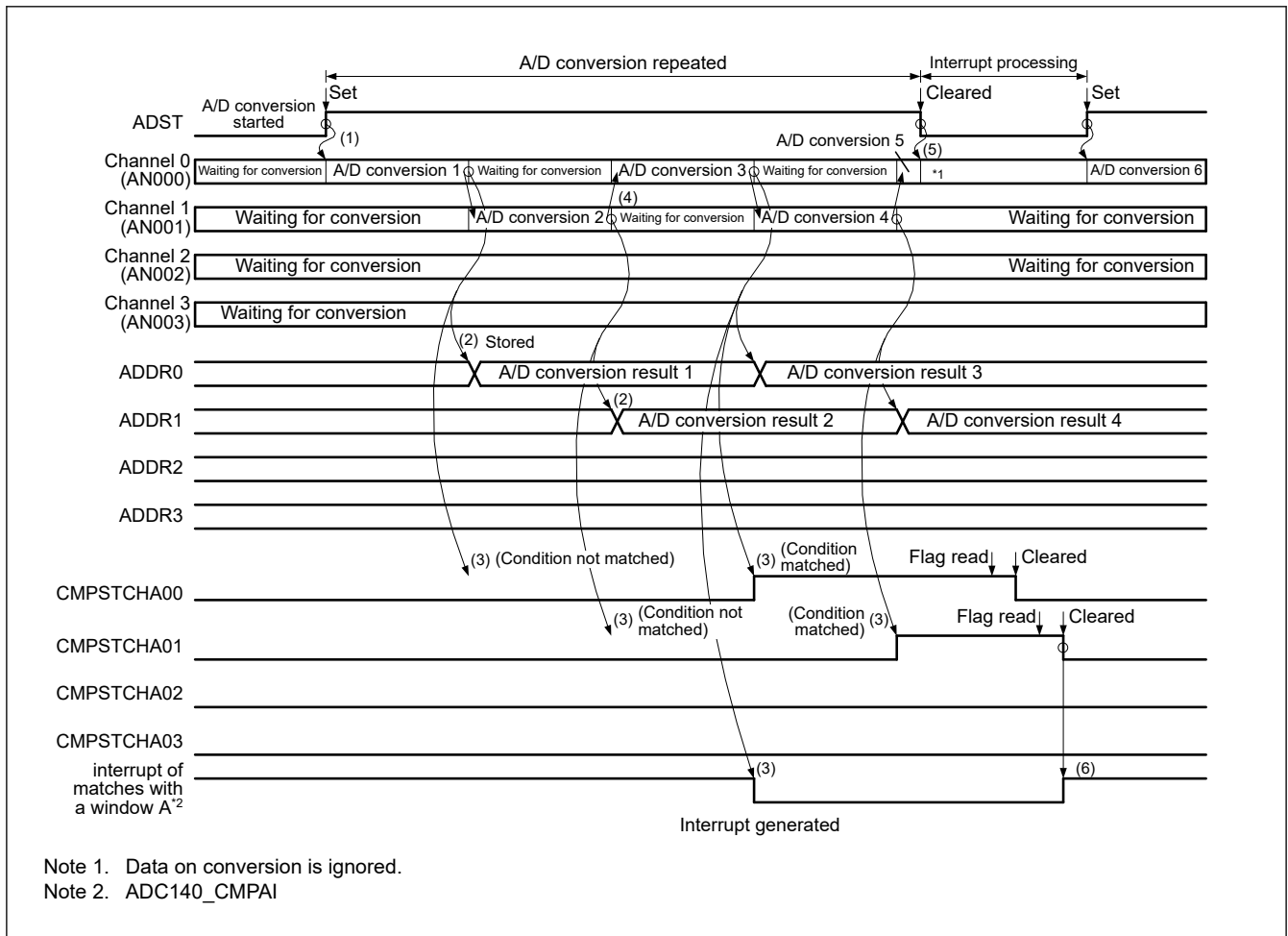


Figure 44.27 Example of compare function operation, when AN000 to AN003 are compared

44.3.5.2 Event output of compare function

The event output of the compare function specifies the upper-side reference voltage value and the lower-side reference voltage value for window A and window B, respectively. The output compares the A/D converted value of the selected channel with the upper and lower side reference voltage value and outputs events (ADC140_WCMPPM/ADC140_WCMPUM) based on event conditions (A or B, A and B, A xor B) and comparison result of window A and window B.

If more than one channel is selected for window A, and even when one channel in window A meets the comparison condition, the comparison result of window A is met. When using this function, perform A/D conversion in single scan mode.

Any channels from analog input, and temperature sensor output are selectable for window A.

One channel from analog input, and temperature sensor output is selectable for window B.

The following sequence is an example of how to set up and use the event output of the compare function:

1. Confirm that the value in the ADCSR.ADCS bits is 00b (single scan mode).
2. Select the channel for window A in the ADCMPANSR0/1 and ADCMPANSER registers. Set the window comparison conditions in the ADCMPLR0/1 and ADCMPLER registers. Set the upper-side and lower-side reference values in the ADCMPDR0/1 registers.
3. Select the channel and comparison conditions for Window B in the ADCMPBNSR register, and set the upper and lower reference values in the ADWINULB and ADWINLLB registers.

4. Set the composite conditions for window A/B, window A/B operation enable, and interrupt output enable in the ADCMPCR register.

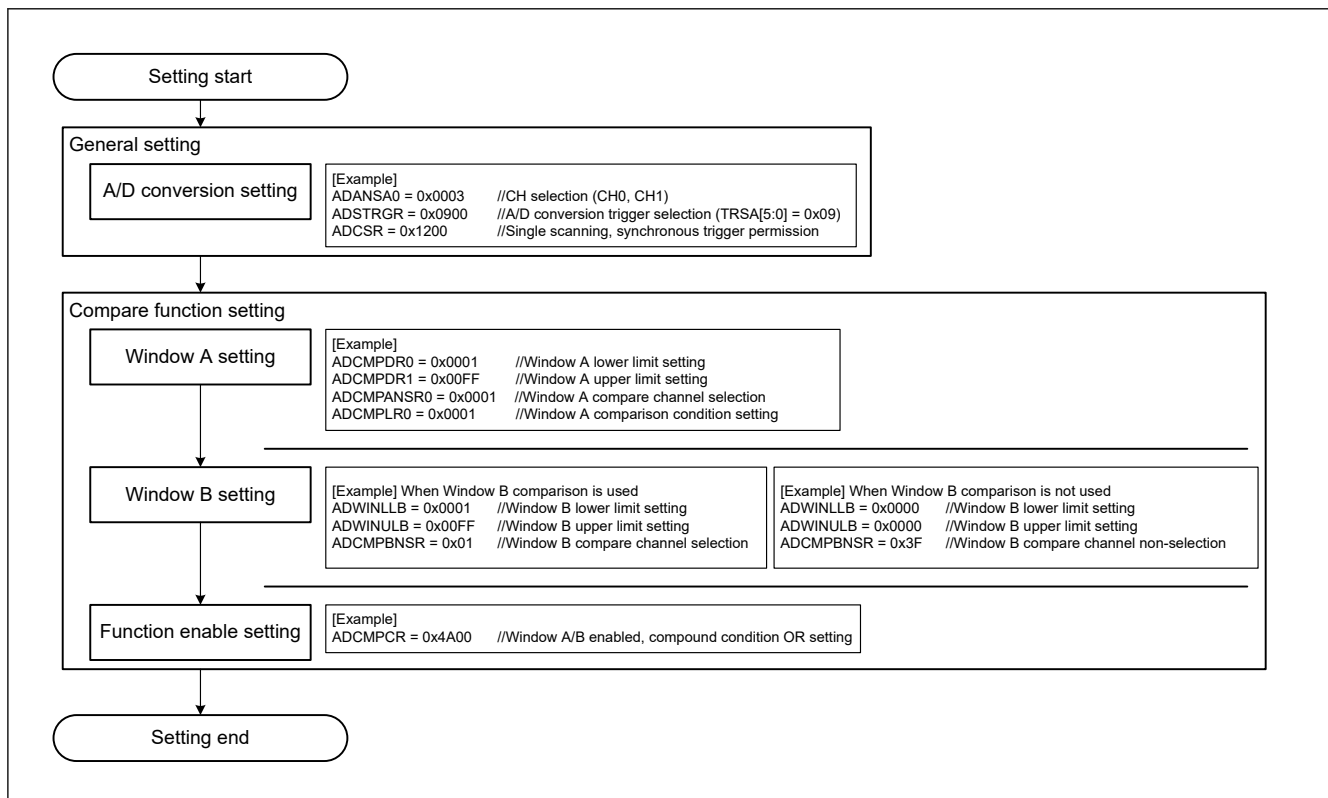


Figure 44.28 Setting example when using the event output of the compare function

For event output usage when using only window A for the compare function, note the following:

- Enable both Window A and Window B (ADCMPCR.CMPAE = 1, ADCMPCR.CMPBE = 1)
- Set the compound condition of Window A and Window B to “OR condition” (ADCMPCR.CMPAB[1:0] = 00b)
- Set the compared channel of Window B to “No selection” (ADCMPBNSR.CMPCHB[5:0] = 0x3F)
- Set the compare condition of Window B to “0 < results < 0 always means mismatch”. (ADCMPCR.WCMPE = 1, ADWINLLB[15:0] = ADWINULB[15:0] = 0x0000, and ADCMPBNSR.CMPLB = 1)

Figure 44.29 shows the event output operation example of compare function.

A scan end event (ADC140_ELC) is output with the same timing as single scan completion. A match or mismatch event (ADC140_WCMPE/ADC140_WCMPUM) is output with 1 PCLKB cycle delay depending on the ADCMPCR.CMPAB[1:0] settings.

Note: The match and mismatch events are exclusive, so both events are never output simultaneously.

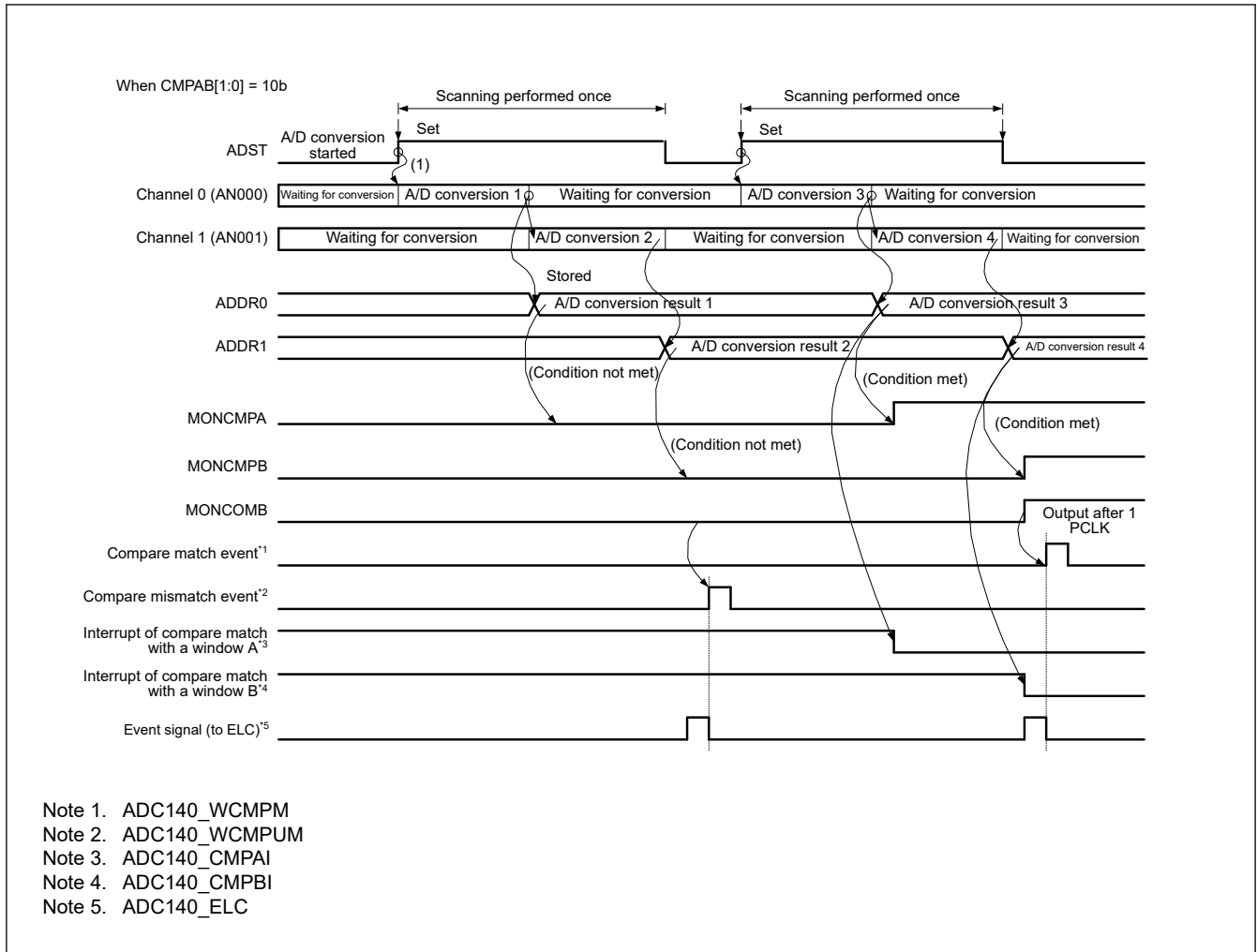


Figure 44.29 Example operation of the compare function event output, when AN000 to AN001 are compared

- Note: Event output of compare function outputs match/mismatch from the comparison results of Window A and Window B, based on the ADCMPCR.CMPAB[1:0] settings.
- Note: The comparison result of Window A is the logical addition of the comparison results of the comparison target channels of Window A. The comparison results of Window A and Window B are updated by each A/D conversion, and are kept even when single scan ends. Set ADCMPCR.CMPAE and ADCMPCR.CMPBE to 0 to clear the comparison results to 0.

44.3.5.3 Restrictions on the compare function

The following constraints apply for the compare function:

- The compare function cannot be used together with the self-diagnosis function or double-trigger mode. (The compare function is not available for ADDR, ADDBLDR.)
- Specify single scan mode when using match/mismatch event outputs.
- When the temperature sensor output is selected for Window A, Window B operations are disabled.
- When the temperature sensor output 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.

44.3.6 Analog Input Sampling and Scan Conversion Time

Scan conversion can be activated either by a software trigger, a synchronous trigger (TMR, ELC), or an asynchronous trigger (ADTRG0). After the start-of-scanning-delay time (t_D) has elapsed, processing for disconnection detection assistance, and processing of conversion for self-diagnosis all proceed, followed by processing for A/D conversion.

Figure 44.30 shows the scan conversion timing, in which scan conversion is activated by a software trigger or a synchronous trigger (TMR, ELC). Figure 44.31 shows the scan conversion timing, in which scan conversion is activated by an asynchronous trigger (ADTRG0). The scan conversion time (t_{SCAN}) includes the start-of-scanning-delay time (t_D), disconnection detection assistance processing time (t_{DIS})^{*1}, self-diagnosis A/D conversion processing time (t_{DIAG} and t_{DSD})^{*2}, A/D conversion processing time (t_{CONV}), and end-of-scanning-delay time (t_{ED}).

The A/D conversion processing time (t_{CONV}) consists of input sampling time (t_{SPL}) and time for conversion by successive approximation (t_{SAM}). The sampling time (t_{SPL}) is used to charge sample-and-hold circuits in the A/D converter. If there is not sufficient sampling time due to the high impedance of an analog input signal source, or if the A/D conversion clock (ADCLK) is slow, sampling time can be adjusted using the ADSSTRn register.

The time for conversion by successive approximation (t_{SAM}) is the following

- 17 ADCLK states with 14-bit accuracy.
- 15 ADCLK states with 12-bit accuracy.

Table 44.27 shows the time for conversion by successive approximation (t_{SAM}).

The scan conversion time (t_{SCAN}) in single scan mode for which the number of selected channels is n can be determined as follows:

$$t_{SCAN} = t_D + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n) + t_{ED}^{*3}$$

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 in the following:

$$(t_{DIS} \times n) + t_{DIAG} + t_{DSD} + (t_{CONV} \times n)^{*3}$$

Note 1. When disconnection detection assistance is not selected, $t_{DIS} = 0$.

Only when the temperature sensor is A/D-converted, the auto-discharge period of 15 ADCLK states is inserted.

Note 2. When the self-diagnosis function is not used, $t_{DIAG} = 0$, $t_{DSD} = 0$.

Note 3. When input sampling times (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 that of t_{SPL} and t_{SAM} set to each selected channel.

Table 44.27 shows the times for conversion during scanning.

Table 44.27 Conversion times during scanning (in numbers of cycles of ADCLK and PCLKB)

Item			Symbol	Type/Conditions*4			Unit
				Synchronous trigger*3	External (asynchronous) trigger	Software trigger	
Scan start delay time*1*2	A/D conversion on a priority group by group priority operation	Lower-priority group is to be stopped (the priority group is activated after the lower-priority group is stopped due to an A/D conversion source of the priority group)	t _D	2 PCLKB + 6 ADCLK	—	—	Cycles
		Lower-priority group is not to be stopped (activation by an A/D conversion source of the priority group)		2 PCLKB + 4 ADCLK	—	—	
	A/D conversion when self-diagnosis is enabled.	When A/D conversion for self-diagnosis is started		2 PCLKB + 6 ADCLK	4 PCLKB + 6 ADCLK	6 ADCLK	
	Other than the above			2 PCLKB + 4 ADCLK	2 PCLKB + 4 ADCLK	4 ADCLK	
Disconnection detection assistance processing time			t _{DIS}		The setting of ADDISCR.ADNDIS[3:0] (initial value = 00h) × ADCLK		
Self-diagnosis conversion processing time*1	Sampling time		t _{DIAG}	t _{SPL}	Setting in ADSSTR00 (initial value = 0Bh) × ADCLK*4*5		
	Time for conversion by successive approximation	12-bit conversion accuracy			t _{SAM}	15 ADCLK	
		14-bit conversion accuracy				17 ADCLK	
	When normal A/D conversion is to be started after completion of A/D conversion for self-diagnosis		t _{DED}	2 ADCLK			
	When A/D conversion for self-diagnosis is to be started after completion of the last channel conversion in continuous-scan mode		t _{DSD}		2 ADCLK		
A/D conversion time*1	Sampling time		t _{CONV}	t _{SPL}	The setting of ADSSTRn (n = 0 to 6, L, T) (initial value 0Fh) × ADCLK*4*5		
	Time for conversion by successive approximation	12-bit conversion accuracy			t _{SAM}	15 ADCLK	
		14-bit conversion accuracy				17 ADCLK	
Scan end processing time*1			t _{ED}		1 PCLKB + 3 ADCLK		

Note 1. See [Figure 44.30](#) and [Figure 44.31](#) for an illustration of times t_D , t_{DIAG} , t_{CONV} , and t_{ED} . t_D and t_{ED} indicate the maximum time.

Note 2. This is the maximum time required from software writing or trigger input to starting A/D conversion.

Note 3. This does not include the time consumed in the path from timer output to trigger input.

Note 4. For this product, PCLKB = ADCLK.

Note 5. The required sampling time is specified by the voltage conditions. see [section 51.4. A/D Conversion Characteristics](#)

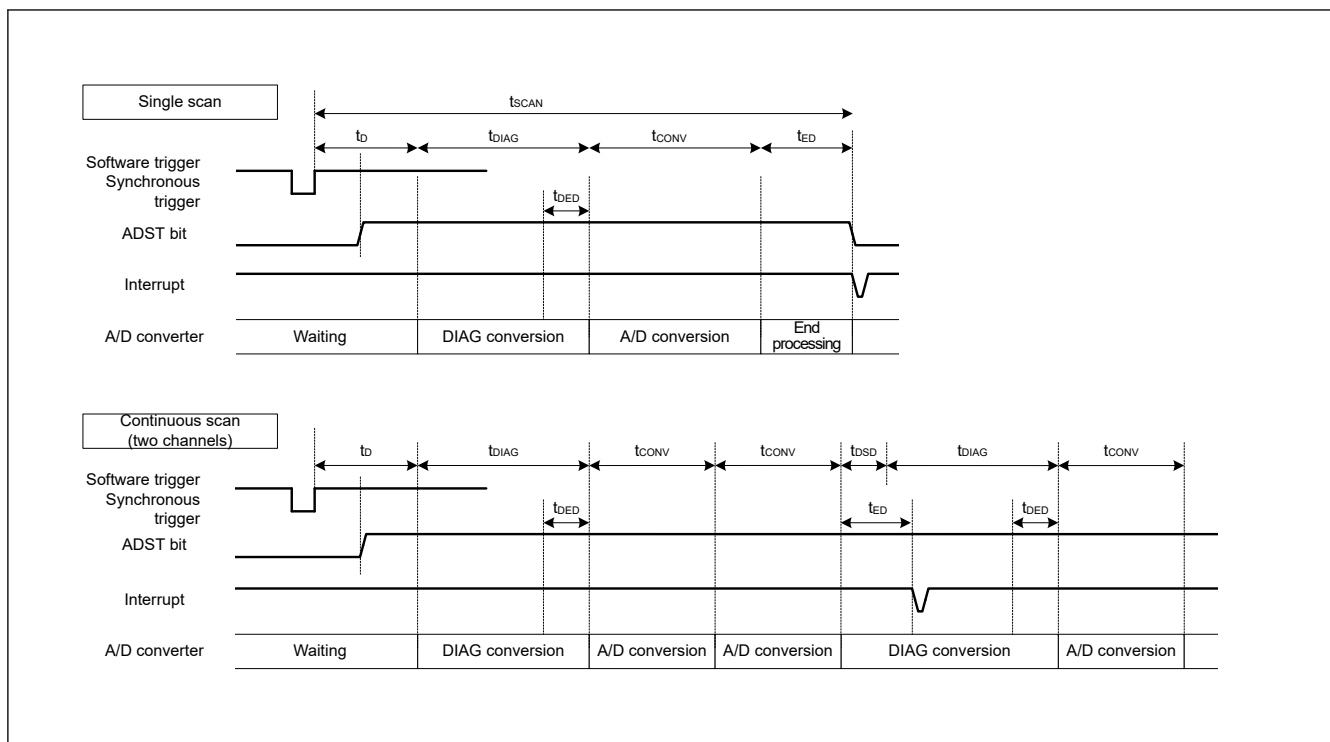


Figure 44.30 Scan conversion timing when activated by software or a synchronous trigger input (ELC)

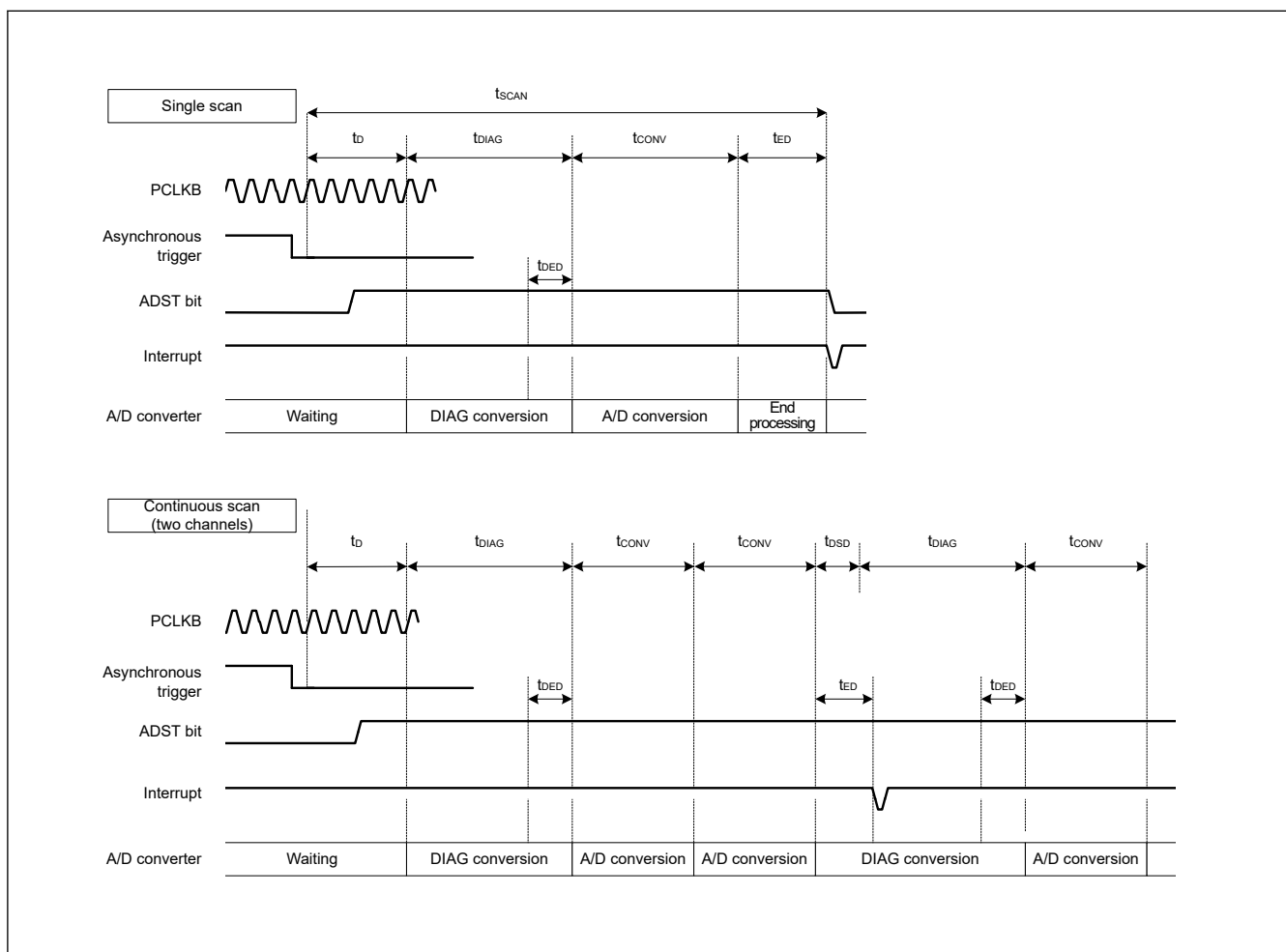


Figure 44.31 Scan conversion timing when activated by an asynchronous trigger input (ADTRG0)

44.3.6.1 Scanning Stop/Start Timing in Group Priority Operations

Group priority operations have the following types of timing of stopping and starting scanning:

1. Timing of stopping scanning of a lower-priority group and then starting scanning of a priority group
2. Timing of restarting the stopped scanning of the lower-priority group. Alternatively, the timing of starting the lower-priority group scanning upon completion of the higher-priority group scanning, by using the lower-priority group trigger accepted during scanning of the higher-priority group.
3. Timing of continuously activating single scan of a lower-priority group

Figure 44.32 shows each type of timing.

The timing of stopping and starting the scanning of groups A and C or groups B and C is the same as the timing of stopping and starting the scanning of groups A and B shown in Figure 44.32. The timing of continuously activating single scan is the same for group B (two groups are used) and group C (three groups are used).

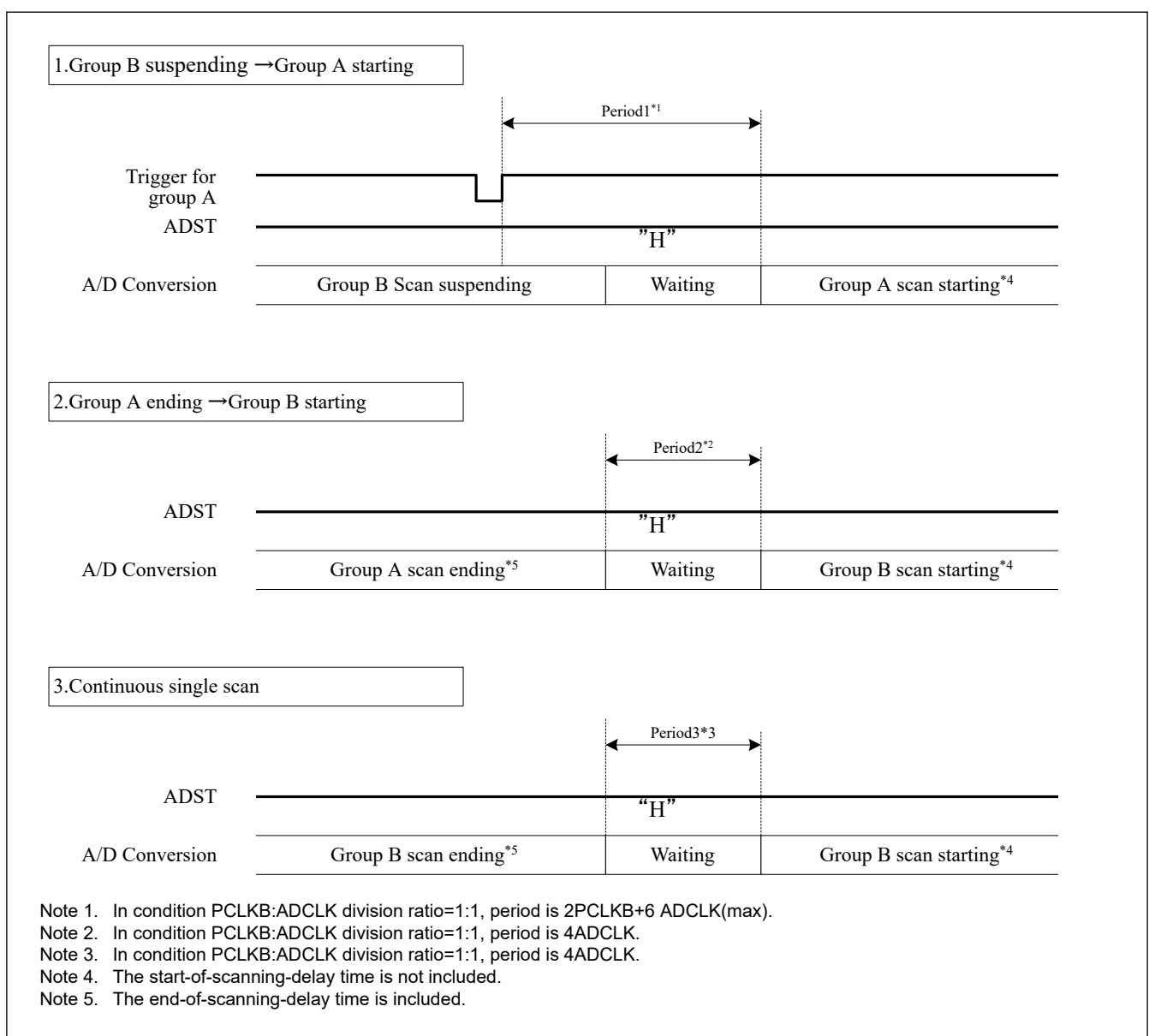


Figure 44.32 Scanning stop/start timing in group priority operations

44.3.7 Usage Example of A/D Data Register Automatic Clearing Function

Setting the ADCER.ACE bit to 1 automatically clears the A/D data registers (ADDRy, ADDR, ADDBLDR, ADTSDR) to 0x0000 when the A/D data registers are read by the CPU or DTC or DMAC.

This function enables detection of update failures of the A/D data registers (ADDRy, ADDR, ADDBLDR, ADTSDR). This section describes examples in which the function to automatically clear the ADDRy register is enabled and disabled.

- If the ADCER.ACE bit is 0 (automatic clearing is disabled) and for some reason, if the A/D conversion result (0x0222) is not written to the ADDRy register, the ADDRy value retains the old data (0x0111). In addition, if this ADDRy value is read into a general-purpose register using an A/D scan end interrupt, the old data (0x0111) can be saved in the general-purpose register. When checking whether there is an update failure, it is necessary to frequently save the old data in SRAM or in a general-purpose register.
- If the ADCER.ACE bit is 1 (automatic clearing is enabled), when ADDRy = 0x0111 is read by the CPU or DTC or DMAC, ADDRy is automatically set to 0x0000. Next, if the A/D conversion result of 0x0222 cannot be transferred to ADDRy for some reason, the cleared data (0x0000) remains as the ADDRy value. If this ADDRy value is read into a general-purpose register using an A/D scan end interrupt, 0x0000 is saved in the general-purpose register. Occurrence of an ADDRy update failure can be determined by checking that the read data value is 0x0000.

44.3.8 A/D-Converted Value Addition/Average Mode

A/D-converted value addition/average mode can be used when A/D conversion of the analog input of the selected channels, the temperature sensor output is selected.

In A/D-converted value addition mode, the same channel is A/D-converted 1, 2, 4, or 16 consecutive times, and the sum of the converted values is stored in the data register. The conversion count of the addition function can be set to 16 only when 12-bit accuracy is selected. In A/D-converted value average mode, the same channel is A/D-converted 1, 2, 4, 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.

The A/D-converted value addition/average function can be used when A/D conversion of the analog inputs of the selected channels or A/D conversion of the temperature sensor output is selected. The A/D-converted value addition/average function can also be used for channels for which the double-trigger function is selected.

The addition function for self-diagnosis is not provided.

44.3.9 Disconnection Detection Assist Function

The ADC14 incorporates a function to fix the charge for sampling capacitance to the specified state VREFH0 or VREFL0 before the start of A/D conversion. This function enables disconnection detection in wiring of analog inputs.

Figure 44.33 shows the A/D conversion operation when the disconnection detection assist function is used. Figure 44.34 shows an example of disconnection detection when precharge is selected. Figure 44.35 shows an example of disconnection detection when discharge is selected.

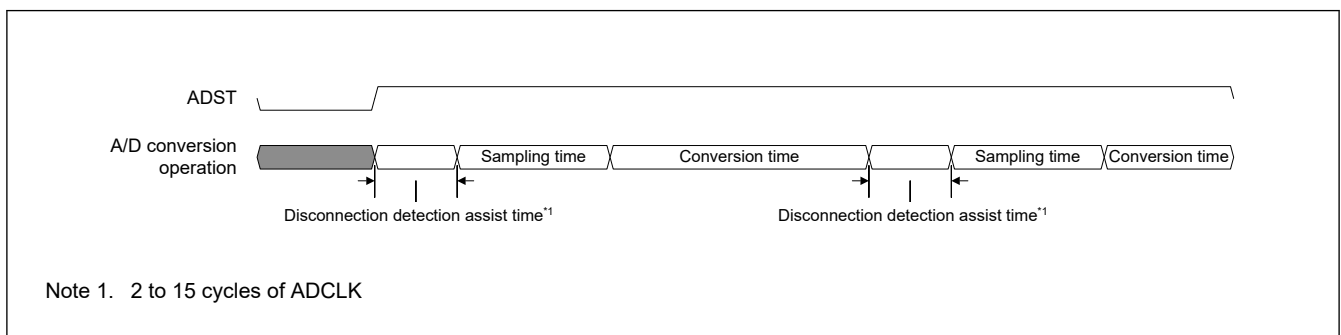


Figure 44.33 Operation of A/D conversion when disconnection detection assist function is used

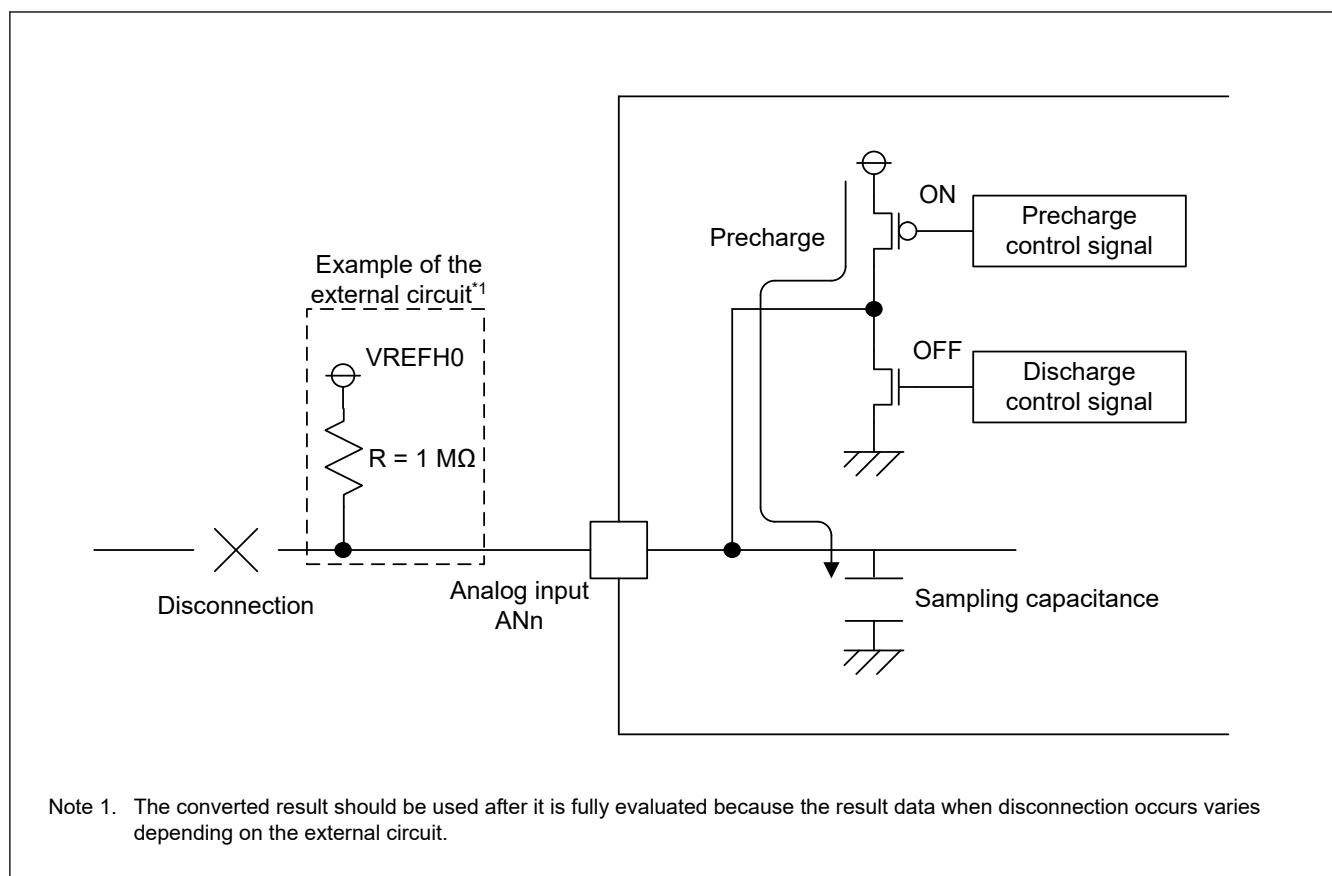


Figure 44.34 Example of disconnection detection when precharge is selected

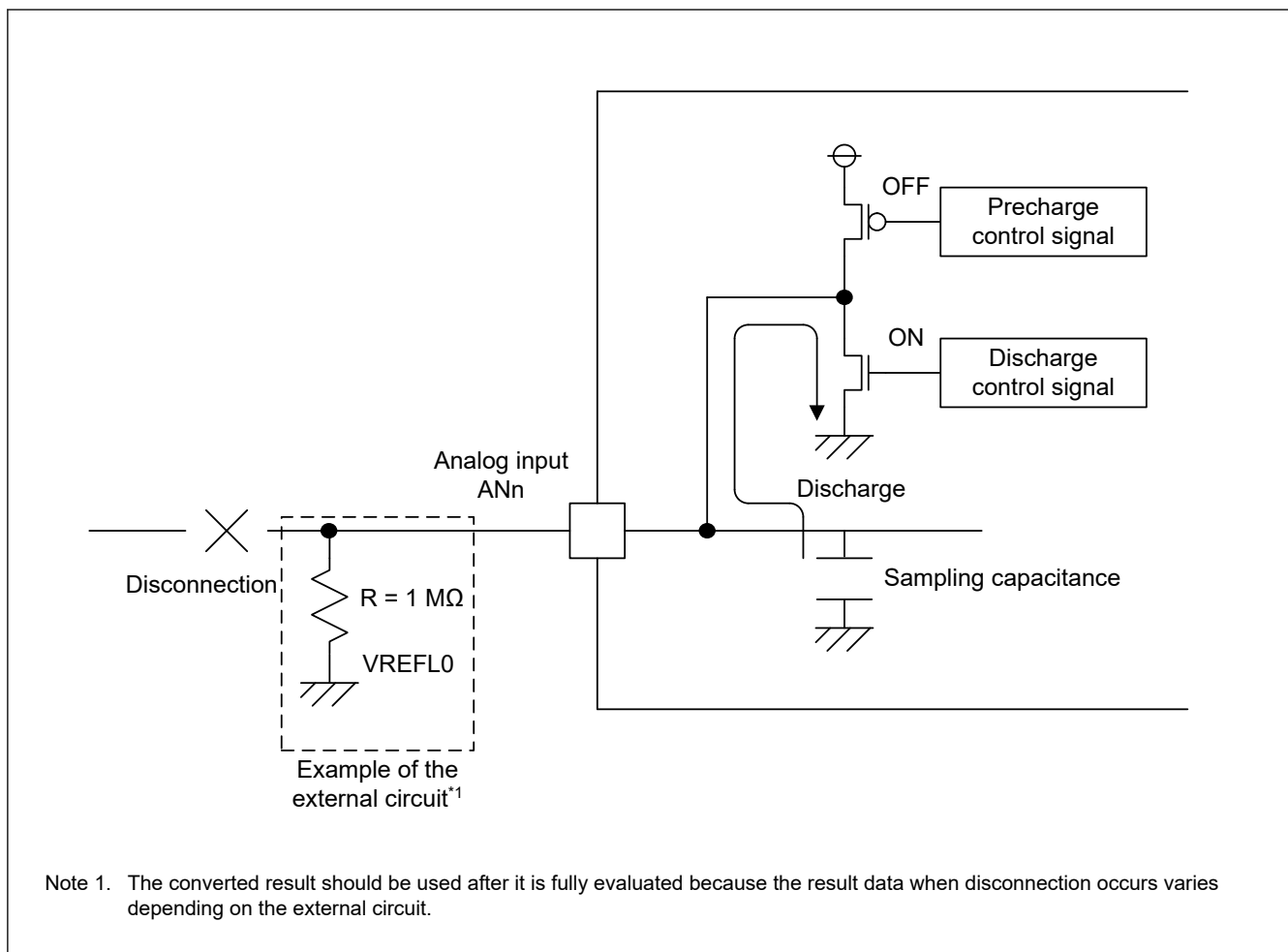


Figure 44.35 Example of disconnection detection when discharge is selected

44.3.10 Starting A/D Conversion with an Asynchronous Trigger

A/D conversion can be started by the input of an asynchronous trigger. To start A/D conversion by an asynchronous trigger, set the pin function in the $PmnPFS$ register, set the A/D Conversion Start Trigger Select bits ($ADSTRGR.TRSA[5:0]$) to 0x00, then input a high-level signal to the asynchronous trigger ($ADTRG0$ pin). Finally, set both the $ADCSR.TRGE$ and $ADCSR.EXTRG$ bits to 1. Figure 44.36 shows timing of the asynchronous trigger input.

An asynchronous trigger cannot be selected in the A/D conversion start trigger for group B, C used in group scan mode. For details on setting the pin function, see [section 22, I/O Ports](#).

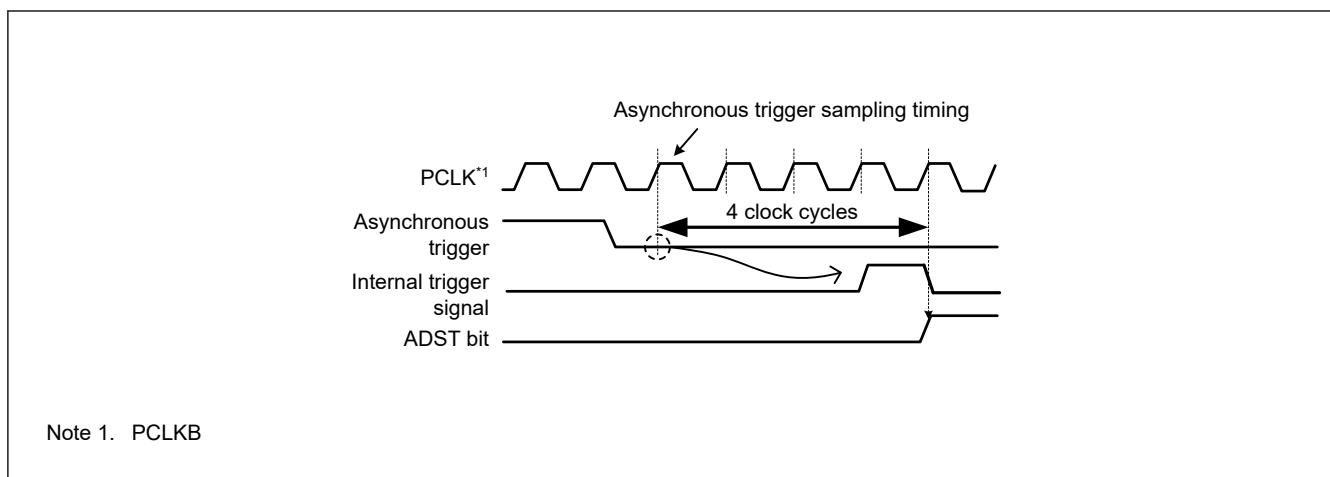


Figure 44.36 Asynchronous trigger input timing

44.3.11 Starting A/D Conversion with a Synchronous Trigger from a Peripheral Module

A/D conversion can be started by a synchronous trigger (TMR, ELC). To do this, set the ADCSR.TRGE bit to 1 and the ADCSR.EXTRG bit to 0, and select the relevant sources in the ADSTRGR.TRSA[5:0] and ADSTRGR.TRSB[5:0] bits.

44.3.12 Arbitrary Channel Order Change Function

This function does not use a concept of scan groups, but replaces the conversion order of physical channels according to the setting of the ADSCSn register.

Channels that can be replaced by this function are the channel of analog input. Setting the same channel in ADSCSn is prohibited. Note that the conversion order relating to self-diagnosis or temperature sensor cannot be changed.

The Conversion Order Setting Register is used only for the arbitrary channel order change function for physical channels. This register does not apply to setting registers (such as ADANSA0) for each physical channel.

Table 44.28 Example1 of conversion order when SCSm for scan group A has an initial value

Setting of ADSCSn		A/D conversion order			ADANSA0 setting
Symbol	Initial value	Physical channel	Priority	Conversion order	Channel Select bit
SCS00	0x00	AN000	High ↓ ↓ ↓	1	ANSA00 = 1
SCS01	0x01	AN001		2	ANSA01 = 1
SCS02	0x02	AN002		3	ANSA02 = 1
SCS03	0x03	AN003		4	ANSA03 = 1
SCS04	0x04	AN004	Low	5	ANSA04 = 1

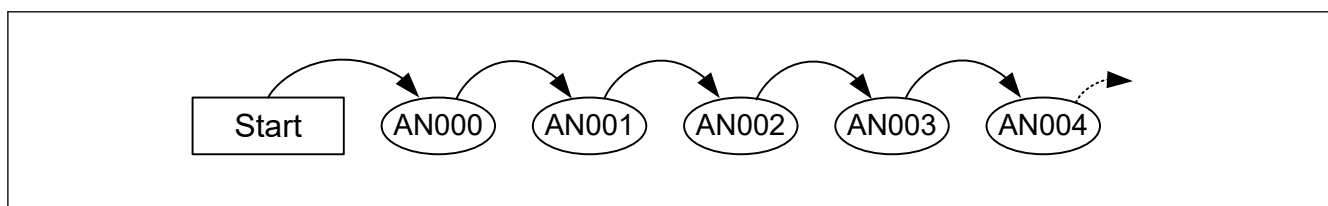


Figure 44.37 A/D conversion order of Example 1

Table 44.29 Example2 of conversion order when the value of SCSm for scan group A is changed

Setting of ADSCSn		A/D conversion order			ADANSA0 setting
Symbol	Initial value	Physical channel	Priority	Conversion order	Channel Select bit
SCS00	0x04	AN004	High ↓ ↓ ↓	1	ANSA00 = 1
SCS01	0x02	AN002		2	ANSA01 = 1
SCS02	0x00	AN000		3	ANSA02 = 1
SCS03	0x01	AN001		4	ANSA03 = 1
SCS04	0x03	AN003	Low	5	ANSA04 = 1

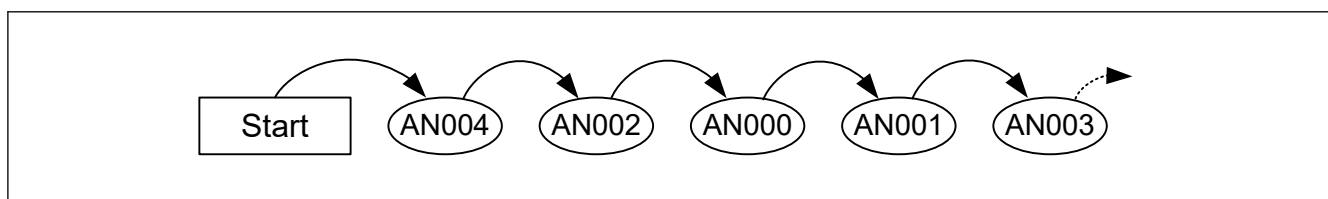


Figure 44.38 A/D conversion order of Example 2

44.3.13 Offset Calibration Function

The offset calibration function corrects offset errors generated in an analog circuit in the A/D converter.

Performing offset calibration can remove stationary offset errors during A/D conversion that is performed after offset calibration.

Figure 44.39 shows a flowchart for using the offset calibration function.

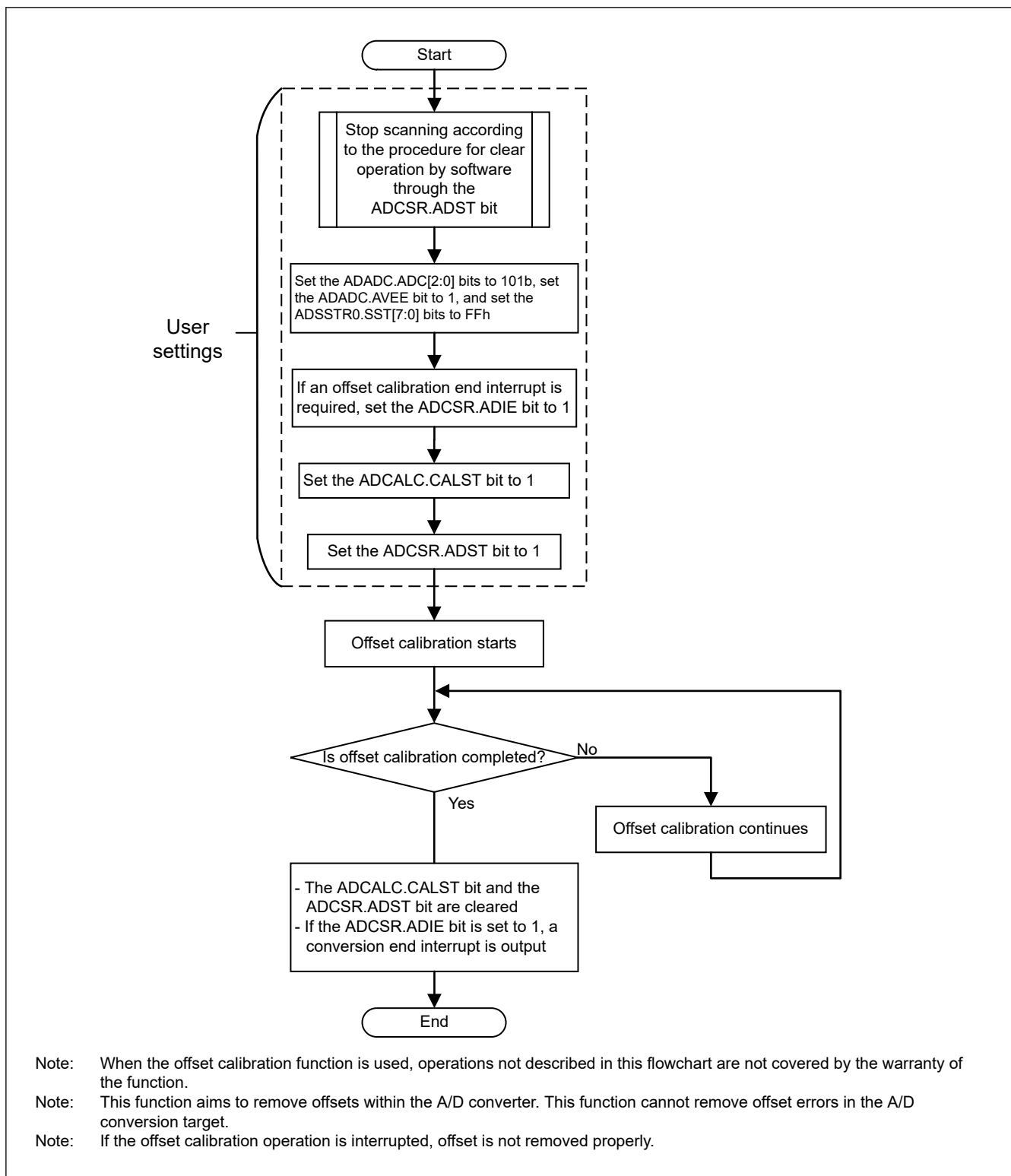


Figure 44.39 Flowchart for setting the offset calibration function

44.4 Interrupt Sources and DTC, DMAC Transfer Requests

44.4.1 Interrupt Requests

The ADC14 can send scan end interrupt requests ADC140_ADI and ADC140_GBADI to the CPU. The ADC14 also generates the ADC140_CMPAI/ADC140_CMPBI interrupt for the CPU in response to matches with a condition for comparison.

An ADC140_ADI interrupt is always generated. An ADC140_GBADI interrupt can be generated by setting the ADCSR.GBADIE bit to 1. Similarly, ADC140_CMPAI and ADC140_CMPBI interrupts can be generated by setting the ADCMPCR.CMPAIE and ADCMPCR.CMPBIE bit to 1.

In addition, the DTC or DMAC can be started when an ADC140_ADI or an ADC140_GBADI interrupt is generated. Using an ADC140_ADI or ADC140_GBADI interrupt to activate the DTC or DMAC to read the converted data enables continuous conversion without a burden on software.

Table 44.30 describes the interrupt sources and ELC events available for the ADC14.

Table 44.30 The interrupt source and ELC event of ADC14 (1 of 2)

Operation			Interrupt request or ELC event	Interrupt request	DTC or DMAC activation	ELC event request	Function
Scan mode	Double trigger mode	Compare function Window A/B					
Single scan mode	Deselected	Deselected	ADC140_ADI, ADC140_ELC	✓	✓	✓	ADC140_ADI, ADC140_ELC generated at the end of single scan
		Selected	ADC140_ADI, ADC140_ELC	✓	✓	✓	ADC140_ADI, ADC140_ELC generated at the end of single scan
			ADC140_CMPAI	✓	—	—	ADC140_CMPAI generated on a match comparison condition of Window A
			ADC140_CMPBI	✓	—	—	ADC140_CMPBI generated on a match comparison condition of Window B
			ADC140_WCMPPM	—	✓	✓	ADC140_WCMPPM generated on a match condition of the Window A/B compare function
			ADC140_WCMPUM	—	✓	✓	ADC140_WCMPUM generated on a mismatch condition of the Window A/B compare function
	Selected	Deselected	ADC140_ADI, ADC140_ELC	✓	✓	✓	ADC140_ADI, ADC140_ELC generated at the end of scans in the even numbered times
Continuous scan mode	Deselected	Deselected	ADC140_ADI, ADC140_ELC	✓	✓	✓	ADC140_ADI, ADC140_ELC generated at the end of scan of all selected channels
		Selected	ADC140_CMPAI	✓	—	—	ADC140_CMPAI generated on a match comparison condition of Window A
			ADC140_CMPBI	✓	—	—	ADC140_CMPBI generated on a match comparison condition of Window B

Table 44.30 The interrupt source and ELC event of ADC14 (2 of 2)

Operation			Interrupt request or ELC event	Interrupt request	DTC or DMAC activation	ELC event request	Function
Scan mode	Double trigger mode	Compare function Window A/B					
Group scan mode	Deselected	Deselected	ADC140_ADI, ADC140_ELC	✓	✓	✓	ADC140_ADI, ADC140_ELC generated at the end of group A scan
			ADC140_GBADI	✓	✓	—	ADC140_GBADI dedicated to group B generated at the end of group B scan
			ADC140_GCADI	✓	✓	—	ADC140_GCADI dedicated to group C generated at the end of group C scan
		Selected	ADC140_ADI, ADC140_ELC	✓	✓	✓	ADC140_ADI, ADC140_ELC generated at the end of group A scan
			ADC140_GBADI	✓	✓	—	ADC140_GBADI dedicated to group B generated at the end of group B scan
			ADC140_GCADI	✓	✓	—	ADC140_GCADI dedicated to group C generated at the end of group C scan
			ADC140_CMPAI	✓	—	—	ADC140_CMPAI generated on a match comparison condition of Window A
			ADC140_CMPBI	✓	—	—	ADC140_CMPBI generated on a match comparison condition of Window B
	Selected	Deselected	ADC140_ADI, ADC140_ELC	✓	✓	✓	ADC140_ADI, ADC140_ELC generated at the end of group A scans in the even-numbered times
			ADC140_GCADI	✓	✓	—	ADC140_GCADI dedicated to group C generated at the end of group C scan
			ADC140_GBADI	✓	✓	—	ADC140_GBADI dedicated to group B generated at the end of group B scan

Note: ✓ available
 —: unavailable

For details on DTC settings, see [section 20, Data Transfer Controller \(DTC\)](#).

44.5 Event Link Function

44.5.1 Event Output to the ELC

The ELC uses the ADC140_ADI interrupt request signal as an event signal ADC140_ELC, enabling link operation for the preset module. The ADC140_GBADI interrupt and ADC140_CMPAI/ADC140_CMPBI interrupts cannot be used as an event signal. For details, see [Table 44.30](#).

An event signal can be output regardless of the settings of the corresponding interrupt request enable bits. For the scan end event(ADC140_ELC), a high-level pulse for one PCLKB cycle is output at the same output timing as the interrupt output (ADC140_ADI) shown in [Table 44.30](#), irrespective of the setting of ADCSR.ADIE. For a compare match

(ADC140_WCMPPM) and mismatch event (ADC140_WCMPUM) to the ELC, a high-level pulse for one PCLKB cycle is output at the timing delayed by one cycle (PCLKB) from the interrupt output (ADC140_ADI) shown in [Table 44.30](#).

To use compare match (ADC140_WCMPPM) or mismatch event (ADC140_WCMPUM) to the ELC, specify single-scan mode.

44.5.2 ADC14 Operation through an Event from the ELC

The ADC14 can start A/D conversion by the preset event specified in the ELSRn settings for the ELC as follows:

- Select the ADC140_ELC signal in the ELC.ELSR8 register

If an ELC event occurs during A/D conversion, the event is disabled.

44.6 Selecting Reference Voltage

The ADC14 can select AVCC0 or VREFH0 as the high-potential reference voltage, and can select VREFL0 or AVSS0 as the low-potential reference voltage. Set these reference voltages before starting A/D conversion. For details on the settings, see [section 44.2.41. ADHVREFCNT : A/D High-Potential/Low-Potential Reference Voltage Control Register](#).

To select an external reference voltage as the high-potential reference voltage, set the ADHVREFCNT.HVSEL [1:0] bits to 01b and apply the external reference voltage to the VREFH pin. To select the voltage from voltage generation circuit, set the ADHVREFCNT.HVSEL [1:0] bits to 01b and connect an external capacitor of about 10 μ F to the VREFH pin. For details, see [section 45, Reference Voltage Generation Circuit \(VREF\)](#). When selecting VREFL0 as the low-potential reference voltage, set the ADHVREFCNT.LVSEL bit to 1. When selecting AVSS0, set the ADHVREFCNT.LVSEL bit to 0.

44.7 Usage Notes

44.7.1 Constraints on Setting the Registers

Set each register while the ADCSR.ADST bit is 0.

44.7.2 Constraints on Reading the Data Registers

The following registers must be read in halfword units:

- A/D Data Registers
- A/D Data Duplexing Register
- A/D Temperature Sensor Data Register
- A/D Self-Diagnosis Data Register

If a register is read twice in byte units, that is, the upper byte and lower byte are read separately, the A/D-converted value read initially might disagree with the A/D-converted value read subsequently. To prevent this, never read the data registers in byte units.

44.7.3 Constraints on Stopping A/D Conversion

(1) A/D Conversion Stop Procedure

To stop A/D conversion when an asynchronous trigger or a synchronous trigger is selected as the condition for starting A/D conversion, follow the procedure shown in [Figure 44.40](#).

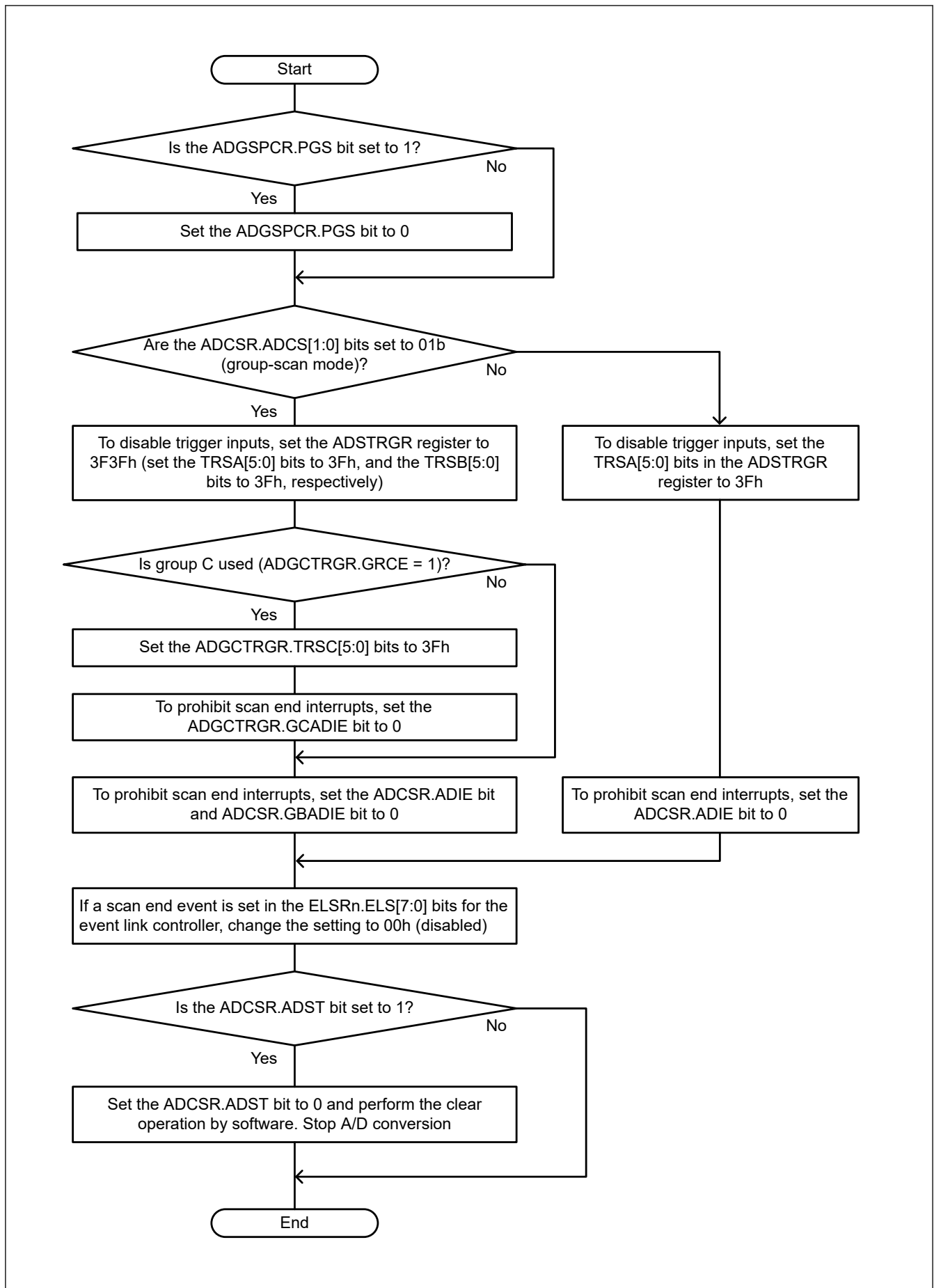


Figure 44.40 Procedures for clearing the ADCSR.ADST bit by software

To specify the following settings after performing the clear operation by software, provide a wait period for at least two ADCLK cycles.

- Enabling scan end interrupts
- Enabling scan end events for the event link controller
- Starting A/D conversion by software
- Enabling trigger input

(2) Notes on Modes and Status Bits

If necessary, individually initialize or set again the voltage status for self-diagnosis, the judgment of the even number or odd number specified for double-trigger mode, and the monitor flags of the compare function.

- To set again the voltage status for self-diagnosis, set the ADCER.DIAGLD bit to 1 and then set a desired value in the ADCER.DIAGVAL[1:0] bits.
- If the setting of the ADCSR.DBLE bit is changed from 0 to 1, the double-trigger mode operation starts from the first scanning.
- To initialize the monitor flags of the compare function (MONCMPA, MONCMPB, and MONCOMB), set the ADCMPCR.CMPAE and ADCMPCR.CMPBE bits to 0.

44.7.4 A/D Conversion Restart and Termination Timing

A maximum of 6 ADCLK cycles is required for the idle analog unit of the ADC14 to restart on setting the ADCSR.ADST bit to 1. A maximum of 2 ADCLK cycles is required for the operating analog unit of the ADC14 to terminate on setting the ADCSR.ADST bit to 0.

44.7.5 Constraints on Scan End Interrupt Handling

When scanning the same analog input twice using any trigger, the first A/D-converted data is overwritten with the second A/D-converted data. This occurs when the CPU does not complete the reading of the A/D-converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated.

44.7.6 Settings for the Module-Stop Function

The Module Stop Control Register can enable or disable ADC14 operation. The ADC14 is initially stopped after a reset. The registers become accessible on release from the module-stop state. After release from the module-stop state, wait for at least 1 μ s before starting A/D conversion. For details, see [section 13, Power-Saving Functions](#).

44.7.7 Notes on Entering the Low-Power States

Before entering the module-stop state or Software Standby mode, be sure to stop A/D conversion. Set the ADCSR.ADST bit in ADCSR to 0 and secure a period of time until the analog unit of the ADC14 stops. Follow the procedure shown in [Figure 44.40](#) to clear the ADCSR.ADST bit with software. Then, wait for 2 clock cycles of ADCLK before entering the module-stop state or Software Standby mode.

44.7.8 Error in Absolute Accuracy When Disconnection Detection Assistance Is in Use

Using disconnection detection assistance leads to an error in absolute accuracy of the ADC14. This error arises because an erroneous voltage is input to the analog input pins due to the resistive voltage division between the pull-up or pull-down resistor (R_p) and the resistance of the signal source (R_s). This error in absolute accuracy is calculated from the following formula:

Maximum error in absolute accuracy (LSB) = $4095 \times R_s/R_p$

Only use disconnection detection assistance after thorough evaluation.

44.7.9 Notes on Canceling Software Standby Mode

After software standby mode is canceled, wait at least 1 μ s after the stabilization time for the oscillator elapses and before starting A/D conversion. For details, see [section 13, Power-Saving Functions](#)

44.7.10 Notes on Changing the Reference Voltage

If the reference power supply pin (VREFH0) selected as the high-potential reference voltage of the ADC14 is changed to AVCC0, calibration must be performed again when the offset calibration function is used.

45. Reference Voltage Generation Circuit (VREF)

45.1 Overview

The MCU incorporates a reference voltage generation circuit (VREF) that can be used as the reference voltage for a 14-bit A/D converter.

Table 45.1 shows the reference voltage generation circuit specifications, Figure 45.1 shows a block diagram, and Table 45.2 shows the I/O pins.

Table 45.1 VREF specifications

Parameter	Specifications
Output voltage	1.25 V or 2.5 V (AVCC0 ≥ 2.8 V) 1.25 V (AVCC0 < 2.8 V)
Function for reducing power consumption	The module-stop state can be set

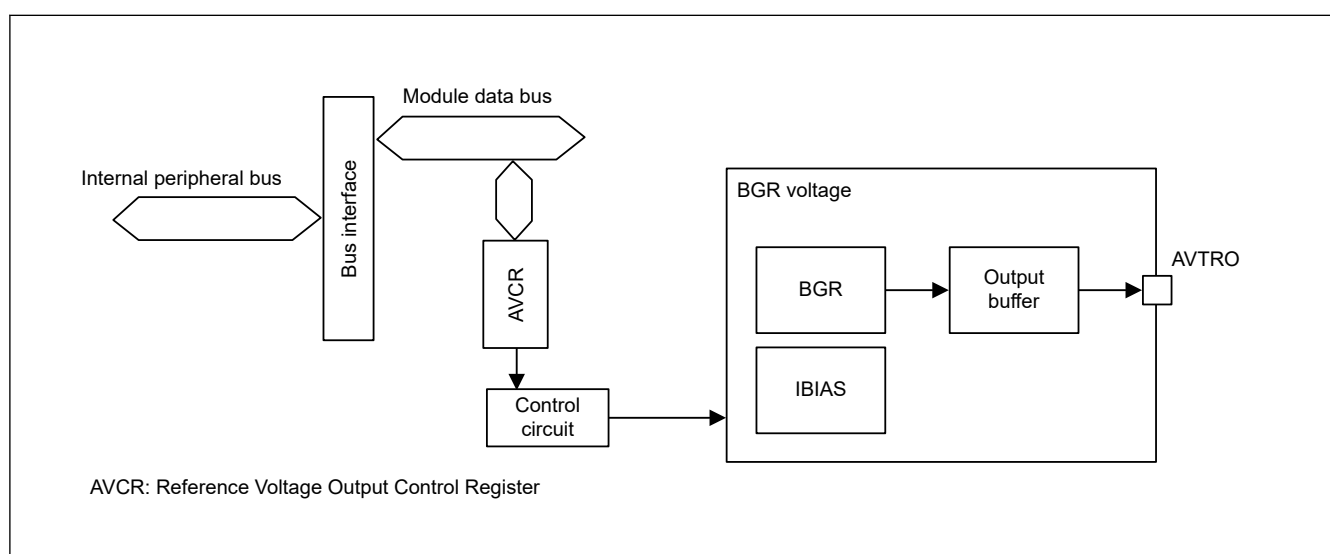


Figure 45.1 VREF block diagram

Table 45.2 VREF I/O pins

Pin name	I/O	Function
AVTRO	Output	Reference voltage output pin

45.2 Register Descriptions

45.2.1 AVCR : Reference Voltage Output Control Register

Base address: VREF = 0x4008_6A00

Offset address: 0x80

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	STDM D	—	LPMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	AVSEL	—	—	—	VREF EN	—	—	—	IBIAS EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IBIASEN	Reference Current Source Enable* ¹ 0: Reference current source stop 1: Reference current source start	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	VREFEN	Reference Voltage Generation Circuit Enable* ¹ 0: Reference voltage output stop 1: Reference voltage output start	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	AVSEL	Output Select 0: 1.25 V output from the AVTRO pin 1: 2.5 V output from the AVTRO pin* ²	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	LPMD	Low Power Consumption Mode 0: Standard operation mode (initial setting) 1: Low power consumption mode * ³	R/W
17	—	This bit is read as 0. The write value should be 0.	R/W
18	STDMD	Standard Operation Mode 0: Low power consumption mode 1: Standard operation mode (initial setting)* ³	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Always set the IBIASEN bit and the VREFEN bit to the same value.

Note 2. When AVCC0 < 2.8 V, 2.5 V output cannot be selected. For the lower limit of AVCC0, see [section 51, Electrical Characteristics](#).

Note 3. The LPMD and the STDMD bit settings are mutually exclusive. When either the LPMD or the STDMD bit is 1, always set the other bit to 0. The low power consumption mode can be selected only when [section 44, 14-Bit A/D Converter \(ADC14\)](#) is used and PCLKB is 32.768 kHz.

IBIASEN bit (Reference Current Source Enable)

The IBIASEN bit sets start or stop of the reference current source. Set this bit to the same value as the VREFEN bit.

VREFEN bit (Reference Voltage Generation Circuit Enable)

The VREFEN bit sets start or stop of the reference voltage output. Set this bit to the same value as the IBIASEN bit.

AVSEL bit (Output Select)

The AVSEL bit selects the output value of the reference voltage.

LPMD bit (Low Power Consumption Mode)

Set the LPMD bit to 1 when PCLKB is 32.768 kHz and the 14-bit A/D converter is used. The STDMD bit must be set to 0.

STDMD bit (Standard Operation Mode)

Set the STDMD bit to 1 when PCLKB is not 32.768 kHz.

45.3 Operation

45.3.1 Operating States and Register Settings

[Table 45.3](#) provides the relationship between the AVCR register settings, pin states, and operation states of the VREF. To enable output of the VREF, it is necessary to set both the AVCR.IBIASEN bit and the AVCR.VREFEN bit to 1 at the same time.

Table 45.3 Register settings, pin states, and VREF states (1 of 2)

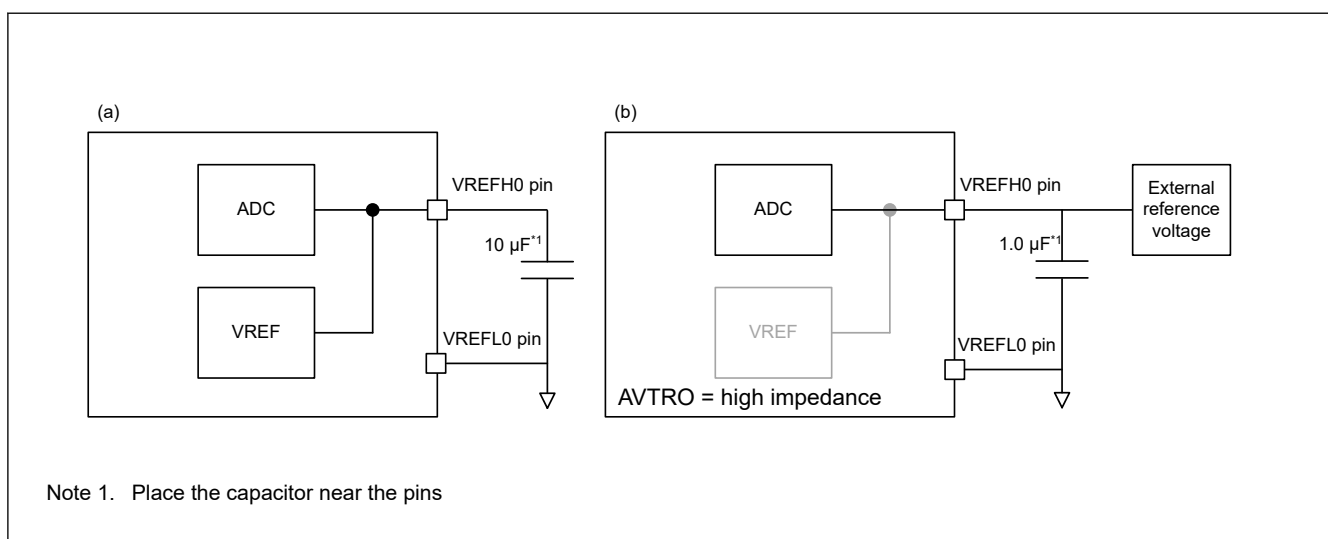
IBIASEN	VREFEN	AVTRO	VREF state
0	0	High impedance	Stopped
0	1	Undefined	Undefined (setting prohibited)

Table 45.3 Register settings, pin states, and VREF states (2 of 2)

IBIASEN	VREFEN	AVTRO	VREF state
1	0	High impedance	Undefined (setting prohibited)
1	1	Output	Operating

45.3.2 Usage

Output voltage of the VREF can be used as a reference voltage for the 14-bit A/D converter. As a reference voltage, the 14-bit A/D converter can select either the VREF output or the external reference voltage supplied from outside of the MCU. About how to set, see [section 44.6. Selecting Reference Voltage](#). [Figure 45.2](#) shows examples of connections. For the connection of the smoothing capacitor, see also [section 1.5. Pin Functions](#) and .

**Figure 45.2 Example connections**

Example (a) in [Figure 45.2](#) shows the case where output from the VREF is used as a reference voltage of the 14-bit A/D converter. External capacitance of approximate 10 µF must be connected to the VREFH0 pin. Set both the AVCR.IBIASEN bit and VREFEN bit to 1, and then wait for the circuit startup stabilization wait time $t_{VREFSTP}$ before use. In Example (b) of the figure, external reference voltage is applied to the VREFH0 pin. Set the register as shown in [Table 45.3](#) so that the AVTRO pin is high impedance.

45.4 Usage Notes

45.4.1 Settings for the Module-Stop Function

The Module Stop Control Register D (MSTPCRD) can enable or disable VREF operation. The VREF module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

45.4.2 Notes on AVCC0 Potential and AVTRO Pin Output

For the electrical characteristics, a restriction of $AVTRO = VREFH0 \leq AVCC0$ applies. When the external capacitance is connected to the VREFH0 pin and the output from the VREF is used as reference voltage for the 14-bit A/D converter (as shown in the connection example (a) of [Figure 45.2](#)), the reference voltage for the 14-bit A/D converter transitions to $AVCC0 < (\text{reference voltage for the 14-bit A/D converter})$. This occurs because of the external capacitance even when the AVCC0 voltage is decreased.

Note: $AVCC0 < (\text{reference voltage for the 14-bit A/D converter})$ is outside of the guaranteed range for the electrical characteristics. For the lower limit of AVCC0, see [section 51, Electrical Characteristics](#).

45.4.3 Notes on 14-bit A/D Converter Usage

When the reference voltage for the 14-bit A/D converter changed, you must perform offset recalibration for the 14-bit A/D converter. For more information, see [section 44, 14-Bit A/D Converter \(ADC14\)](#).

46. Temperature Sensor (TSN)

46.1 Overview

The on-chip Temperature Sensor (TSN) determines and monitors the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is fairly linear. The output voltage is provided to the ADC14 for conversion and can be further used by the end application.

Table 46.1 lists the TSN specifications, and Figure 46.1 shows a block diagram.

Table 46.1 TSN specifications

Item	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the 14-bit A/D converter
Module-stop function	Module stop state can be set
Temperature sensor calibration data	Reference data measured for each chip at factory shipment is stored in a register

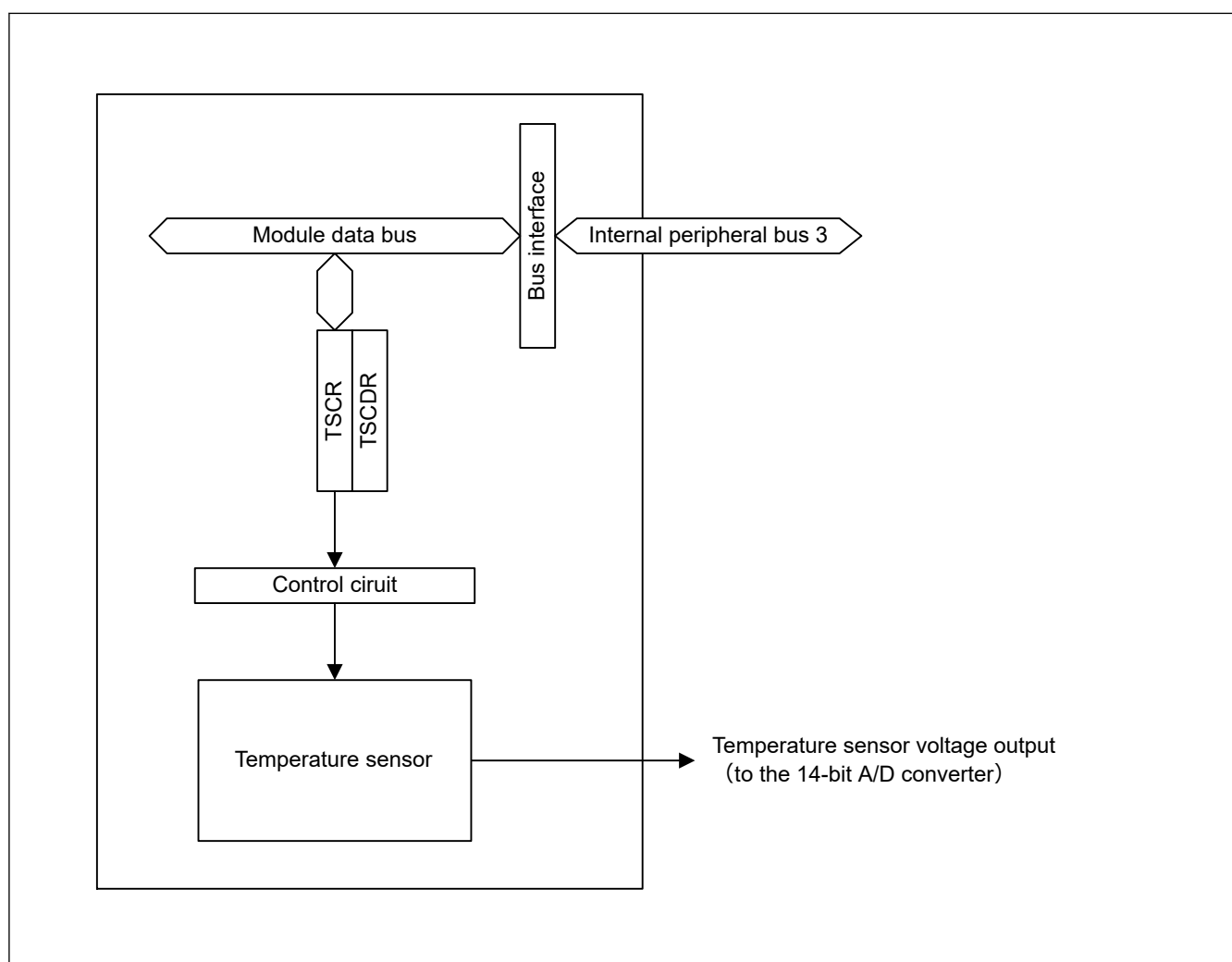


Figure 46.1 TSN block diagram

46.2 Register Descriptions

46.2.1 TSCR : Temperature Sensor Control Register

Base address: TSN = 0x4005_D000

Offset address: 0x00

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	TSEN	Temperature Sensor Enable 0: Stop the temperature sensor 1: Start the temperature sensor.	R/W

The TSCR is a register which controls the temperature sensor. The timing constraints shown in [Figure 46.4](#) apply to the settings of the TSCR register.

TSEN bit (Temperature Sensor Enable)

The TSEN bit starts or stops the temperature sensor.

46.2.2 TSCDR : Temperature Sensor Calibration Data Register

Address: 0x0100_817C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSCDR[15:0]															
Value after reset:	Chip-specific value															

Bit	Symbol	Function	R/W
15:0	TSCDR[15:0]	Temperature Sensor Calibration Data Chip-specific value	R
31:16	—	These bits are read as 0.	R

The TSCDR register stores temperature sensor calibration data measured for each chip at factory shipment.

Temperature sensor calibration data is the output voltage of the temperature sensor under the conditions $T_j = 97^\circ\text{C}$ and $AVCC0 = 3.3\text{ V}$ converted to a digital value by the 14-bit A/D converter.

The TSCDR register is a read-only 32-bit register. Read from this register in 32-bit units.

46.3 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 14-bit A/D converter. To obtain the die temperature, convert this value into the temperature.

46.3.1 Preparation for Using the Temperature Sensor

The ambient temperature (T) is proportional to the temperature sensor voltage output (V_s), so ambient temperature is calculated with the following formula:

$$T = (V_s - V_1) / \text{slope} + T_1$$

- T: Ambient temperature of MCU as calculation result (°C)
- Vs: Voltage output by the temperature sensor on temperature measurement (V)
- T1: Temperature experimentally measured at one point (°C)
- V1: Voltage output by the temperature sensor on measurement of T1 (V)
- T2: Temperature experimentally measured at a second point (°C)
- V2: Voltage output by the temperature sensor on measurement of T2 (V)
- Slope: Temperature gradient of the temperature sensor (V / °C), slope = (V2 - V1) / (T2 - T1)

Characteristics vary between sensors, so Renesas recommends measuring two different sample temperatures as follows:

1. Use the 14-bit A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1.
2. Again use the 14-bit A/D converter to measure the voltage V2 output by the temperature sensor at a different temperature T2.
3. Obtain the temperature gradient (slope = (V2 - V1) / (T2 - T1)) from these results.
4. Subsequently, obtain temperatures by substituting the slope into the formula for the temperature characteristic ($T = (V_s - V_1) / \text{slope} + T_1$).

If you are using the temperature gradient given in [section 51, Electrical Characteristics](#), use the A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1, then calculate the temperature characteristic using the following formula:

$$T = (V_s - V_1) / \text{slope} + T_1$$

- T: Measured temperature (°C)
- Vs: Output voltage by the temperature sensor when the temperature is measured (V)
- T1: Temperature experimentally measured at a first point (°C)
- V1: Output voltage by the temperature sensor on measurement of T1 (V)
- Slope: Temperature gradient of the temperature sensor described in [section 51.5. Temperature Sensor Characteristics](#) ÷ 1000 (V/°C)

Note: This method produces less accurate temperatures than measurement at two points.

In this MCU, the TSCDR register stores the temperature value (CAL₉₇) of the temperature sensor measured under the condition Ta = Tj = 97°C and AVCC0 = 3.3 V. If you use this value as the sample measurement result at the first point, you can omit the preparation before using the temperature sensor.

V1 is calculated from CAL₉₇:

$$V_1 = 3.3 \times \text{CAL}_{97} / 16384 \text{ [V]} \text{ (In case of 14 bit accuracy)}$$

Using this value, the measured temperature can be calculated according to the following formula:

$$T = (V_s - V_1) / \text{slope} + 97 \text{ [°C]}$$

- T: Ambient temperature of MCU as calculation result (°C)
- Vs: Voltage output by the temperature sensor when the temperature is measured (V)
- V1: Voltage output by the temperature sensor when Ta = Tj = 97°C and AVCC0 = 3.3 V (V)
- Slope: Temperature gradient of the temperature sensor^{*1} / 1000 (V/°C)

Note 1. See [section 51, Electrical Characteristics](#)

[Figure 46.2](#) shows the error in the measured temperature. The variation range is 3σ.

Regarding the characteristics of the 14-bit A/D converter, the typical values are used. See [section 51.4. A/D Conversion Characteristics](#).

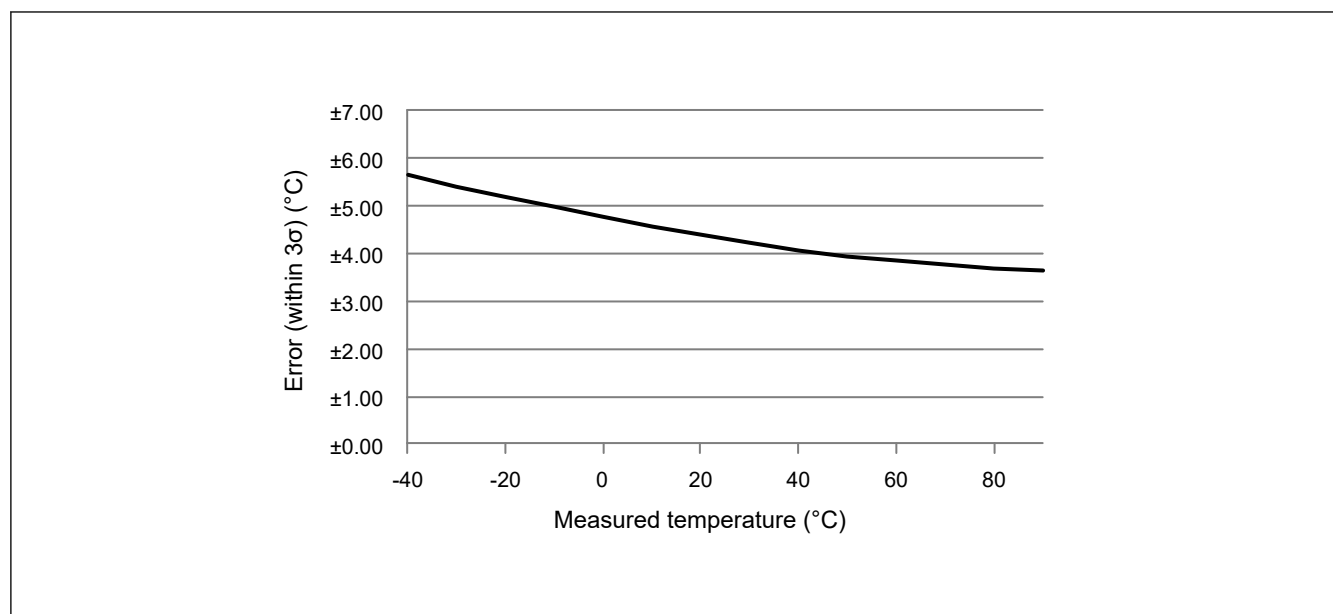
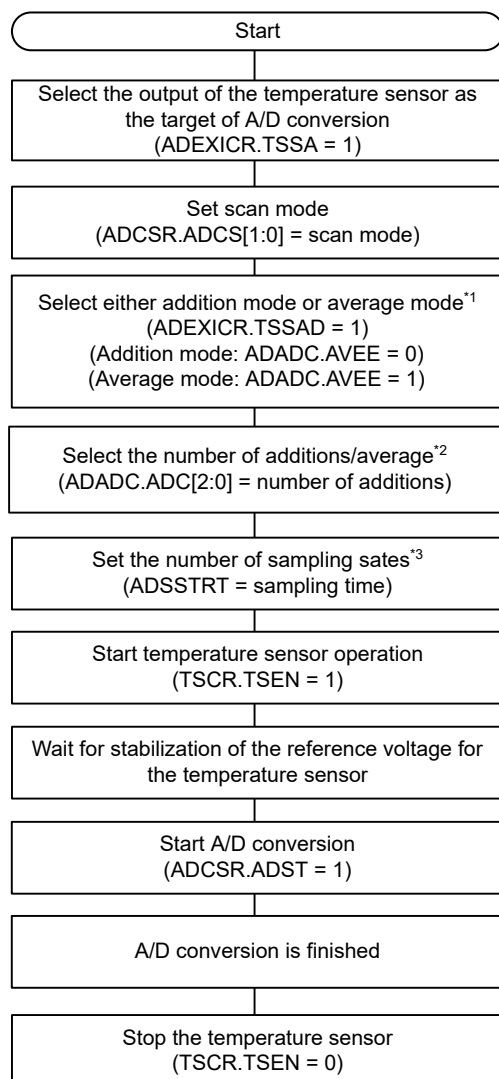


Figure 46.2 Error in the measured temperature (designed values)

46.3.2 Procedures for Using the Temperature Sensor

[Figure 46.3](#) shows the procedure for using the TSN.

For details, see [section 44, 14-Bit A/D Converter \(ADC14\)](#).



Note 1. This setting is not required if addition/average mode is not set.

Note 2. The ADADC.ADC[2:0] setting is limited to some values in additions/average mode. For details on the available ADADC.ADC[2:0] settings in additions/average mode, see [section 44, 14-Bit A/D Converter \(ADC14\)](#).

Note 3. Set the sampling time to more than the value described in [section 51, Electrical Characteristics](#).

Figure 46.3 Procedure example for using the TSN

[Figure 46.4](#) shows the timing from the start of temperature sensor operation until the completion of A/D conversion when the ADC14 is in single scan mode (the conversion target is the temperature sensor output only). The times shown in the figure are described in [Table 46.2](#)

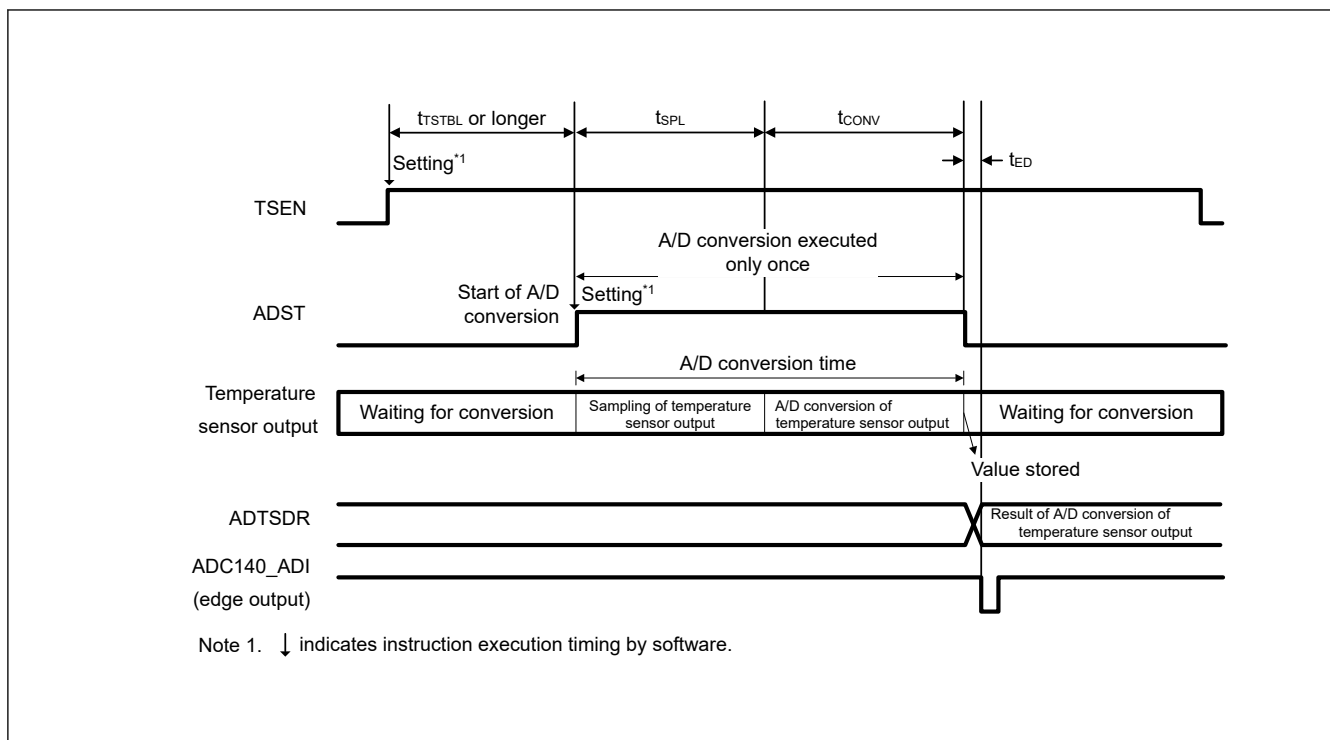


Figure 46.4 Timing from start of temperature sensor operation until completion of A/D conversion

Table 46.2 Time until completion of A/D conversion after start of temperature sensor operation

Parameter	Symbol	Time
Wait time for temperature sensor reference voltage stabilization	t_{STBL}	See section 51, Electrical Characteristics .
A/D converter input sampling time	t_{SPL}	ADSSTRT setting × ADCLK cycles
A/D conversion time	t_{CONV}	See the table in section 44.3.6. Analog Input Sampling and Scan Conversion Time .
Scan conversion end delay	t_{ED}	

46.4 Usage Notes

46.4.1 Settings for the Module-Stop Function

TSN operation can be disabled or enabled using the associated bit in Module Stop Control Register D (MSTPCRD). The TSN is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

47. Data Operation Circuit (DOC)

47.1 Overview

The Data Operation Circuit (DOC) compares, adds, and subtracts 16-bit data. When a selected condition applies, 16-bit data is compared and an interrupt can be generated. Table 47.1 lists the DOC specifications and Figure 47.1 shows a block diagram.

Table 47.1 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 0xFFFF • The result of data subtraction is less than 0x0000

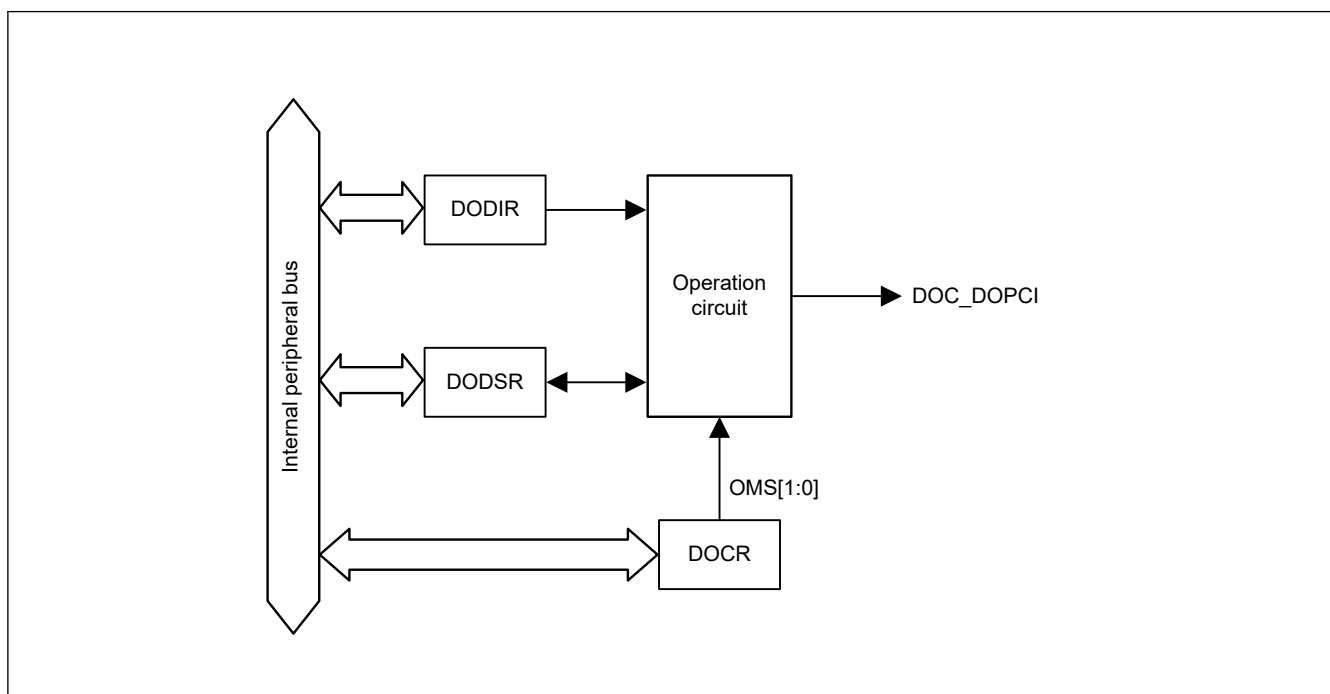


Figure 47.1 DOC block diagram

47.2 DOC Register Descriptions

47.2.1 DOCR : DOC Control Register

Base address: DOC = 0x4005_4100

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	DOPC FCL	DOPC F	—	—	DCSE L	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OMS[1:0]	Operating Mode Select 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
2	DCSEL *1	Detection Condition Select 0: Set DOPCF flag when data mismatch is detected 1: Set DOPCF flag when data match is detected	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	DOPCF	DOC Flag Indicates the result of an operation.	R
6	DOPCFCL	DOPCF Clear 0: Retain DOPCF flag state 1: Clear DOPCF flag	R/W
7	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only valid when data comparison mode is selected.

OMS[1:0] bits (Operating Mode Select)

The OMS[1:0] bits select the operating mode of the DOC.

DCSEL bit (Detection Condition Select)

The DCSEL bit selects the detection condition in data comparison mode. This bit is only valid when data comparison mode is selected.

DOPCF flag (DOC Flag)

The DOPCF flag indicates the result of an operation.

[Setting conditions]

- The result of data comparison matches the condition selected in the DCSEL bit
- A data addition result is greater than 0xFFFF
- A data subtraction result is less than 0x0000

[Clearing condition]

- Writing 1 to the DOPCFCL bit

DOPCFCL bit (DOPCF Clear)

Setting the DOPCFCL bit to 1 clears the DOPCF flag. This bit is read as 0.

47.2.2 DODIR : DOC Data Input Register

Base address: DOC = 0x4005_4100

Offset address: 0x02

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	n/a	It stores 16-bit data used in the operations.	R/W

47.2.3 DODSR : DOC Data Setting Register

Base address: DOC = 0x4005_4100

Offset address: 0x04

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

Bit	Symbol	Function	R/W
15:0	n/a	It stores 16-bit data used as a reference in data comparison mode. This register also stores the results of operations in data addition and subtraction modes.	R/W

47.3 Operation

47.3.1 Data Comparison Mode

Figure 47.2 shows an example operation in data comparison mode operation by the DOC. The following sequence is an example operation when DCSEL is set to 0 (data mismatch is detected as a result of data comparison):

1. Write 00b to the DOCR.OMS[1:0] bits to select data comparison mode.
2. Set 16-bit reference data in DODSR.
3. Write the 16-bit data for comparison to DODIR.
4. Continue writing the 16-bit data until all data for comparison is written to DODIR.
5. If a value written to DODIR does not match that in DODSR, the DOCR.DOPCF flag is set to 1.

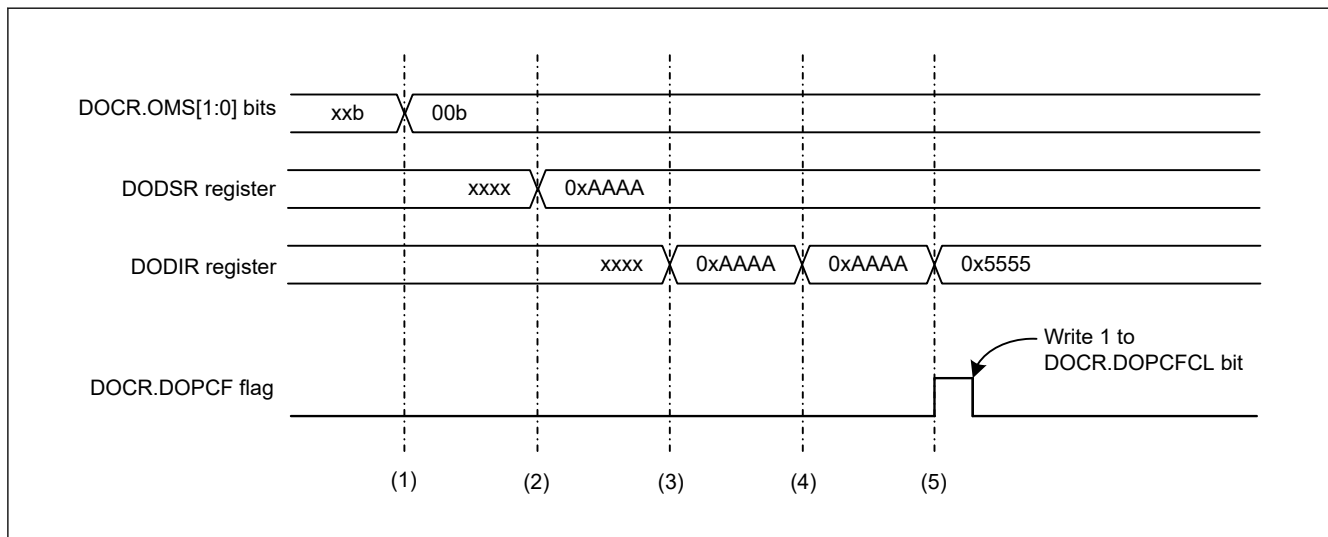


Figure 47.2 Example of operation in data comparison mode

47.3.2 Data Addition Mode

Figure 47.3 shows an example operation in data addition mode. The steps are as follows:

1. Write 01b to the DOCR.OMS[1:0] bits to select data addition mode.
2. Set 16-bit data as the initial value in the DODSR register.
3. Write the 16-bit data to be added to the DODIR register. The result of the operation is stored in the DODSR register.
4. Continue writing the 16-bit data until all data to be added is written to the DODIR.
5. If the result of an operation is greater than 0xFFFF, the DOCR.DOPCF flag is set to 1.

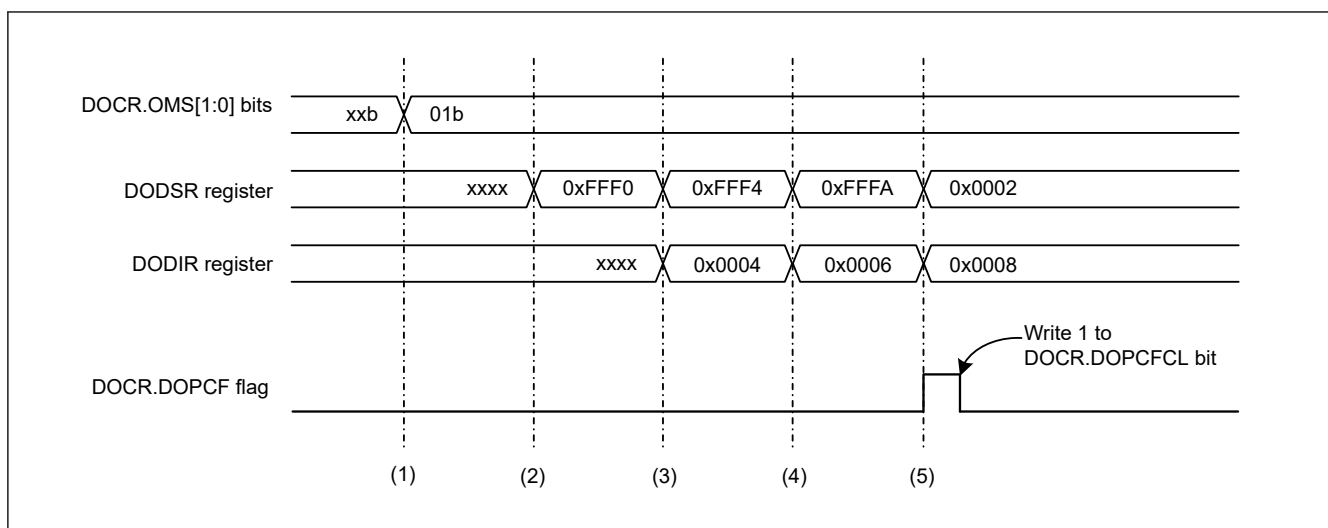


Figure 47.3 Example of operation in data addition mode

47.3.3 Data Subtraction Mode

Figure 47.4 shows an example operation in data subtraction mode. The steps are as follows:

1. Write 10b to the DOCR.OMS[1:0] bits to select data subtraction mode.
2. Set 16-bit data as the initial value in the DODSR register.
3. Write the 16-bit data to be subtracted to the DODIR register. The result of the operation is stored in DODSR.
4. Continue writing the 16-bit data to the DODIR register until all data to be subtracted is written.
5. If the result of an operation is less than 0x0000, the DOCR.DOPCF flag is set to 1.

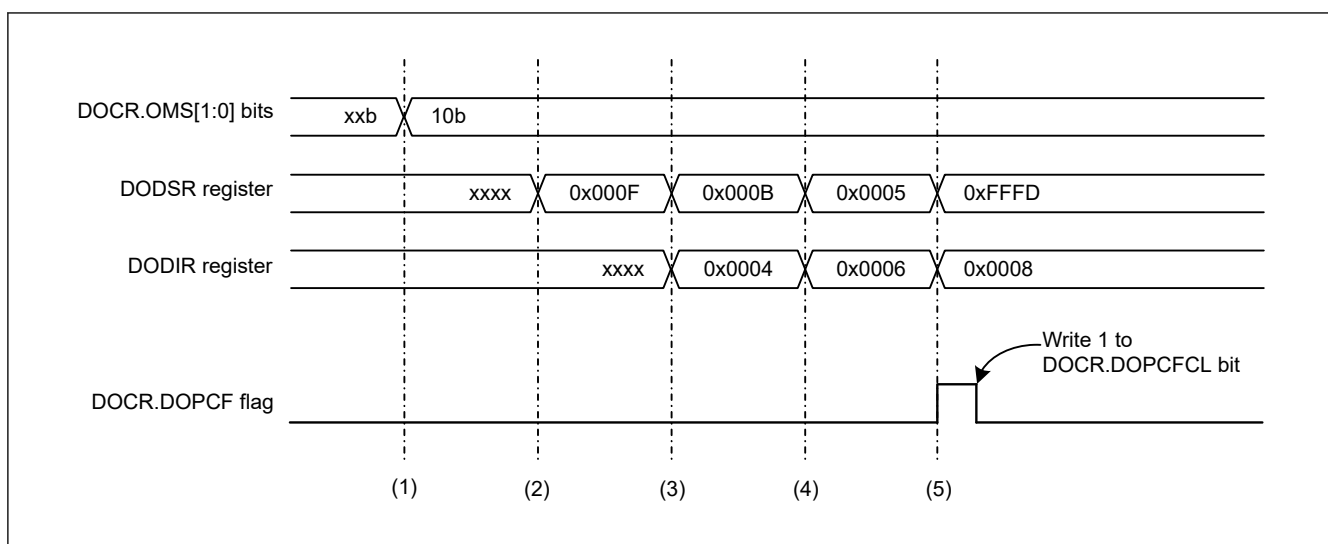


Figure 47.4 Example of operation in data subtraction mode

47.4 Interrupt Source

The DOC generates the DOC interrupt (DOC_DOPCI) as an interrupt request. Table 47.2 describes the DOC interrupt request.

Table 47.2 Interrupt request from DOC

Interrupt request	Status flag	Interrupt source
DOC interrupt	DOPCF	• The result of data comparison matches the condition selected in the DOCR.DCSEL bit. • The result of data addition is greater than 0xFFFF. • The result of data subtraction is less than 0x0000.

47.5 Output of an Event Signal to the Event Link Controller (ELC)

The DOC outputs an event signal for the ELC under the following conditions:

- The compared values either match or mismatch
- The data addition result is greater than 0xFFFF
- The data subtraction result is less than 0x0000

This signal can be used to initiate operations by other modules selected in advance and can also be used as an interrupt request. When an event signal is generated, the DOC Flag (DOCR.DOPCF) is set to 1.

47.6 Usage Notes

47.6.1 Settings for the Module-Stop State

The module Stop Control Register C (MSTPCRC) can enable or disable DOC operation. The DOC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 13, Power-Saving Functions](#).

48. Memory Mirror Function (MMF)

48.1 Overview

You can use the Memory Mirror Function (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. The user application code is developed and linked to run from this MMF destination address. The user application code is not required to know the load location where it is stored in the code flash memory.

Table 48.1 lists the MMF specifications.

Table 48.1 MMF specifications

Parameter	Specifications
Memory mirror space	8 Mbytes 0x0200_0000 to 0x027F_FFFF
Memory mirror boundary	128 bytes

48.2 Register Descriptions

48.2.1 MMSFR : MemMirror Special Function Register

Base address: MMF = 0x4000_1000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	KEY[7:0]								—	MEMMIRADDR[15:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MEMMIRADDR[15:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
22:7	MEMMIRADDR[15:0]	Memory Mirror Address 0x0000 to 0xFFFF (8 Mbytes)	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
31:24	KEY[7:0]	MMSFR Key Code These bits enable or disable rewriting of the MEMMIRADDR[15:0] bits.	W

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

The MEMMIRADDR[15:0] bits specify bits [22:7] of the memory mirror address. These bits configure where the start address of the memory mirror space 0x0200_0000 is linked to. When rewriting these bits, write 0xDB simultaneously 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 retained. These bits are read as 0. The KEY code and the MEMMIRADDR[15:0] bits must be written in the same cycle.

48.2.2 MMEN : MemMirror Enable Register

Base address: MMF = 0x4000_1000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EN	Memory Mirror Function Enable 0: Disable MMF 1: Enable MMF	R/W
23:1	—	These bits are read as 0. The write value should be 0.	R/W
31:24	KEY[7:0]	MMEN Key Code These bits enable or disable rewriting of the EN bit	W

EN bit (Memory Mirror Function Enable)

When rewriting the EN bit, write 0xDB simultaneously to the KEY[7:0] bits.

An attempt at access to the memory mirror space when the EN bit is set to 0 is handled as an illicit address access error. For details, see [section 17.4.3. Conditions for issuing illegal Address Access Errors](#).

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 retained. These bits are read as 0. The KEY code and the EN bit must be written in the same cycle.

48.3 Operation

48.3.1 Memory Mirror Function

The Memory Mirror Function (MMF) links the memory mirror space (0x0200_0000 to 0x027F_FFFF) to the code flash area. If MMEN.EN = 1, the CPU can access code flash memory using both normal addresses (starting at 0x0000_0000) and memory mirror space (starting at 0x0200_0000).

[Figure 48.1](#) shows an overview of the MMF. The MMSFR.MEMMIRADDR[15:0] bits configure where the start address of the memory mirror space (0x0200_0000) is linked to. [Figure 48.2](#), [Figure 48.3](#), and [Figure 48.4](#) show the MMF operation. [Figure 48.5](#) shows the MMF setting procedure.

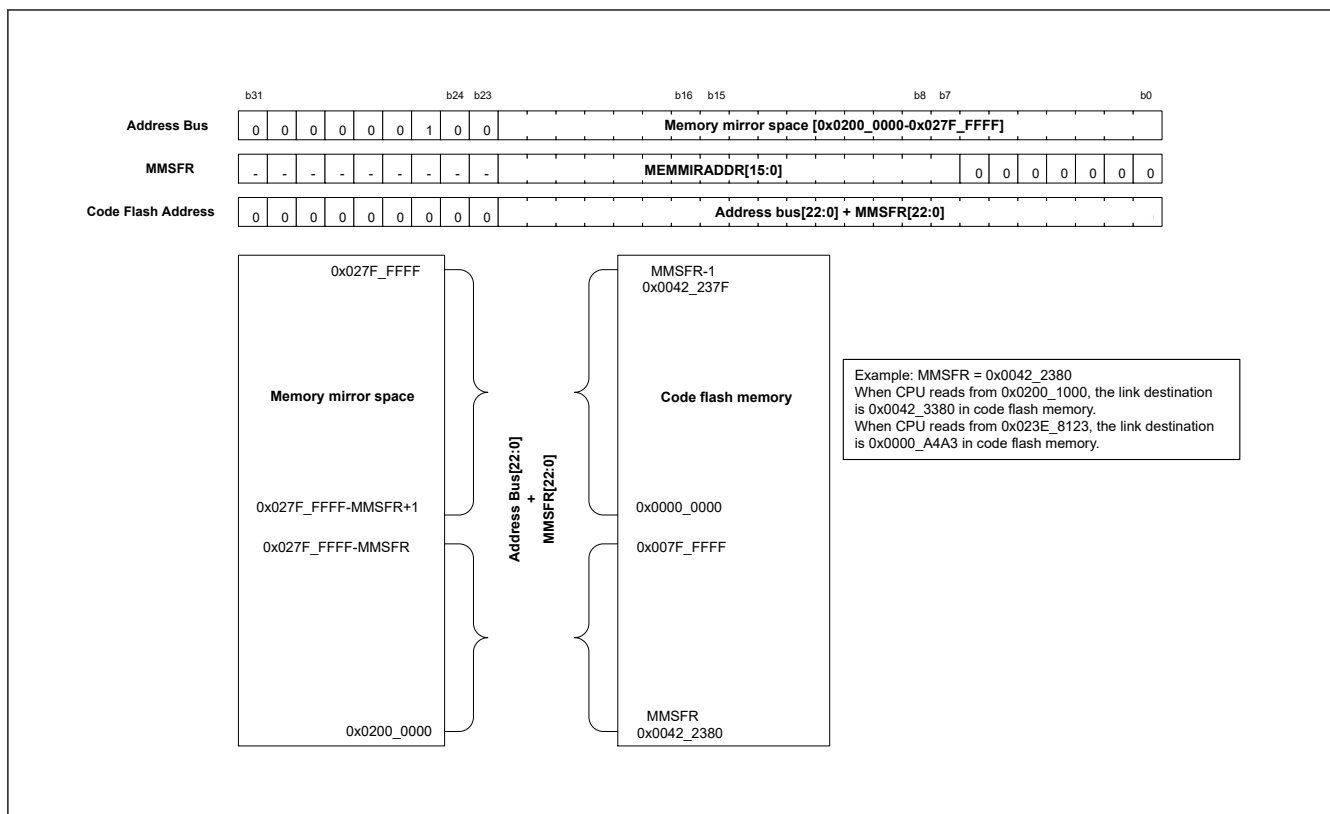
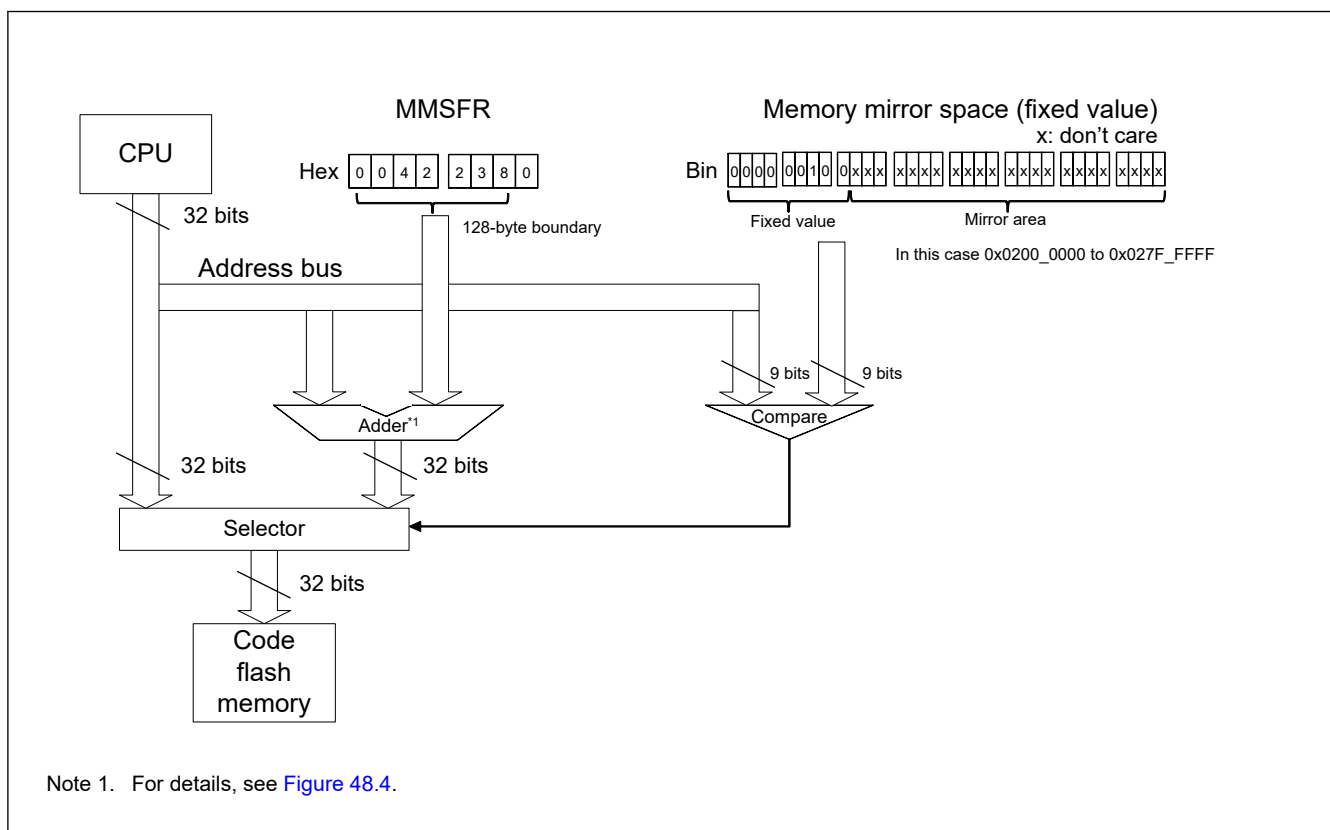


Figure 48.1 MMF overview



Note 1. For details, see [Figure 48.4](#).

Figure 48.2 MMF block diagram

Figure 48.3 shows the addresses handled by each module. The Arm® MPU uses the original address of the CPU. The security MPU and code flash memory through the memory mirror function use an address after the conversion.

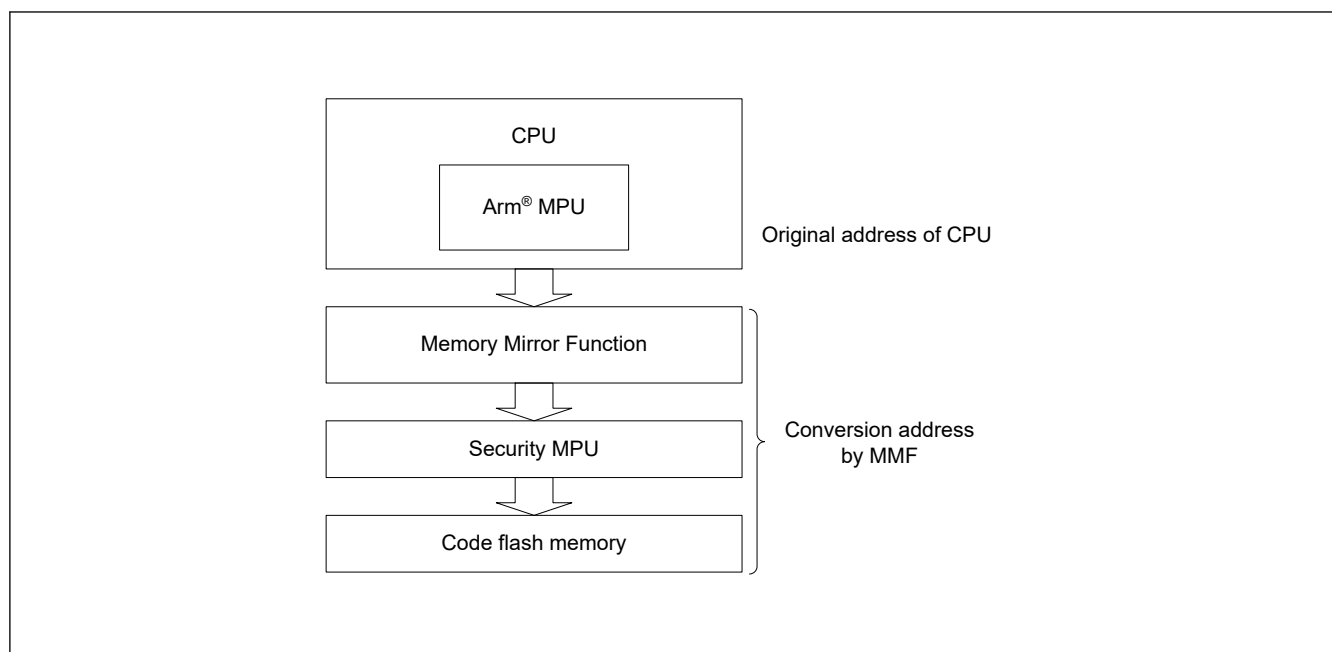


Figure 48.3 MMF address conversion

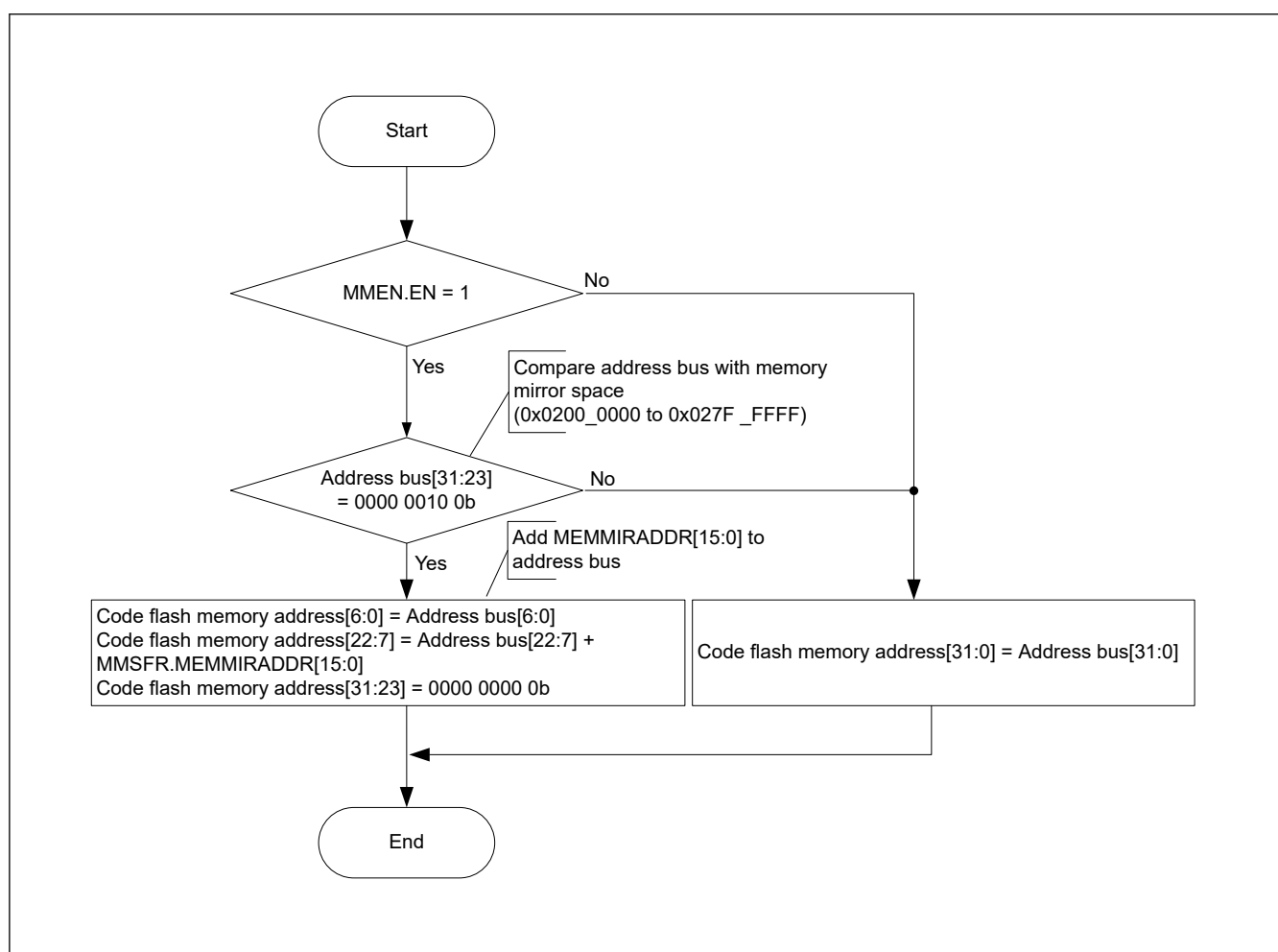


Figure 48.4 MMF flowchart

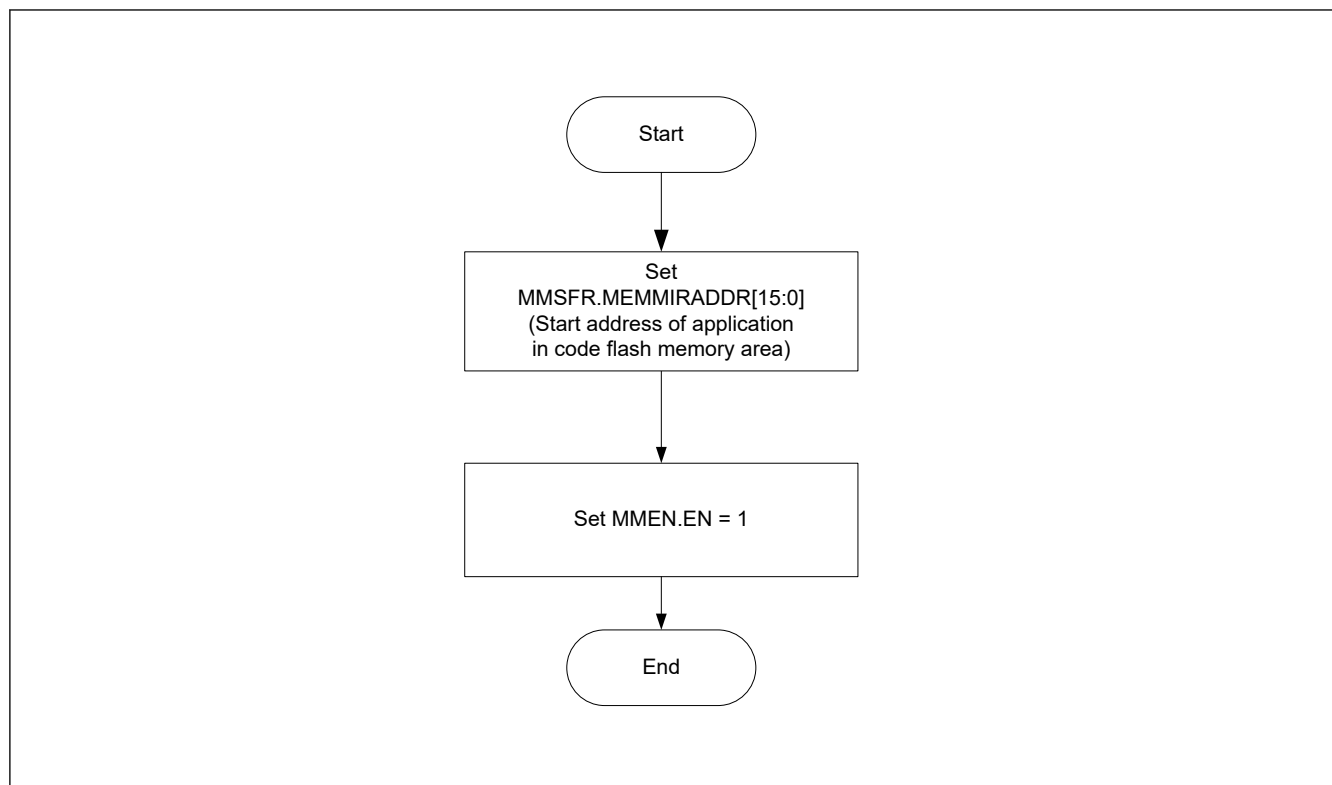
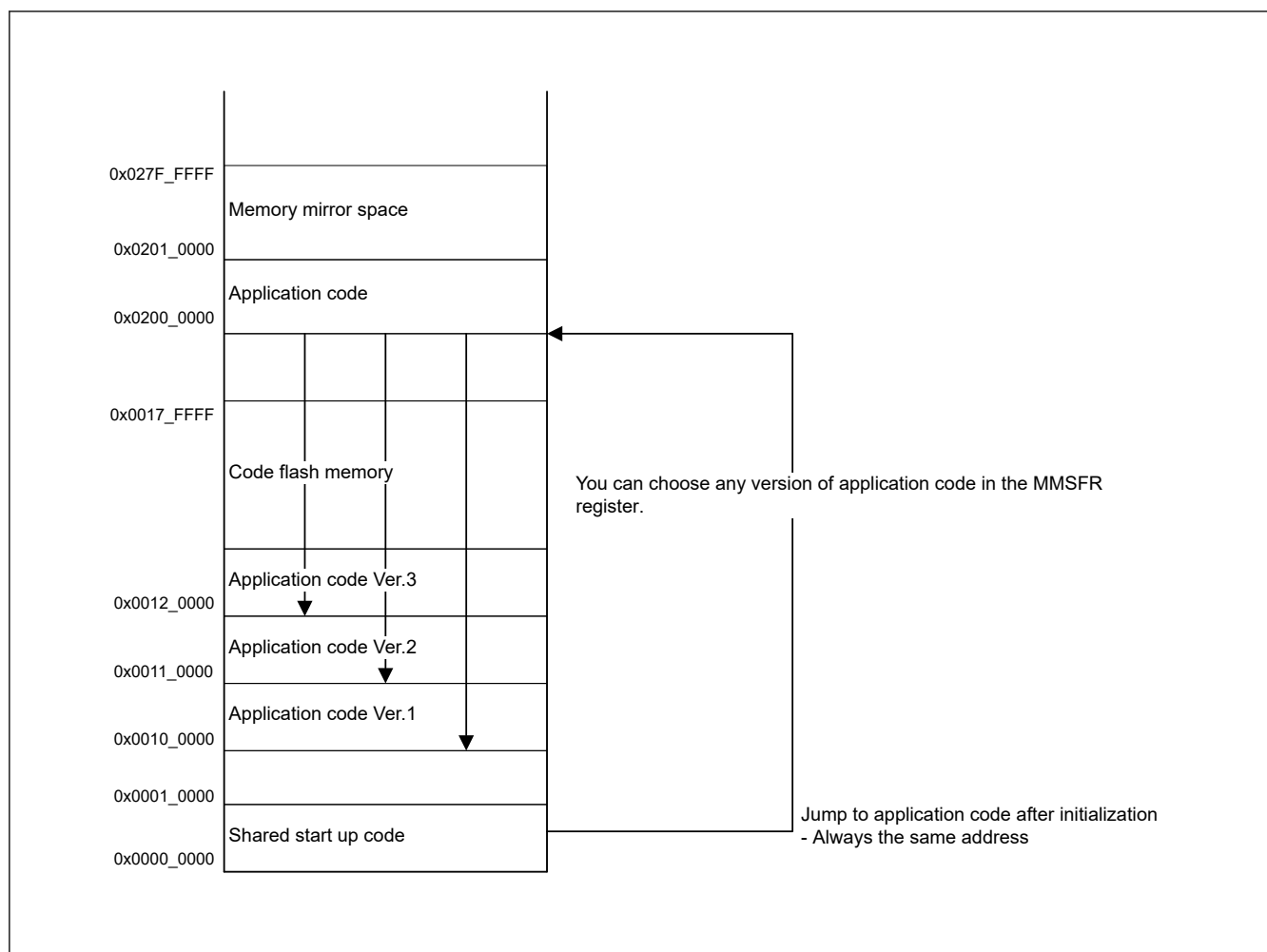


Figure 48.5 MMF setting procedure

48.3.2 Setting Example

[Figure 48.6](#) shows an example of how to use the MMF.

**Figure 48.6 MMF example**

The application code in the MMSFR.MEMMIRADDR[15:0] address on the code flash memory can be accessed from address 0x0200_0000 on the memory mirror space by the MMSFR.MEMMIRADDR[15:0] and MMEN.EN = 1 bit settings.

Example settings for the MMSFR.MEMMIRADDR memory mirror addresses:

- When ver. 1 of the application code is selected, set the MMSFR register to 0xDB100000.
- When ver. 2 of the application code is selected, set the MMSFR register to 0xDB110000.
- When ver. 3 of the application code is selected, set the MMSFR register to 0xDB120000.

49. SRAM

49.1 Overview

The MCU provides an on-chip, high-density SRAM module with parity-bit checking. Parity check is performed on the all SRAM areas.

[Table 49.1](#) lists the SRAM specifications.

Table 49.1 SRAM specifications

Parameter	Description
SRAM capacity	Total : 128 KB SRAM0: 32 KB SRAM1: 96 KB
SRAM address	SRAM0: 0x2000_0000 to 0x2000_7FFF SRAM1: 0x2000_8000 to 0x2001_FFFF
Access*1	0 wait for both reading and writing
RAM shut-off function	The RAM can be shut off in Software Standby mode*2

Note: SRAM0 and Trace RAM are shared. For the Trace RAM specifications, see *CoreSight™ MTB-M0+ Technical Reference Manual (Revision: r0p1)*.

Note 1. For details, see [section 49.3.1. Access Cycle](#).

Note 2. The RAM shut-off function can be selected in the RAMSDCR register. For details, see [section 13, Power-Saving Functions](#).

49.2 Register Descriptions

49.2.1 Trace Control (for the MTB)

The Micro Trace Buffer (MTB) has programmable registers to control the behavior of the trace features and the POSITION, MASTER, FLOW, and BASE registers. [Table 49.2](#) shows the registers in offset order from the base address.

Table 49.2 Address of MTB registers

Address	Register	Value on reset
MTB_BASE + 0x000	MTB_POSITION	Bits [31:0] = UNKNOWN
MTB_BASE + 0x004	MTB_MASTER	Bits [31] = 0, Bits [30:10] = UNKNOWN, Bits [9:8] = 0, Bits [7] = 1, Bits [6:5] = 0, Bits [4:0] = UNKNOWN
MTB_BASE + 0x008	MTB_FLOW	Bits [31:2] = UNKNOWN, Bits [1:0] = 0
MTB_BASE + 0x00C	MTB_BASE	Bits [31:0] = 0x2000_0000

Note: MTB_BASE = 0x4001_9000

For more information on these registers, see the *CoreSight™ MTB-M0+ Technical Reference Manual (Revision: r0p1)*.

Note: Do not attempt to access reserved or unused address locations.

The MTB for trace is limited from 0x2000_0000 to 0x2000_7FFF.

49.2.2 CoreSight™ (for MTB)

See the *ARM® CoreSight™ Architecture Specification* for more information about the registers and access types. [Table 49.3](#) shows the registers in offset order from the base address.

Table 49.3 Address of CoreSight (1 of 2)

Address	Register
MTB_BASE + 0xFF0 to 0xFFC	Component ID
MTB_BASE + 0xFE0 to 0xFDC	Peripheral ID

Table 49.3 Address of CoreSight (2 of 2)

Address	Register
MTB_BASE + 0xFCC	Device Type Identifier
MTB_BASE + 0xFC8	Device Configuration
MTB_BASE + 0xFBC	Device Architecture
MTB_BASE + 0xFB8	Authentication Status
MTB_BASE + 0xFB4	Lock Status
MTB_BASE + 0xFB0	Lock Access

Note: MTB_BASE = 0x4001_9000

For more information on these registers, see the *CoreSight™ MTB-M0+ Technical Reference Manual (Revision: r0p1)*.

Note: Do not attempt to access reserved or unused address locations.

49.3 Operation

49.3.1 Access Cycle

Table 49.4 SRAM access cycle

Read (cycles)		Write (cycles)	
Word access	Halfword/Byte access	Word access	Halfword/Byte access
	2		2

50. Flash Memory

50.1 Overview

This product provides up to 256-Kbyte code flash memory. The flash control unit (FCU) controls the programming and erasure of the code flash memory. The flash application command interface (FACI) controls the FCU in accordance with the specified FACI commands.

[Table 50.1](#) lists the specification of the code flash memory. [Figure 50.1](#) shows a block diagram of the related modules.

Table 50.1 Specifications of Code Flash Memory

Item	Specification
Memory capacity	Up to 256 Kbytes
Read cycle	- ICLK frequency ≤ 32 MHz (normal mode) 1 cycle - ICLK frequency ≤ 64 MHz (boost mode) 2 cycles
Value after erasure	0xFF
Programming/erasing (P/E) method	- Programming and erasing of code flash memory handled by FACI commands specified in the FACI command issuing area (0x407E_0000) - Programming by dedicated flash-memory programmer through a serial interface (serial programming) - Programming of code flash memory by a user program (self-programming)*1.
Security function	Protection against illicit tampering or reading of data in code flash memory
Protection	Protection against erroneous overwriting of code flash memory
Programming and erasing units	8/256-byte units for programming in user area - Block units for erasure in user area.
Other functions	Interrupts can be accepted during self-programming
	An expansion area (option bytes) of code flash memory can be set in the initial device settings
On-board programming (three types)	Programming in serial programming mode (SCI boot mode): - Asynchronous serial interface (SCI9) used - Transfer rate adjusted automatically. - 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 flash memory programming within the user program: - Allows code flash memory programming without resetting the system.
Unique ID	ID code of 16 bytes for every device

Note 1. For the conditions where self-programming is available, see [Table 50.3](#).

50.1.1 Serial Programming Mode

In serial programming mode, the code flash memory can be programmed by using a dedicated flash-memory programmer through an SCI interface. For the conditions where the code flash memory can be programmed, see [Table 50.3](#).

50.1.2 On-chip Debug Mode (OCD Mode)

In on-chip debug mode, the code flash memory can be programmed by using a dedicated flash-memory programmer or an emulator through an SWD interface. For the conditions where the code flash memory can be programmed, see [Table 50.3](#).

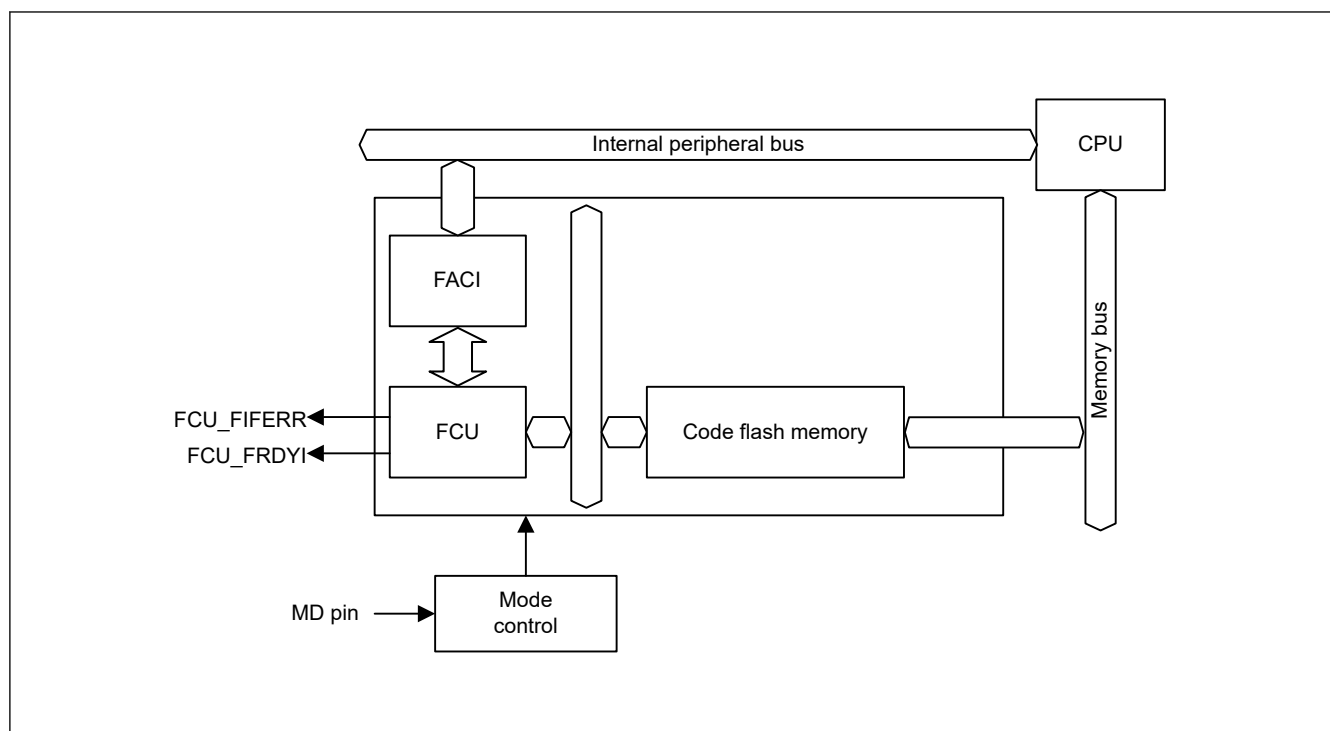


Figure 50.1 Block Diagram of Flash Memory-related Modules

50.2 Memory Structure

Figure 50.2 shows the mapping of the code flash memory, and Table 50.2 shows the read address and programming/ erasure (P/E) addresses by product. The user space of the code flash memory is divided into 4-Kbyte blocks, which serve as the units of erasure. The user area is available for storing user programs

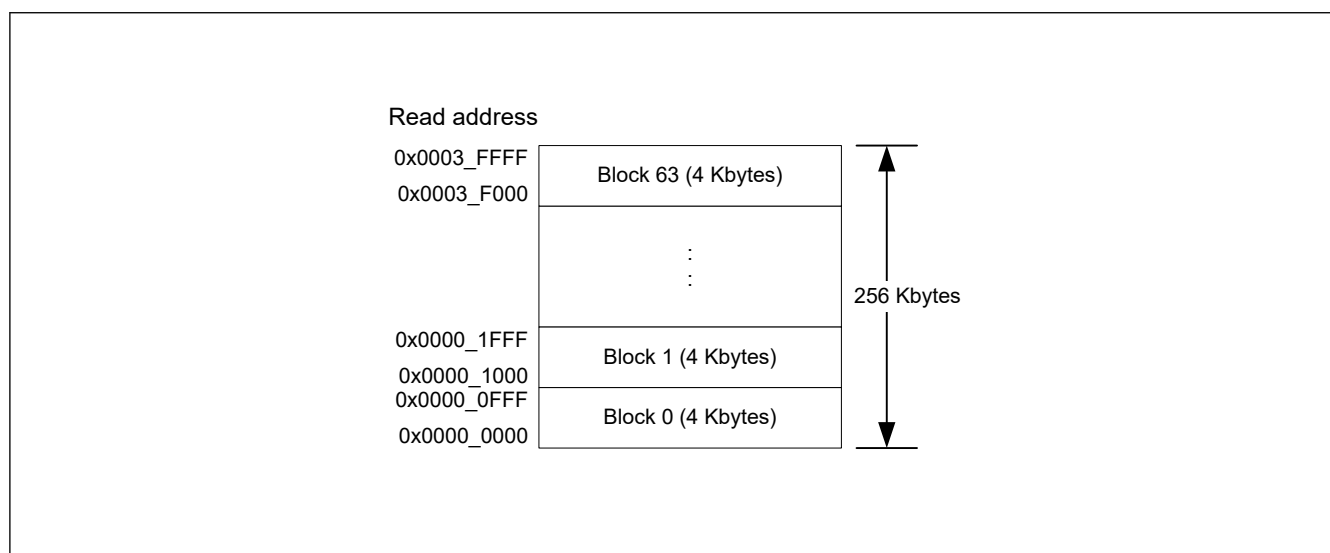


Figure 50.2 Mapping of the Code Flash Memory

Table 50.2 Read and P/E Addresses by Product for the Code Flash Memory

Size of Code Flash Memory	Read Address	P/E Address	Number of Blocks
256-Kbyte product	0x0000_0000 to 0x0003_FFFF	0x0000_0000 to 0x0003_FFFF	0 to 63

50.3 Register Descriptions

50.3.1 FLWT : Flash Wait Cycle Register

Base address: SYSF = 0x4001_C000

Offset address: 0x11C

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

Bit	Symbol	Function	R/W
2:0	FLWT[2:0]	Flash Wait Cycle 0 0 0: No cycles of waiting (ICLK ≤ 32 MHz) 0 0 1: One cycle of waiting (ICLK ≤ 64 MHz)*1 Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: Specify no cycles of waiting when rewriting the flash memory.

Note 1. One cycle of waiting can only be specified in the boost mode. Do not specify one cycle of waiting in the normal mode or low leakage current operation mode.

The FLWT register specifies the access wait count for the flash memory.

When changing the access wait count, be sure to do it while ICLK frequency ≤ 32 MHz. The procedure for setting the FLWT.FLWT[2:0] bits is as follows:

To set ICLK frequency ≤ 64 MHz (one cycle of waiting):

1. Set the FLWT.FLWT[2:0] bits to 001b (one cycle of waiting)
2. Change the ICLK frequency to ≤ 64 MHz

To set ICLK frequency ≤ 32 MHz (no cycles of waiting):

1. Change the ICLK frequency to ≤ 32 MHz
2. Set the FLWT.FLWT[2:0] bits to 000b (no cycles of waiting)

Note: The Settings shown above should be made in boost mode. For details on the boost mode, see [section 13, Power-Saving Functions](#). For details on the change of the ICLK frequency, see [section 9, Clock Generation Circuit](#).

50.3.2 UIDRn : Unique ID Registers n (n = 0 to 3)

Address: 0x0100_8190 + n × 4

Bit position:	31	0
Bit field:	UID	
Value after reset:	Chip-specific value	

Bit	Symbol	Function	R/W
31:0	UID	Unique ID	R

UIDRn is a read-only register that stores an ID code (unique ID) of 16 bytes for identifying each device. UIDRn must be accessed in a word.

50.3.3 FWEPROR : Flash Write Erase Protect Register

Base address: SYSC = 0x4001_E000

Offset address: 0x416

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

Bit	Symbol	Function	R/W
1:0	FLWE[1:0]	Flash Write Erase Enable 0 0: Programming and erasing are disabled. 0 1: Programming and erasing are enabled. 1 0: Programming and erasing are disabled. 1 1: Programming and erasing are disabled.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

The FWEPROR register enables or disables the command execution of programming and erasure of the code flash memory by hardware.

This register is initialized by a reset due to the signal on the RES# pin, a power-on reset, a reset sequence monitor reset, a voltage-monitoring 0 reset, an independent watchdog timer reset, a watchdog timer reset, a voltage-monitoring 1 reset, a voltage-monitoring BAT reset, and a software reset, and by transitions to deep software standby and software standby modes.

To execute commands for programming and erasure, the conditions shown in [Table 50.3](#) should be satisfied.

Table 50.3 Conditions where Self-programming is Available

Power Supply Mode	Power Control Mode	Operating Frequency Range	Flash Read	Flash P/E	Note
ALLPWON	Boost	32 MHz to 64 MHz	✓	—	—
		1 MHz to 32 MHz	✓	✓	—
		32.768 kHz to 1 MHz	✓	—	—
		Less than 32.768 kHz	—	—	—
	Normal	1 MHz to 32 MHz	✓	✓	—
		32.768 kHz to 1 MHz	✓	—	—
		Less than 32.768 kHz	—	—	—
	VBB	32.768 kHz	✓	—	—
		Less than 32.768 kHz	—	—	—
EXFPWON		Entire range of operating frequencies	—	—	—
MINPWON		Entire range of operating frequencies	—	—	—

50.3.4 FASTAT : Flash Access Status Register

Base address: FACL = 0x407F_E000

Offset address: 0x10

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

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	CMDLK	Command Lock Flag 0: The flash sequencer is not in the command-locked state 1: The flash sequencer is in the command-locked state.	R
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	CFAE	Code Flash Memory Access Violation Flag 0: No code flash memory access violation has occurred 1: A code flash memory access violation has occurred.	R/W ^{*1}

Note 1. Only 0 can be written to clear the flag after 1 is read.

The FASTAT register indicates whether the access violation of a code flash memory has occurred. If either of the CFAE and CMDLK flags is set to 1, the flash sequencer enters the command-locked state (see [section 50.6.3.2. Relationship between the Flash Sequencer State and FACL Commands](#)). To release it from the command-locked state, a status clear command or a forced stop command must be issued by the FACL.

CMDLK bit (Command Lock Flag)

The CMDLK bit indicates that the flash sequencer is in the command-locked state.

[Setting conditions]

- The flash sequencer detects an error and enters the command-locked state.

[Clearing conditions]

- When the flash sequencer starts to process the status clear or forced stop command.

CFAE bit (Code Flash Memory Access Violation Flag)

The CFAE bit indicates whether a code flash memory access violation has occurred. When this bit is set to 1, the ILGLERR bit in the FSTATR register is set to 1, placing the flash sequencer in the command-locked state.

[Setting conditions]

- When an FACL command is issued to the reserved area or unmounted area in the code flash P/E mode.

[Clearing conditions]

- After 1 is read, when 0 is written or when a status clear command or a forced stop command is issued.

50.3.5 FAEINT : Flash Access Error Interrupt Enable Register

Base address: FACL = 0x407F_E000

Offset address: 0x14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CFAEI E	—	—	CMDL KIE	—	—	—	—
Value after reset:	1	0	0	1	1	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
4	CMDLKIE	Command Lock Interrupt Enable 0: Generation of an FCU_FIFERR interrupt request is disabled when FASTAT.CMDLK is set to 1 1: Generation of an FCU_FIFERR interrupt request is enabled when FASTAT.CMDLK is set to 1.	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W

The FAEINT register enables or disables generation of a flash access error (FCU_FIFERR) interrupt request.

The CMDLKIE bit enables or disables generation of an FCU_FIFERR interrupt request when the flash sequencer enters the command-locked state, setting the CMDLK bit in the FASTAT register to 1.

The CFAEIE bit enables or disables generation of an FCU_FIFERR interrupt request when a code flash memory access violation occur, setting the CFAE bit in the FASTAT register to 1.

Offset address: 0x18

Value after reset: 0 0 0 0 0 0 0 0

The FRDYIE register enables or disables generation of a flash ready (FCU FRDYI) interrupt.

The FRDYIE bit enables or disables the generation of an FCU_FRDYI interrupt request when the flash sequencer completes the writing/erasure command processing and the FSTATR.FRDY flag is changed from 0 to 1.

Offset address: 0x30

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Note 1. Writing to these bits is only possible when the FRDY bit in the FSTATR register is 1. Writing to these bits while the FRDY bit = 0 is ignored. Note that bit [0] and bit [1] are read-only.

The FSADDR register specifies the address where the target area for command processing starts when the FACL command for programming, block erasure or configuration setting is issued.

The FSADDR value is initialized when the SUINIT bit in the FSUINITR register is set to 1. It is also initialized by a reset.

FSADDR[31:0] bits (Start Address for FACL Command Processing)

Bits 31 to 24 are ignored in FACL command processing for the code flash memory. Bits corresponding to address bits of lower order than the corresponding boundary listed below are also ignored.

Command	Size	Address boundary
Programming (Code flash memory)	8 bytes	8 bytes
	256 bytes	256 bytes
Block erasure (Code flash memory)	—	4 Kbyte
Configuration setting	—	16 bytes

50.3.8 FSTATR : Flash Status Register

Base address: FACL = 0x407F_E000

Offset address: 0x80

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	ILGCO MERR	FESE TERR	SECE RR	OTER R	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FRDY	ILGLE RR	ERSE RR	PRGE RR	SUSRDY	DBFUL L	ERSS PD	PRGS PD	—	FLWE ERR	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	FLWEERR	Flash Write/Erase Protect Error Flag 0: An error has not occurred 1: An error has occurred.	R
7	—	These bits are read as 0. The write value should be 0.	R/W
8	PRGSPD	Programming Suspend Status Flag 0: The flash sequencer is in a state other than those corresponding to the value 1 1: The flash sequencer is in the programming suspension processing state or programming suspended state.	R
9	ERSSPD	Erase Suspend Status Flag 0: The flash sequencer is in a state other than those corresponding to the value 1 1: The flash sequencer is in the erasure suspension processing state or the erasure suspended state.	R
10	DBFULL	Data Buffer Full Flag 0: The data buffer is empty 1: The data buffer is full.	R
11	SUSRDY	Suspend Ready Flag 0: The flash sequencer cannot receive P/E suspend commands 1: The flash sequencer can receive P/E suspend commands.	R
12	PRGERR	Programming Error Flag 0: Programming has completed successfully 1: An error has occurred during programming.	R
13	ERSERR	Erase Error Flag 0: Erasure has completed successfully 1: An error has occurred during erasure.	R

Bit	Symbol	Function	R/W
14	ILGLERR	Illegal Command Error Flag 0: The flash sequencer has not detected an illegal FACL command or illegal flash memory access 1: The flash sequencer has detected an illegal FACL command or illegal flash memory access.	R
15	FRDY	Flash Ready Flag 0: Program, block erase, P/E suspend, P/E resume, forced stop, or configuration set command processing is in progress 1: None of the above is in progress.	R
19:16	—	These bits are read as 0. The write value should be 0.	R/W
20	OTERR	Other Error 0: A status clear or forced stop command processing is complete 1: An error has occurred.	R
21	SECERR	Security Error 0: A status clear or forced stop command processing is complete 1: An error has occurred.	R
22	FESETERR	FENTRY Setting Error 0: A status clear or forced stop command processing is complete 1: An error has occurred.	R
23	ILGCOMERR	Illegal Command Error 0: A status clear or forced stop command processing is complete 1: An error has occurred.	R
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The FSTATR register indicates the state of the flash sequencer.

FLWEERR flag (Flash Write/Erase Protect Error Flag)

The FLWEERR flag indicates a violation of the flash memory overwrite protection setting in the FWEPROR register. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- When a program/erase command is executed during program/erase prohibition is set

[Clearing condition]

- When the flash sequencer starts processing the forced stop command.

PRGSPD flag (Programming Suspend Status Flag)

The PRGSPD flag indicates that the flash sequencer is in the programming suspension processing state or programming suspended state.

[Setting condition]

- When the flash sequencer starts processing in response to the programming suspend command.

[Clearing conditions]

- When the flash sequencer has received a P/E resume command (when the write access to the FACL command-issuing area is completed)
- When the flash sequencer starts processing the forced stop command.

ERSSPD flag (Erasure Suspend Status Flag)

The ERSSPD flag indicates that the flash sequencer is in the erasure suspension processing state or erasure suspended state.

[Setting condition]

- When the flash sequencer starts the erasure suspension processing during an execution of an erasure command.

[Clearing condition]

- When the flash sequencer has received a P/E resume command (when the write access to the FACI command-issuing area is completed).
- When the flash sequencer starts the processing of a forced stop command.

DBFULL flag (Data Buffer Full Flag)

The DBFULL flag indicates the state of the data buffer when the program command is issued. The flash sequencer incorporates a buffer for write data (data buffer). When data for writing to the flash memory are written to the FACI command-issuing area while the data buffer is full, the flash sequencer inserts a wait cycle in the peripheral bus.

[Setting condition]

- When the data buffer becomes full while a programming command or a configuration setting command is being issued.

[Clearing condition]

- When the data buffer becomes empty.

SUSRDY flag (Suspend Ready Flag)

The SUSRDY flag indicates whether the flash sequencer can receive a P/E suspend command.

[Setting condition]

- After starting programming/erasure processing, the flash sequencer enters a state in which P/E suspend commands can be received.

[Clearing conditions]

- When the flash sequencer accepts the P/E suspend command or forced stop command (when the write access to the FACI command-issuing area is completed)
- When the flash sequencer enters the command-locked state during writing or erasure
- When writing or erasure is completed

PRGERR flag (Programming Error Flag)

The PRGERR flag indicates the result of programming of the flash memory. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- When an error has occurred during programming.

[Clearing condition]

- When the flash sequencer starts processing of the status clear or forced stop command.

ERSERR flag (Erasure Error Flag)

The ERSERR flag indicates the result of erasure of the flash memory. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- When an error has occurred during erasure.

[Clearing condition]

- When the flash sequencer starts processing of the status clear or forced stop command.

ILGLERR flag (Illegal Command Error Flag)

The ILGLERR flag indicates that the flash sequencer has detected an illegal FACI command or flash memory access. If this flag is 1, the flash sequencer is in the command-locked state.

[Setting conditions]

- When the flash sequencer detects an illegal command
- When the flash sequencer detects illegal flash memory access

- When the setting of the FENTRYR register is detected as illegal

[Clearing condition]

- When the flash sequencer starts the processing of a status clear or a forced stop command.

FRDY flag (Flash Ready Flag)

The FRDY flag indicates the command processing state of the flash sequencer.

[Setting conditions]

- When the flash sequencer completes command processing
- When the flash sequencer receives a P/E suspend command and suspends programming of the flash memory
- When the flash sequencer received the forced stop command and ended command processing.

[Clearing conditions]

- When the flash sequencer received an FACY command
- In the case of the setting of the program and configuration, when the first write access is made to the FACY command-issuing area
- In the case of other commands, when the last write access is made to the FACY command issuing area.

OTERR flag (Other Error)

When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- When an error is detected.

[Clearing condition]

- When the processing of a status clear or a forced stop command is executed.

SECERR flag (Security Error)

When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- When an error is detected.

[Clearing condition]

- When the processing of a status clear or a forced stop command is executed.

FESERR flag (FENTRY Setting Error)

When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- When an error is detected.

[Clearing condition]

- When the processing of a status clear or a forced stop command is executed.

ILGCOMERR flag (Illegal Command Error)

When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- When an error is detected.

[Clearing condition]

- When the processing of a status clear or a forced stop command is executed.

50.3.9 FENTRYR : Flash P/E Mode Entry Register

Base address: FACL = 0x407F_E000

Offset address: 0x84

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

Bit	Symbol	Function	R/W
0	FENTRYC	Code Flash P/E Mode Entry 0: Code flash is in read mode 1: Code flash is in P/E mode.	R/W ¹
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code The KEY[7:0] bits enable or disable rewriting of the FENTRYC bit.	R/W ²

Note 1. Writing to these bits is possible only when the FSTATR.FRDY flag is 1. Writing to these bits while the FRDY flag = 0 is ignored.
Writing to these bits is possible only when 0xAA is written to the KEY[7:0] bits in a 16-bit access.

Note 2. Written values are not retained by these bits (always read 0x00).

The FENTRYR register is used to specify the code flash P/E mode.

To specify the code flash P/E mode so that the flash sequencer can receive FACL commands, set the FENTRYC bit to 1 to place the flash sequencer in the code flash P/E mode.

The FENTRYR register is initialized when the FSUINITR.SUINIT bit is set to 1.

In the case where writing to the FENTRYR register is executed and the value of the FENTRYC bit is modified, read the FENTRYR register to confirm that the desired values have been written to the FENTRYR register before programming/erasing or accessing the flash memory. Note that values written to the FENTRYR register cannot be read immediately after writing. In such cases, read the FENTRYR register a short time later.

In the case where the set value of the FENTRYC bit has been changed from 1 to 0, confirm it has been certainly changed to 0 before branching to a program on the code flash memory.

Before changing the set value of the FENTRYC bit from 0 to 1, execute a program allocated to memory such as internal SRAM (other than the code flash memory) so that the code flash is not accessed.

FENTRYC bit (Code Flash P/E Mode Entry)

The FENTRYC bit specifies P/E mode for the code flash memory.

[Setting condition]

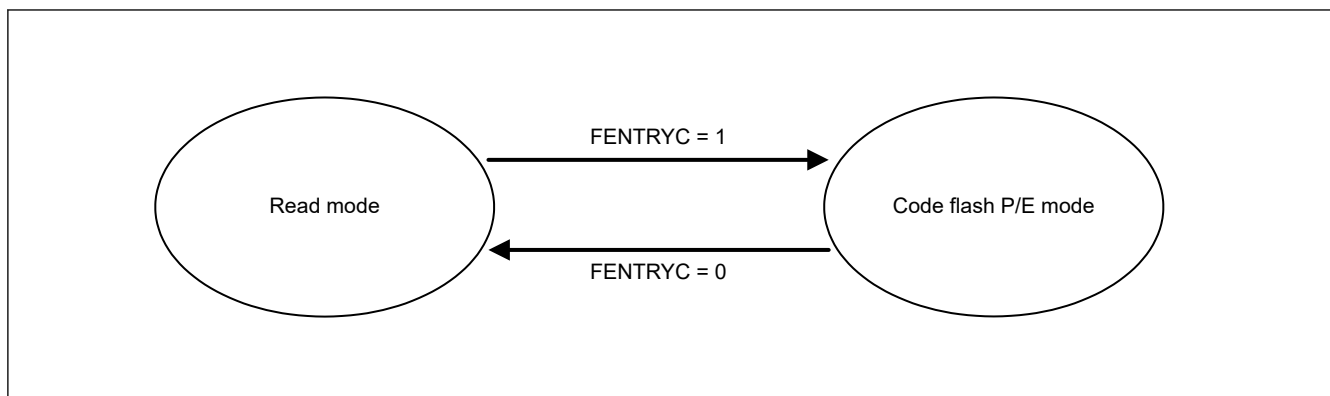
- When write 1 to the FENTRYC bit while writing to FENTRYR is enabled and FENTRYR is 0x0000.

[Clearing conditions]

- Write 8 bits to FENTRYR while the FRDY bit is 1
- When a value other than 0xAA is specified in the KEY[7:0] bits and 16 bits are written to FENTRYR while the FRDY bit is 1
- When 0 is written to the FENTRYC bit while writing to the FENTRYR register is enabled
- When the FENTRYR register is written while writing to the FENTRYR register is enabled and the value of the FENTRYR register is other than 0x0000

The flash sequencer is provided with modes shown in [Figure 50.3](#). The mode transition is executed by writing to the FENTRYR register. When the FENTRYC bit is 0, the flash sequencer is in the read mode. In the read mode, the FACL command is not received. Data can be read out to the code flash memory.

When the FENTRYC bit is 1, the flash sequencer is in the code flash P/E mode. In the code flash P/E mode, the FACL command can be used to write/erase data on the code flash memory.

**Figure 50.3 Flash Sequencer Mode**

50.3.10 FSUINTR : Flash Sequencer Setup Initialization Register

Base address: FACL = 0x407F_E000

Offset address: 0x8C

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

Bit	Symbol	Function	R/W
0	SUINIT	Set-Up Initialization This bit initializes the following flash sequencer set-up registers. FSADDR, FENTRYR 0: The above flash sequencer set-up registers keep their current values. 1: The above flash sequencer set-up registers are initialized.	R/W ¹
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code The KEY[7:0] bits enable or disable rewriting of the SUINIT bit.	R/W ²

Note 1. Writing to these bits is possible only when the FSTATR.FRDY flag is 1. Writing to these bits while the FRDY flag = 0 is ignored.
Writing to these bits is possible only when 0x2D is written to the KEY[7:0] bits in a 16-bit access.

Note 2. Written values are not retained by these bits. Read values are always 0x00

The FSUINTR is used for initialization of the flash sequencer setup.

50.3.11 FCMDR : FACL Command Register

Base address: FACL = 0x407F_E000

Offset address: 0xA0

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

Bit	Symbol	Function	R/W
7:0	PCMDR[7:0]	Pre-command Flag These bits store the command received immediately before the last command received by the FACL.	R
15:8	CMDR[7:0]	Command Flag These bits store the latest command received by the FACL.	R

The FCMDR register indicates the command received by the FACL.

Table 50.4 States of FCMDR after receiving commands

Command	CMDR[7:0]	PCMDR[7:0]
Program	0xE8	Previous command
Block erase	0xD0	0x20
P/E suspend	0xB0	Previous command
P/E resume	0xD0	Previous command
Status clear	0x50	Previous command
Forced stop	0xB3	Previous command
Configuration set	0x40	Previous command

50.3.12 FAWMON : Flash Access Window Monitor Register

Base address: FACI = 0x407F_E000

Offset address: 0x0DC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BTFL G	—	—	—	—	—	FAWE[9:0]									
Value after reset:	0/1	0	0	0	0	0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FSPR	—	—	—	—	—	FAWS[9:0]									
Value after reset:	0/1	0	0	0	0	0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

Bit	Symbol	Function	R/W
9:0	FAWS[9:0]	Flash Access Window Start Address These bits indicate the start block address for the access window.	R
14:10	—	These bits are read as 0.	R
15	FSPR	Access Window Protection Flag This bit indicates whether there is protection to the configuration setting command for access window setting and to writing of the FSUACR register. 0: Protection available (execution is impossible) 1: Protection unavailable (execution is possible)	R
25:16	FAWE[9:0]	Flash Access Window End Address These bits indicate the end block address for the access window. The value of flash access window end address indicates a block next to a writable/erasable block that has been set by the access window.	R
30:26	—	These bits are read as 0.	R
31	BTFLG	Startup Area Select Flag This bit indicates whether or not a startup area is changed by using the startup area select function. 0: Block 8 to block 15 are used as a startup area 1: Block 0 to Block 7 are used as a startup area (initial value).	R

The FAWMON register indicates a value of the write protection flag and a startup area select flag for setting the flash access window start address, the flash access window end address, and the access window.

When a reset or a configuration program command is executed, the FACI transfers data from the flash memory to the FAWMON register. The FAWMON register holds a copy of the value of the AWS register. In the case of writing, see [section 7, Option-Setting Memory](#).

See [section 50.6.1.4. Startup Area Select](#) for the startup area select function, and see [section 50.6.5.4. Protection by Access Window](#) for the access window.

50.3.13 FPCKAR : Flash Sequencer Processing Clock Notification Register

Base address: FACL = 0x407F_E000

Offset address: 0xE4

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

Bit	Symbol	Function	R/W
7:0	PCKA[7:0]	Flash Sequencer Operating Clock Notification These bits are used to set the operating frequency of the flash sequencer while processing FACL commands.	R/W ¹
15:8	KEY[7:0]	Key Code	R/W ²

Note 1. Writing to these bits is only possible when the FRDY bit in the FSTATR register is 1. Writing to these bits while the FRDY bit = 0 is ignored. Writing to these bits is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0x1E.

Note 2. Written values are not retained by these bits (always read 0x00).

The FPCKAR register is used to specify the flash sequencer operating frequency in the FACL command processing. The initial value is set to the maximum operating frequency for every product.

PCKA[7:0] bits (Flash Sequencer Operating Clock Notification)

The PCKA[7:0] bits specify the operating frequency of the flash sequencer while processing FACL commands. Set the desired frequency for these bits before issuing an FACL command. Specifically, convert the frequency in MHz to a binary number and set it for these bits.

Example: When a frequency is 25.9 MHz (PCKA[7:0] = 0x1A)

Round up the first decimal place of 25.9 MHz to a whole number (= 26).

Convert 26 into a binary number.

If the value set in these bits is smaller than the actual operating frequency of the flash sequencer, the flash memory overwriting characteristics cannot be guaranteed. If the value set in these bits is greater than the actual operating frequency of the flash sequencer, the flash memory overwriting characteristics can be guaranteed but the FACL command processing time such as the time overwriting takes increases. The minimum FACL command processing time is obtained when the operating frequency of the flash sequencer is the same as the PCKA value.

50.3.14 FSUACR : Flash Startup Area Control Register

Base address: FACL = 0x407F_E000

Offset address: 0xE8

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

Bit	Symbol	Function	R/W
1:0	SAS[1:0]	Startup Area Select These bits are used to select a startup area. 0 x: A startup area is selected in accordance with the setting of the BTFLG flag of the AWS register in section 7, Option-Setting Memory 1 0: The startup area of blocks 0 to 7 is used regardless of the setting of the BTFLG flag of the AWS register in section 7, Option-Setting Memory . In the case of a reset after the setting of the above value, a startup area conforms to the setting of the BTFLG flag. 1 1: The startup area of blocks 8 to 15 is used regardless of the setting of the BTFLG flag of the AWS register in section 7, Option-Setting Memory . In the case of a reset after the setting of the above value, a startup area conforms to the setting of the BTFLG flag.	R/W*2 *3
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code The KEY[7:0] bits enable or disable rewriting of the SAS[1:0] bits	R/W*1

Note 1. Written values are not retained by these bits (always read 0x00).

Note 2. Writing to these bits is possible only when 0x66 is written to the KEY[7:0] bits in a 16-bit access. Writing to these bits is possible only when the FSPR flag of the AWS register in [section 7, Option-Setting Memory](#) is 1. Writing to these bits while the FSPR flag = 0 is ignored.

Note 3. When changing a startup area, set the FLWT register to 0x000 (no cycles of waiting).

The FSUACR register is used by the startup area select function to select a block where execution begins.

50.4 Operating Modes Associated with the Flash Memory

Figure 50.4 shows a diagram of the mode transitions associated with the flash memory. For information on setting up the modes, see [section 3, Startup Modes](#).

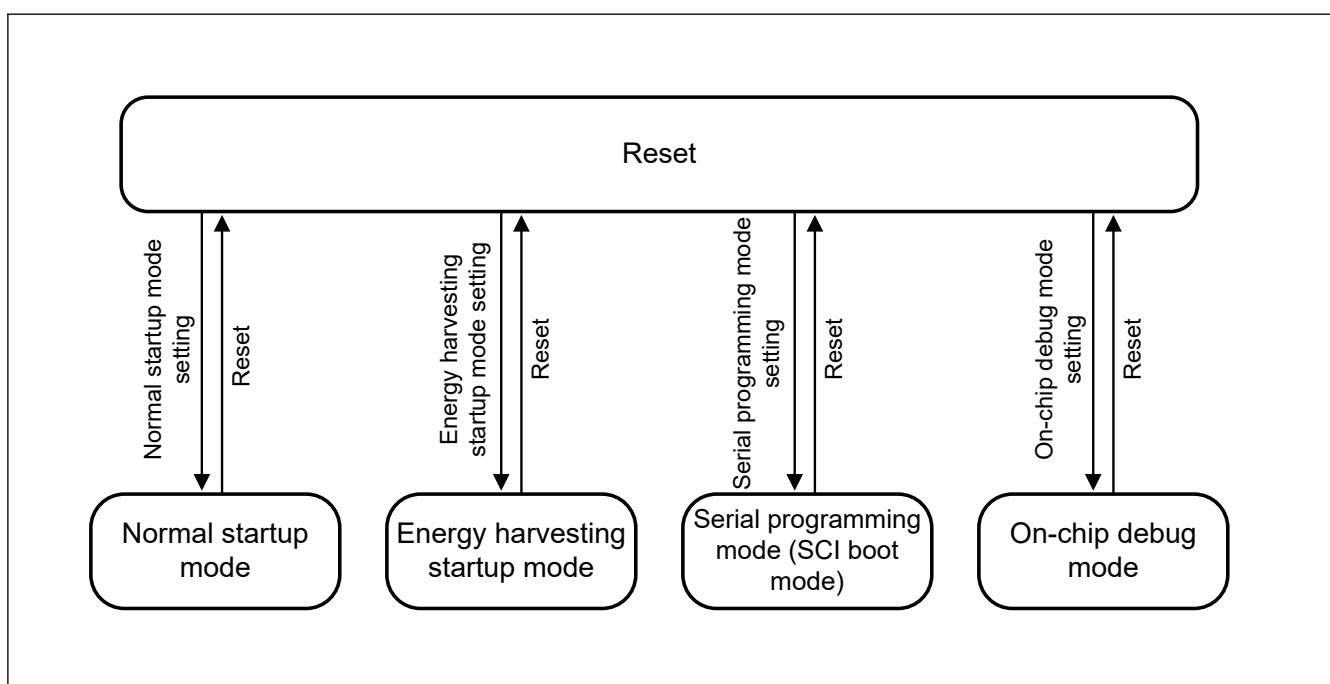


Figure 50.4 Mode Transitions Associated with Flash Memory

The flash memory areas where programming and erasure are permitted, and where the boot program executes on reset, depend on the mode. [Table 50.5](#) shows the differences between the modes.

Table 50.5 Difference between Modes

Item	Normal Startup Mode	Serial Programming Mode (SCI Boot Mode)	On-chip Debug Mode (SWD Boot Mode)
Programmable and erasable areas	Code flash memory	Code flash memory	Code flash memory
Block erase	Possible	Possible	Possible
Boot program on reset	User area program	Embedded program for serial programming	Depends on debug command

50.4.1 Normal Startup Mode

The flash memory can be programmed by executing a user program written to the code flash memory in advance by serial programming. Background operation can be used by the code flash memory to read from and write to the code flash memory itself, but only when the target address ranges of the code flash memory to be programmed and read satisfy particular conditions. When the conditions are met, a program resident in the range for reading of the code flash memory can be executed to program the range for writing of the code flash memory.

50.5 Serial Programming Mode

The serial programming mode includes:

- SCI boot mode with SCI9

[Table 50.6](#) lists the I/O pins for the flash memory-related modules.

Table 50.6 Basic Functions

Pin Name	I/O	Applicable Modes	Function
MD	Input	SCI boot mode (Serial programming mode)	Selection of operating mode
RXD9_A*1	Input		For host communication, to receive data through SCI
TXD9_A*1	Output		For host communication, to transmit data through SCI

Note 1. For the port allocated destination of the RXD9_A/TXD9_A pin, see [section 22, I/O Ports](#).

50.5.1 SCI Boot Mode

In SCI boot mode, the host sends control commands and data for programming, and the code flash memory area is programmed or erased accordingly. An on-chip SCI handles transfer between the host and this product in asynchronous mode. Tools for transmission of control commands and the programming data must be prepared in the host.

When this product is activated in SCI boot mode, the embedded program for serial programming is executed. This program automatically adjusts the bit rate of the SCI and controls programming and erasure by receiving control commands from the host.

[Figure 50.5](#) shows the system configuration for operations in SCI boot mode

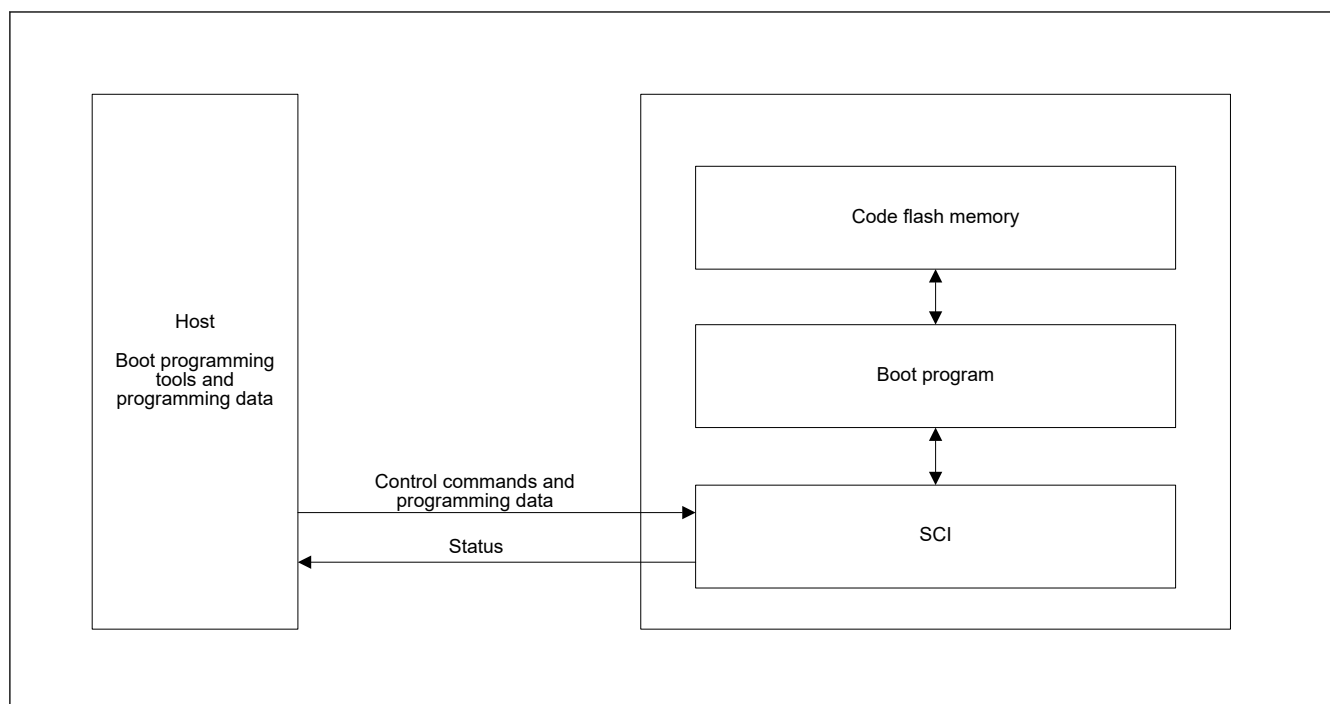


Figure 50.5 System Configuration in SCI Boot Mode

50.5.2 On-chip Debug Mode (OCD mode)

In on-chip debug mode, a dedicated programming adapter allows off-board programming of the flash memory before the device is mounted on the target system. A dedicated flash-memory programmer or an on-chip debugger through an SWD interface enables on-board programming of the flash memory after the device is mounted on the target system.

50.6 Functions

50.6.1 Basic Functions

50.6.1.1 Functions of the Flash Memory

Table 50.7 lists the functions of the flash memory. The functions in serial programming are realized by serial programmer commands. On the other hand, the functions in self-programming are realized by reading of the code flash memory by a FSCI command or the user program.

Table 50.7 Basic Functions (1 of 2)

Function	Functional Overview	Support Status	
		Serial Programming	Self-programming
Block erase	Erases the memory contents in the specified block	Supported	Supported
Programming	Writes to the specified address	Supported	Supported
Read	Reads data programmed in the flash memory	Supported	Not supported (Read by user program is possible)
ID code check	Compares the ID code sent by the host with the code stored in the code flash memory, and if the two match, the FCU enters the wait state for programming and erasure commands from the host.	Supported	Not supported (ID authentication is not performed)

Table 50.7 Basic Functions (2 of 2)

Function	Functional Overview	Support Status	
		Serial Programming	Self-programming
Security configuration	Configures the security function for serial programming	Supported with conditions (Only switching from enabled to disabled is possible)	Supported with conditions (Only switching from enabled to disabled is possible)
Protection configuration	Configures the access window for area protection in the code flash memory	Supported	Supported

50.6.1.2 ID Code Security Function

The flash memory supports the ID code security function. Authentication of ID codes is a security function for use with serial programming and SWD programming. [Table 50.8](#) lists the security functions supported by the flash memory, and [Table 50.9](#) lists the available operations and security settings.

Table 50.8 Security Functions

Function	Functional Overview
ID authentication	The result of ID authentication can be used to control the connection of a serial programmer for serial programming

Table 50.9 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 - Security configuration commands not supported - Protection configuration commands not supported. When ID codes match: - Block erasure commands supported - Programming commands supported - Security configuration commands supported - Protection configuration commands supported.	(ID authentication is not performed) Block erasure supported Programming supported Security configuration supported Protection configuration supported.	ID authentication is not performed

50.6.1.3 Suspend Operation

When a P/E suspend command is issued to suspend the programming or erasure of the code flash memory, reading from the memory is enabled. The P/E resume command is available for resuming suspended programming or erasure.

50.6.1.4 Startup Area Select

The startup area select function allows the boot program to be safely updated. The size of the startup area is 32 Kbytes and the startup area is located in the user area. FSCI controls the address of the startup area based on the startup area select flag (AWS.BTFLG). The startup area can be locked by the AWS.FSPR flag*1.

Note 1. For details on the AWS.FSPR flag, see [section 7.2.4. AWS : Access Window Setting Register](#).

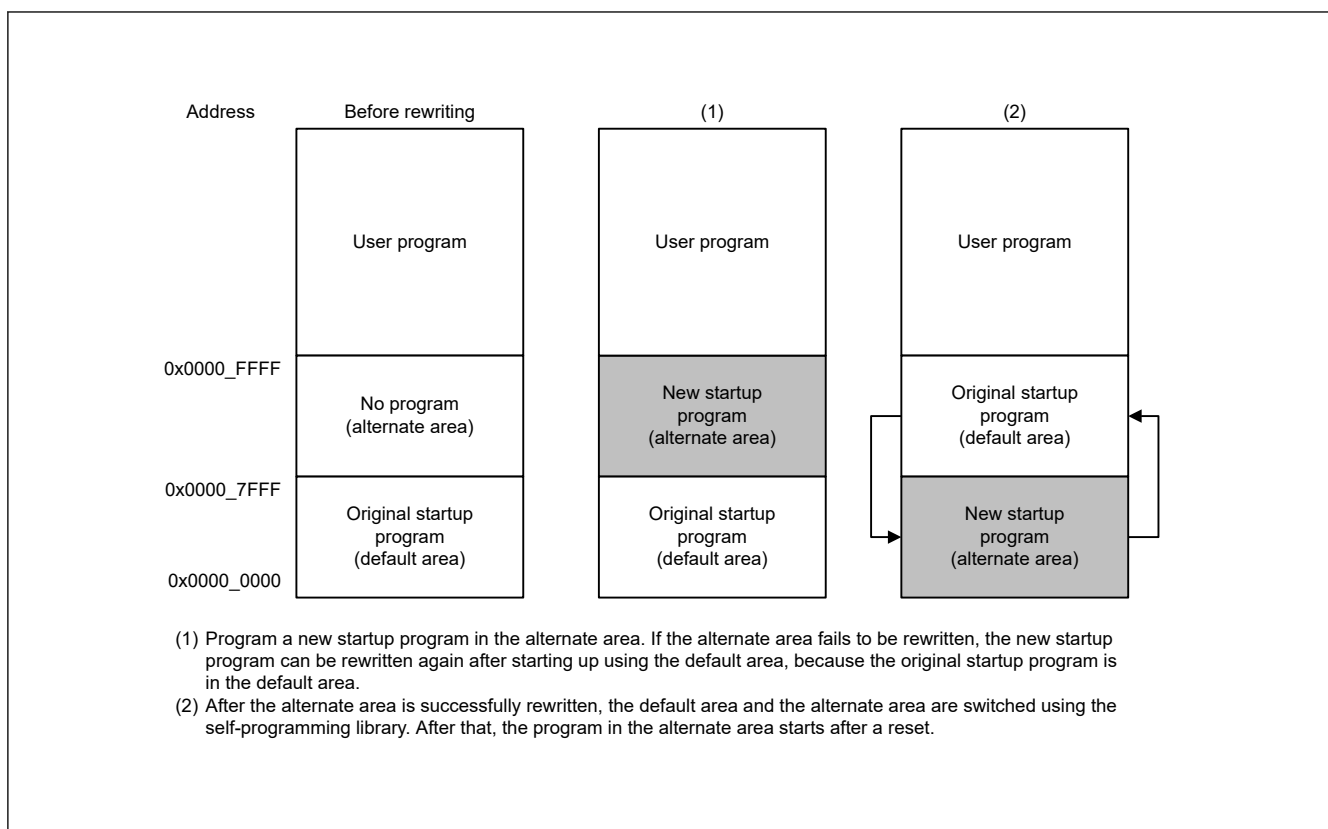


Figure 50.6 Overview of Startup Program Protection

50.6.2 Programming Methods

By using a dedicated flash-memory programmer to program the on-chip flash memory through a serial communication interface (serial programming mode) or through an 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 written to the on-chip flash memory are incorporated to 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 the programming can proceed while processing external communications and other functions. Table 50.10 lists the programming methods and the corresponding operating modes.

Table 50.10 Programming Methods (1 of 2)

Programming Method	Functional Overview	Operating Mode
Serial programming	A dedicated flash-memory programmer through the SCI 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 through the SCI interface and a dedicated programming adapter board allow off-board programming of the flash memory before the device is mounted on the target system.	
Self-programming ^{*1}	The flash memory can also be programmed by executing a user program written to memory in advance by serial programming. Instructions in the code flash memory cannot be fetched and data cannot be accessed while the code flash memory is being programmed by self-programming. In such cases, a program for programming from the SRAM or external memory must be transferred in advance and executed.	Normal startup mode

Table 50.10 Programming Methods (2 of 2)

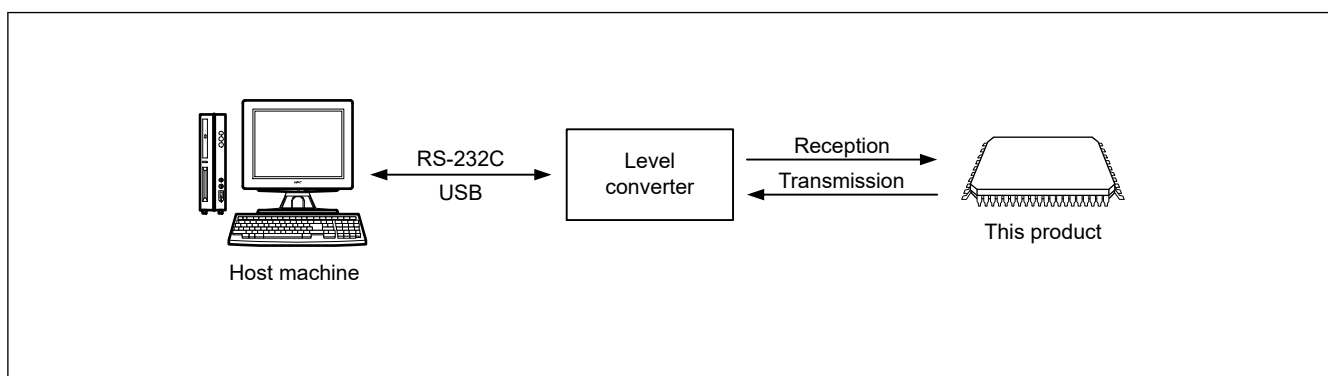
Programming Method	Functional Overview	Operating Mode
SWD programming	A dedicated flash-memory programmer or an on-chip debugger through SWD enables on-board programming of the flash memory after the device is mounted on the target system.	On-chip debug mode
	A dedicated flash-memory programmer or an on-chip debugger through SWD and a dedicated programming adapter allow off-board programming of the flash memory, for example, programming of the device before it is mounted on the target system.	

Note 1. Contact Renesas for register information individually.

50.6.2.1 Serial Programming

A dedicated flash memory programmer can be used to program the flash memory in serial programming mode. This device is mounted on the serial board, and a connector to the board enables programming by the flash memory programmer.

Figure 50.7 shows the recommended environment for programming the flash memory of the device with data

**Figure 50.7 Environments for Writing Programs to the Flash Memory**

50.6.2.2 Self-Programming

This device supports programming of the code flash memory by the user program itself. The programming commands can be used with user programs for writing to the code flash memory. This enables upgrading of user programs and overwriting of constant data fields.

The programming program can be copied to the SRAM or external memory before the programming operation, and executed from the copy destination to program the code flash memory.

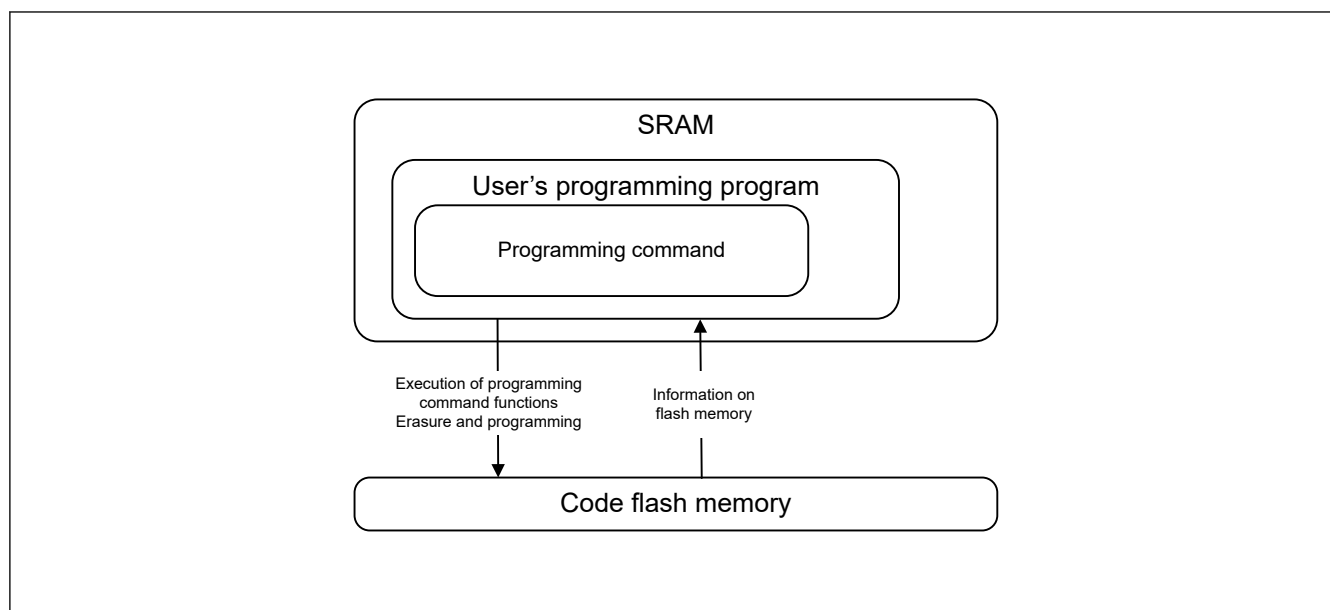


Figure 50.8 Schematic View of Self-programming

50.6.2.3 SWD Programming

A dedicated flash-memory programmer or an on-chip debugger through an SWD interface enables programming of the flash memory in the on-chip debug mode.

50.6.3 Programming Commands

[Figure 50.9](#) shows the configuration of the flash memory-related modules. The flash sequencer is mainly composed of the flash control unit (FCU) and the flash application command interface (FACI). The FCU executed the basic control of flash memory rewriting. The FACI controls the FCU in accordance with the FACI commands that have been received via peripheral buses.

When a reset operation is executed, the FACI transfers data from the flash memory to the option byte storage register (FACI reset transfer

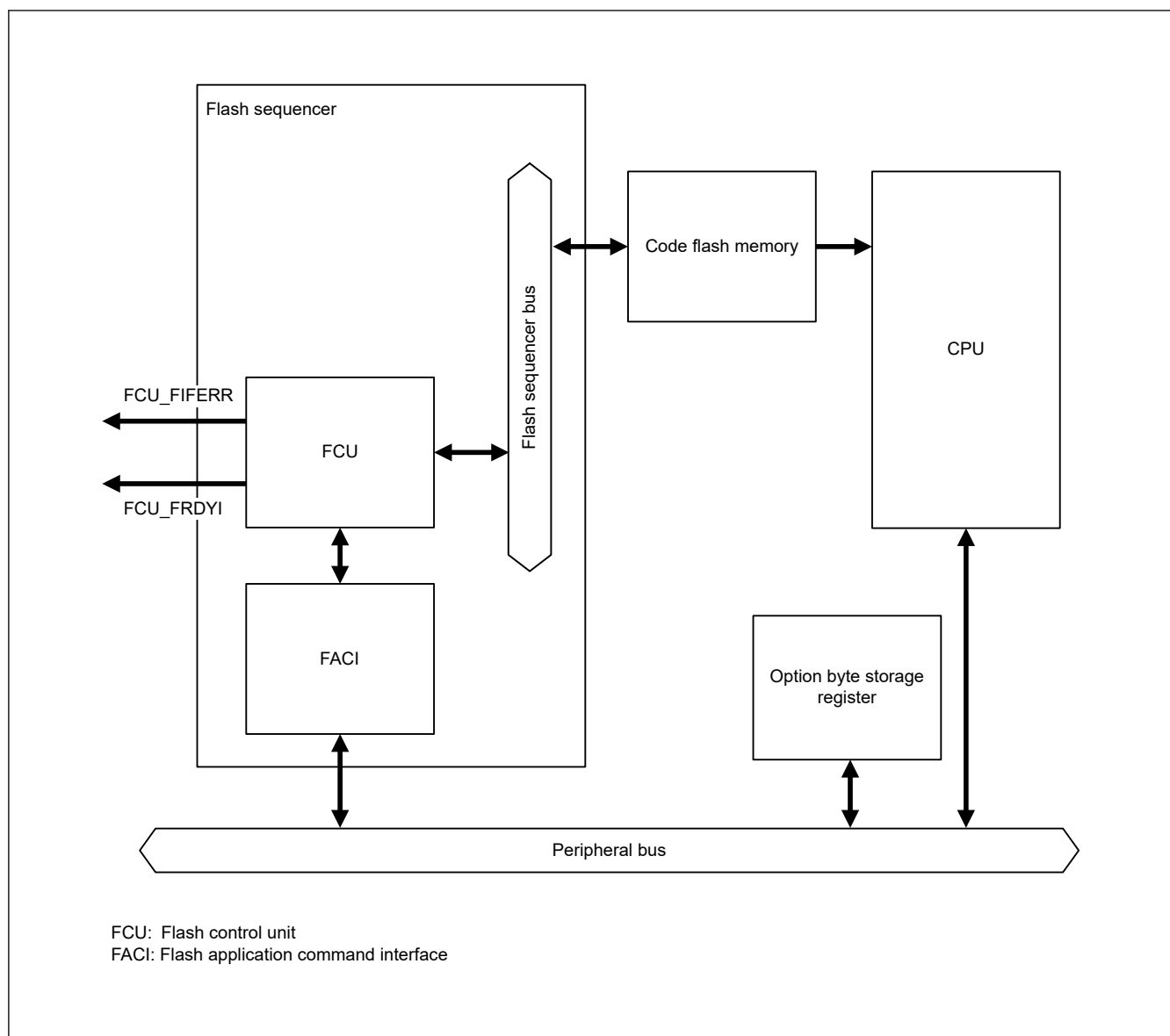


Figure 50.9 Block Diagram of Flash Memory-related Modules

50.6.3.1 List of FACI Commands

Table 50.11 List of FACI Commands

FACI Command	Description
Programming	This is used to write to the user area. Units of writing are 8 or 256 bytes for the user area.
Block erase	This is used to erase the user area. The unit of erasure is one block (4 Kbytes).
P/E suspend	This suspends writing or erasure processing.
P/E resume	This resumes suspended writing or erasure processing.
Status clear	This initializes the ILGLERR/ERSERR/PRGERR/ILGCOMERR/FESETERR/SECERR/OTERR flags in the FSTATR register and CMDLK/CFAE flags in the FASTAT register, and releases the flash sequencer from the command-locked state.
Forced stop	This forcibly stops processing of FACI commands and initializes the FSTATR/FASTAT register.
Configuration setting	This is used to set the security function, safety function, and other configuration function. Units of setting: 16 bytes.

FACI commands are issued by writing to the FACI command-issuing area. When each write access shown in [Table 50.12](#) is made in a specified state, the flash sequencer executes the processing corresponding to the given command (see [section 50.6.3.2. Relationship between the Flash Sequencer State and FACI Commands](#)).

Table 50.12 List of FACI Commands

FACI Command	Write Count	Data to be Written to the FACI Command-issuing Area			
		1st Access	2nd Access	3rd to (N + 2)th Access	(N + 3)th Access
Programming (user area) 8-byte writing: N = 4 256-byte writing: N = 128	N + 3	0xE8	0x04 (= N) 0x80 (= N)	WD ₁ to WD _N	0xD0
Block erase (user area)	2	0x20	0xD0	—	—
P/E suspend	1	0xB0	—	—	—
P/E resume	1	0xD0	—	—	—
Status clear	1	0x50	—	—	—
Forced stop	1	0xB3	—	—	—
Configuration setting N = 8	11	0x40	0x08 (= N)	WD ₁ to WD _N	0xD0

Note: WD_N (N = 1, 2,...): Nth 16-bit data to be written

50.6.3.2 Relationship between the Flash Sequencer State and FACI Commands

Each FACI command can be accepted in a specific mode or state of the flash sequencer. FACI commands should be issued after the transition of the flash sequencer to the code flash P/E mode and checking of the state of the flash sequencer. Use the FSTATR and FASTAT registers to check the state of the flash sequencer. When ILGLERR, ILGCOMERR, FESETERR, SECERR, OTERR, ERSERR, PRGERR, or FLWEERR flag in the FSTATR register is 1, the CMDLK flag of the FASTAT register is 1. The occurrence of errors in general can be checked by reading the CMDLK flag of the FASTAT register.

[Table 50.13](#) lists the available commands in each operating mode.

Table 50.13 List of FACI Commands

Mode	FENTRYC	Available Commands
Read Mode	0	None
Code flash P/E mode	1	Programming Block erase P/E suspend P/E resume Status clear Forced stop Configuration setting

[Table 50.14](#) shows the state of the flash sequencer and acceptable FACI commands. An appropriate mode is assumed to be set before the commands are executed.

Table 50.14 Acceptable FACI Commands and the State of the Flash Sequencer (1 of 2)

	Processing of Writing or Erasure	Processing of Configuration Setting	Processing to Suspend Writing or Erasure	Writing Suspended	Erasure Suspended	Writing while Erasure is Suspended	Command-Locked State (FRDY = 1)	Command-Locked State (FRDY = 0)	Processing of Forced Stop Command	Other State
FRDY flag	0	0	0	1	1	0	1	0	0	1
SUSRDY flag	1	0	0	0	0	0	0	0	0	0

Table 50.14 Acceptable FACI Commands and the State of the Flash Sequencer (2 of 2)

	Processing of Writing or Erasure	Processing of Configuration Setting	Processing to Suspend Writing or Erasure	Writing Suspended	Erasure Suspended	Writing while Erasure is Suspended	Command-Locked State (FRDY = 1)	Command-Locked State (FRDY = 0)	Processing of Forced Stop Command	Other State
ERSSPD flag	0	0	0/1	0	1	1	0/1	0/1	0	0
PRGSPD flag	0	0	0/1	1	0	0	0/1	0/1	0	0
CMDLK flag	0	0	0	0	0	0	1	1	0	0
Programming	x	x ^{*3}	x	x	✓ ^{*2}	x	x	x	x	✓
Block erase	x	x ^{*3}	x	x	x	x	x	x	x	✓
P/E suspend	✓	x ^{*3}	x	x	x	x	—	x	x	—
P/E resume	x	x ^{*3}	x	✓	✓	x	x	x	x	x
Status clear	x	x ^{*3}	x	✓	✓	x	✓	x	x	✓
Forced stop	✓	✓ ^{*3}	✓	✓	✓	✓	✓	✓	✓	✓
Configuration setting	x	x ^{*3}	x	x	x	x	x	x	x	✓ ^{*1}

Note: : ✓ Acceptable, x: Not acceptable (the sequencer in the command-locked state), —: Ignored

Note 1. Acceptable only in code flash P/E mode

Note 2. Acceptable unless an area that is a target of the programming command is a block that is a target of erasure suspend. Command-locked state when a block that is a target of erasure suspend is included.

Note 3. When FSTATR.DBFULL flag is 1 during the processing of the configuration setting command, do not issue the command to the FACI

50.6.4 Usage of FACI Commands

This section gives an overview of the usage of FACI commands.

50.6.4.1 Overview of Command Usage in Code Flash memory P/E Mode

Figure 50.10 shows an overview of FACI command usage in code flash P/E mode. For which commands are available in code flash P/E mode, see Table 50.13. When the setting value of the FPCKAR register is lower than the operating frequency of the flash sequencer, the FPCKAR register must be modified. When the execution time of the FACI command is longer than an estimated value due to the setting value of the FPCKAR register, the FPCKAR register can be modified. The FPCKAR register cannot be modified during an execution of the FACI command. Consequently, modify the FPCKAR register before the FACI command is issued.

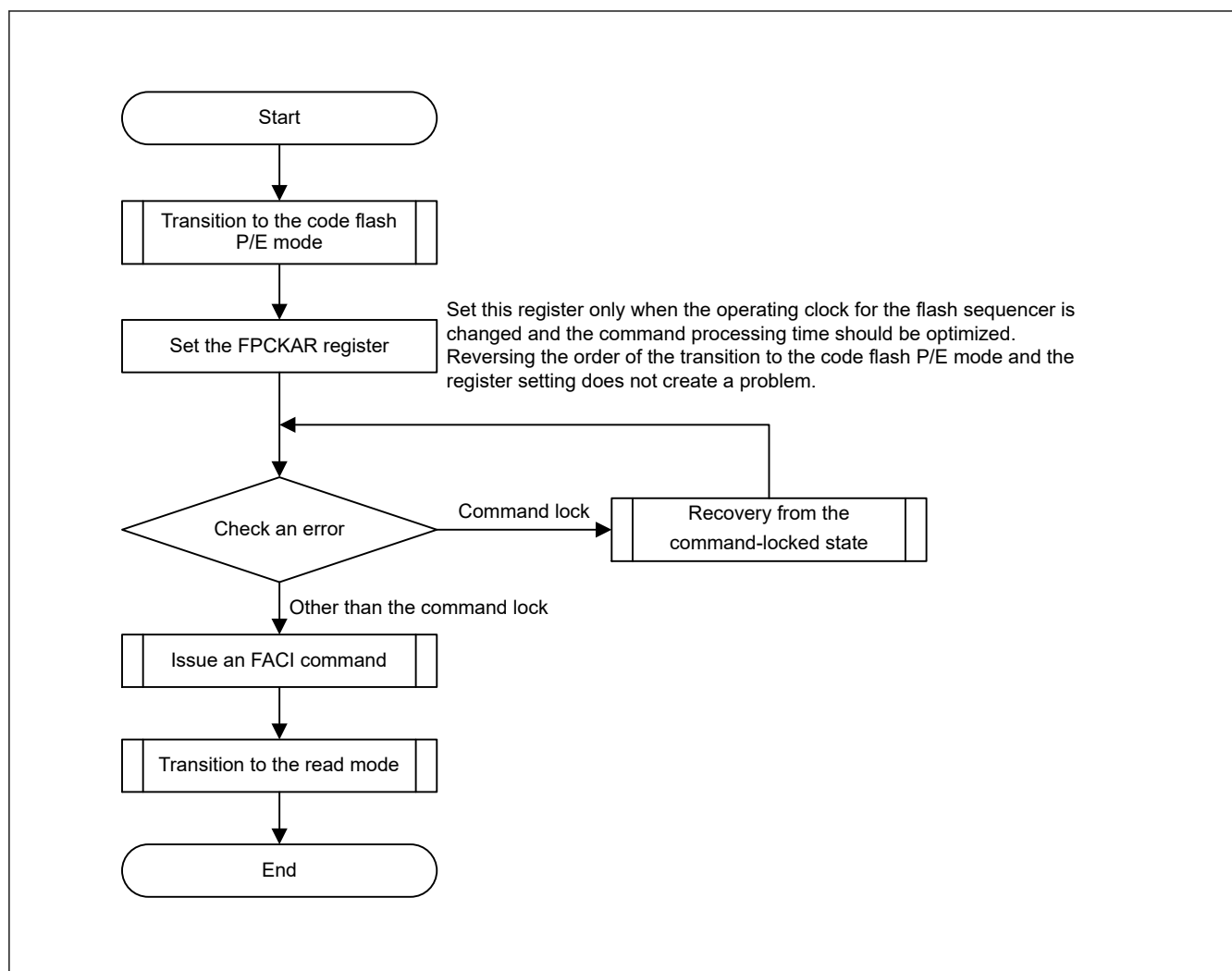


Figure 50.10 Overview of Command Usage in Code Flash P/E Mode

50.6.4.2 Transition to Code Flash P/E Mode

To use the FACL commands for the code flash memory, a transition to code flash P/E mode is required. To cause shift to code flash P/E mode, set the FENTRYR.FENTRYC bit to 1.

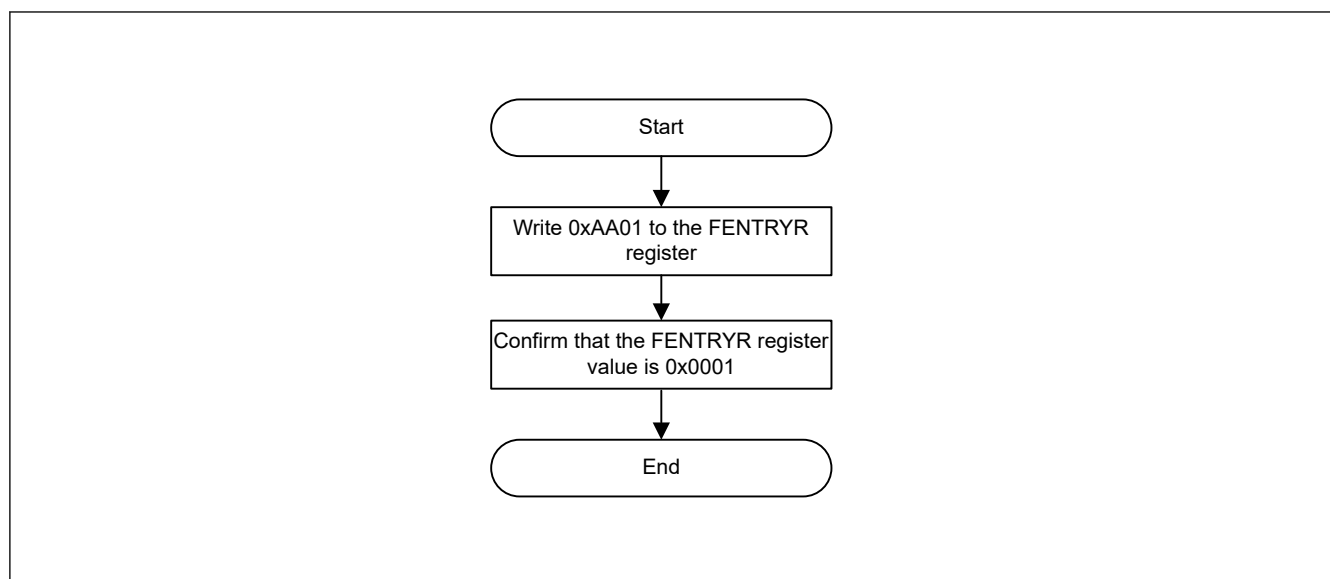


Figure 50.11 Procedure for Transition to Code Flash Memory P/E Mode

50.6.4.3 Transition to Read Mode

To read the flash memory, a transition to read mode is required. To shift to read mode, set the FENTRYR register to 0x0000. The transition to read mode should be made after processing by the flash sequencer is completed and while operation is in other than in the command-locked state.

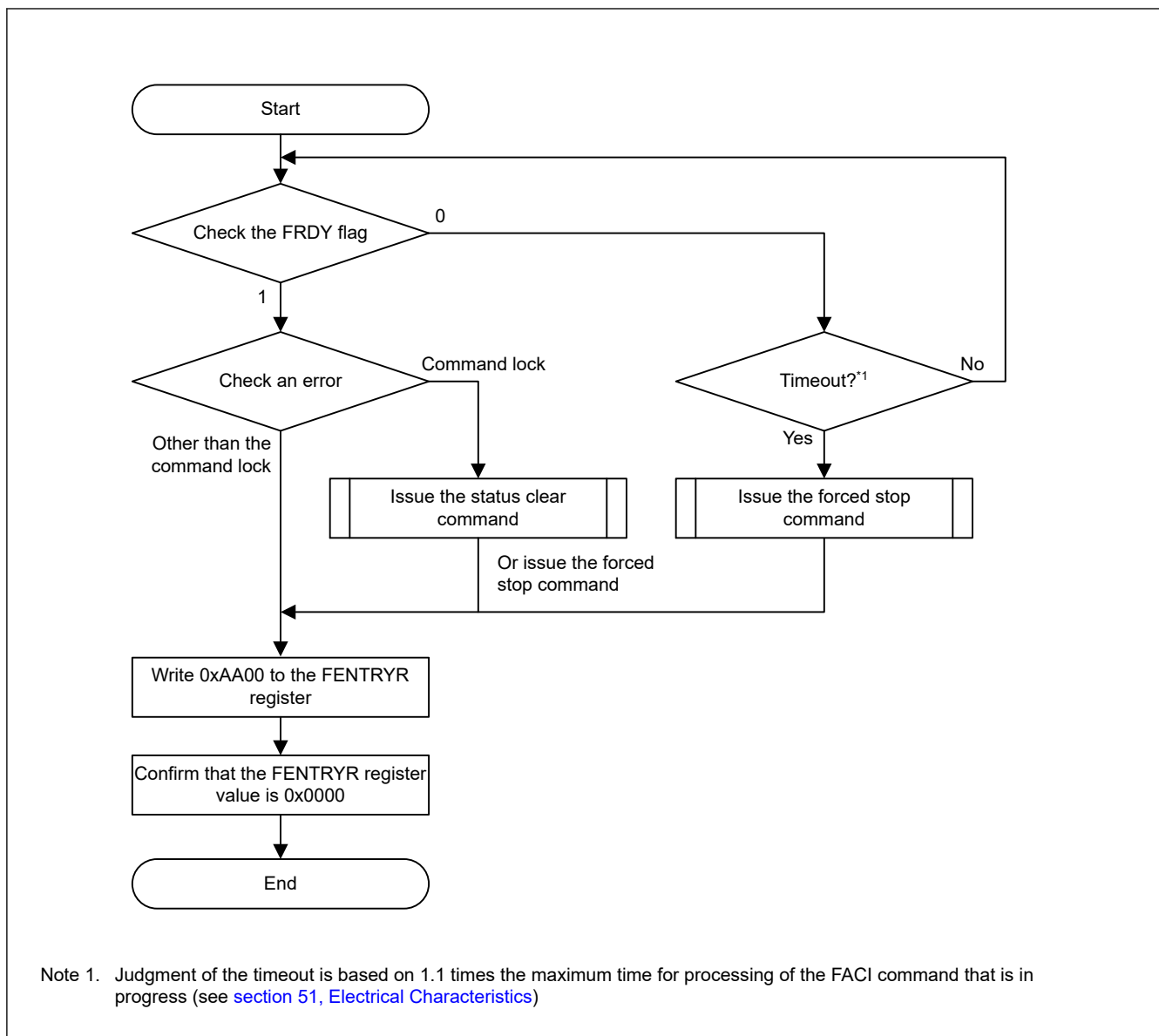


Figure 50.12 Procedure for Transition to Read Mode

50.6.4.4 Recovery from the Command-Locked State

When the flash sequencer enters the command-locked state, FACL commands cannot be accepted. To release the sequencer from the command-locked state, use the status clear command or forced stop command.

When the command-locked state is detected by checking for an error before issuing the P/E suspend command, the FSTAT.FRDY flag may hold 0 as the command processing has not been completed. If processing is not completed within the maximum writing/erasure time specified in the electrical characteristics, this can be considered a timeout, and the flash sequencer should be stopped by the forced stop command.

The FLWEERR flag in the FSTAT register is not changed from 1 to 0 by the status clear command. When this flag is set to 1, use the forced stop command for release from the command-locked state. The other flags that indicate the command-locked state can be changed from 1 to 0 by the status clear or forced stop command.

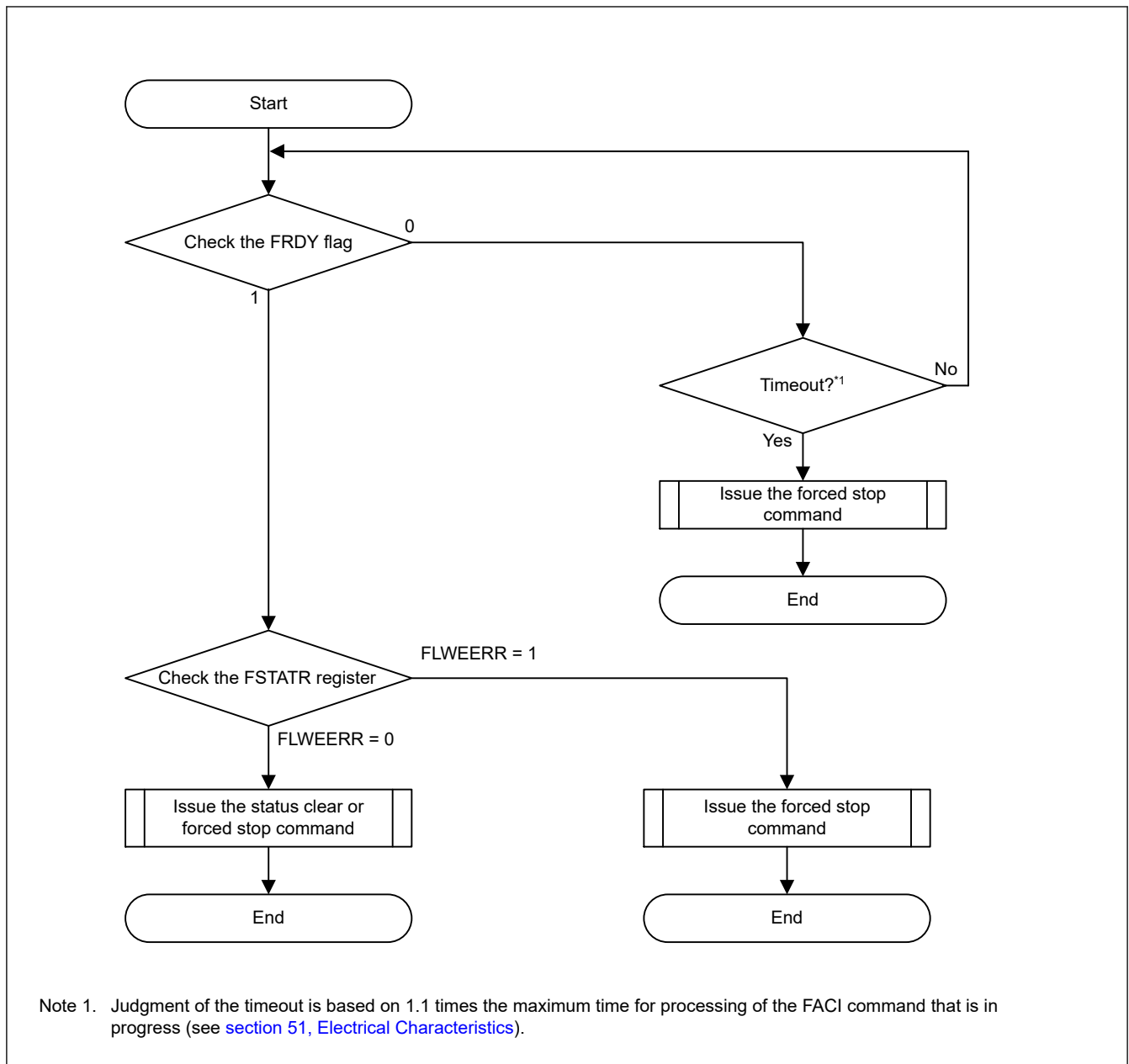


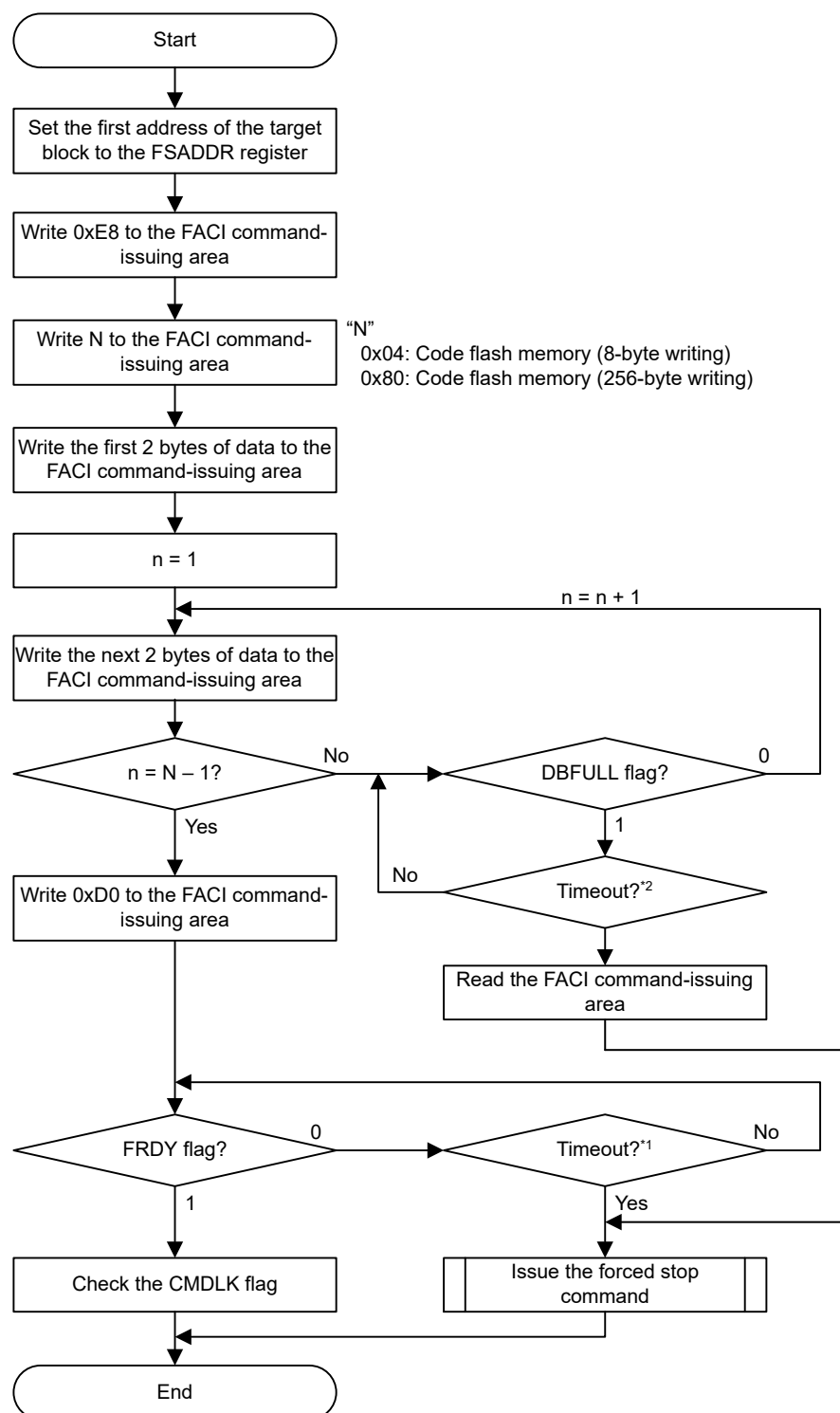
Figure 50.13 Recovery from the Command-locked State

50.6.4.5 Programming Commands

A programming command is used for writing to the user area.

Before issuing a programming command, set the first address of the target block in the FSADDR register. Writing 0xD0 to the FACL command-issuing area at the final access of the FACL command-issuing starts the programming command processing. The end of the command processing can be checked by reading the FRDY flag of the FSTATR register. If the target area of programming command processing contains the area not for writing, write 0xFFFF to the corresponding area.

Issuing a programming command consecutively while the FACL internal data buffer is full leads to a wait on the peripheral bus and this may affect the communication performance of the other peripheral modules. To avoid the generation of such a wait, issue an FACL command while the FSTATR.DBFULL flag is 0.



Note 1. Judgment of the timeout is based on 1.1 times the maximum value of writing time T_{prog} (see [section 51, Electrical Characteristics](#))

Note 2. Judged for 0.22 μs . (when ICLK = 32 MHz)

Figure 50.14 Usage of the Programming Command

50.6.4.6 Block Erase Command

A block erase command is used for erasing a user area. A unit of erasure is one block of a target area. Before issuing a block erase command, set the first address of the target block in the FSADDR register. Writing 0x20 and 0xD0 to the FACL command-issuing area starts processing of a block erase command. The end of the command processing can be checked by reading the FRDY flag of the FSTATR register.

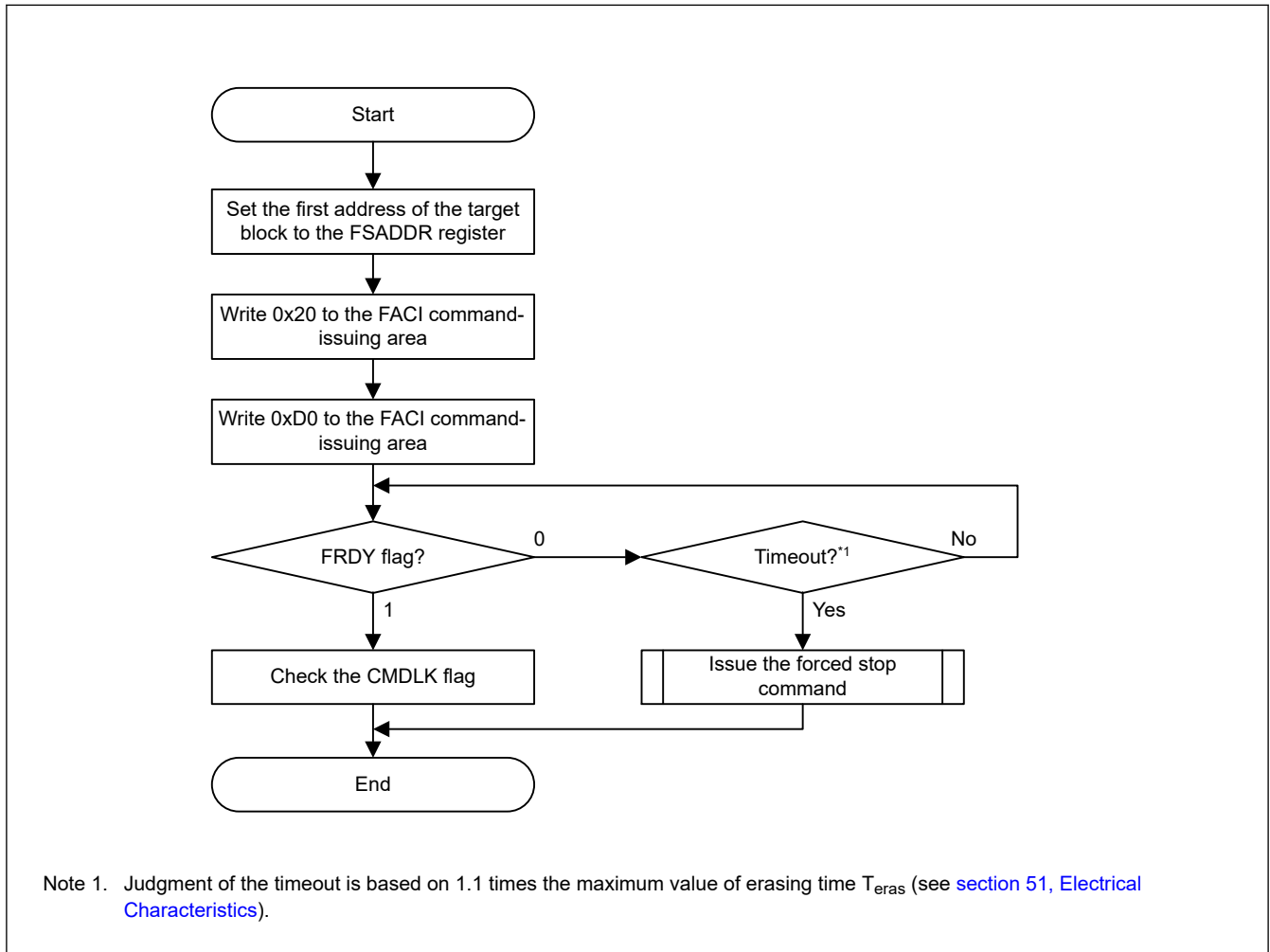


Figure 50.15 Usage of the Block Erase Command

50.6.4.7 P/E Suspend Command

The P/E suspend command is used to suspend writing or erasure.

Before issuing a P/E suspend command, check that the FASTAT.CMDLK flag is 0, and the execution of writing/erasure is normally performed. To confirm that the P/E suspend command can be received, also check that the FSTATR.SUSRDY flag is 1.

After issuing a P/E suspend command, read the FASTAT.CMDLK flag to confirm that an error has not occurred.

If an abnormality occurs during writing/erasure, the CMDLK flag is set to 1. If writing or erasure processing has been completed between when the state of the SUSRDY flag is 1 is confirmed and when the P/E suspend command is accepted, any of errors does not occur and the device does not enter the suspended state (the FSTATR.FR DY flag is 1, and the FSTATR.ERSSPD and PRGSPD flags are 0).

When a P/E suspend command is received and then the writing/erasure suspend processing finishes normally, the flash sequencer enters the suspended state, the FRDY flag is set to 1, and the ERSSPD or PRGSPD flag is set to 1. After issuing a P/E suspend command, check that the ERSSPD or PRGSPD flag is 1 and that the suspended state is entered, and then decide the subsequent flow. If a P/E resume command is issued in the subsequent flow although the suspended state is not

entered, an illegal command error occurs and the flash sequencer shifts to the command-locked state (see [section 50.7.2. Error Protection](#)).

If the erasure suspended state is entered, writing to blocks targeted for other than erasure can be performed. Additionally, the writing and erasure suspended states can shift to read mode by clearing the FENTRYR register.

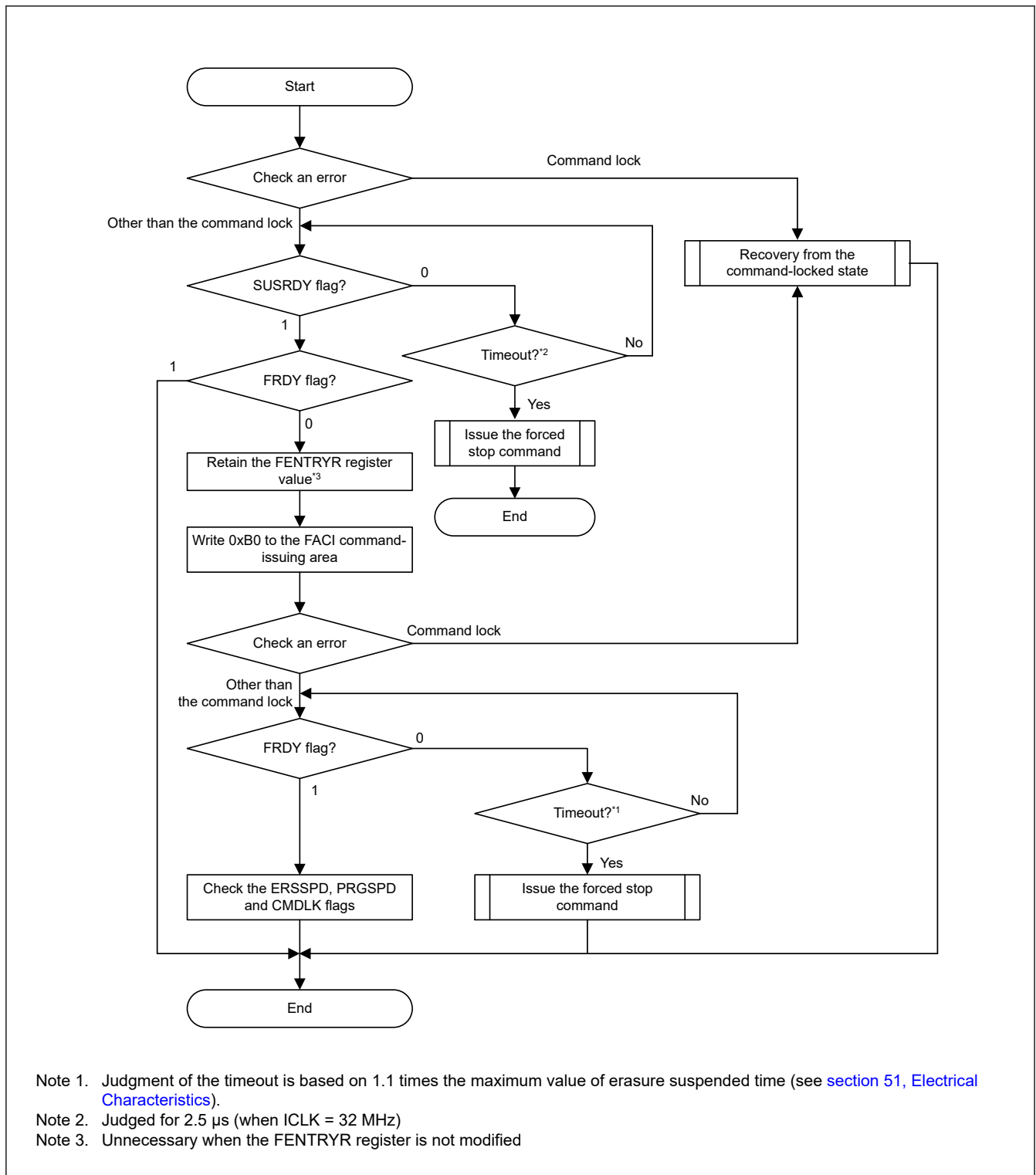


Figure 50.16 Usage of the P/E Suspend Command

(1) Suspension during writing

When issuing a P/E suspend command during the flash memory writing, the flash sequencer suspends writing processing.

Figure 50.17 shows the suspend operation of writing. When receiving a writing-related command, the flash sequencer clears the FSTAT.FRDY flag to 0 to start writing. When the flash sequencer enters the state in which the P/E suspend command can be received after starting writing, it sets the FSTAT.SUSRDY flag to 1. When a P/E suspend command is issued, the flash sequencer receives the command and clears the SUSRDY flag to 0. When the flash sequencer receives a P/E suspend command while a writing pulse is being applied, the flash sequencer continues applying the pulse. After the specified pulse application time, the flash sequencer finishes pulse application, and starts the writing suspend processing and sets the FSTAT.PRGSPPD flag to 1.

When the suspend processing finishes, the flash sequencer sets the FRDY flag to 1 to enter the writing suspended state. When receiving a P/E resume command in the writing suspended state, the flash sequencer clears the FRDY and PRGSPPD flags to 0 and resumes writing.

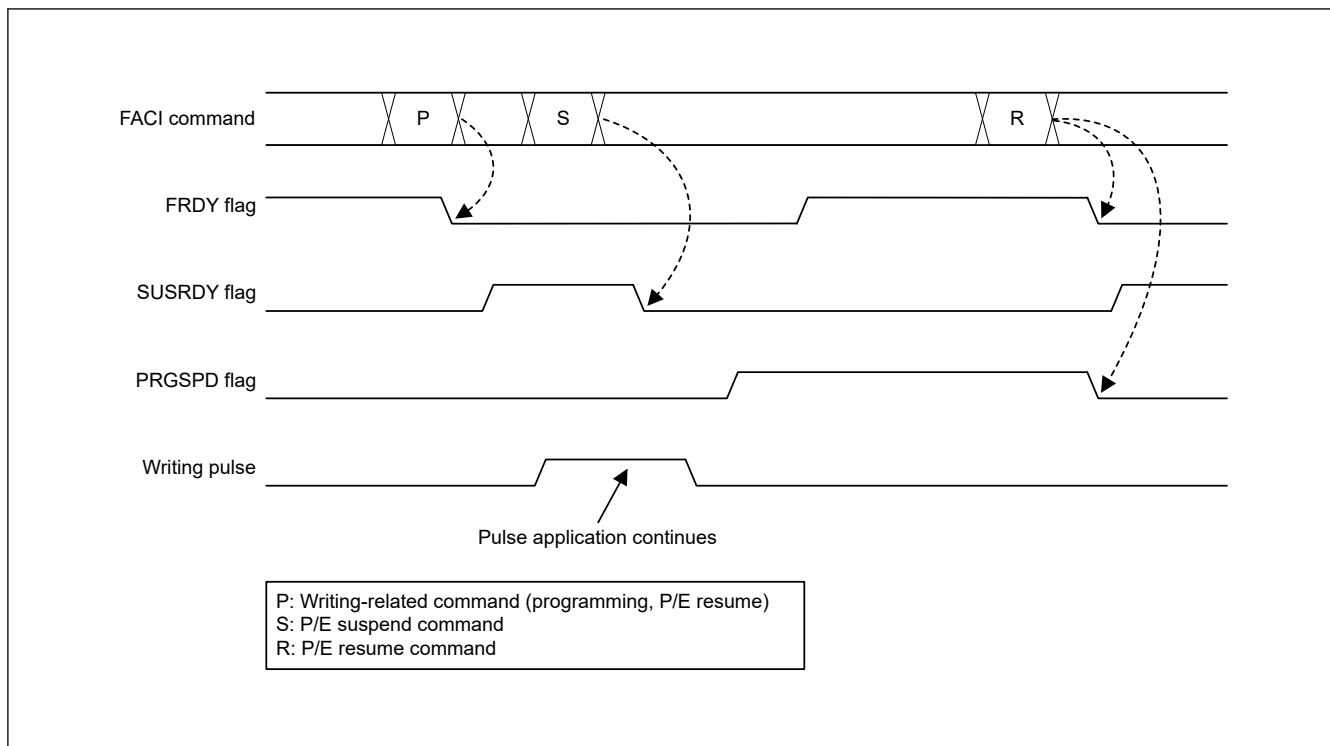


Figure 50.17 Suspension during Writing

(2) Suspension during erasure

Figure 50.18 shows the suspend operation of erasure. When receiving an erasure-related command, the flash sequencer clears the FSTAT.FRDY flag to 0 to start erasure. When the FCU enters the state in which the P/E suspend command can be received after starting erasure, it sets the FSTAT.SUSRDY flag to 1. When a P/E suspend command is issued, the flash sequencer receives the command and clears the SUSRDY flag to 0. When receiving a suspend command during erasure, the flash sequencer starts the suspend processing and sets the FSTAT.ERSSPD flag to 1 even if it is applying an erasure pulse. When the suspend processing is completed, the flash sequencer sets the FRDY flag to 1 to enter the erasure suspended state. When receiving a P/E resume command in the erasure suspended state, the flash sequencer clears the FRDY and ERSSPD flags to 0 and resumes erasure.

In the erasure suspended state, when receiving the first P/E suspend command while erasure pulse is being applied, the flash sequencer suspends the application of erasure pulse and enters the erasure suspended state. When receiving the second P/E suspend command while reapplying erasure pulse after erasure is resumed by a P/E resume command, the flash sequencer continues applying erasure pulse until the specified time elapses. After the specified pulse application time, the flash sequencer suspends erasure pulse application and enters the erasure suspended state. When the flash sequencer then receives the next P/E resume command and receives a P/E suspend command again, an operation equivalent to that of the second P/E suspend command is executed.

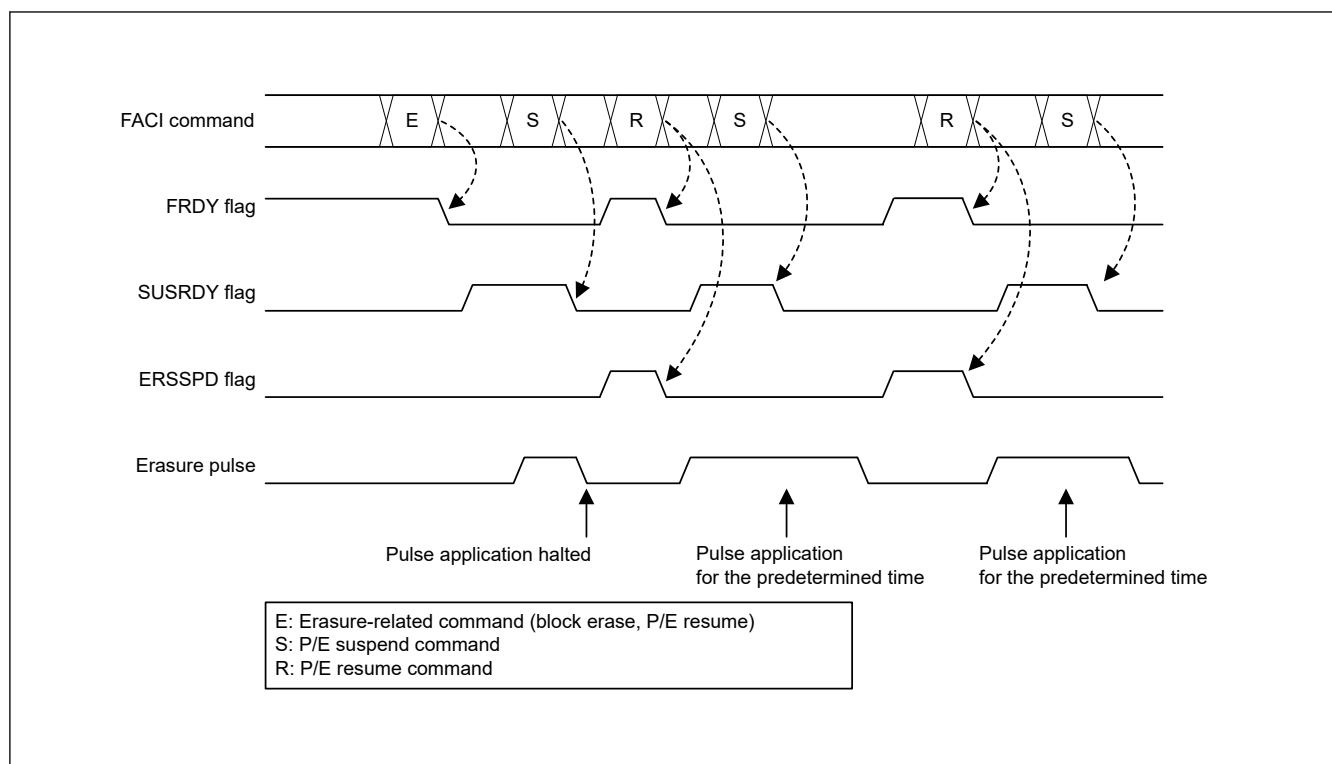
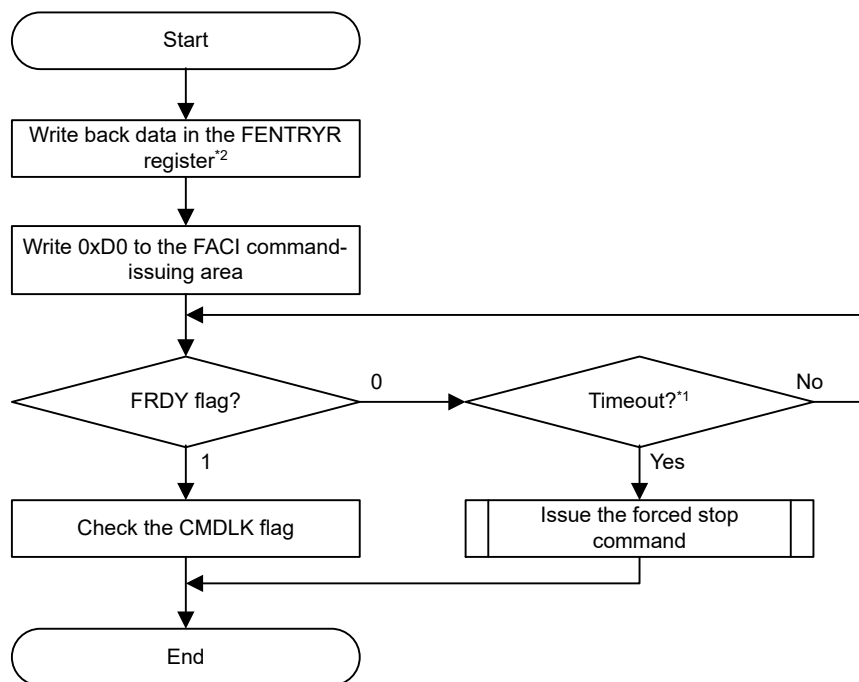


Figure 50.18 Suspension during Erasure

50.6.4.8 P/E Resume Command

To resume suspended writing or erasure, use the P/E resume command. When the settings of the FENTRYR register are changed during suspension, reset the setting of the FENTRYR register to the value immediately before the P/E suspend command was issued, and then issue a P/E resume command. The end of the command processing that has been resumed can be checked by reading the FRDY flag of the FSTATR register



Note 1. Judgment of the timeout is based on 1.1 times the maximum time for suspended processing of the FACL command (see [section 51, Electrical Characteristics](#)).

Note 2. Unnecessary when the FENTRYR register is not modified.

Figure 50.19 Usage of the P/E Resume Command

50.6.4.9 Status Clear Command

If the ILGLERR, ILGCOMERR, FESETERR, SECERR, OTERR, ERSERR, PRGERR, or FLWEERR flag of the FSTATR register is set to 1, the flash sequencer is placed in the command-locked state. In the command-locked state, the flash sequencer can receive only the status clear or forced stop command.

The status clear command is used to clear the command-locked state (see [section 50.6.4.4. Recovery from the Command-Locked State](#)). To clear the ILGLERR, ILGCOMERR, FESETERR, SECERR, OTERR, ERSERR, and PRGERR flags in the FSTATR register in the command-locked state, the status clear command is available. The FLWEERR flag cannot be cleared by the status clear command, and can be cleared only by the forced stop command.

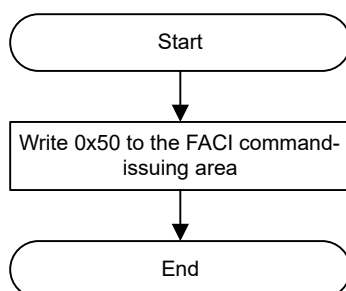


Figure 50.20 Usage of the Status Clear Command

50.6.4.10 Forced Stop Command

The forced stop command forcibly ends command processing by the flash sequencer. Although this command can halt command processing in higher speed than the P/E suspension command, values from the area where writing or erasure was in progress are not guaranteed. Furthermore, resumption of processing is not possible. Processing of writing or erasure that was halted by the forced stop command is also defined as one round of writing.

Executing a forced stop command also initializes the whole FCU and a part of the FACL. In addition, the FSTATR and FASTAT registers are also initialized. Accordingly, this command can be used in the procedure for recovery from the command-locked state and in processing in response to a time-out of the flash sequencer (see [section 50.6.4.4. Recovery from the Command-Locked State](#)).

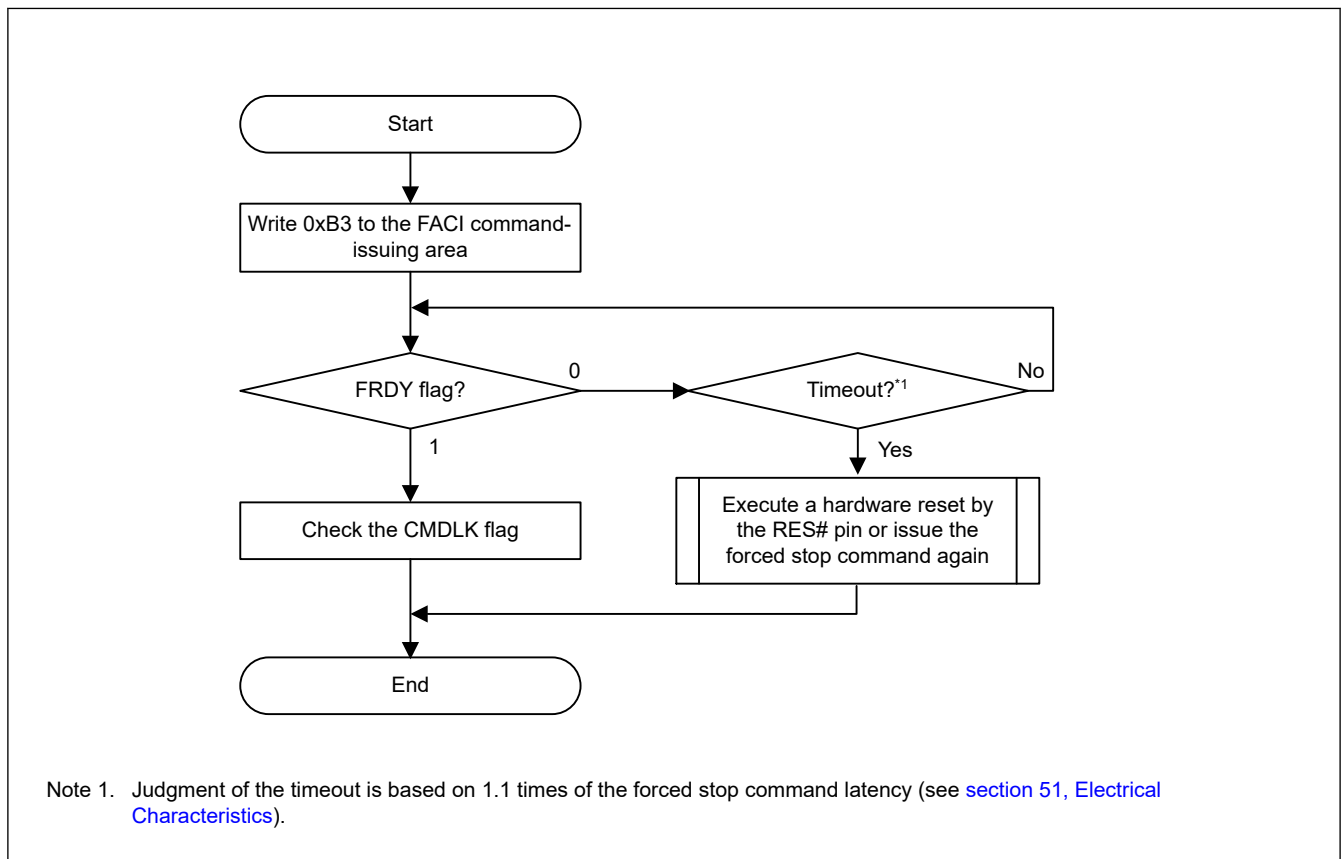


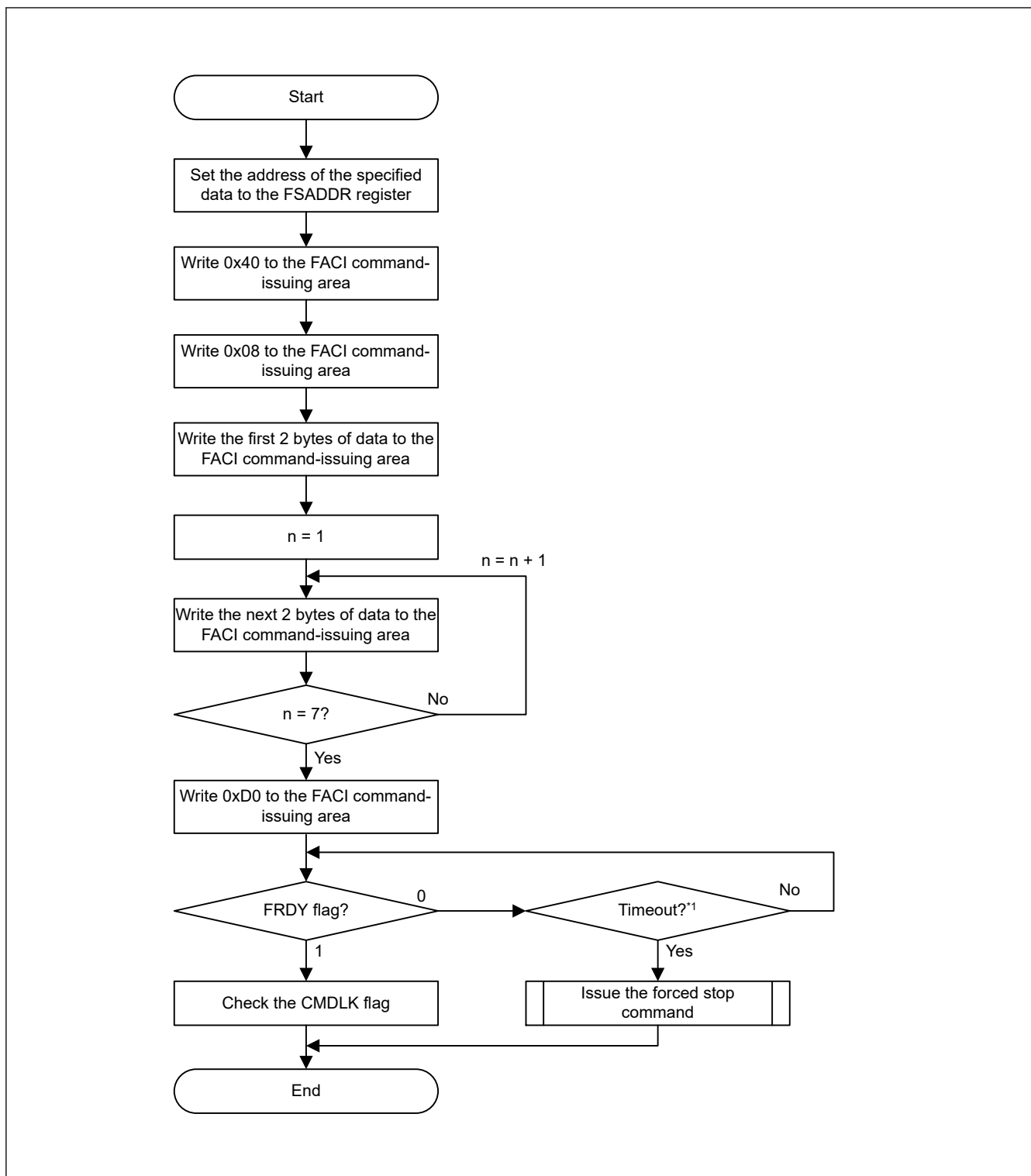
Figure 50.21 Usage of the Forced Stop Command

Usage of the forced stop command during issuing a command.

In the case where suspension due to the forced stop command is executed when a timeout occurs based on the DBFULL flag determination of the programming command, writing to the FACL command issuing area might be handled as written data of the programming command. In this case, after reading the FACL command issuing area to generate a command lock intentionally, issue the forced stop command in accordance with the recovery method from the command-locked state. Even in the case where an access size of reading the FACL command issuing area is any one of 8 bit, 16 bit, and 32 bit, a command lock can be generated.

50.6.4.11 Configuration Set Command

The configuration set command is used to set the security selection, safety setting, and system configuration setting. Before issuing a configuration set command, set the specified address (shown in [Table 50.15](#)) in the FSADDR register. Writing 0xD0 to the FACL command-issuing area in the final access for issuing the FACL command starts processing of the configuration set command.

**Figure 50.22 Usage of the Configuration Set Command**

The correspondence between the possible target data for configuration setting and the address value set in the FSADDR register is shown in [Table 50.15](#)

Table 50.15 Address Used by Configuration Set Command (1 of 2)

Address	Setting Data	Additional Writing Operation	
		FSPR = 1	FSPR = 0
0x0100_A150	ID for authentication	Possible	Possible

Table 50.15 Address Used by Configuration Set Command (2 of 2)

Address	Setting Data	Additional Writing Operation	
		FSPR = 1	FSPR = 0
0x0100_A164	Access window setting, startup area select flag, and FSPR flag setting	Possible	Impossible *1

Note 1. After the FSPR flag is set to 0, it is impossible that the configuration set command changes the bit to 1. Consequently, the access window and the startup area select flag cannot be set any more. (When the configuration set command is executed for the 0x0100_A164 address, a command lock occurs.) The FSPR flag should be used with the utmost care

The data mapping of the ID for authentication is shown below.

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
0x0100_A150	ID4[7:0]								ID3[7:0]							
0x0100_A154	ID8[7:0]								ID7[7:0]							
0x0100_A158	ID12[7:0]								ID11[7:0]							
0x0100_A15C	ID16[7:0]								ID15[7:0]							

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0x0100_A150	ID2[7:0]								ID1[7:0]							
0x0100_A154	ID6[7:0]								ID5[7:0]							
0x0100_A158	ID10[7:0]								ID9[7:0]							
0x0100_A15C	ID14[7:0]								ID13[7:0]							

The data mapping of the access window setting, the startup area select flag, and the FSPR flag setting is shown below

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
0x0100_A160																
0x0100_A164	BTF LG						FAWE[9:0]									
0x0100_A168																
0x0100_A16C																

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0x0100_A160																
0x0100_A164	FSP R						FAWS[9:0]									
0x0100_A168																
0x0100_A16C																

50.6.5 Security Function (Preventing Tampering)

50.6.5.1 Serial Programming Mode Protection

The serial programming mode protection is executed by ID authentication. The FSCI protects the receiving of the flash sequencer command in accordance with the result of ID authentication. When the ID authentication is passed, the FSCI enables the flash sequencer command

50.6.5.2 OCD Mode Protection

In the OCD mode, the flash memory is protected by ID authentication. The entry system of the on-chip debugger controls the protection with ID authentication. The FSCI outputs the ID authentication results to the above system.

50.6.5.3 Protection by ID Code

ID code protection compares the ID code (128 bits) sent from the host with the ID code (128 bits) stored in the OSIS register, and enables the connection between the host and the device only when the two ID codes match. The device enables ID code protection when bit 127 of the ID code is 1, and compares values from bit 125 to bit 0 to determine whether the ID code authentication is passed or failed. When the ID code protection is enabled, the ALeRASE command can be used if bit 126 is 1. [Table 50.16](#) shows the details. For details of the OSIS register, see [section 7.2.5. OSIS : OCD/Serial Programmer ID Setting Register](#).

Table 50.16 Specifications for ID Code Protection

Operating Mode at Startup	ID Code	ALeRASE Command	ID Authentication	Connection between OCD/Serial Programmer and the device
Serial programming mode (SCI boot mode) On-chip debug mode (SWD boot mode)	Bit 127 = 1 and Bit 126 = 1	Available ^{*1}	Passed	The on-chip debugger (OCD) or the serial programmer can be connected with the device.
			Failed	The on-chip debugger (OCD) or the serial programmer cannot be connected with the device.
	Bit 127 = 1 and Bit 126 = 0	Not available	Passed	The on-chip debugger (OCD) or the serial programmer can be connected with the device.
			Failed	The on-chip debugger (OCD) or the serial programmer cannot be connected with the device.
	Bit 127 = 0	Not available	Failed	The on-chip debugger (OCD) or the serial programmer cannot be connected with the device.
			Failed	The on-chip debugger (OCD) or the serial programmer cannot be connected with the device.

Note 1. When the ID code sent from the on-chip debugger (OCD) or the serial programmer is ALeRASE in ASCII code (0x414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFF), the content of the user flash area is erased and all bits in the OSIS register become 1. However, the content of the user flash area is not erased when the AWS.FSPR flag is 0 or the security MPU is enabled. For details on the AWS.FSPR flag, see [section 7.2.4. AWS : Access Window Setting Register](#).

50.6.5.4 Protection by Access Window

An access window can only be specified within the user area of the code flash memory. Issuing a command for programming or erasing a code flash memory area outside the specified access window results in the flash sequencer being placed in the command-locked state. When the FSPR flag is 0, issuing a configuration setting command for specifying the access window or modifying the startup area select flag also causes a command-locked state. This function can be enabled in the self-programming mode, serial programming mode, or OCD mode.

Use both the AWS.FAWS[9:0] and AWS.FAWE[9:0] bits^{*1} to specify the access window. The following shows the correspondence between the access window settings and their protection coverage.

- FAWE = FAWS: The program and erasure commands can be executed in the entire user area of the code flash memory.
- FAWE > FAWS: The program and erasure commands can only be executed in the area from the block specified by the FAWS bits to the block immediately before that specified by the FAWE bits.
- FAWE < FAWS: The program and erasure commands cannot be executed anywhere within the user area of the code flash memory.

Note 1. For details on the AWS.FAWS and AWS.FAWE bits, see [section 7.2.4. AWS : Access Window Setting Register](#).

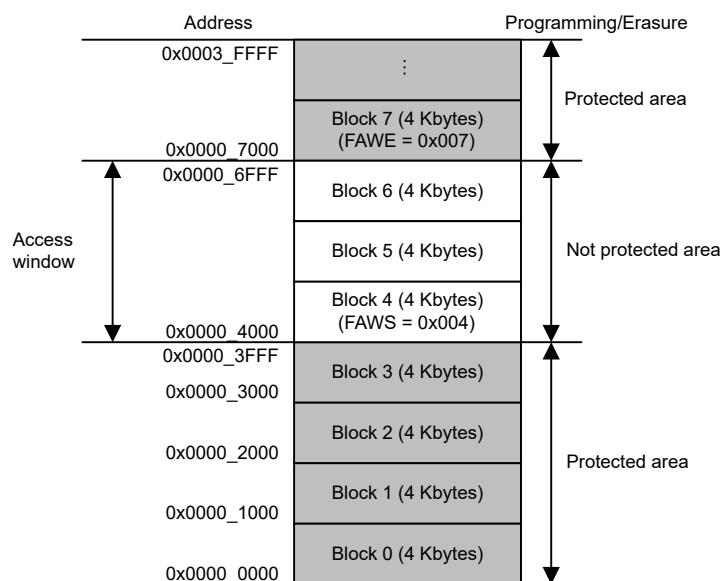


Figure 50.23 Start Block Address (FAWS) and End Block Address (FAWE) of Access Window (Setting Example)

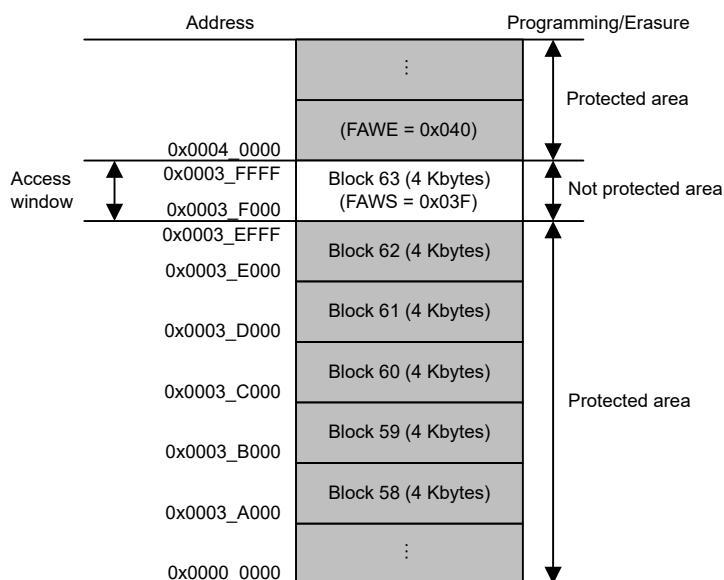


Figure 50.24 Start Block Address (FAWS) and End Block Address (FAWE) of Access Window when the Access Window Only Includes the Final Block

50.7 Protection (Preventing Unintended Writing)

50.7.1 Software Protection

Software protection disables writing and erasure for the code flash memory through the settings of control registers. If an attempt is made to issue an FACI command against software protection, the flash sequencer enters the command-locked state.

50.7.1.1 Protection through the FWEPROR Register

Rewriting is prohibited for the flash memory unless the FWEPROR register is set to 0x0001. In the rewriting prohibition state, the FACI command cannot be received. If an FACI command is issued in the rewriting prohibition state, the flash sequencer enters the command-locked state.

50.7.1.2 Protection through the FENTRYR Register

When the FENTRYR register is set to 0x0000, the flash sequencer enters read mode. In read mode, FACI commands cannot be accepted. If an attempt is made to issue an FACI command in read mode, the flash sequencer enters the command-locked state.

50.7.2 Error Protection

Error protection detects erroneous issuance of FACI commands, unauthorized access, and flash sequencer malfunction. FACI command acceptance is disabled (command-locked state) in response to the detection of these errors. The code flash memory cannot be written or erased while the flash sequencer is in the command-locked state. To release it from the command-locked state, a status clear command or a forced stop command must be issued. The status clear command can only be used while the FSTATR.FRDY flag is 1. The forced stop command can be used regardless of the value of the FRDY flag. In the case where the value of the CMDLKIE bit of the FAEINT register becomes 1, when the flash sequencer enters the command-locked state (the FASTAT.CMDLK flag is 1), a flash access error (FCU_FIFERR) interrupt is generated.

When the flash sequencer enters the command-locked state in response to a command other than the P/E suspend command during writing or erasure processing, the flash sequencer continues the processing for writing or erasure. In this state, the P/E suspend command cannot be used to suspend the processing for writing or erasure. If a command is issued in the command-locked state, the ILGLERR flag and the ILGCOMERR flag become 1 and the other flags retain the values set due to previous error detection.

Table 50.17 shows error protection types and status flag values after error detection.

Table 50.17 Error Protection Type (1 of 2)

Error Type	Description	ILGCOMERR	FESEERR	SECERR	OTERR	ILGLERR	ERSERR	PRGERR	FLWERR	CFAE
FENTRYR setting error	The FENTRYR register setting at suspension disagrees with that at resumption.	0	1	0	0	1	0	0	0	0

Table 50.17 Error Protection Type (2 of 2)

Error Type	Description	ILGC OMER R	FESE TERR	SECE RR	OTER R	ILGL ERR	ERSE RR	PRGE RR	FLW EER R	CFAE
Illegal command error	Access with an undefined size in the first access of the FACL command (is not byte write)	1	0	0	0	1	0	0	0	0
	An undefined code is written in the first access of the FACL command.	1	0	0	0	1	0	0	0	0
	The value specified in the last access of the multiple-access FACL command is other than 0xD0.	1	0	0	0	1	0	0	0	0
	The value (N) specified by the second write access of the FACL command in the programming or configuration setting is incorrect.	1	0	0	0	1	0	0	0	0
	The FACL command not acceptable in each mode has been issued (see Table 50.13).	1	0	0	0	1	0	0	0	0
	The program and block erase commands have been issued to an area that is protected by the access window.	1	0	0	0	1	0	0	0	0
	The program has been issued to an area that is being erased in erasure suspended.	1	0	0	0	1	0	0	0	0
	The FACL command has been issued when command acceptance conditions are not satisfied (see Table 50.14). ^{*1}	0/1	0/1	0/1	0/1	1	0/1	0/1	0/1	0/1
Erase error	An error has occurred during erasure.	0	0	0	0	0	1	0	0	0
Writing error	An error has occurred during writing.	0	0	0	0	0	0	1	0	0
Code flash memory access violation	The program and block erase commands have been issued to the reserved portion of the user area in the code flash P/E mode. (See section 50.3.3 . FWEPROR : Flash Write Erase Protect Register)	0	0	0	0	1	0	0	0	1
	The configuration set command has been issued to the reserved area.	0	0	0	0	1	0	0	0	1
Security error	ID authentication in the serial programming mode has resulted in failure, and the FACL command has been issued.	0	0	1	0	1	0	0	0	0
	When the FSPR flag is 0, the configuration set command has been issued for the access window setting and BTFLG flag setting.	0	0	1	0	1	0	0	0	0
Others	The FACL command-issuing area has been accessed in read mode.	0	0	0	1	1	0	0	0	0
	The FACL command-issuing area has been read in the code flash P/E mode.	0	0	0	1	1	0	0	0	0
Flash write/erase protect error	Programming erasure is prohibited by the flash write erase protect register (FWEPROR) during the FACL command processing.	0	0	0	0	0	0/1	0/1	1	0

Note 1. The information on command issuance is inherited.

50.7.3 Boot Program Protection

This device provides two functions for protecting user boot programs from unintended update and erasure: the startup area select function and the protection by the access window. For details on the protection by the access window, see [section 50.6.5.4. Protection by Access Window](#).

The startup area select function switches block 0 to 7 and block 8 to 15 that are allocated to addresses 0x0000_0000 to 0x0000_FFFF including a reset vector, thus providing a program updating method that is safe for suspension of rewriting. [Table 50.18](#) shows the startup area select function, and [Figure 50.26](#) shows an example of usage procedures. The BTFLG flag of the AWS register and SAS[1:0] bits of the FSUACR register described in [section 7, Option-Setting Memory](#),

determine whether or not switching (boot swap) is executed. When changing a startup area, set the FLWT register to no cycles of waiting (FLWT = 0x000). In the boot swap state, targets of rewriting (P/E) by FACL commands are also switched.

The startup area select function can fix the boot swap state by the FSPR flag. However, since reconfiguring is impossible after fixing the boot swap state, the FSPR flag should be used with the utmost care. (See [Table 50.15](#).)

Table 50.18 Startup Area Select Function

SAS[1]	SAS[0]	BTFLG	Boot Swap
1	0	*	Normal
	1	*	Boot swap
0	*	0	Boot swap
	*	1	Normal

Note: * Don't care

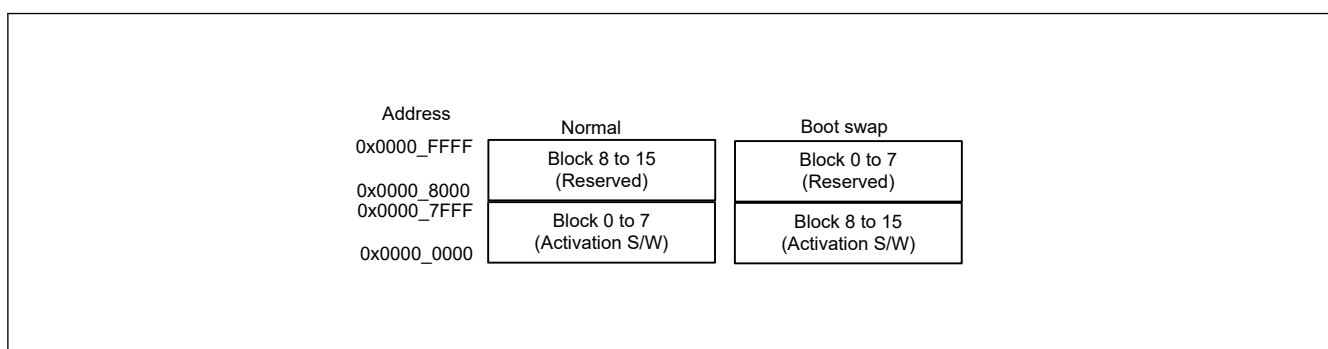


Figure 50.25 Configuration of block 0 and block 1

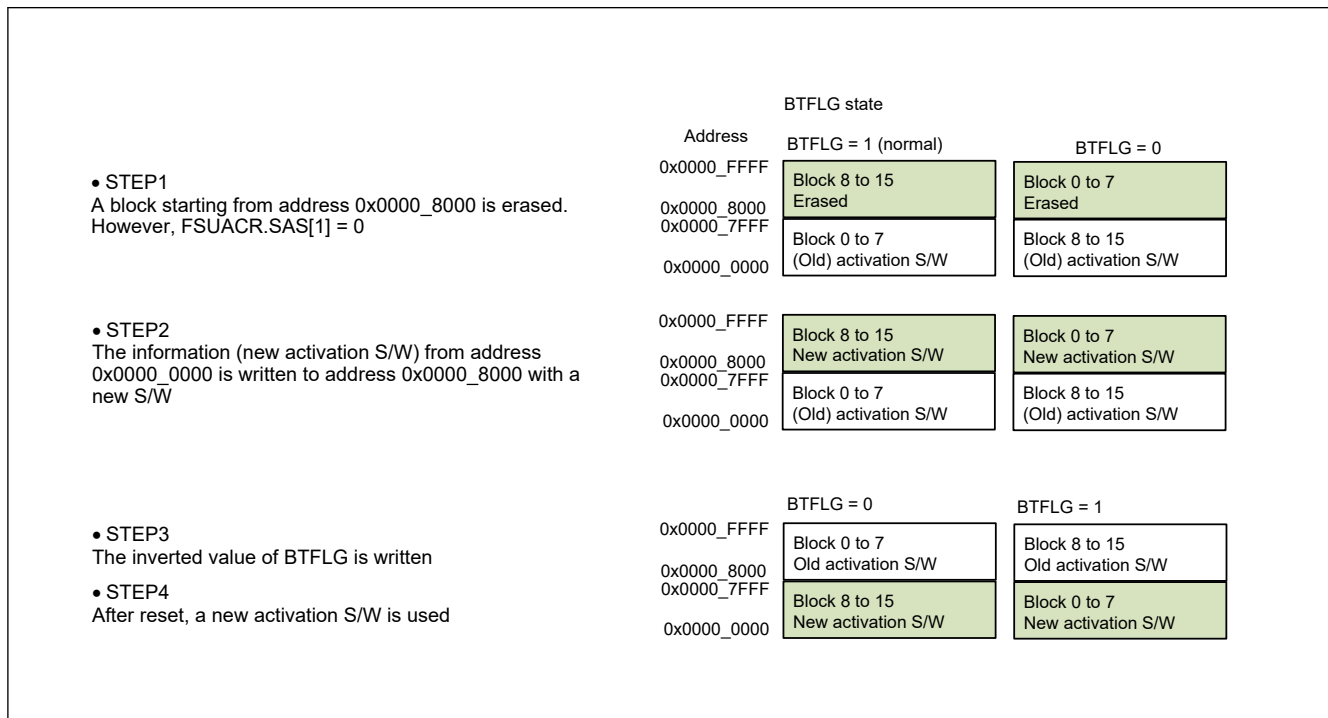


Figure 50.26 Usage Procedures of the Startup Area Select Function

50.8 Usage Notes

50.8.1 Suspension of Programming/Erase

When processing of programming/erase is stopped by issuing the P/E suspend command, the programming/erase processing can be resumed by issuing the P/E resume command. If the flash sequencer enters the command-locked state for any reason and issues the forced stop command after the suspended processing is normally completed and the ERSSPD flag or PRGSPD flag is set to 1, the suspended processing cannot be resumed. In addition, the values in the area where the processing was suspended are not guaranteed. Erase that area.

50.8.2 Reading Areas Where Programming or Erase was Interrupted

When programming or erase of an area of code flash memory is interrupted, the data stored in the area become undefined. To avoid faulty operation caused by reading undefined data, take care not to fetch instructions or read data from areas where programming/erase was suspended.

50.8.3 Allocation of Vectors for Interrupts and Other Exceptions during Programming and Erase

Generation of an interrupt or other exception during programming or erase might lead to fetching of the vector from the code flash memory. Set the address for vector fetching to an address that is not in the code flash memory.

50.8.4 Notes on Additional Writes

Be sure to erase a code flash memory area before attempting to write to that area. However, the configuration area can be overwritten.

50.8.5 Notes on Programming/Erase

Configure the following settings for programming and erase.

- Release the flash write erase protect (FWEPROR.FLWE[1:0] = 01b).
- Do not generate the voltage monitor 1 interrupt reset (LVD1CR0.RIE = 0).
- Do not generate the voltage monitor BAT interrupt reset (LVDBATCR0.RIE = 0).
- Set the all power supply mode (ALLPWON) for the power supply mode.
- Even during programming/erase, maintain the system clock (ICLK) values set before the execution of programming/erase.
- When programming or erasing the code flash memory, set the ICLK frequency to a value from 1 MHz to 32MHz.

51. Electrical Characteristics

The electrical characteristics are defined under the following conditions unless otherwise specified:

- $VCC = AVCC0 = IOVCC0 = IOVCC1 = 1.62$ to 3.6 V
- $1.62\text{ V} \leq VREFH0 \leq AVCC0$
- $VSS = AVSS0 = VREFL0 = 0$ V
- $T_a = T_{opr}$
- The load capacitance of each I/O pin is 30 pF.

51.1 Absolute Maximum Ratings

Table 51.1 Absolute maximum ratings

Item	Symbol	Value	Unit
Power-supply voltage	Power-supply voltage	VCC	-0.3 to 4.6
	Input voltage for EHC	VSC_VCC	-0.3 to 4.6
	Input voltage of secondary battery for EHC	VBAT_EHC	-0.3 to 4.6
	Power-supply voltage for I/O	IOVCC, IOVCC0, IOVCC1	-0.3 to 4.6
Input voltage	V_{in}	-0.3 to VCC + 0.3 (max. 4.6 V)	V
Reference power supply voltage	VREFH0	-0.3 to AVCC0 + 0.3 (max. 4.6 V)	V
	VREFL0	-0.3 to AVSS0 + 0.3	V
Analog power supply voltage	AVCC0	-0.3 to 4.6	V
Junction temperature	T_j	-40 to +95	°C
Storage Temperature	T_{stg}	-55 to +125	°C

Caution: Permanent damage to the LSI might result if absolute maximum ratings are exceeded.

Table 51.2 Recommended operating conditions

Item	Symbol	Min.	Typ.	Max.	Unit
Power-supply voltage	VCC	1.62	—	3.6	V
	VSS	—	0	—	V
Input voltage for EHC	VSC_VCC	1.62	—	3.6	V
Input voltage of secondary battery for EHC	VBAT_EHC*1	1.62	—	3.6	V
Analog power supply voltage	AVCC0	1.62	—	3.6	V
	AVSS0	—	0	—	V
	VREFH0	1.62	—	AVCC0	V
	VREFL0	—	0	—	V
Power-supply for I/O	IOVCC, IOVCC0, IOVCC1	1.62	—	3.6	V
Power-supply voltage	T_{opr}	-40	—	85	°C

Note 1. The voltage of the secondary battery to be connected to VBAT_EHC is 2.4 V, 2.5 V, 2.6 V, 2.7 V, 2.8 V, 2.9 V, 3.0 V, or 3.1 V.

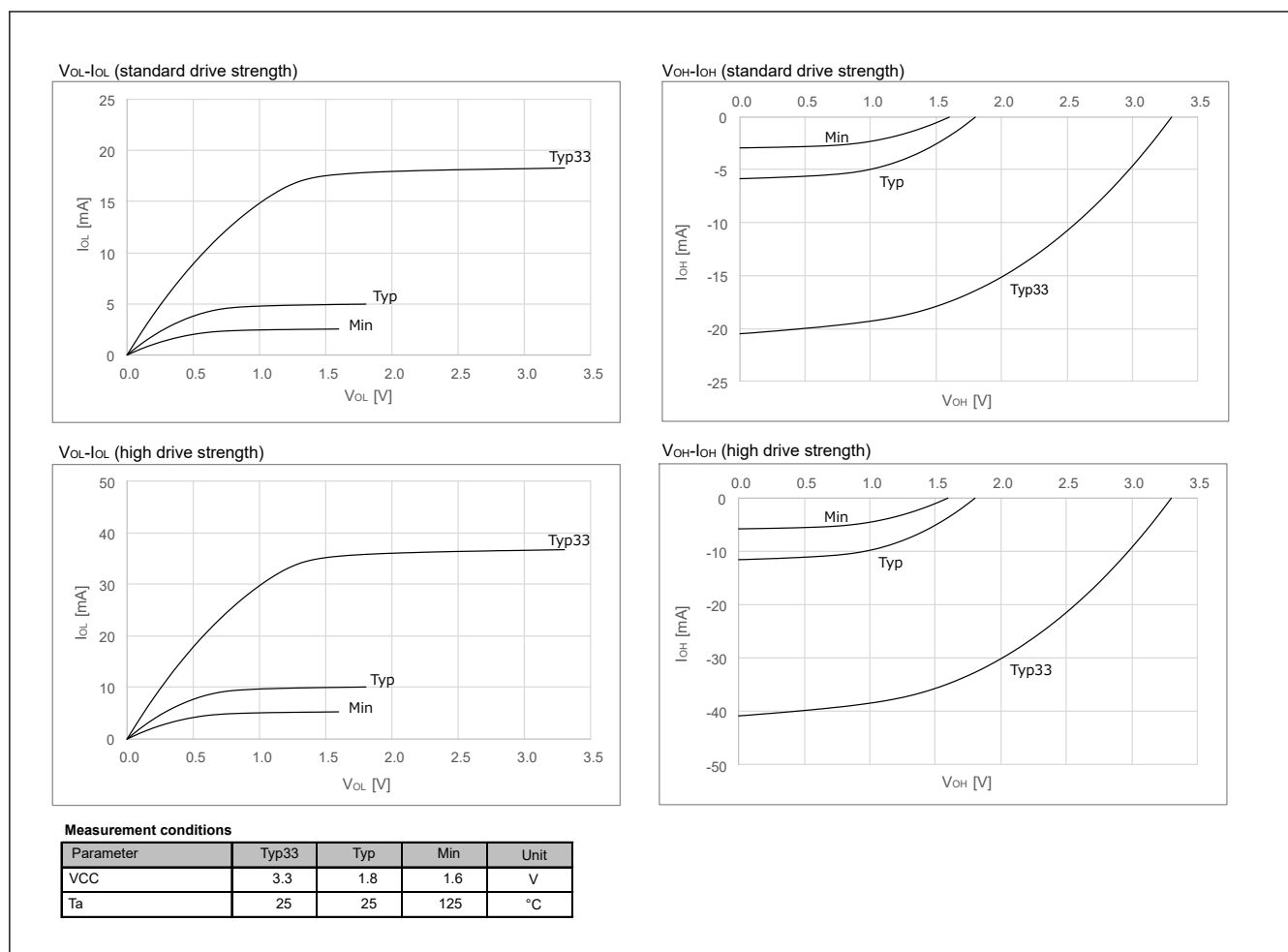
51.2 DC Characteristics

51.2.1 I/O input characteristics (V_{IH} , V_{IL})Table 51.3 I/O input characteristics (V_{IH} , V_{IL})

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Schmitt trigger input voltage	Input pins of RES#, NMI, IRQn, and peripheral functions (except for IIC)	V_{IH}	$VCC \times 0.8$	—	—	V	—
		V_{IL}	—	—	$VCC \times 0.2$		
		ΔV_T	0.3	—	—		
	IIC	V_{IH}	$VCC \times 0.7$	—	—		VCC = 3.0 to 3.6 V
		V_{IL}	—	—	$VCC \times 0.3$		
		ΔV_T	$VCC \times 0.05$	—	—		
Input voltage (except for Schmitt trigger input pin)	EXTAL, MD, EHMD, General-Purpose I/O Ports	V_{IH}	$VCC \times 0.8$	—	—		—
		V_{IL}	—	—	$VCC \times 0.2$		

51.2.2 I/O output characteristics (V_{OH} , V_{OL}) (1)Table 51.4 I/O output characteristics (V_{OH} , V_{OL}) (1)

Item	Register settings	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Output high level voltage	Standard drive (PmnPFS.DSCR[1:0] = 10b)	V_{OH}	$VCC - 0.6$	—	—	V	$I_{OH} = 2 \text{ mA}$
	High drive (PmnPFS.DSCR[1:0] = 11b)		$VCC - 0.5$	—	—		$I_{OH} = 2 \text{ mA}$
Output low level voltage	Standard drive (PmnPFS.DSCR[1:0] = 10b)	V_{OL}	—	—	0.6		$I_{OL} = 2 \text{ mA}$
	High drive (PmnPFS.DSCR[1:0] = 11b)		—	—	0.5		$I_{OL} = 2 \text{ mA}$

Figure 51.1 V_{OH} - I_{OH} and V_{OL} - I_{OL} characteristics51.2.3 I/O output characteristics (V_{OL}) (2)Table 51.5 I/O output characteristics (V_{OL}) (2)

Condition: VCC = 3.0 to 3.6 V

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Output low level voltage	IIC	—	—	0.4	V	$I_{OL} = 3 \text{ mA}$
		—	—	0.6		$I_{OL} = 6 \text{ mA}$

51.2.4 Pull-up Resistors

Table 51.6 Pull-up resistors

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Pull-up resistor	I_P	120	200	—	k Ω	VCC = 2.5 V

51.2.5 Pin Capacitance

Table 51.7 Pin capacitance

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Pin Related to IIC	P012, P013, P700, P701	C _{in}	—	—	8	pF	—
EXTAL, XTAL	P412, P413						
All of pins other than above			—	—	16		

Note: [section 1, Overview Table 1.15](#) shows the detail.

51.2.6 Operating and Standby Current

Table 51.8 Operating and Standby Current (1 of 6)Measurement condition of maximum values: VCC = AVCC0 = AVCC1 = VREFH0 = 3.6 V, T_a = T_{opr} = 85°CTypical measurement conditions: VCC = AVCC0 = AVCC1 = VREFH0 = 3.3 V, T_a = T_{opr} = 25°C

Power supply mode	Power control mode/low power mode		Setting value of operating frequency	Clock source	Typ.	Max.	Unit
The mode code of all the power supply (ALLPWON) is executed from the flash memory.	BOOST	Maximum operation*1	ICLK/PCLKB = 64/32 MHz	HOCO	—	14	mA
			ICLK/PCLKB = 32/16 MHz		—	9.1*3	
		while(1) operation (peripheral clock is supplied.)	ICLK/PCLKB = 32/32 MHz	MOSC	3.6	—	
			ICLK/PCLKB = 16/16 MHz		1.9	—	
			ICLK/PCLKB = 32/16 MHz		3.0	—	
			ICLK/PCLKB = 64/32 MHz	HOCO	5.7	—	
			ICLK/PCLKB = 32/16 MHz		2.9	—	
		CoreMark (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 64/1 MHz	HOCO	2.3	—	
			ICLK/PCLKB = 32/0.5 MHz		1.2	—	
		while(1) operation (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 64/32 MHz	HOCO	2.0	—	
			ICLK/PCLKB = 32/16 MHz		1.1	—	
			ICLK/PCLKB = 32/32 MHz	MOSC	1.2	—	
			ICLK/PCLKB = 16/16 MHz		0.7	—	
			ICLK/PCLKB = 64/1 MHz	HOCO	1.8	—	
			ICLK/PCLKB = 32/0.5 MHz		1.0	—	
		Sleep mode (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 64/32 MHz	HOCO	0.9	—	
			ICLK/PCLKB = 32/16 MHz		0.5	—	
			ICLK/PCLKB = 64/1 MHz		0.7	—	
			ICLK/PCLKB = 32/0.5 MHz		0.5	—	
	Normal	Maximum operation*1	ICLK/PCLKB = 32/32 MHz	MOSC	—	8.3	mA
			ICLK/PCLKB = 16/16 MHz		—	6.1*3	
		while(1) operation (peripheral clock is supplied.)	ICLK/PCLKB = 32/32 MHz	MOSC	2.9	7.6	
			ICLK/PCLKB = 16/16 MHz		1.5	5.7*3	
			ICLK/PCLKB = 32/32 MHz	HOCO	2.8	7.5	
			ICLK/PCLKB = 16/16 MHz		1.6	5.8*3	
		CoreMark (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 32/0.50 MHz	MOSC	1.1	—	
			ICLK/PCLKB = 16/0.25 MHz		0.61	—	
		while(1) operation (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 32/32 MHz	MOSC	0.97	—	
			ICLK/PCLKB = 16/16 MHz		0.56	—	
			ICLK/PCLKB = 32/0.50 MHz		0.84	5.7	
			ICLK/PCLKB = 16/0.25 MHz		0.49	—	
		Sleep mode (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 32/32 MHz	MOSC	0.56	—	
			ICLK/PCLKB = 16/16 MHz		0.35	—	
			ICLK/PCLKB = 32/0.50 MHz		0.45	—	
			ICLK/PCLKB = 16/0.25 MHz		0.3	—	

Table 51.8 Operating and Standby Current (2 of 6)Measurement condition of maximum values: VCC = AVCC0 = AVCC1 = VREFH0 = 3.6 V, T_a = T_{opr} = 85°CTypical measurement conditions: VCC = AVCC0 = AVCC1 = VREFH0 = 3.3 V, T_a = T_{opr} = 25°C

Power supply mode	Power control mode/low power mode		Setting value of operating frequency	Clock source	Typ.	Max.	Unit
The mode code of all the power supply (ALLPWON) is executed from the flash memory.	VBB	Maximum operation*1	ICLK/PCLKB = 32.7/32.7 kHz	LOCO	—	120*3	μA
		while(1) operation (peripheral clock is supplied.)	ICLK/PCLKB = 32.7/32.7 kHz		44	—	
			ICLK/PCLKB = 32.7/0.51 kHz		43	—	
		Sleep mode (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 32.7/0.51 kHz		40	—	
		while(1) operation (peripheral clock is supplied.)	ICLK/PCLKB = 32.768/32.768 kHz	SOSC (standard CL)	44	—	
		Sleep mode (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 32.768/0.512 kHz		40	—	
		while(1) operation (peripheral clock is supplied.)	ICLK/PCLKB = 32.768/32.768 kHz	SOSC (low CL)	43	—	
		while(1) operation (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 32.768/32.768 kHz		43	—	
			ICLK/PCLKB = 32.768/0.512 kHz		43	—	
		Sleep mode (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 32.768/32.768 kHz		39	—	
			ICLK/PCLKB = 32.768/0.512 kHz		39	—	

Table 51.8 Operating and Standby Current (3 of 6)Measurement condition of maximum values: VCC = AVCC0 = AVCC1 = VREFH0 = 3.6 V, T_a = T_{opr} = 85°CTypical measurement conditions: VCC = AVCC0 = AVCC1 = VREFH0 = 3.3 V, T_a = T_{opr} = 25°C

Power supply mode	Power control mode/low power mode			Setting value of operating frequency	Clock source	Typ.	Max.	Unit			
The mode code of power supply other than flash (EXFPWON) is executed from the SRAM.	BOOST	Software standby mode*4	When VCC = 3.3 V	ICLK/PCLKB = 32.7/32.7 kHz	LOCO	39	—	μA			
			When VCC = 1.8 V	ICLK/PCLKB = 32.7/32.7 kHz		38	—				
	BOOST_V BB	Software standby mode*4	When VCC = 3.3 V	ICLK/PCLKB = 32.7/32.7 kHz	LOCO	14	—	μA			
			When VCC = 1.8 V	ICLK/PCLKB = 32.7/32.7 kHz		13	—				
	Normal	High-speed mode	Maximum operation*1	ICLK/PCLKB = 32/32 MHz	MOSC	—	7.3*3	mA			
				ICLK/PCLKB = 16/16 MHz		—	5.8*3				
			while(1) operation (peripheral clock is supplied.)	ICLK/PCLKB = 32/32 MHz		MOSC	2.8		—		
				ICLK/PCLKB = 16/16 MHz			1.5		—		
				ICLK/PCLKB = 32/32 MHz	HOCO	2.8	—				
						ICLK/PCLKB = 16/16 MHz	1.5		—		
			while(1) operation (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 32/32 MHz	MOSC	0.93	—				
				ICLK/PCLKB = 16/16 MHz		0.52	—				
				ICLK/PCLKB = 32/0.50 MHz		0.8	—				
				ICLK/PCLKB = 16/0.25 MHz		0.45	—				
			Sleep mode (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 32/32 MHz	MOSC	0.52	—				
				ICLK/PCLKB = 16/16 MHz		0.32	—				
				ICLK/PCLKB = 32/0.50 MHz		0.41	—				
				ICLK/PCLKB = 16/0.25 MHz		0.26	—				
			Low-speed mode	Maximum operation*1	ICLK/PCLKB = 2/2 MHz	MOSC	—		4.4*3	mA	
					ICLK/PCLKB = 1/1 MHz		—		4.3*3		
				while(1) operation (peripheral clock is supplied.)	ICLK/PCLKB = 2/2 MHz	MOSC	0.22		—		
					ICLK/PCLKB = 1/1 MHz		0.13		—		
					ICLK/PCLKB = 2/2 MHz	MOCO	0.2		—		
							ICLK/PCLKB = 1/1 MHz		0.12		—
				while(1) operation (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 2/2 MHz	MOSC	0.10		—		
					ICLK/PCLKB = 1/1 MHz		0.07		—		
					ICLK/PCLKB = 2000/31.25 kHz		0.09		—		
					ICLK/PCLKB = 1000/31.25 kHz		0.07		—		
				Sleep mode (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 2/2 MHz	MOSC	0.07		—		
					ICLK/PCLKB = 1/1 MHz		0.06		—		
					ICLK/PCLKB = 2000/31.25 kHz		0.07		—		
					ICLK/PCLKB = 1000/31.25 kHz		0.05		—		
				Software standby mode*4	When VCC = 3.3 V	LOCO	24		—		μA
							24		—		
	SOSC (standard CL)	24				—					
		24				—					
	SOSC (low CL)	23				—					
		23				—					

Table 51.8 Operating and Standby Current (4 of 6)Measurement condition of maximum values: VCC = AVCC0 = AVCC1 = VREFH0 = 3.6 V, T_a = T_{opr} = 85°CTypical measurement conditions: VCC = AVCC0 = AVCC1 = VREFH0 = 3.3 V, T_a = T_{opr} = 25°C

Power supply mode	Power control mode/low power mode		Setting value of operating frequency	Clock source	Typ.	Max.	Unit
The mode code of power supply other than flash (EXFPWON) is executed from the SRAM.	VBB	Maximum operation*1	ICLK/PCLKB = 32.7/32.7 kHz	LOCO	—	26*3	μA
		while(1) operation (peripheral clock is supplied.)	ICLK/PCLKB = 32.7/32.7 kHz		4.1	—	
		Sleep mode (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 32.7/0.51 kHz		1.6	—	
		Software standby mode*4	When VCC = 3.3/3.6 V		1.4	22*3	
			When VCC = 1.8 V		1.3	—	
		while(1) operation (peripheral clock is supplied.)	ICLK/PCLKB = 32.768/32.768 kHz	SOSC (standard CL)	4.7	—	
		Sleep mode (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 32.768/0.512 kHz		2.2	—	
		Software standby mode*4	When VCC = 3.3 V		2.0	—	
			When VCC = 1.8 V		1.9	—	
		while(1) operation (peripheral clock is supplied.)	ICLK/PCLKB = 32.768/32.768 kHz	SOSC (low CL)	4.0	—	
		while(1) operation (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 32.768/32.768 kHz		4.0	—	
		Sleep mode (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 32.768/0.512 kHz		1.5	—	
		Software standby mode*4	When VCC = 3.3 V		1.3	—	
			When VCC = 1.8 V		1.2	—	

Table 51.8 Operating and Standby Current (5 of 6)Measurement condition of maximum values: VCC = AVCC0 = AVCC1 = VREFH0 = 3.6 V, T_a = T_{opr} = 85°CTypical measurement conditions: VCC = AVCC0 = AVCC1 = VREFH0 = 3.3 V, T_a = T_{opr} = 25°C

Power supply mode	Power control mode/low power mode			Setting value of operating frequency	Clock source	Typ.	Max.	Unit
The mode code of the minimum power supply (MINPWON) is executed from the SRAM.	BOOST	Software standby mode*4	When VCC = 3.3 V	ICLK/PCLKB = 32.7/32.7 kHz	LOCO	29	—	μA
			When VCC = 1.8 V	ICLK/PCLKB = 32.7/32.7 kHz		28	—	
	BOOST_V BB	Software standby mode*4	When VCC = 3.3 V	ICLK/PCLKB = 32.7/32.7 kHz	LOCO	14	—	μA
			When VCC = 1.8 V	ICLK/PCLKB = 32.7/32.7 kHz		13	—	
	Normal	High-speed mode	Maximum operation*1	ICLK/PCLKB = 32/32 MHz	MOSC	—	4.6*3	mA
				ICLK/PCLKB = 16/16 MHz		—	3.8*3	
			while(1) operation (peripheral clock is supplied.)	ICLK/PCLKB = 32/32 MHz	MOSC	1.3	—	
				ICLK/PCLKB = 16/16 MHz		0.72	—	
			while(1) operation (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 32/32 MHz	MOSC	0.9	—	
				ICLK/PCLKB = 16/16 MHz		0.5	—	
				ICLK/PCLKB = 32/0.5 MHz		0.78	3.7*3	
				ICLK/PCLKB = 16/0.5 MHz		0.44	—	
			Sleep mode (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 32/32 MHz	MOSC	0.5	—	
				ICLK/PCLKB = 16/16 MHz		0.3	—	
				ICLK/PCLKB = 32/0.5 MHz		0.39	—	
				ICLK/PCLKB = 16/0.5 MHz		0.25	—	
		Low-speed mode	Maximum operation*1	ICLK/PCLKB = 2/2 MHz	MOSC	—	3000*3	μA
				ICLK/PCLKB = 1/1 MHz		—	2900*3	
			while(1) operation (peripheral clock is supplied.)	ICLK/PCLKB = 2/2 MHz	MOCO	108	—	
				ICLK/PCLKB = 1/1 MHz		60	—	
			while(1) operation (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 2000/31.25 kHz	MOSC	78	—	
				ICLK/PCLKB = 1000/31.25 kHz		52	—	
				ICLK/PCLKB = 2/2 MHz	MOCO	68	—	
				ICLK/PCLKB = 1/1 MHz		46	—	
				ICLK/PCLKB = 2000/31.25 kHz		60	—	
				ICLK/PCLKB = 1000/31.25 kHz		42	—	
			Sleep mode (stopping clock supply to peripheral functions*2)	ICLK/PCLKB = 2/2 MHz	MOCO	43	—	
				ICLK/PCLKB = 1/1 MHz		34	—	
				ICLK/PCLKB = 2000/31.25 kHz		36	—	
				ICLK/PCLKB = 1000/31.25 kHz		30	—	
	Software standby mode*4		When VCC = 3.3 V		LOCO	14	—	μA
						14	—	
			When VCC = 3.3 V		SOSC (standard CL)	14	—	
						14	—	
			When VCC = 3.3 V		SOSC (low CL)	14	—	
						13	—	

Table 51.8 Operating and Standby Current (6 of 6)Measurement condition of maximum values: VCC = AVCC0 = AVCC1 = VREFH0 = 3.6 V, T_a = T_{opr} = 85°CTypical measurement conditions: VCC = AVCC0 = AVCC1 = VREFH0 = 3.3 V, T_a = T_{opr} = 25°C

Power supply mode	Power control mode/low power mode		Setting value of operating frequency	Clock source	Typ.	Max.	Unit		
The mode code of the minimum power supply (MINPWON) is executed from the SRAM.	VBB	while(1) operation (peripheral clock is supplied.)		SOSC (standard CL)	2.4	10 ⁻³	μA		
		Sleep mode (stopping clock supply to peripheral functions*2)			1.4	—			
		Software standby mode*4	When VCC = 3.3 V		1.2	—			
			When VCC = 1.8 V		1.1	—			
		while(1) operation (peripheral clock is supplied.)		LOCO	2	10 ⁻³	μA		
		Sleep mode (stopping clock supply to peripheral functions*2)			0.9	8.5 ⁻³			
		Software standby mode	VCC = 3.3 V (Typ.)/ 3.6 V (Max.)		LOCO	0.6	8.4	μA	
			When VCC = 1.8 V			0.5	7.3 ⁻³		
		while(1) operation (peripheral clock is supplied.)		SOSC (low CL)	1.7	—	μA		
		while(1) operation (stopping clock supply to peripheral functions*2)			1.7	—			
		Sleep mode (stopping clock supply to peripheral functions*2)			0.7	—			
		Software standby mode*4	When VCC = 3.3 V		0.5	7 ⁻³			
			When VCC = 1.8 V		0.4	5.8 ⁻³			
		Software standby mode*4			Increment when IWDT is used (OFS0.IWDTSTRT = 0)			81	—
		Increase when peripheral modules are in use		Increase for using AGT and AGTW (AGTCR.TSTART = 1)			38	—	
		(No dependency on VCC)		Increase for each 32 KB of SRAM in use (set by the RAMSDCR register)			12	—	
Deep software standby			VCC = 3.3 V (Typ.)/ 3.6 V (Max.)	—	—	120	1600 ⁻³	nA	
			When VCC = 1.8 V	—	—	100	1200 ⁻³		
			Increment when SOSC is used (VCC = 3.3 V)		SOSC (low CL)	160	—		
			Increment when SOSC is used (VCC = 1.8 V)			100	—		
Increment on peripheral function during the standby mode			Increase for using LVD0 (OFS1.LVDAS = 0)			48	—	nA	
			Increase for using LVD1 (LVCMPCR.LVD1E = 1)			66	—		
			Increase for using LVDBAT (LVCMPCR.LVDBATR = 1)			66	—		
			Increase for using CCC (CADJUSCEN = 1 and ADUSTEN = 1) (3.3 V)			35	—		
			Increase for using CCC (CADJUSCEN = 1 and ADUSTEN = 1) (1.8 V)			12	—		
			Increase for using WUPT (TCR.TCST = 1 and TCR.TCCE = 1) (3.3 V)			65	—		
			Increase for using WUPT (TCR.TCST = 1 and TCR.TCCE = 1) (1.8 V)			30	—		
			Increase for using RTC (RCR4.R32KMD = 0 and RCR2.CNTMD = 1) (3.3 V)			200	—		
			Increase for using RTC (RCR4.R32KMD = 0 and RCR2.CNTMD = 1) (1.8 V)			100	—		
			Increase for using RTC (RCR4.R32KMD = 1 and RCR2.CNTMD = 1) (3.3 V)			280	—		
			Increase for using RTC (RCR4.R32KMD = 1 and RCR2.CNTMD = 1) (1.8 V)			150	—		

- Note 1. The value for current in a "Maximum operation" row is for a case where the DMAC is handling transfer in every cycle and the CPU is repeatedly executing a multiply instruction while all modules are released from the module-stop state. The value does not include the supply of current for the pins.
- Note 2. The value for current in a row with a label that includes "stopping clock supply to peripheral functions" is for a case where the peripheral circuits have been placed in the module-stop state following the settings for frequency-division of ICLK and PCLKB.
- Note 3. We do not inspect this value before shipment. The values presented in this manual are only for reference.
- Note 4. The supply of the clock signals is stopped in this mode regardless of the operating frequency settings.

Table 51.9 Analog operating current (AVCC0) and standby current

Maximum measurement conditions: VCC = AVCC0 = VREFH0 = 3.6 V, T_a = T_{opr} = 85°C

Typical measurement conditions: VCC = AVCC0 = VREFH0 = 3.3 V, T_a = T_{opr} = 25°C (when VREF circuit is not used.)

Typical measurement conditions: VCC = AVCC0 = 3.3 V, AVTRO = 1.25 V, T_a = T_{opr} = 25°C (when VREF circuit is used.)

Item	Operation state of circuit			Symbol	Typ.	Max.	Unit	Measurement conditions
	A/D	Temperature sensor	VREF					
AVCC0 power supply current	Under conversion	Under operation	Under operation	I _{AVCC0}	81	—	μA	PCLKB = 16 MHz Sampling time is 1 μs. (ADSSTRn.SST[7:0] = 0x10)
		Stopped	Under operation		77	—		
		Under operation	Stopped		69	—		
		Stopped	Stopped		53	—		
		Stopped	Stopped		0.19	—		
	Waiting for conversion	Stopped	Stopped		22	—	nA	PCLKB = 16 MHz* ¹
	Standby mode				22	1900		Clock supply is stopped.
Reference power supply current	Under conversion	Stopped	Stopped	I _{REFH0}	18	—	μA	PCLKB = 16 MHz
					0.08	—		PCLKB = 32.768 kHz
	Waiting for conversion	Stopped	Stopped		22	—	nA	PCLKB = 16 MHz* ¹
	Standby mode				22	—		Clock supply is stopped.

Note 1. This indicates that the clock is supplied to the A/D converter, but the A/D conversion is not performed.

Table 51.10 IOVCC wait current

Maximum measurement conditions: VCC = IOVCCn = 3.6 V, T_a = T_{opr} = 85°C

Typical measurement conditions: VCC = IOVCCn = 3.3 V, T_a = T_{opr} = 25°C

Item	Symbol	Typ.	Max.	Unit	Measurement conditions
IOVCC0 wait current	I _{IOVCC0ST}	10	—	nA	—
IOVCC1 wait current	I _{IOVCC1ST}	18	—		—
IOVCC0 and IOVCC1 wait current (total value)	—	—	1500		—

51.2.7 VCC Rise and Fall Gradient

Table 51.11 Rise and fall gradient characteristics

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
VCC rising gradient	SrVCC	0.02	—	20	ms/V	—
Allowable voltage change rising and falling gradient	dt/dVCC	2	—	20	ms/V	—

51.2.8 Internal Liner Regulator Characteristics

Table 51.12 Internal Liner Regulator Characteristics

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
LDO startup time	t_{LDO}	220	—	—	μs	Figure 51.2
LDO stabilization time	t_{LDOWT}	60	—	—	μs	Figure 51.2

Note: The device should not be consume large currents during the LDO stabilization time to ensure stable operation.

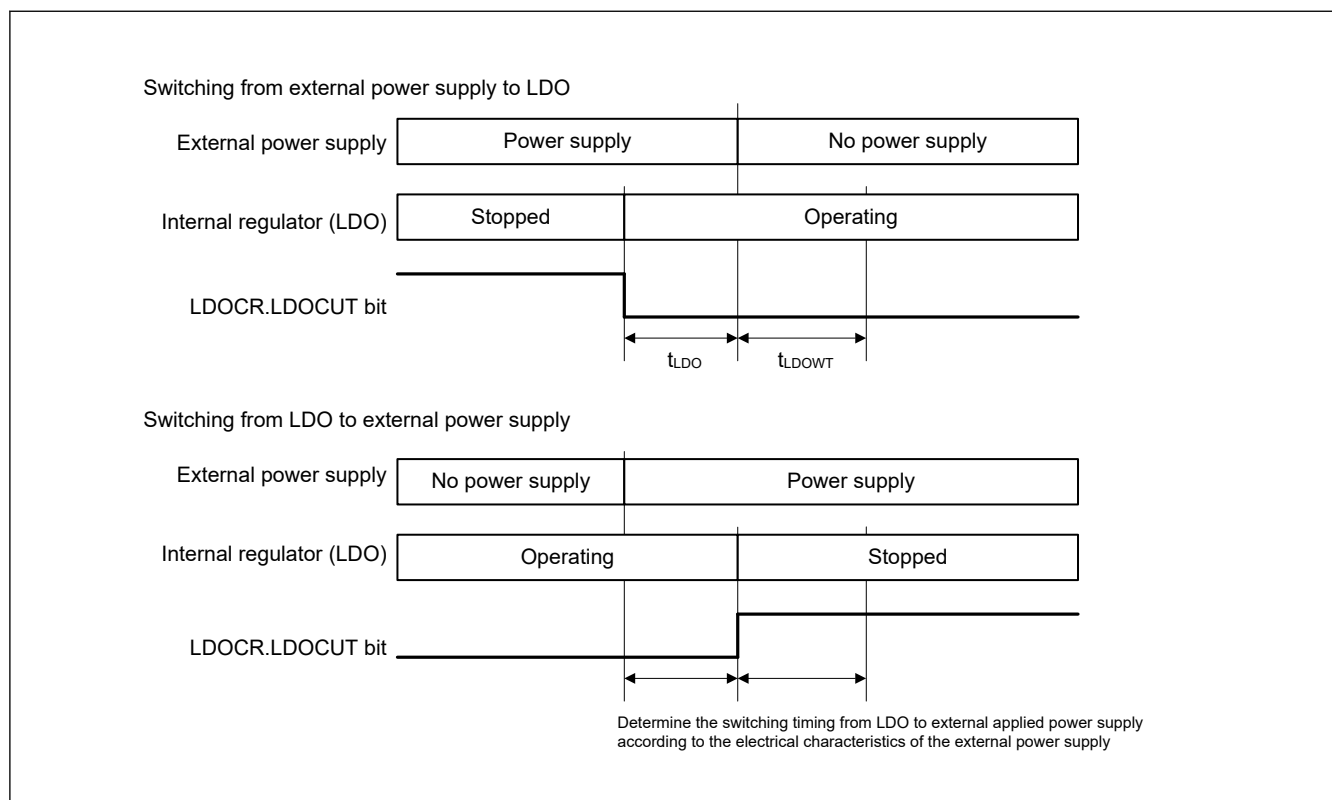


Figure 51.2 Switching timing between external power supply and LDO

51.3 AC Characteristics

51.3.1 Operating Frequency

Table 51.13 Operating frequency in each mode

Power control mode		Clock source	Symbol	Min.	Typ.	Max.	Unit
BOOST		System clock (ICLK)	f	—	—	64	MHz
		Peripheral module clock A (PCLKA)		—	—	64	
		Peripheral module clock B (PCLKB)		—	—	32	
NORMAL	High-speed	System clock (ICLK)		—	—	32	
		Peripheral module clock A (PCLKA)		—	—	32	
		Peripheral module clock B (PCLKB)		—	—	32	
	Low-speed	System clock (ICLK)		—	*1	2.3	
		Peripheral module clock A (PCLKA)		—	*1	2.3	
		Peripheral module clock B (PCLKB)		—	*1	2.3	
VBB		System clock (ICLK)		—	*2	37.6	kHz
		Peripheral module clock A (PCLKA)		—	*2	37.6	
		Peripheral module clock B (PCLKB)		—	*2	37.6	

Note: The minimum ICLK frequency is 1 MHz during programming or erasure of flash memory.

Note: Restriction on the clock frequency settings: ICLK/PCLKA ≥ PCLKB

Restriction on the clock frequency ratio (N: integer, and up to 64): ICLK/PCLKA:PCLKB = N:1

PCLKA and ICLK are at the same speed.

Note 1. The value is 2.0 MHz when the MOCO is selected as the clock source and the frequency is not being divided.

Note 2. The value is 32.768 kHz when the sub-clock oscillator is selected as the clock source and the frequency is not being divided.

51.3.2 Clock Timing

Table 51.14 Clock timing except for sub-clock oscillator (1 of 2)

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
EXTAL external clock input cycle time	t _{EXcyc}	39	—	—	ns	Figure 51.3
EXTAL external clock input high-level pulse width	t _{EXH}	15	—	—	ns	
EXTAL external clock input low-level pulse width	t _{EXL}	15	—	—	ns	
EXTAL external clock input rise time	t _{EXr}	—	—	4.5	ns	
EXTAL external clock input fall time	t _{EXf}	—	—	4.5	ns	
Main clock oscillator frequency	f _{MAIN}	8	—	32	MHz	—
Main clock oscillation stabilization wait time (crystal)*1	t _{MAINOSCWT}	—	—	—	ms	Figure 51.4
LOCO clock oscillation frequency	f _{LOCO}	27.8	32.7	37.6	kHz	—
LOCO clock oscillation stabilization wait time	t _{LOCOWT}	—	—	130	μs	Figure 51.5
IWDT-dedicated clock oscillation frequency	f _{IWDTLOCO}	13.9	16.35	18.8	kHz	—
MOCO clock oscillation frequency	f _{MOCO}	1.4	2	2.3	MHz	—
MOCO clock oscillation stabilization wait time	t _{MOCOWT}	—	—	16	μs	—

Table 51.14 Clock timing except for sub-clock oscillator (2 of 2)

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
HOCO clock oscillation frequency ^{*3}	FLL correction function is disabled.	f _{HOCO24}	23.64	24	24.36	MHz	0 ≤ T _a ≤ +85°C
		f _{HOCO32}	31.52	32	32.48		
		f _{HOCO48}	47.28	48	48.72		
		f _{HOCO64}	63.04	64	64.96		
		f _{HOCO24}	23.64	24	24.36		-40 ≤ T _a ≤ 0°C
		f _{HOCO32}	31.52	32	32.48		
		f _{HOCO48}	47.28	48	48.72		
		f _{HOCO64}	63.04	64	64.96		
	FLL correction function is enabled.	f _{HOCO24}	23.88	24	24.12		-40 ≤ T _a ≤ +85°C
		f _{HOCO32}	31.84	32	32.16		
		f _{HOCO48}	47.76	48	48.24		
		f _{HOCO64}	63.68	64	64.32		
HOCO clock oscillation stabilization wait time ^{*2}		t _{HOCOWT}	—	—	320	μs	—
FLL correction function stabilization wait time		f _{FLLWT}	—	—	1800	μs	—

Note 1. For setting up the main clock oscillator, we recommend consulting the oscillator manufacturer regarding the results of oscillation evaluation and use the results for the oscillation stabilization time. The value of the MOSCWTCR register should correspond to at least that value.

After changing the setting in the MOSCCR.MOSTP bit to start main clock operation, read the OSCSF.MOSCSF flag to confirm that it is 1, and then start using the main clock oscillator.

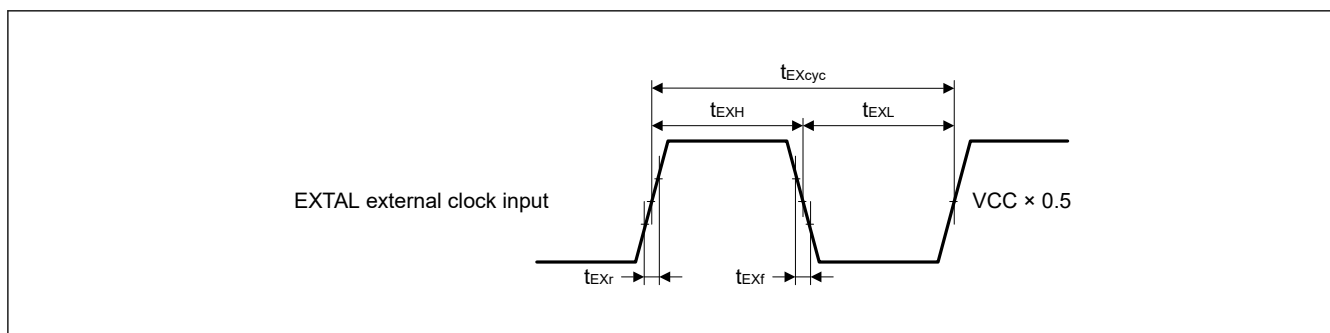
Note 2. This is the time period between when HOCOCR.HCSTP is changed to 0 and when OSCSF.HOCOSF is changed to 1.

Note 3. The guaranteed values stated for this item apply to products in packages. If you are using WLBGA samples, note that the characteristics deteriorate once the device has been mounted on your system due to fluctuations in stress.

Table 51.15 Clock timing for sub-clock oscillator

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Sub-clock frequency	f _{SUB}	—	32.768	—	kHz	—
Sub-clock oscillation stabilization wait time	t _{SUBOSCWT}	—	—	— ^{*1}	s	Figure 51.6

Note 1. For setting up the sub-clock oscillator, we recommend consulting the oscillator manufacturer regarding the results of oscillation evaluation and use the results for the oscillation stabilization time. After changing the setting in the SOSCCR.SOSTP flag to start sub-clock operation, only start using the sub-clock oscillator after the sub-clock oscillation stabilization time elapses with an adequate margin. We recommend using two times the value of the results of oscillation evaluation by the oscillator manufacturer.

**Figure 51.3 EXTAL external clock input timing**

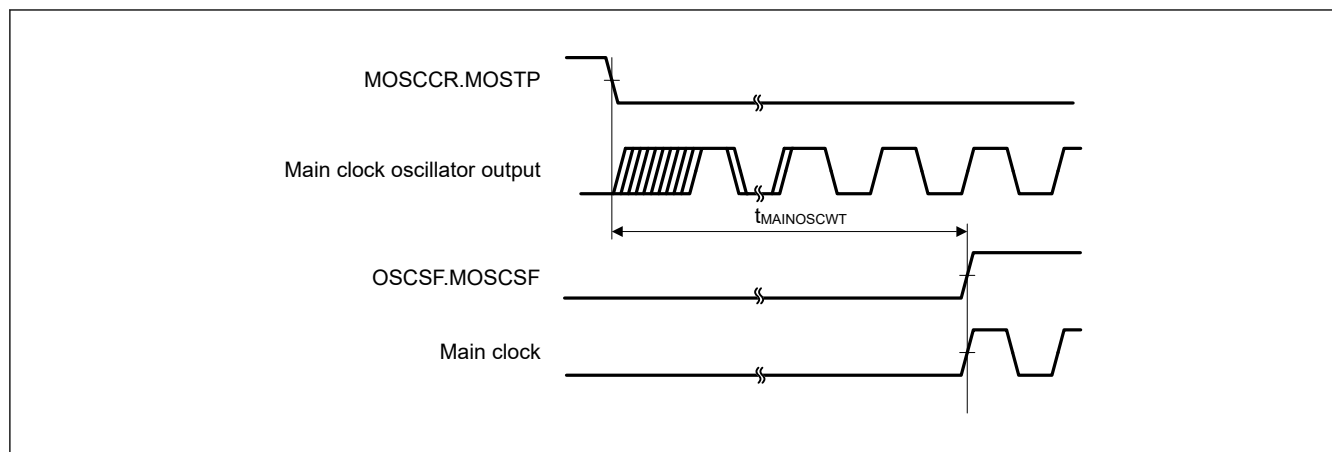


Figure 51.4 Main clock oscillation start timing

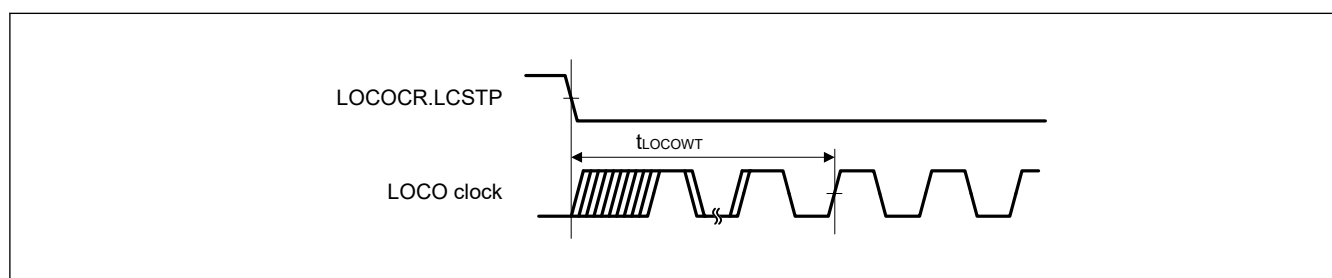


Figure 51.5 LOCO clock oscillation start timing

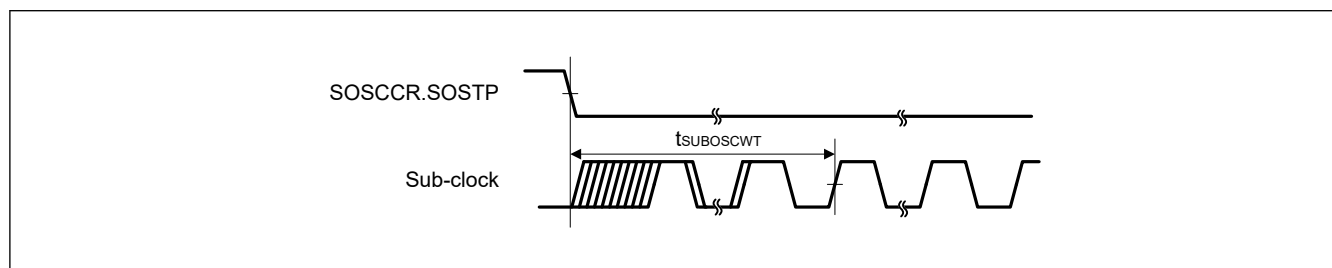


Figure 51.6 Sub-clock oscillation start timing

51.3.3 Reset Timing

Table 51.16 Reset timing

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
RES# pulse width	Power-on (in the normal start-up mode)	t_{RESWP}	44	—	—	ms	Figure 51.7
	Deep software standby mode	t_{RESWD}	7.7	—	—	ms	Figure 51.8
	Software standby mode	t_{RESWS}	1.2	—	—	ms	
	ALLPWON	Operation in boost mode	t_{RESW}	0.15	—	ms	
		Operation in normal mode	t_{RESW}	0.14	—	ms	
		Operation in low leakage current mode	t_{RESW}	0.62	—	ms	
		Transition between boost mode and normal mode	t_{RESW}	0.99	—	ms	
		Transition between normal mode and low leakage current mode	t_{RESW}	0.84	—	ms	
	EXFPWON	Operation in normal mode	t_{RESW}	0.46	—	ms	
		Operation in low leakage current mode	t_{RESW}	0.58	—	ms	
		Transition between normal mode and low leakage current mode	t_{RESW}	0.87	—	ms	
	MINPWON	Operation in normal mode	t_{RESW}	0.46	—	ms	
		Operation in low leakage current mode	t_{RESW}	0.58	—	ms	
		Transition between normal mode and low leakage current mode	t_{RESW}	0.87	—	ms	
	Transition between ALLPWON and EXFPWON in normal mode		t_{RESW}	0.78	—	ms	
	Transition between EXFPWON and MINPWON in normal mode		t_{RESW}	0.44	—	ms	
	Transition between ALLPWON and MINPWON in normal mode		t_{RESW}	0.78	—	ms	
	Transition between ALLPWON and EXFPWON in low leakage current mode		t_{RESW}	0.80	—	ms	
	Transition between EXFPWON and MINPWON in low leakage current mode		t_{RESW}	1.04	—	ms	
	Transition between ALLPWON and MINPWON in low leakage current mode		t_{RESW}	1.01	—	ms	
Wait time after release from the RES# pin reset		t_{RESWT}	—	19	22	ms	Figure 51.7, Figure 51.8

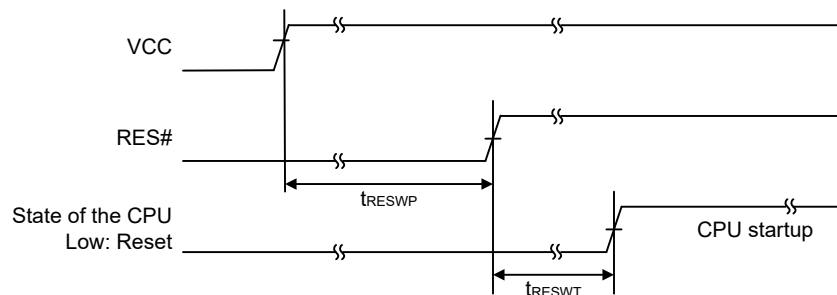


Figure 51.7 Timing of input through the reset pin when power is supplied

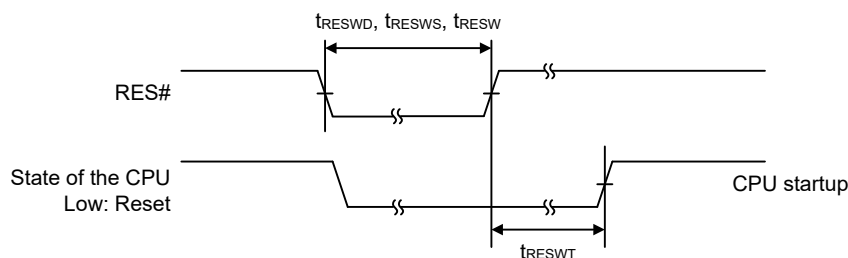


Figure 51.8 Reset input timing

51.3.4 Wakeup Timing

Table 51.17 Wakeup time from low power modes (standby modes) (1 of 7)

Item	Power control mode	System clock source	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Wakeup time from Software Standby mode (EXFPWON) to Operating mode (ALLPWON) *1*3	Normal	MOSC	t _{SBYMC}	—	—	3.1	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 0
		HOCO*2	t _{SBYHO}	—	—	0.9		
		MOCO	t _{SBYMO}	—	—	0.8		Figure 51.9 The division ratio of all oscillators is 1. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 0
		SOSC	t _{SBYSC}	—	—	3.0		
		LOCO	t _{SBYLO}	—	—	3.5		
	BOOST	MOSC	t _{SBYMC}	—	—	3.0	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 0
		HOCO*2	t _{SBYHO}	—	—	0.9		
		MOCO	t _{SBYMO}	—	—	0.7		Figure 51.9 The division ratio of all oscillators is 1. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 0
	VBB	SOSC	t _{SBYSC}	—	—	3.0	ms	
		LOCO	t _{SBYLO}	—	—	3.5		
Wakeup time from Software Standby mode (EXFPWON) to Operating mode (EXFPWON) *1*3	Normal, High-speed	MOSC	t _{SBYMC}	—	—	2.7	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 0
		HOCO*2	t _{SBYHO}	—	—	0.6		
		MOCO	t _{SBYMO}	—	—	0.4		Figure 51.9 The division ratio of all oscillators is 1. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 0
	Normal, Low-speed	MOCO	t _{SBYMO}	—	—	0.05	ms	
		SOSC	t _{SBYSC}	—	—	0.4		
		LOCO	t _{SBYLO}	—	—	0.5		
	VBB	SOSC	t _{SBYSC}	—	—	0.4	ms	
		LOCO	t _{SBYLO}	—	—	0.5		

Table 51.17 Wakeup time from low power modes (standby modes) (2 of 7)

Item	Power control mode	System clock source	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Wakeup time from Software Standby mode (MINPWON) to Operating mode (MINPWON) *1*3	Normal, High-speed	MOSC	t _{SBYMC}	—	—	2.7	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 0
		MOCO	t _{SBYMO}	—	—	0.4		Figure 51.9
	Normal, Low-speed	MOCO	t _{SBYMO}	—	—	0.05	ms	Figure 51.9 The division ratio of all oscillators is 1. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 0
		SOSC	t _{SBYSC}	—	—	0.4		
		LOCO	t _{SBYLO}	—	—	0.5		
	VBB	SOSC	t _{SBYSC}	—	—	0.4	ms	
		LOCO	t _{SBYLO}	—	—	0.5	ms	
	Normal	MOSC	t _{SBYMC}	—	—	3.1	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 1, SSBYVBB = 0, SSBYACC = 0
		HOCO*2	t _{SBYHO}	—	—	0.9		Figure 51.9 The division ratio of all oscillators is 1. SSBYPWG = 1, SSBYVBB = 0, SSBYACC = 0
		MOCO	t _{SBYMO}	—	—	0.8		
		SOSC	t _{SBYSC}	—	—	3.0		
		LOCO	t _{SBYLO}	—	—	3.5		
Wakeup time from Software Standby mode (MINPWON) to Operating mode (ALLPWON) *1*3	BOOST	MOSC	t _{SBYMC}	—	—	3.0	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 1, SSBYVBB = 0, SSBYACC = 0
		HOCO*2	t _{SBYHO}	—	—	0.9		Figure 51.9 The division ratio of all oscillators is 1. SSBYPWG = 1, SSBYVBB = 0, SSBYACC = 0
		MOCO	t _{SBYMO}	—	—	0.7		
	VBB	SOSC	t _{SBYSC}	—	—	3.2	ms	
		LOCO	t _{SBYLO}	—	—	3.7		
	Normal, High-speed	MOSC	t _{SBYMC}	—	—	2.8	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 1, SSBYVBB = 0, SSBYACC = 0
		HOCO*2	t _{SBYHO}	—	—	0.6		Figure 51.9 The division ratio of all oscillators is 1. SSBYPWG = 1, SSBYVBB = 0, SSBYACC = 0
		MOCO	t _{SBYMO}	—	—	0.5		
Wakeup time from Software Standby mode (MINPWON) to Operating mode (EXFPWON) *1*3	Normal, Low-speed	MOCO	t _{SBYMO}	—	—	0.2	ms	
		SOSC	t _{SBYSC}	—	—	0.5		
		LOCO	t _{SBYLO}	—	—	0.6		
	VBB	SOSC	t _{SBYSC}	—	—	1.0	ms	
		LOCO	t _{SBYLO}	—	—	1.1		

Table 51.17 **Wakeup time from low power modes (standby modes) (3 of 7)**

Item	Power control mode	System clock source	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Wakeup time from Software Standby mode (MINPWON) to Operating mode (MINPWON) *1*3	Normal, High-speed	MOSC	t _{SBYMC}	—	—	2.7	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 1, SSBYVBB = 0, SSBYACC = 0
		MOCO	t _{SBYMO}	—	—	0.4		Figure 51.9
	Normal, Low-speed	MOCO	t _{SBYMO}	—	—	0.05	ms	The division ratio of all oscillators is 1. SSBYPWG = 1, SSBYVBB = 0, SSBYACC = 0
		SOSC	t _{SBYSC}	—	—	0.4		
		LOCO	t _{SBYLO}	—	—	0.5		
	VBB	SOSC	t _{SBYSC}	—	—	0.4	ms	
		LOCO	t _{SBYLO}	—	—	0.5		
Wakeup time from Software Standby mode (VBB MINPWON) to Operating mode (ALLPWON) *1*3	Normal	MOSC	t _{SBYMC}	—	—	3.2	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 0
		HOCO*2	t _{SBYHO}	—	—	1.0		
		MOCO	t _{SBYMO}	—	—	0.8		
		SOSC	t _{SBYSC}	—	—	3.1		Figure 51.9 The division ratio of all oscillators is 1. Minimum current condition: SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 0
		LOCO	t _{SBYLO}	—	—	3.6		
	BOOST	MOSC	t _{SBYMC}	—	—	3.0	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 0
		HOCO*2	t _{SBYHO}	—	—	0.9		
		MOCO	t _{SBYMO}	—	—	0.7		
	VBB	SOSC	t _{SBYSC}	—	—	3.2	ms	Figure 51.9 The division ratio of all oscillators is 1. Minimum current condition: SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 0
		LOCO	t _{SBYLO}	—	—	3.7		

Table 51.17 Wakeup time from low power modes (standby modes) (4 of 7)

Item	Power control mode	System clock source	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Wakeup time from Software Standby mode (VBB MINPWON) to Operating mode (EXFPWON) *1*3	Normal, High-speed	MOSC	t _{SBYMC}	—	—	2.9	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 0
		HOCO*2	t _{SBYHO}	—	—	0.7		
		MOCO	t _{SBYMO}	—	—	0.6		
	Normal, Low-speed	MOCO	t _{SBYMO}	—	—	0.7	ms	Figure 51.9 The division ratio of all oscillators is 1. Minimum current condition: SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 0
		SOSC	t _{SBYSC}	—	—	1.0		
		LOCO	t _{SBYLO}	—	—	1.1		
	VBB	SOSC	t _{SBYSC}	—	—	1.0	ms	
		LOCO	t _{SBYLO}	—	—	1.1		
Wakeup time from Software Standby mode (VBB MINPWON) to Operating mode (MINPWON) *1*3	Normal, High-speed	MOSC	t _{SBYMC}	—	—	2.8	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 0
		MOCO	t _{SBYMO}	—	—	0.5		
	Normal, Low-speed	MOCO	t _{SBYMO}	—	—	0.6	ms	Figure 51.9 The division ratio of all oscillators is 1. Minimum current condition: SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 0
		SOSC	t _{SBYSC}	—	—	0.9		
		LOCO	t _{SBYLO}	—	—	1.1		
	VBB	SOSC	t _{SBYSC}	—	—	0.4	ms	
		LOCO	t _{SBYLO}	—	—	0.5		

Table 51.17 Wakeup time from low power modes (standby modes) (5 of 7)

Item	Power control mode	System clock source	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Wakeup time in fast transition from Software Standby mode (EXFPWON) to Operating mode (ALLPWON) *1*3	Normal	MOSC	t _{SBYMC}	—	—	3.0	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 1
		HOCO*2	t _{SBYHO}	—	—	0.9		
		MOCO	t _{SBYMO}	—	—	0.7		Figure 51.9 The division ratio of all oscillators is 1. Minimum transition time condition: SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 1
		SOSC	t _{SBYSC}	—	—	3.0		
		LOCO	t _{SBYLO}	—	—	3.5		
	BOOST	MOSC	t _{SBYMC}	—	—	3.0	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 1
		HOCO*2	t _{SBYHO}	—	—	0.9		
		MOCO	t _{SBYMO}	—	—	0.7		Figure 51.9 The division ratio of all oscillators is 1. Minimum transition time condition: SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 1
		VBB	SOSC	t _{SBYSC}	—	—		
	LOCO		t _{SBYLO}	—	—	3.5		
Wakeup time in fast transition from Software Standby mode (EXFPWON) to Operating mode (EXFPWON) *1*3	Normal, High-speed	MOSC	t _{SBYMC}	—	—	2.4	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 1
		HOCO*2	t _{SBYHO}	—	—	0.3		
		MOCO	t _{SBYMO}	—	—	0.05		Figure 51.9 The division ratio of all oscillators is 1. Minimum transition time condition: SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 1
	Normal, Low-speed	MOCO	t _{SBYMO}	—	—	0.05	ms	
		SOSC	t _{SBYSC}	—	—	0.4		
		LOCO	t _{SBYLO}	—	—	0.5		
	VBB	SOSC	t _{SBYSC}	—	—	0.4	ms	
		LOCO	t _{SBYLO}	—	—	0.5		

Table 51.17 Wakeup time from low power modes (standby modes) (6 of 7)

Item	Power control mode	System clock source	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Wakeup time in fast transition from Software Standby mode (MINPWON) to Operating mode (MINPWON) *1*3	Normal, High-speed	MOSC	t _{SBYMC}	—	—	2.4	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 1
		MOCO	t _{SBYMO}	—	—	0.05		Figure 51.9
	Normal, Low-speed	MOCO	t _{SBYMO}	—	—	0.05	ms	The division ratio of all oscillators is 1. Minimum transition time condition: SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 1
		SOSC	t _{SBYSC}	—	—	0.4		
		LOCO	t _{SBYLO}	—	—	0.5		
	VBB	SOSC	t _{SBYSC}	—	—	0.4	ms	
		LOCO	t _{SBYLO}	—	—	0.5		
Wakeup time in fast transition from Software Standby mode (VBB MINWON) to Operating mode (ALLPWON) *1*3	Normal	MOSC	t _{SBYMC}	—	—	3.0	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 1
		HOCO*2	t _{SBYHO}	—	—	0.9		Figure 51.9 The division ratio of all oscillators is 1. Minimum transition time condition: SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 1
		MOCO	t _{SBYMO}	—	—	0.7		
		SOSC	t _{SBYSC}	—	—	3.0		
		LOCO	t _{SBYLO}	—	—	3.5		
	BOOST	MOSC	t _{SBYMC}	—	—	3.0	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 1
		HOCO*2	t _{SBYHO}	—	—	0.9		Figure 51.9 The division ratio of all oscillators is 1. Minimum transition time condition: SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 1
		MOCO	t _{SBYMO}	—	—	0.7		
	VBB	SOSC	t _{SBYSC}	—	—	3.0	ms	
		LOCO	t _{SBYLO}	—	—	3.5		

Table 51.17 Wakeup time from low power modes (standby modes) (7 of 7)

Item	Power control mode	System clock source	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Wakeup time in fast transition from Software Standby mode (VBB MINPWON) to Operating mode (EXFPWON) *1*3	Normal, High-speed	MOSC	t _{SBYMC}	—	—	2.6	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 1
		HOCO*2	t _{SBYHO}	—	—	0.4		
		MOCO	t _{SBYMO}	—	—	0.3		
	Normal, Low-speed	MOCO	t _{SBYMO}	—	—	0.3	ms	Figure 51.9 The division ratio of all oscillators is 1. Minimum transition time condition: SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 1
		SOSC	t _{SBYSC}	—	—	0.6		
		LOCO	t _{SBYLO}	—	—	0.7		
	VBB	SOSC	t _{SBYSC}	—	—	0.5	ms	
		LOCO	t _{SBYLO}	—	—	0.7		
Wakeup time in fast transition from Software Standby mode (VBB MINPWON) to Operating mode (MINPWON) *1*3	Normal, High-speed	MOSC	t _{SBYMC}	—	—	2.5	ms	Figure 51.9 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 1
		MOCO	t _{SBYMO}	—	—	0.2		
	Normal, Low-speed	MOCO	t _{SBYMO}	—	—	0.2	ms	Figure 51.9 The division ratio of all oscillators is 1. Minimum transition time condition: SSBYPWG = 1, SSBYVBB = 1, SSBYACC = 1
		SOSC	t _{SBYSC}	—	—	0.5		
		LOCO	t _{SBYLO}	—	—	0.6		
	VBB	SOSC	t _{SBYSC}	—	—	0.4	ms	
		LOCO	t _{SBYLO}	—	—	0.5		
	Wait time after cancellation of Software Standby mode			t _{SBYWT}	—	—	*4	ms
Wakeup time from Deep Software Standby mode (on normal start-up mode)			t _{DSBY}	—	—	6.8	ms	Figure 51.10
Wait time after cancellation of Deep Software Standby mode			t _{DSBYWT}	—	—	22.0	ms	

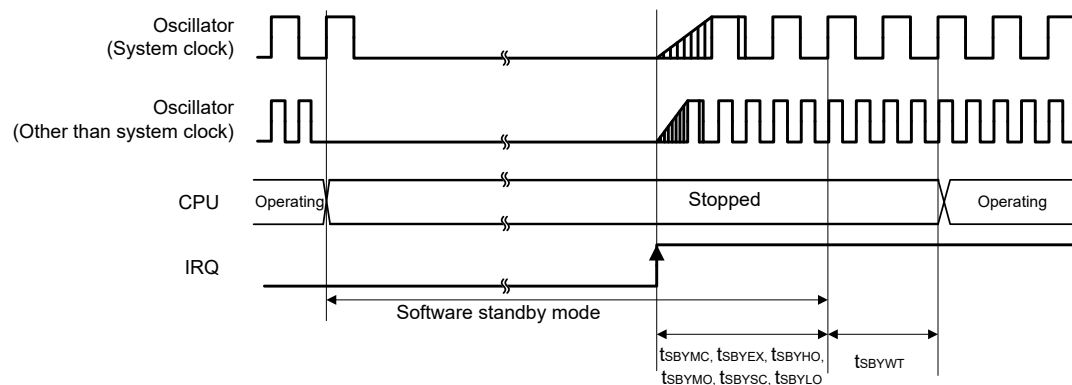
Note 1. The wakeup time is determined by the system clock source. When multiple oscillators are active, the wakeup time can be determined with the following equation:
 Total wakeup time = wakeup time for an oscillator as the system clock source + the longest oscillation stabilization time of any oscillators requiring longer stabilization times than the system clock source + 2 LOCO cycles (when LOCO is operating) + 3 SOSC cycles (when Subosc is oscillating and MSTPCRC.MSTPC0 = 0 (cancel the CAC module-stop state)).

Note 2. HOCO clock frequency = 32 MHz

Note 3. This value is a reference value because the shipment test is not performed.

Note 4. Wait time = 3 × PCLKB period + 14 × ICLK period

(1) When the oscillation stable waiting time for system clock is short



(2) When the oscillation stable waiting time for system clock is long

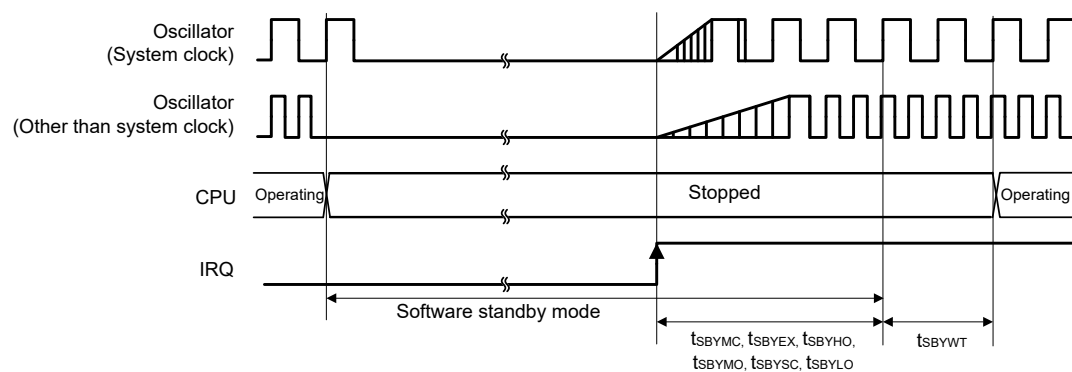


Figure 51.9 Software standby mode cancellation timing

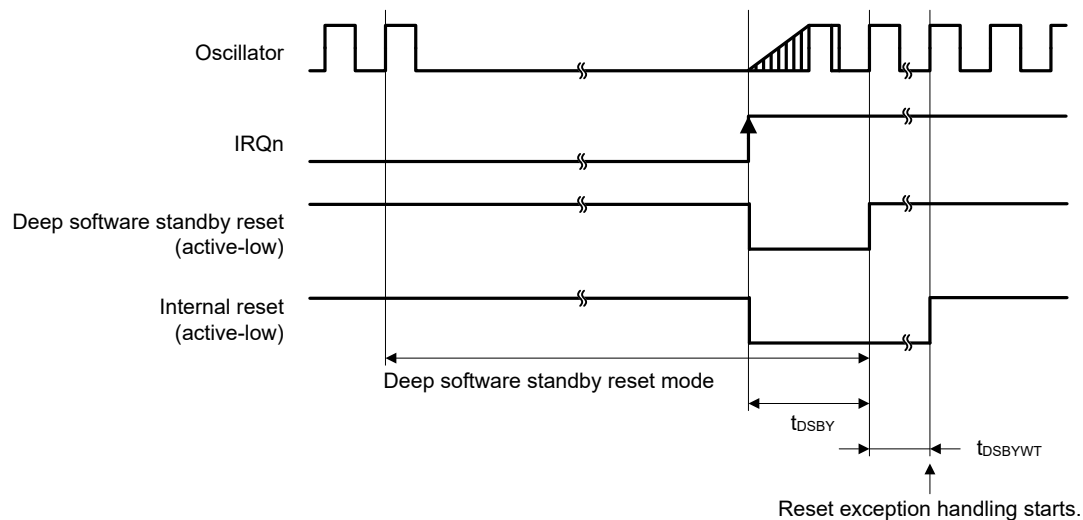


Figure 51.10 Deep software standby mode cancellation timing

Table 51.18 **Wakeup time from software standby mode to snooze mode**

Item	Power control mode	System clock source	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Wakeup time from Software Standby mode (EXFPWON) to Snooze mode (ALLPWON) *1*3	Normal	MOSC	t _{SBYMC}	—	—	3.1	ms	Figure 51.11 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 0
		HOCO*2	t _{SBYHO}	—	—	0.9		
		MOCO	t _{SBYMO}	—	—	0.8		Figure 51.11 The division ratio of all oscillators is 1. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 0
		SOSC	t _{SBYSC}	—	—	3.0		
		LOCO	t _{SBYLO}	—	—	3.4		
	BOOST	MOSC	t _{SBYMC}	—	—	3.0	ms	Figure 51.11 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 0
		HOCO*2	t _{SBYHO}	—	—	0.9		
		MOCO	t _{SBYMO}	—	—	0.7		Figure 51.11 The division ratio of all oscillators is 1. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 0
	VBB	SOSC	t _{SBYSC}	—	—	3.0	ms	
		LOCO	t _{SBYLO}	—	—	3.5		
Wakeup time from Software Standby mode (EXFPWON) to Snooze mode (EXFPWON) *1*3	Normal, High-speed	MOSC	t _{SBYMC}	—	—	2.7	ms	Figure 51.11 Each division ratio of ICLK/PCLKA and PCLKB is 1/8. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 0
		HOCO*2	t _{SBYHO}	—	—	0.6		
		MOCO	t _{SBYMO}	—	—	0.4		Figure 51.11 The division ratio of all oscillators is 1. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 0
	Normal, Low-speed	SOSC	t _{SBYSC}	—	—	0.4	ms	
		LOCO	t _{SBYLO}	—	—	0.5		
	VBB	SOSC	t _{SBYSC}	—	—	0.4	ms	
		LOCO	t _{SBYLO}	—	—	0.5		
	Wakeup time from Software Standby mode (MINPWON) to Snooze mode (MINPWON) *1*3	Normal, High-speed	MOSC	t _{SBYMC}	—	—	2.7	ms
MOCO			t _{SBYMO}	—	—	0.4	Figure 51.11 The division ratio of all oscillators is 1. SSBYPWG = 0, SSBYVBB = 0, SSBYACC = 0	
Normal, Low-speed		SOSC	t _{SBYSC}	—	—	0.4		ms
		LOCO	t _{SBYLO}	—	—	0.5		
VBB		SOSC	t _{SBYSC}	—	—	0.4		ms
		LOCO	t _{SBYLO}	—	—	0.5		

Note: When crystal frequency is 32 MHz (when Main Clock Oscillator Wait Control Register (MOSCWTCR) is 0x05).

Note 1. The wakeup time is determined by the system clock source. When multiple oscillators are active, the wakeup time can be determined with the following equation:

Total wakeup time = wakeup time for an oscillator as the system clock source + the longest oscillation stabilization time of any oscillators requiring longer stabilization times than the system clock source + 2 LOCO cycles (when LOCO is operating) + 3 SOSC cycles (when Subosc is oscillating and MSTPCRC.MSTPC0 = 0 (cancel the CAC module-stop state)).

Note 2. HOCO clock frequency = 32 MHz

Note 3. This value is a reference value because the shipment test is not performed.

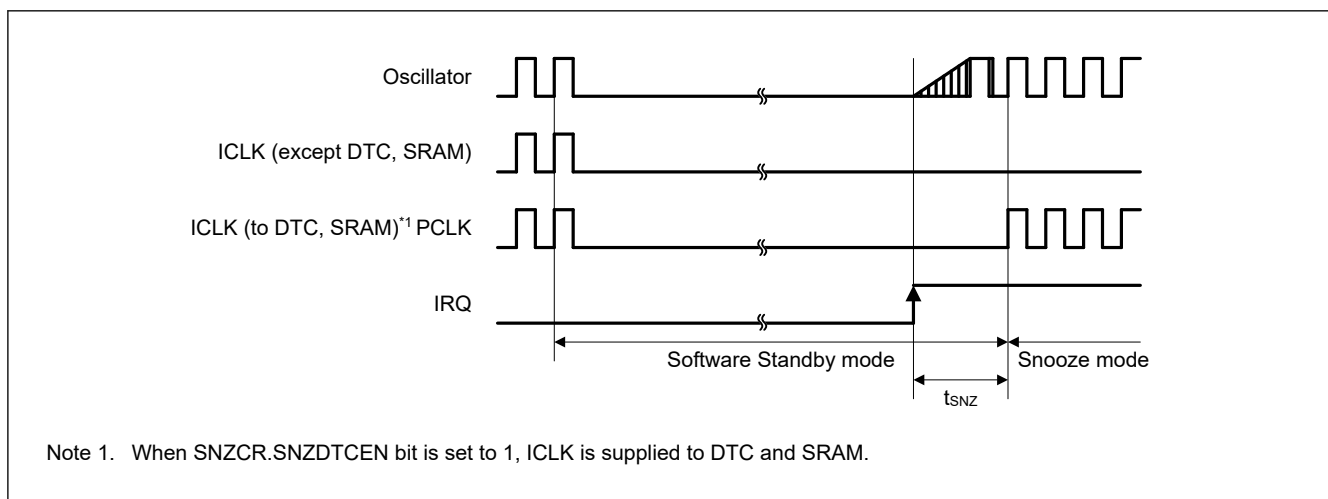


Figure 51.11 Wakeup time from software standby mode to snooze mode

51.3.5 Transition Time Between Operation Modes

Table 51.19 Transition time between each power supply modes (1 of 2)

Item	Power control mode	System clock source	Symbol	Min.	Typ.	Max.	Unit
Transition time from ALLPWON to EXFPWON	Normal (ALLPWON) → Normal High-speed (EXFPWON)	MOSC	t_{MDMC}	—	—	2.7	ms
		HOCO*1	t_{MDHO}	—	—	0.6	
		MOCO	t_{MDMO}	—	—	0.4	
	Normal (ALLPWON) → Normal Low-speed (EXFPWON)	MOCO	t_{MDMO}	—	—	0.5	ms
		SOSC	t_{MDSC}	—	—	1.7	
		LOCO	t_{MDLO}	—	—	2.1	
	VBB (ALLPWON) → VBB (EXFPWON)	SOSC	t_{MDSC}	—	—	1.7	ms
		LOCO	t_{MDLO}	—	—	2.1	
Transition time from EXFPWON to ALLPWON	Normal High-speed (EXFPWON) → Normal (ALLPWON)	MOSC	t_{MDMC}	—	—	3.0	ms
		HOCO*1	t_{MDHO}	—	—	0.9	
		MOCO	t_{MDMO}	—	—	0.7	
	Normal Low-speed (EXFPWON) → Normal (ALLPWON)	MOCO	t_{MDMO}	—	—	0.8	ms
		SOSC	t_{MDSC}	—	—	3.9	
		LOCO	t_{MDLO}	—	—	4.6	
	VBB (EXFPWON) → VBB (ALLPWON)	SOSC	t_{MDSC}	—	—	4.2	ms
		LOCO	t_{MDLO}	—	—	4.7	

Table 51.19 Transition time between each power supply modes (2 of 2)

Item	Power control mode	System clock source	Symbol	Min.	Typ.	Max.	Unit
Transition time from ALLPWON to MINPWON	Normal (ALLPWON) → Normal High-speed (MINPWON)	MOSC	t _{MDMC}	—	—	2.7	ms
		MOCO	t _{MDMO}	—	—	0.4	
	Normal (ALLPWON) → Normal Low-speed (MINPWON)	MOCO	t _{MDMO}	—	—	0.5	ms
		SOSC	t _{MDSC}	—	—	1.7	
		LOCO	t _{MDLO}	—	—	2.1	
	VBB (ALLPWON) → VBB (MINPWON)	SOSC	t _{MDSC}	—	—	1.4	ms
		LOCO	t _{MDLO}	—	—	1.8	
Transition time from MINPWON to ALLPWON	Normal High-speed (MINPWON) → Normal (ALLPWON)	MOSC	t _{MDMC}	—	—	3.0	ms
		MOCO	t _{MDMO}	—	—	0.7	
	Normal Low-speed (MINPWON) → Normal (ALLPWON)	MOCO	t _{MDMO}	—	—	0.8	ms
		SOSC	t _{MDSC}	—	—	3.9	
		LOCO	t _{MDLO}	—	—	4.6	
	VBB (MINPWON) → VBB (ALLPWON)	SOSC	t _{MDSC}	—	—	4.4	ms
		LOCO	t _{MDLO}	—	—	4.9	
Transition time from EXFPWON to MINPWON	Normal High-speed (EXFPWON) → Normal High-speed (MINPWON)	MOSC	t _{MDMC}	—	—	2.4	ms
		MOCO	t _{MDMO}	—	—	0.07	
	Normal High-speed (EXFPWON) → Normal Low-speed (MINPWON)	MOCO	t _{MDMO}	—	—	0.07	ms
		SOSC	t _{MDSC}	—	—	1.3	
		LOCO	t _{MDLO}	—	—	1.7	
	VBB (EXFPWON) → VBB (MINPWON)	SOSC	t _{MDSC}	—	—	1.4	ms
		LOCO	t _{MDLO}	—	—	1.8	
Transition time from MINPWON to EXFPWON	Normal High-speed (MINPWON) → Normal (EXFPWON)	MOSC	t _{MDMC}	—	—	2.5	ms
		MOCO	t _{MDMO}	—	—	0.2	
	Normal Low-speed (MINPWON) → Normal (EXFPWON)	MOCO	t _{MDMO}	—	—	0.2	ms
		SOSC	t _{MDSC}	—	—	1.4	
		LOCO	t _{MDLO}	—	—	1.8	
	VBB (MINPWON) → VBB (EXFPWON)	SOSC	t _{MDSC}	—	—	1.9	ms
		LOCO	t _{MDLO}	—	—	2.3	

Note: The transition time is determined by the system clock source. When multiple oscillators are active, the wakeup time can be determined with the following equation:

Total transition time = stabilization time for an oscillator as the system clock source + the longest oscillation stabilization time of any oscillators requiring longer stabilization times than the system clock source + 2 LOCO cycles (when LOCO is operating) + 3 SOSC cycles (when Subosc is oscillating and MSTPCRC.MSTPC0 = 0 (cancel the CAC module-stop state)).

Note: The division ratio of all oscillators is 1.

Note: This value is a reference value because the shipment test is not performed.

Note 1. HOCO clock frequency = 32 MHz

Table 51.20 Transition time between each power control modes

Item	Power control mode	System clock source	Symbol	Min.	Typ.	Max.	Unit
Transition between Normal and Boost	Normal (ALLPWON) → Boost (ALLPWON)	MOSC	t_{MDMC}	—	—	3.4	ms
		HOCO*1	t_{MDHO}	—	—	1.2	
		MOCO	t_{MDMO}	—	—	1.1	
	Boost (ALLPWON) → Normal (ALLPWON)	MOSC	t_{MDMC}	—	—	2.4	ms
		HOCO*1	t_{MDHO}	—	—	0.3	
		MOCO	t_{MDMO}	—	—	0.07	
Transition between Normal and VBB	Normal (ALLPWON) → VBB (ALLPWON)	SOSC	t_{MDSC}	—	—	1.8	ms
		LOCO	t_{MDLO}	—	—	2.2	
	Normal (EXFPWON) → VBB (EXFPWON)	SOSC	t_{MDSC}	—	—	1.8	ms
		LOCO	t_{MDLO}	—	—	2.2	
	Normal (MINPWON) → VBB (MINPWON)	SOSC	t_{MDSC}	—	—	1.4	ms
		LOCO	t_{MDLO}	—	—	1.8	
	VBB (ALLPWON) → Normal (ALLPWON)	SOSC	t_{MDSC}	—	—	1.7	ms
		LOCO	t_{MDLO}	—	—	2.0	
	VBB (EXFPWON) → Normal (EXFPWON)	SOSC	t_{MDSC}	—	—	1.7	ms
		LOCO	t_{MDLO}	—	—	2.1	
	VBB (MINPWON) → Normal (MINPWON)	SOSC	t_{MDSC}	—	—	1.8	ms
		LOCO	t_{MDLO}	—	—	2.2	
Transition between Boost and VBB	Boost (ALLPWON) → VBB (ALLPWON)	SOSC	t_{MDSC}	—	—	1.8	ms
		LOCO	t_{MDLO}	—	—	2.2	
	VBB (ALLPWON) → Boost (MINPWON)	SOSC	t_{MDSC}	—	—	2.6	ms
		LOCO	t_{MDLO}	—	—	3.0	
Transition from High-speed to Low-speed	ALLPWON		t_{HILOW}	—	—	0.003	ms
	EXFPWON		t_{HILOW}	—	—	0.5	
	MINPWON		t_{HILOW}	—	—	0.5	
Transition from Low-speed to High-speed	ALLPWON		t_{LOWHI}	—	—	0.003	ms
	EXFPWON		t_{LOWHI}	—	—	0.4	
	MINPWON		t_{LOWHI}	—	—	0.4	

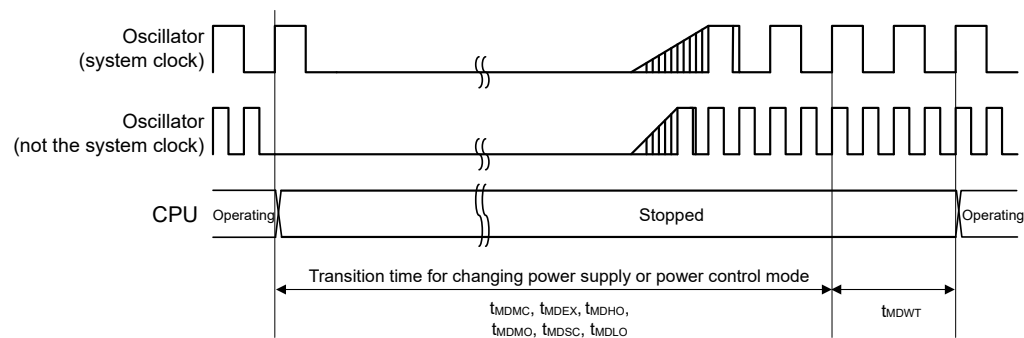
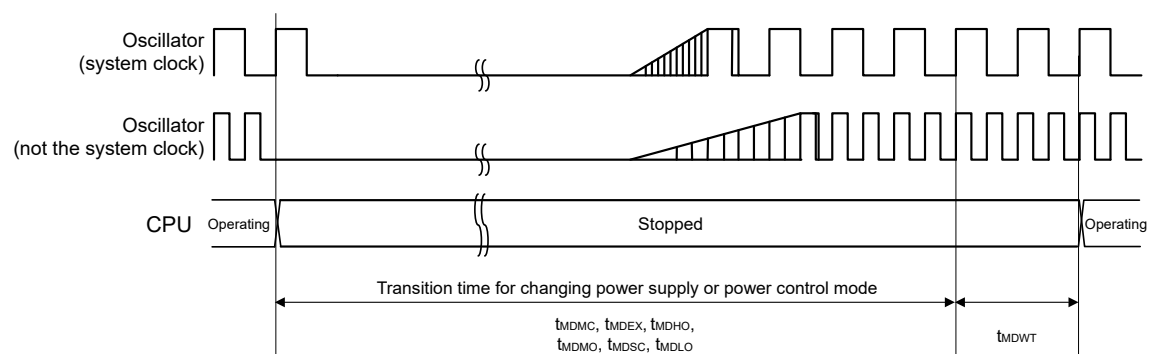
Note: The transition time is determined by the system clock source. When multiple oscillators are active, the wakeup time can be determined with the following equation:

Total transition time = stabilization time for an oscillator as the system clock source + the longest oscillation stabilization time of any oscillators requiring longer stabilization times than the system clock source + 2 LOCO cycles (when LOCO is operating) + 3 SOSC cycles (when Subosc is oscillating and MSTPCRC.MSTPC0 = 0 (cancel the CAC module-stop state)).

Note: The division ratio of all oscillators is 1.

Note: This value is a reference value because the shipment test is not performed.

Note 1. HOCO clock frequency = 32 MHz

When stabilization of the system clock oscillator is slower**When stabilization of an oscillator other than the system clock is slower**

Note: $t_{MDWT} = 3 \times \text{PCLKB period} + 14 \times \text{ICLK period}$

Figure 51.12 Transition timing between operation modes

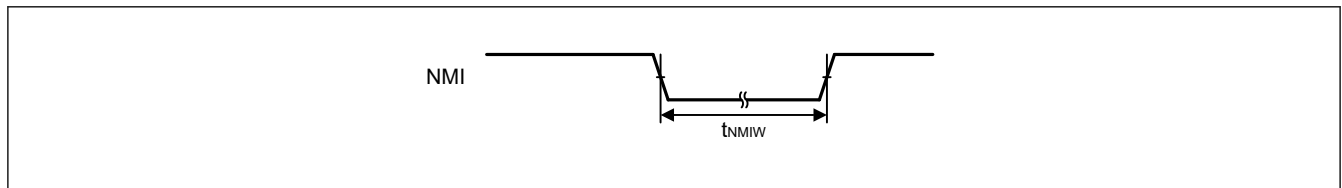
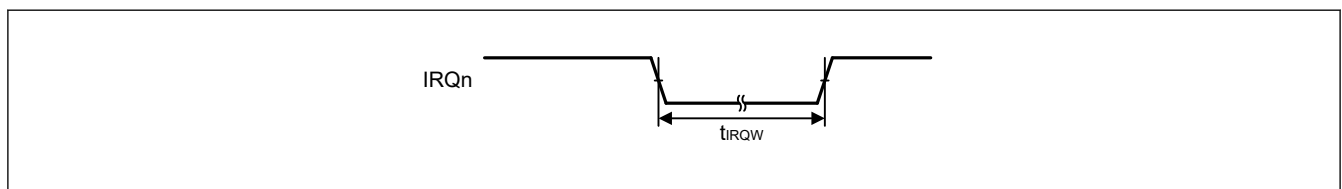
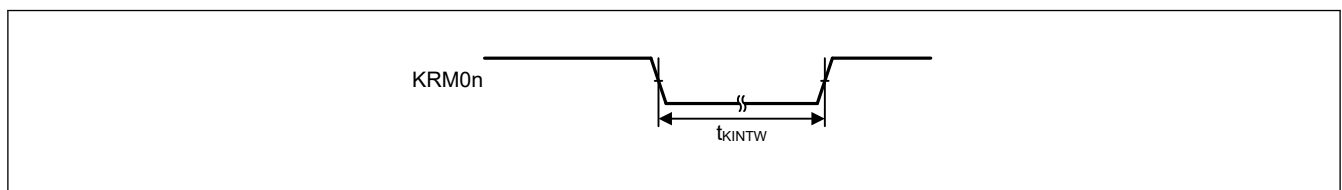
51.3.6 Interrupt Input Timing

Table 51.21 Interrupt input timing (1 of 2)

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
NMI pulse width	t_{NMIW}	6000	—	—	ns	Software Standby mode on VBB mode
		1000	—	—		Software Standby mode other than above
		300	—	—		Deep Software Standby mode
		4	—	—	t_{Pcyc}^{*1}	Other than above
IRQn pulse width	t_{IRQW}	6000	—	—	ns	Software Standby mode on VBB mode
		1000	—	—		Software Standby mode other than above
		300	—	—		Deep Software Standby mode
		4	—	—	t_{Pcyc}^{*1}	Other than above (IRQCRi.IRQMD[1:0] = 00b, 01b)
		5	—	—		Other than above (IRQCRi.IRQMD[1:0] = 10b)

Table 51.21 Interrupt input timing (2 of 2)

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
KINT pulse width	t_{KINTW}	6000	—	—	ns	Software Standby mode on VBB mode
		1000	—	—		Software Standby mode other than above
		4	—	—	t_{Pcyc}^{*1}	Other than above

Note 1. t_{Pcyc} : PCLKB cycle**Figure 51.13** NMI interrupt input timing**Figure 51.14** IRQn interrupt input timing**Figure 51.15** KINT interrupt input timing

51.3.7 Trigger Timing of I/O port, POE, GPT, AGT, and ADC14

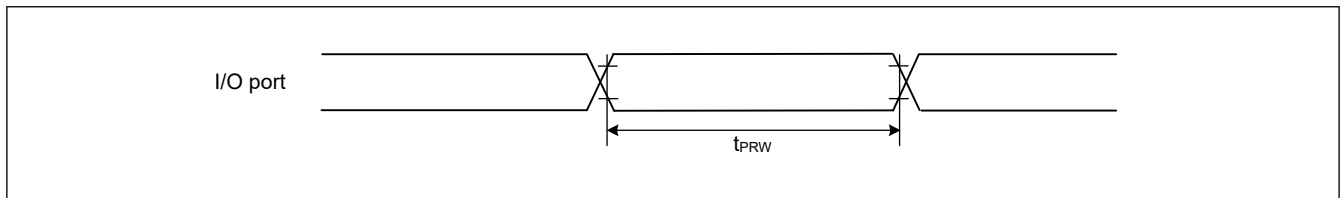
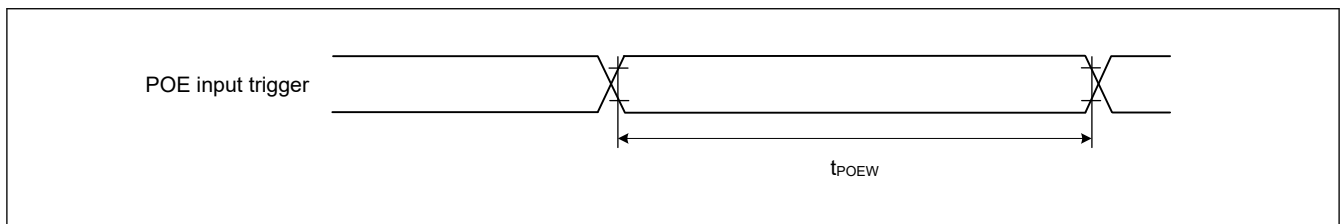
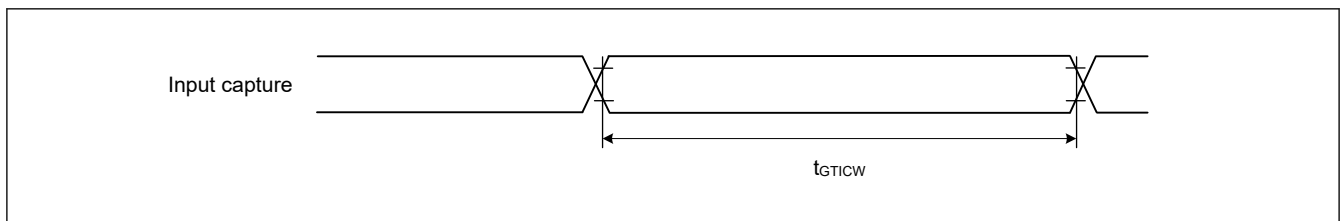
Table 51.22 Trigger timing of I/O port, POE, GPT, AGT, and ADC14 (1 of 2)

Item		Symbol	Min.	Typ.	Max.	Unit ^{*1}	Measurement conditions
I/O port	Input data pulse width	t_{PRW}	2.5	—	—	t_{Pcyc}	Figure 51.16
	ELC event pulse input width		4	—	—		
POE	POE input trigger pulse width	t_{POEW}	1.5	—	—	t_{Pcyc}	Figure 51.17
GPT	Input capture pulse width	t_{GTICW}	1.5	—	—	t_{Pcyc}	Figure 51.18
			2.5	—	—	t_{Pcyc}	

Table 51.22 Trigger timing of I/O port, POE, GPT, AGT, and ADC14 (2 of 2)

Item		Symbol	Min.	Typ.	Max.	Unit ^{*1}	Measurement conditions
AGT/AGTW	AGTIO _n /AGTWIO _n input cycle	t_{ACYC}	4	—	—	t_{Pcyc}	Figure 51.19, AGTMR1.TEDGPL = 0 AGTMR1.TMOD[2:0] = 010b
			9	—	—	t_{Pcyc}	Figure 51.19, AGTMR1.TEDGPL = 1 AGTMR1.TMOD[2:0] = 010b
	AGTIO _n /AGTWIO _n input high-level width, low-level width	t_{ACKWH} , t_{ACKWL}	1	—	—	t_{Pcyc}	Figure 51.19, AGTMR1.TEDGPL = 0 AGTMR1.TMOD[2:0] = 010b
			4	—	—	t_{Pcyc}	Figure 51.19, AGTMR1.TEDGPL = 1 AGTMR1.TMOD[2:0] = 010b
	AGTEEn/AGTWEE _n input high-level width, low-level width	t_{ACKWH} , t_{ACKWL}	—	1	—	t_{ACYC}	Figure 51.19, AGTMR1.TEDGPL = 0 AGTMR1.TMOD[2:0] = 010b
			4	—	—	t_{Pcyc}	Figure 51.19, AGTMR1.TEDGPL = 1 AGTMR1.TMOD[2:0] = 010b
ADC14	14-bit A/D converter trigger input pulse width	t_{TRGW}	1.5	—	—	t_{Pcyc}	Figure 51.20

Note: n = 0, 1

Note 1. t_{Pcyc} : This indicates a clock cycle of PCLKA for GPT port, and PCLKB for I/O port, POE, AGT, and ADC14 ports.**Figure 51.16 I/O port input data pulse width****Figure 51.17 POE input trigger pulse width****Figure 51.18 GPT input capture pulse width**

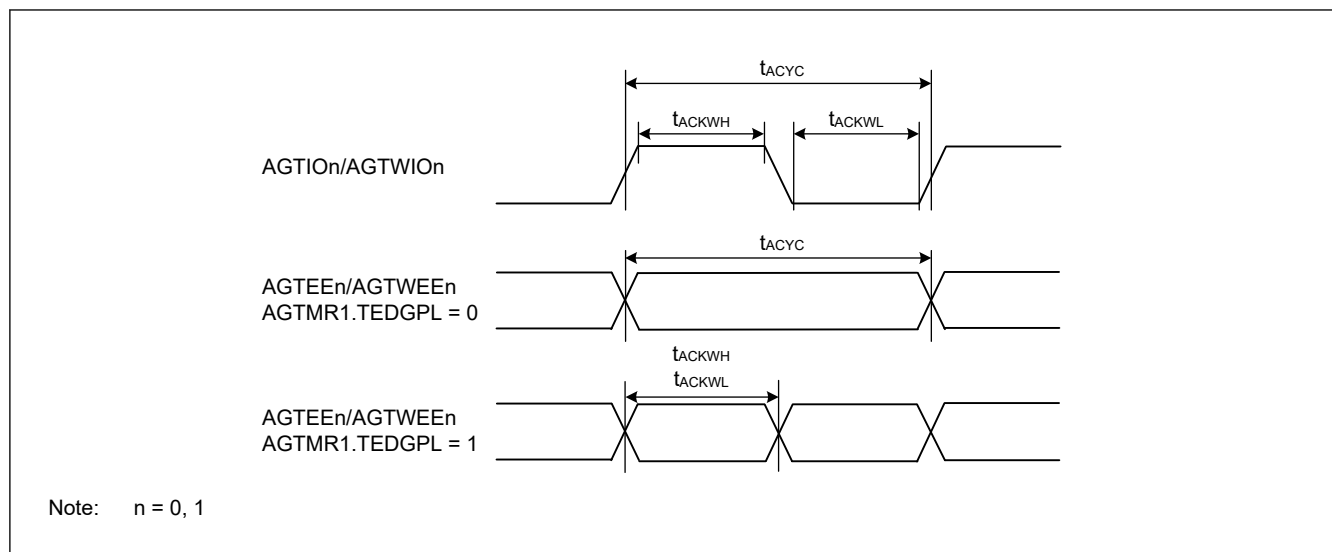


Figure 51.19 AGT/AGTW input timing

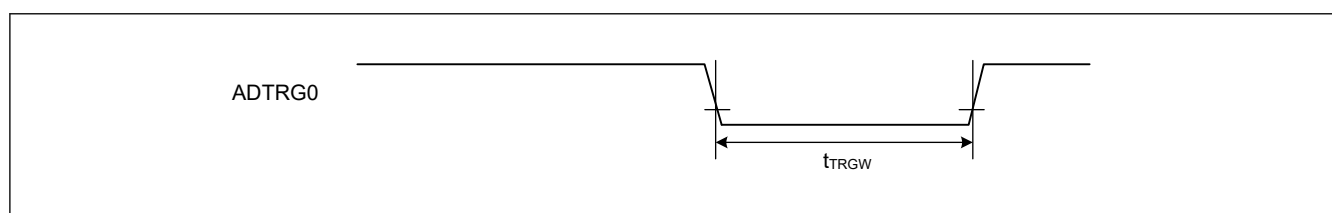


Figure 51.20 ADC14 trigger input timing

51.3.8 CAC Timing

Table 51.23 CAC timing

Item			Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
CAC	CACREF input pulse width	$t_{P_{cyc}}^{*1} \leq t_{cac}^{*2}$	t_{CACREF}	$4.5t_{cac} + 3t_{P_{cyc}}$	—	—	ns	—
		$t_{P_{cyc}}^{*1} > t_{cac}^{*2}$		$5t_{cac} + 6.5t_{P_{cyc}}$	—	—	ns	

Note 1. $t_{P_{cyc}}$: PCLKB clock cycle

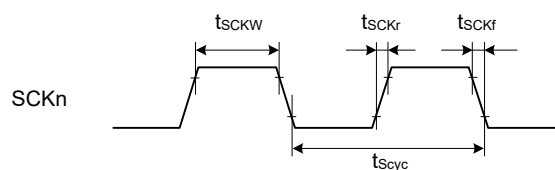
Note 2. t_{cac} : CAC count clock source cycle

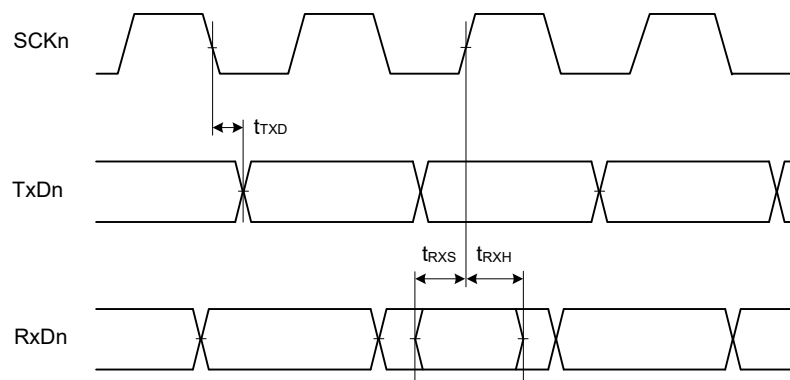
51.3.9 SCI Timing

Table 51.24 SCI timing (1)

Condition: High drive output is selected in the drive capability control bits in PmnPFS register.

Item			Symbol	Min.	Max.	Unit*1	Measurement conditions
SCI	Frequency (SCI0, SCI1)	BOOST	pclkfmax	—	64	MHz	—
		NORMAL		—	32		
		Frequency (other than SCI0 or SCI1)		—	32		
	Input clock cycle	Asynchronous	t _{Scyc}	4	—	t _{Pcyc}	Figure 51.21
		Clock synchronous		6	—		
	Input clock pulse width		t _{SCKW}	0.4	0.6	t _{Scyc}	
	Input clock rise time		t _{SCKr}	—	1 × t _{Pcyc}	ns	
	Input clock fall time		t _{SCKf}	—	1 × t _{Pcyc}	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		t _{SCKr}	—	1 × t _{Pcyc}	ns	
	Output clock fall time		t _{SCKf}	—	1 × t _{Pcyc}	ns	
	Transmit data delay	Master	t _{TXD}	—	40	ns	Figure 51.22
		Slave		—	55	ns	
	Receive data setup time	Master	t _{RXS}	45	—	ns	
		Slave		27	—	ns	
	Receive data hold time	Master	t _{RXH}	5	—	ns	
		Slave		40	—	ns	

Note 1. $t_{P\text{cyc}}$: This indicates a clock cycle of PCLKA for SCI0 and SCI1 ports, and PCLKB for the ports from SCI2 to SCI5 and SCI9 port.Note: $n = 0$ to 5, 9**Figure 51.21** SCK clock input timing



Note: $n = 0$ to $5, 9$

Figure 51.22 SCI input/output timing in clock synchronous mode

Table 51.25 SCI timing (2)

Condition: High drive output is selected in the drive capability control bits in PmnPFS register.

Item			Symbol	Min.	Max.	Unit*1	Measurement conditions	
Simple SPI	Frequency (SCI0, SCI1)	BOOST	pclkfmax	—	64	MHz	—	
		NORMAL		—	32			
		Frequency (other than SCI0 or SCI1)		—	32			
	SCK clock cycle	Master	tSPCyc	4	65536	tPcyc	Figure 51.23	
		Slave		6	—			
	SCK clock high-level pulse width		tSPCKWH	0.4	0.6	tSPCyc		
	SCK clock low-level pulse width		tSPCKWL	0.4	0.6	tSPCyc		
	SCK clock rise and fall time		tSPCKr, tSPCKf	—	1 × tPcyc	ns		
	Data input setup time	Master	tSU	45	—	ns		Figure 51.24 to Figure 51.27
		Slave		27	—			
	Data input hold time	Master	tH	33.3	—	ns		Figure 51.26 Figure 51.27
		Slave		40	—			
	SS input setup time		tLEAD	1	—	tSPCyc		
	SS input hold time		tLAG	1	—	tSPCyc		
	Data output delay	Master	tOD	—	40	ns		
		Slave		—	65			
	Data output hold time	Master	tOH	-10	—	ns		
		Slave		-10	—			
	Data rise and fall time		tDr, tDf	—	1 × tPcyc	ns		
	Slave access time	BOOST	tSA	—	8	tPcyc	Figure 51.26 Figure 51.27	
		NORMAL		—	6			
	Slave output release time	BOOST	tREL	—	8	tPcyc		
		NORMAL		—	6			

Note 1. t_{Pcyc} : This indicates a clock cycle of PCLKA for SCI0 and SCI1 ports, and PCLKB for the ports from SCI2 to SCI5 and SCI9 port.

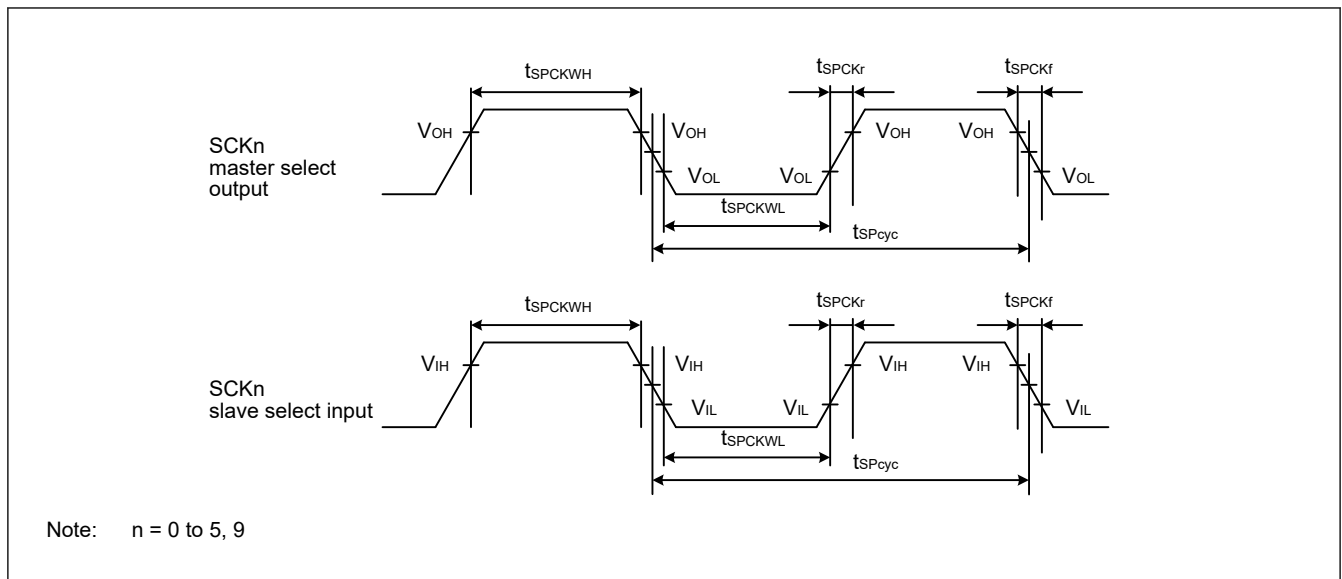


Figure 51.23 SCK clock input/output timing (simple SPI mode)

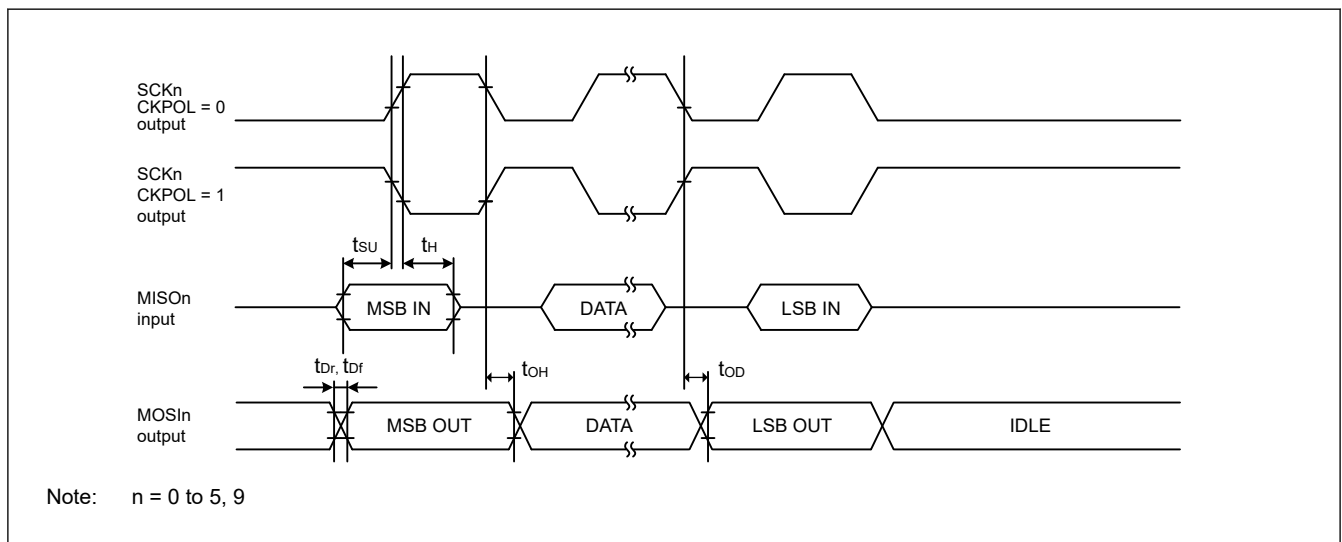


Figure 51.24 SCK input/output timing (simple SPI mode) (master, SPMR.CKPH = 1)

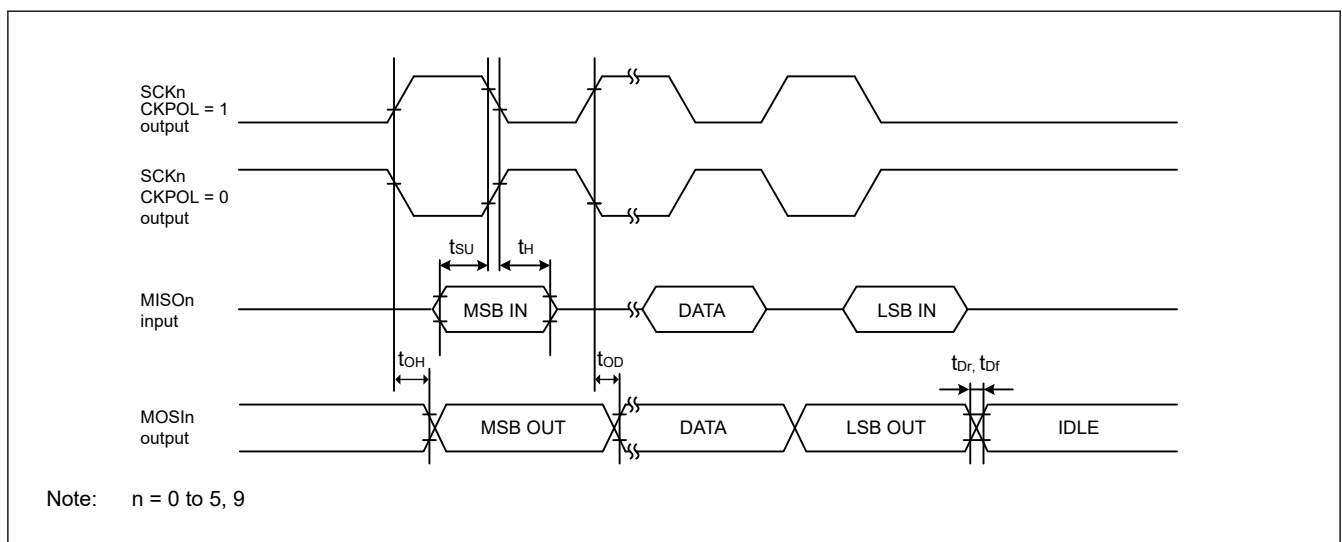


Figure 51.25 SCK input/output timing (simple SPI mode) (master, SPMR.CKPH = 0)

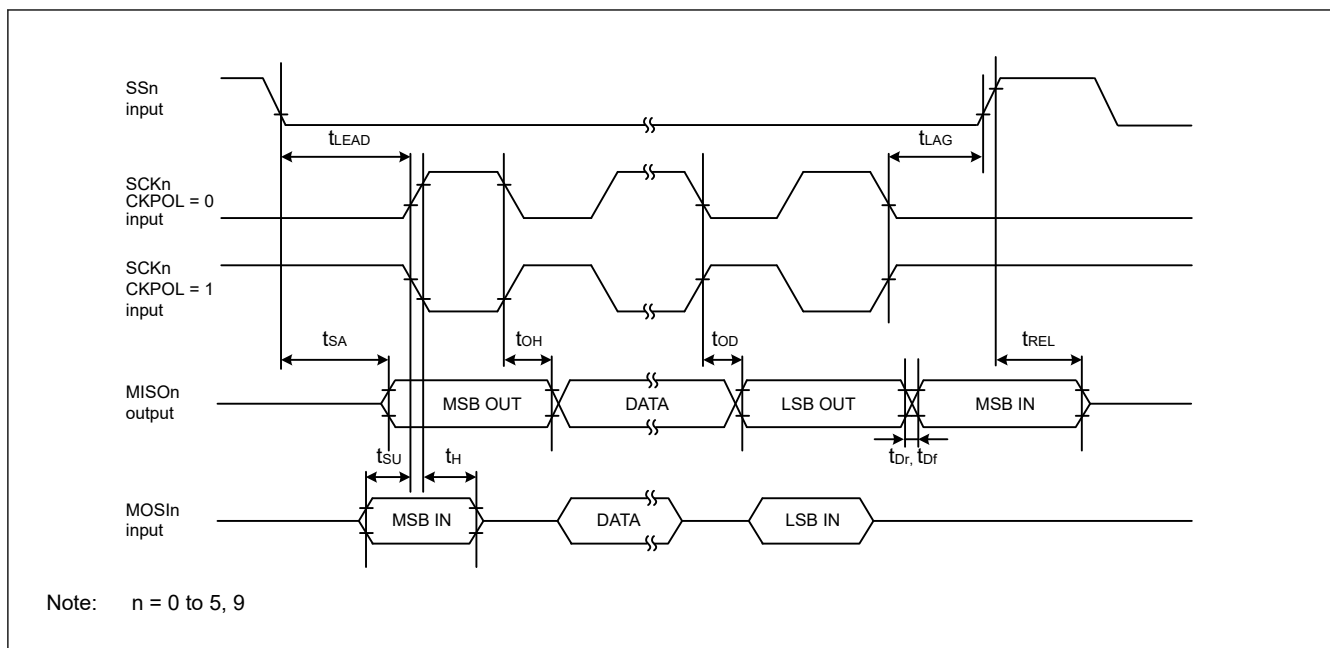


Figure 51.26 SCK input/output timing (simple SPI mode) (slave, SPMR.CKPH = 1)

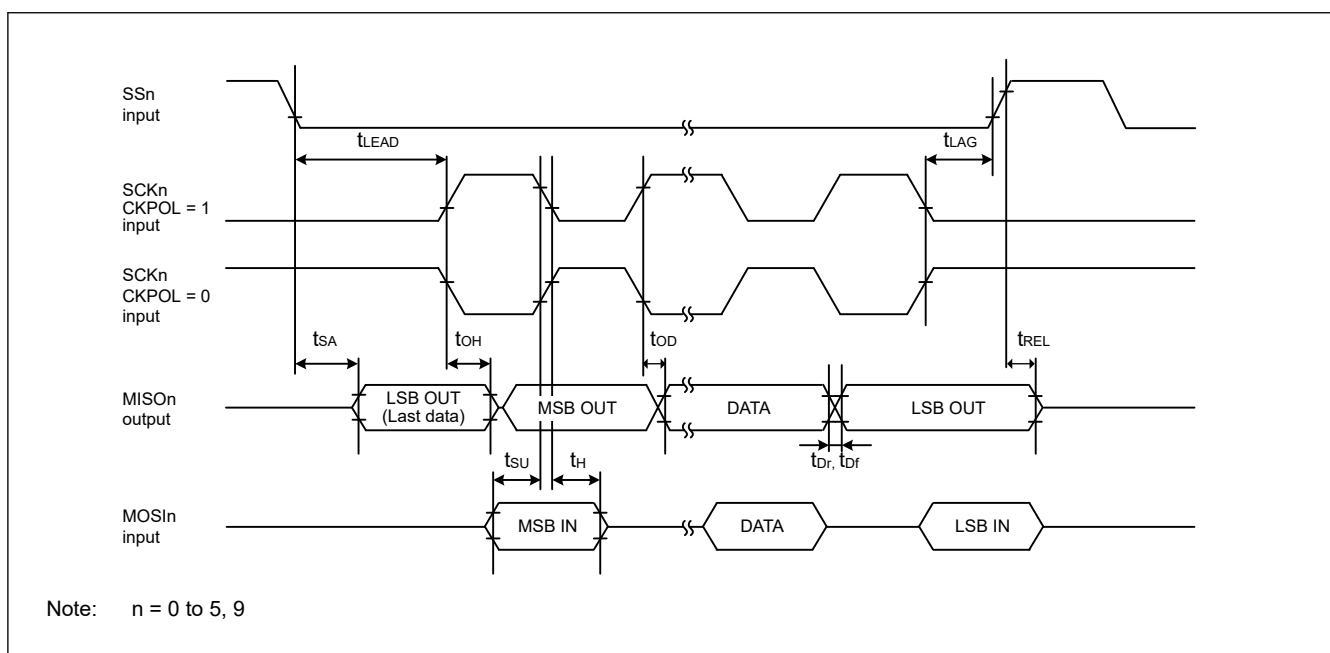
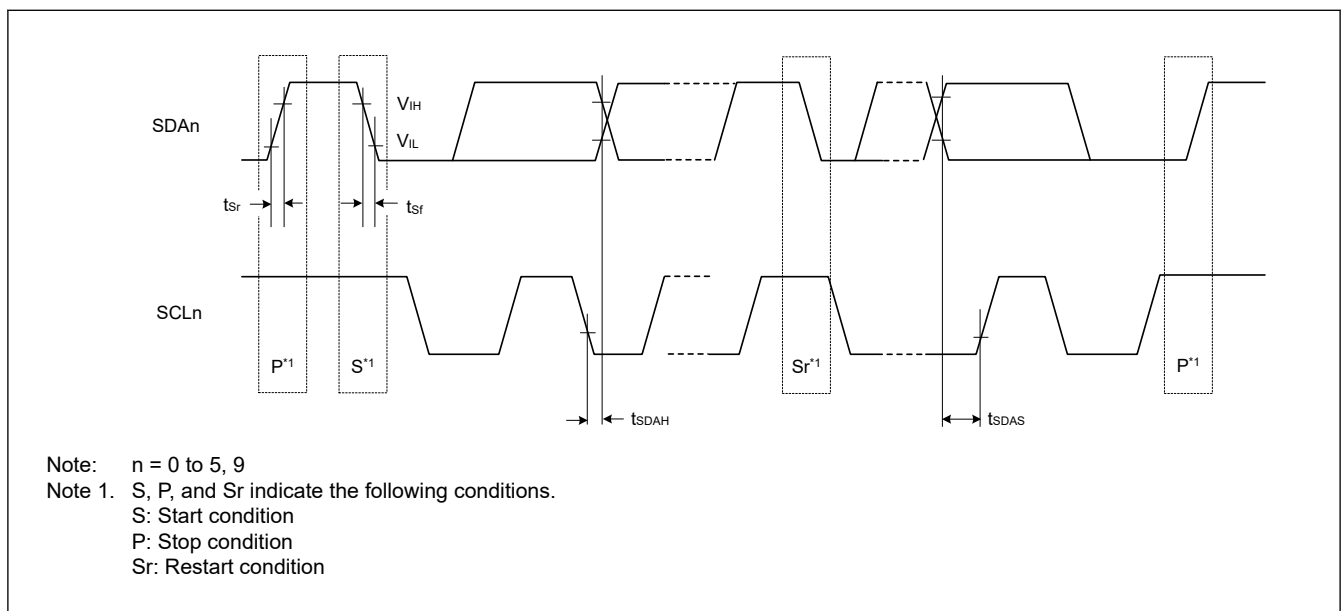


Figure 51.27 SCK input/output timing (simple SPI mode) (slave, SPMR.CKPH = 0)

Table 51.26 SCI timing (3)

Condition: High drive output is selected in the drive capability control bits in PmnPFS register.

Item			Symbol	Min.	Max.	Unit*2	Measurement conditions
Simple IIC (Standard mode)	Frequency (SCI0, SCI1)	BOOST	pclkfmax	—	64	MHz	—
		NORMAL		—	32		
	Frequency (other than SCI0 or SCI1)			—	32		
	SDA input rise time		t_{Sr}	—	1000	ns	Figure 51.28
	SDA input fall time		t_{Sf}	—	300	ns	
	Data input setup time		t_{SDAS}	250	—	ns	Figure 51.28
	Data input hold time		t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load		C_b^{*1}	—	400	pF	
Simple IIC (Fast mode)	Frequency (SCI0, SCI1)	BOOST	pclkfmax	—	64	MHz	—
		NORMAL		—	32		—
	Frequency (other than SCI0 or SCI1)			—	32		—
	SCL, SDA input rise time		t_{Sr}	—	300	ns	Figure 51.28
	SCL, SDA input fall time		t_{Sf}	—	300	ns	
	Data input setup time		t_{SDAS}	100	—	ns	Figure 51.28
	Data input hold time		t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load		C_b^{*1}	—	400	pF	

Note 1. C_b indicates the total capacity of the bus line.Note 2. t_{Pcy} : This indicates a clock cycle of PCLKA for SCI0 and SCI1 ports, and PCLKB for the ports from SCI2 to SCI5 and SCI9 port.**Figure 51.28 SCK input/output timing (simple I²C mode)**

51.3.10 SPI Timing

Table 51.27 SPI timing (1 of 2)

Condition: High drive output is selected in the drive capability control bits in PmnPFS register.

Item			Symbol	Min.	Max.	Unit*1	Measurement conditions
Frequency		BOOST	pclkfmax	—	64	MHz	—
		NORMAL		—	32		
RSPCK clock cycle	Master	BOOST	tSPcyc	4	4096	tPcyc	Figure 51.29
		NORMAL		2	4096		
	Slave			6	4096		
RSPCK clock high-level pulse width	Master		tSPCKWH	(tSPcyc - tSPCKr - tSPCKf)/2 - 3	—	ns	
	Slave			3 × tpcyc	—		
RSPCK clock low-level pulse width	Master		tSPCKWL	(tSPcyc - tSPCKr - tSPCKf)/2 - 3	—	ns	
	Slave			3 × tpcyc	—		
RSPCK clock rise and fall time	Output		tSPCKr, tSPCKf	—	10	ns	Figure 51.29 IOVCCn ≥ 2.7V
	Input				1	μs	
Data input setup time	Master	BOOST	tSU	25	—	ns	Figure 51.30 to Figure 51.35 IOVCCn ≥ 2.7V
		NORMAL		15	—		
	Slave			10	—		
Data input hold time	Master		tHF	0	—	ns	Figure 51.30 to Figure 51.35 PCLKA division ratio is set to 1/2.
			tH	1	—	tPcyc	Figure 51.30 to Figure 51.35 PCLKA division ratio is set to a value other than 1/2.
	Slave			20	—	ns	Figure 51.30 to Figure 51.35
SSL setup time	Master		tLEAD	-30 + N × tSPcyc*2	—	ns	Figure 51.30 to Figure 51.35
	Slave			6 × tpcyc	—	ns	
SSL hold time	Master		tLAG	-30 + N × tSPcyc*3	—	ns	
	Slave			6 × tpcyc	—	ns	
Data output delay	Master		tOD	—	14	ns	Figure 51.30 to Figure 51.35 IOVCCn ≥ 2.7V
	Slave				50		
Data output hold time	Master		tOH	0	—	ns	Figure 51.30 to Figure 51.35
	Slave			0	—		
Successive transmission delay	Master		tTD	tSPcyc + 2 × tpcyc	8 × tSPcyc + 2 × tpcyc	ns	Figure 51.30 to Figure 51.35
	Slave			6 × tpcyc	—		

Table 51.27 SPI timing (2 of 2)

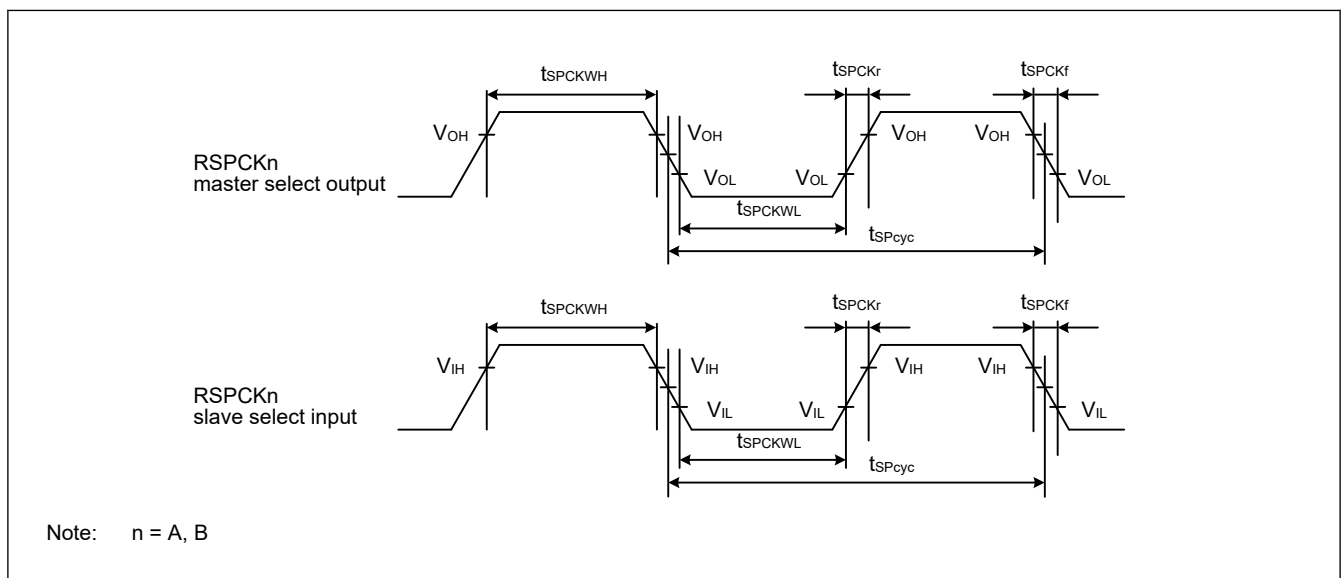
Condition: High drive output is selected in the drive capability control bits in PmnPFS register.

Item		Symbol	Min.	Max.	Unit ^{*1}	Measurement conditions
MOSI and MISO rise and fall time	Output	t_{Dr}, t_{Df}	—	10	ns	Figure 51.30 to Figure 51.35 IOVCCn $\geq$ 2.7V
	Input		—	1	μ s	Figure 51.30 to Figure 51.35
SSL rise and fall time	Output	t_{SSLr}, t_{SSLf}	—	10	ns	Figure 51.30 to Figure 51.35 IOVCCn $\geq$ 2.7V
	Input		—	1	μ s	Figure 51.30 to Figure 51.35
Slave access time		t_{SA}	—	$2 \times t_{pcyc} + 100$	ns	Figure 51.34, Figure 51.35 IOVCCn $\geq$ 2.7V
Slave output release time		t_{REL}	—	$2 \times t_{pcyc} + 100$	ns	Figure 51.34, Figure 51.35 IOVCCn $\geq$ 2.7V

Note 1. t_{pcyc} indicates the clock cycle of PCLKA.

Note 2. "N" is the number of delay cycles for RSPCK clock set at SPCKD register.

Note 3. "N" is the number of delay cycles for RSPCK clock set at SSLND register.

**Figure 51.29 SPI clock timing**

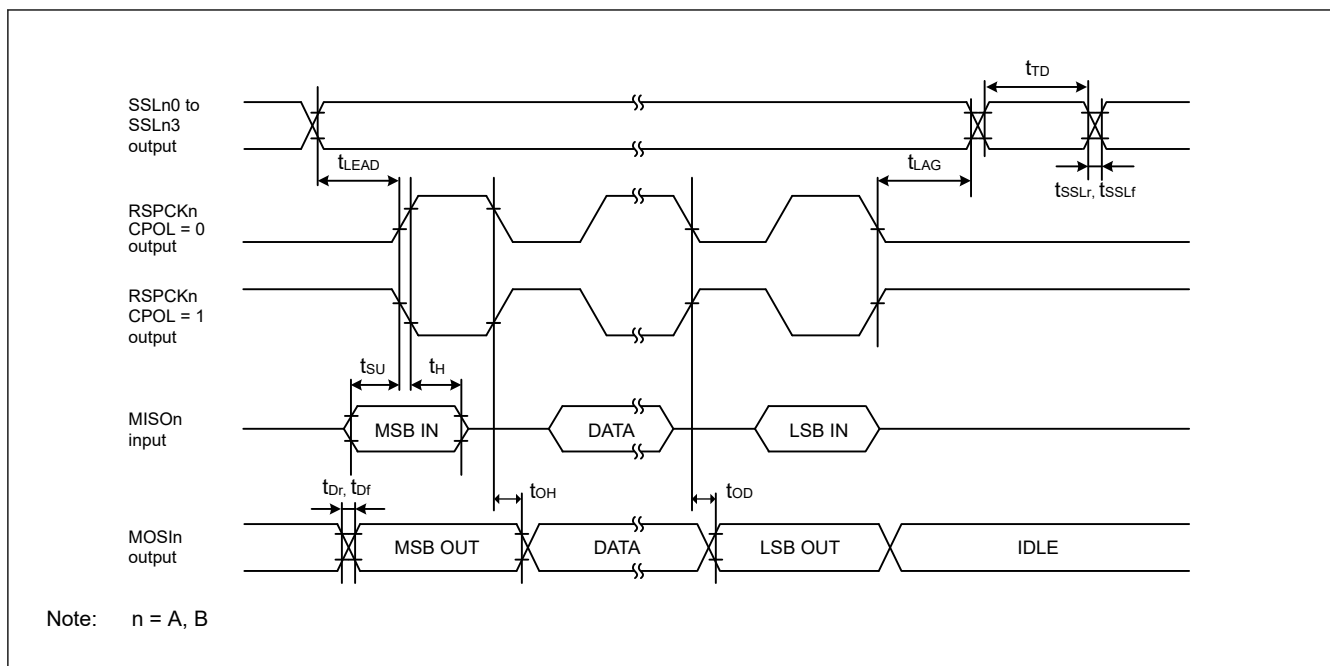


Figure 51.30 SPI timing (master, CPHA = 0) (bit rate: PCLKA division ratio is set to a value other than 1/2)

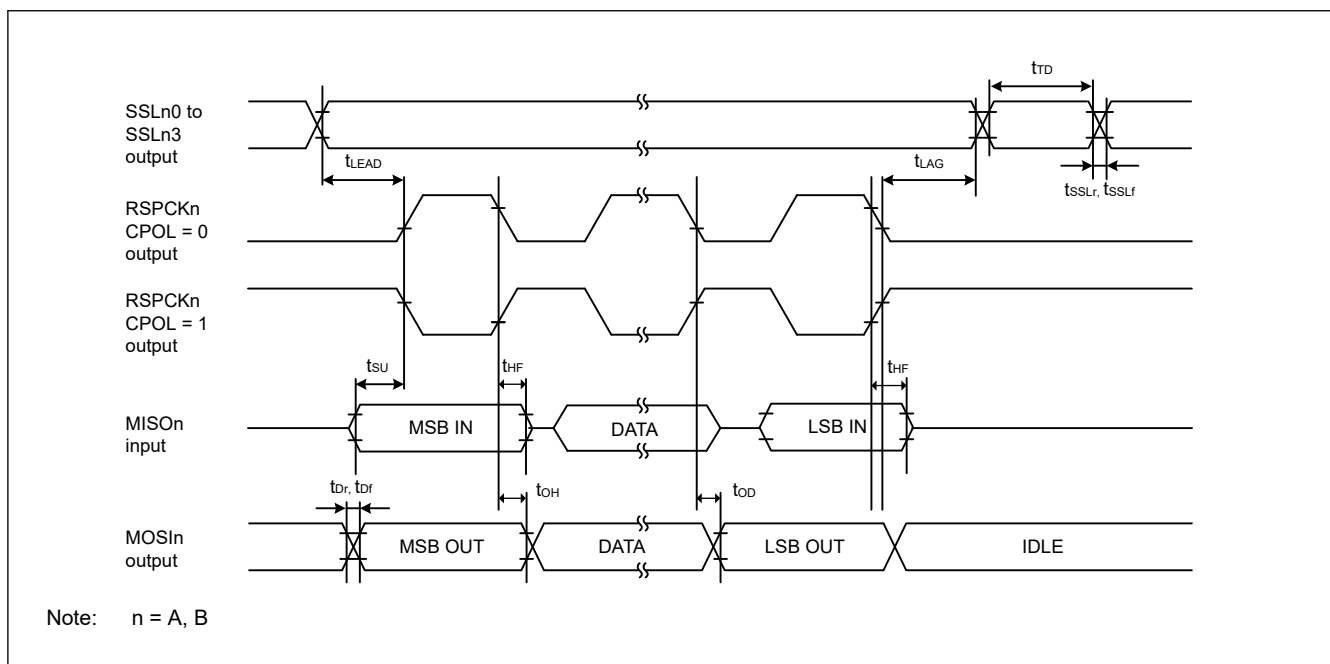


Figure 51.31 SPI timing (master, CPHA = 0) (bit rate: PCLKA division ratio is set to 1/2)

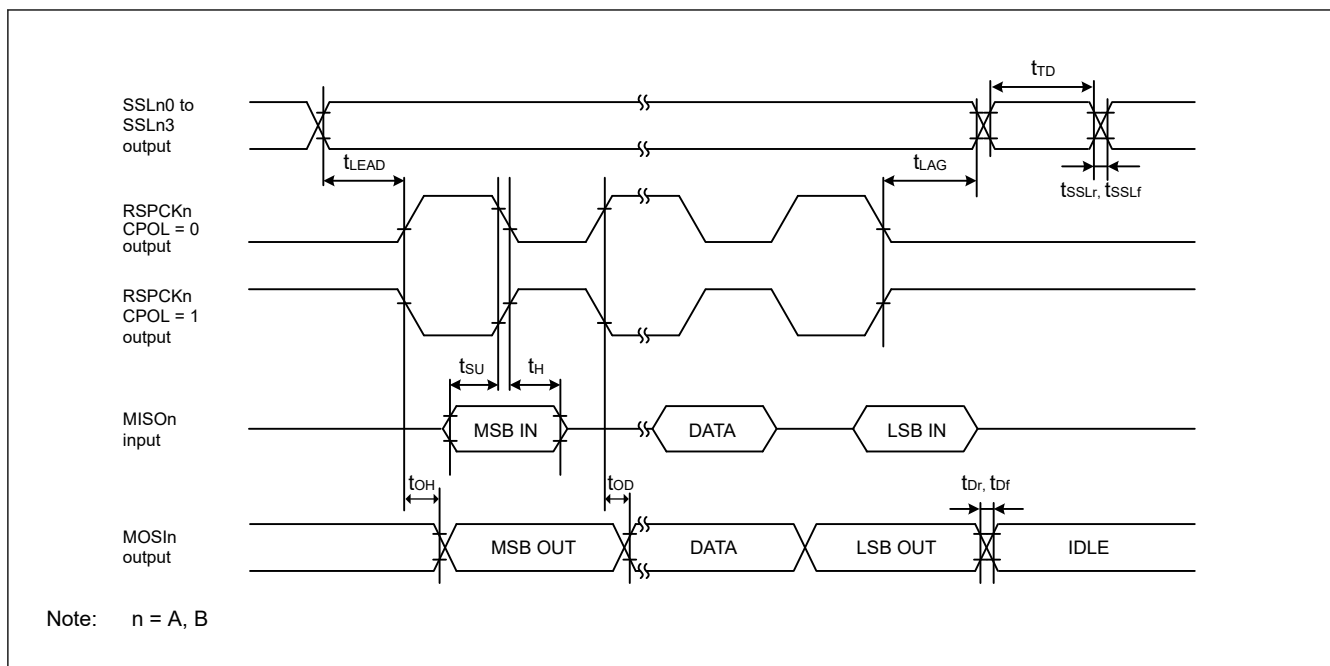


Figure 51.32 SPI timing (master, CPHA = 1) (bit rate: PCLKA division ratio is set to a value other than 1/2)

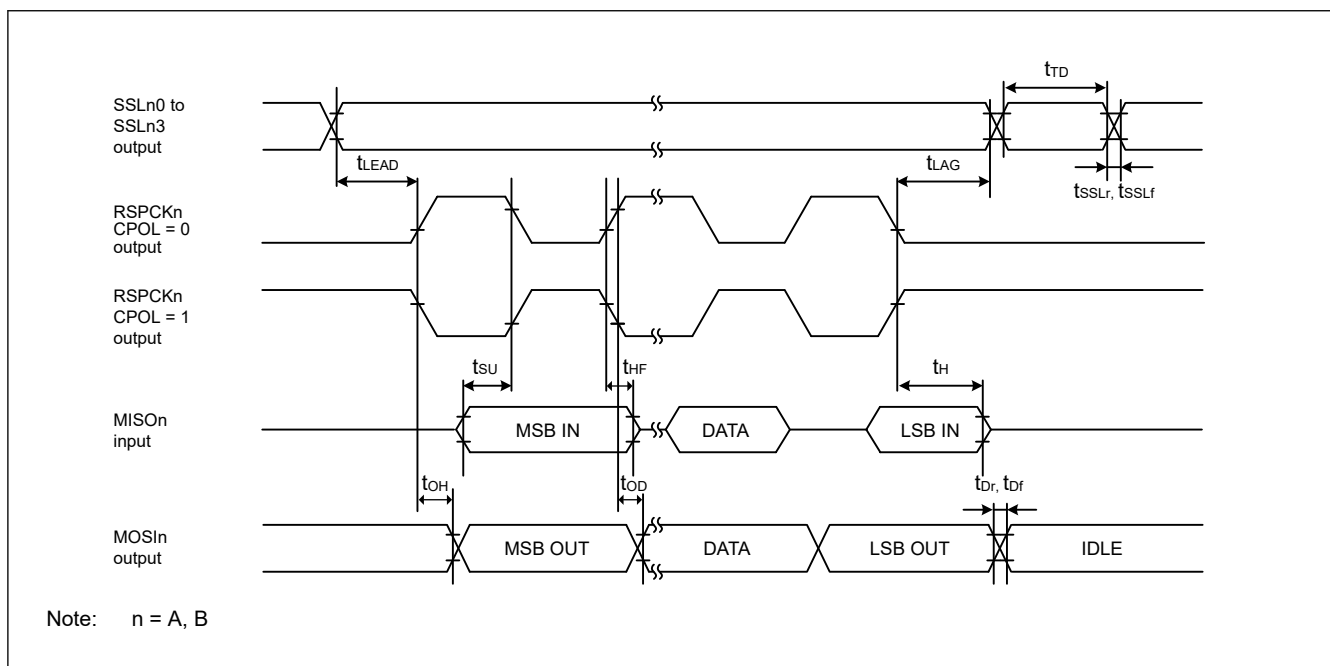


Figure 51.33 SPI timing (master, CPHA = 1) (bit rate: PCLKA division ratio is set to 1/2)

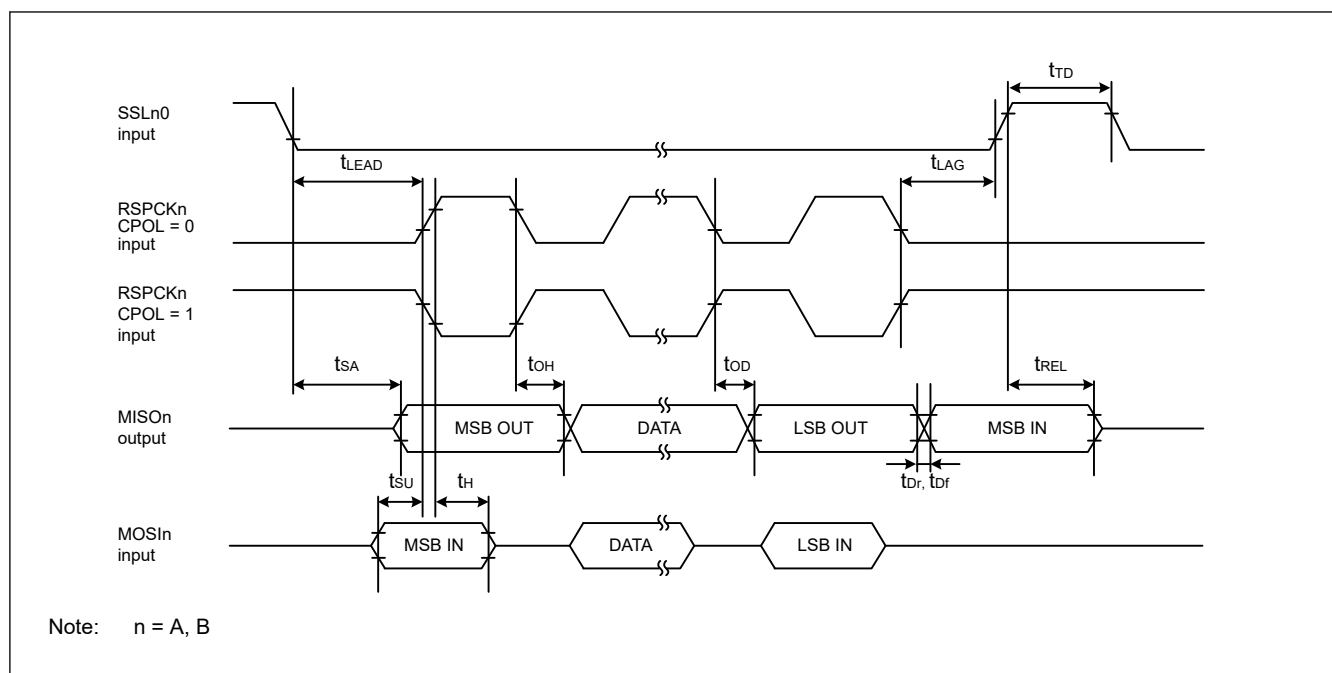


Figure 51.34 SPI timing (slave, CPHA = 0)

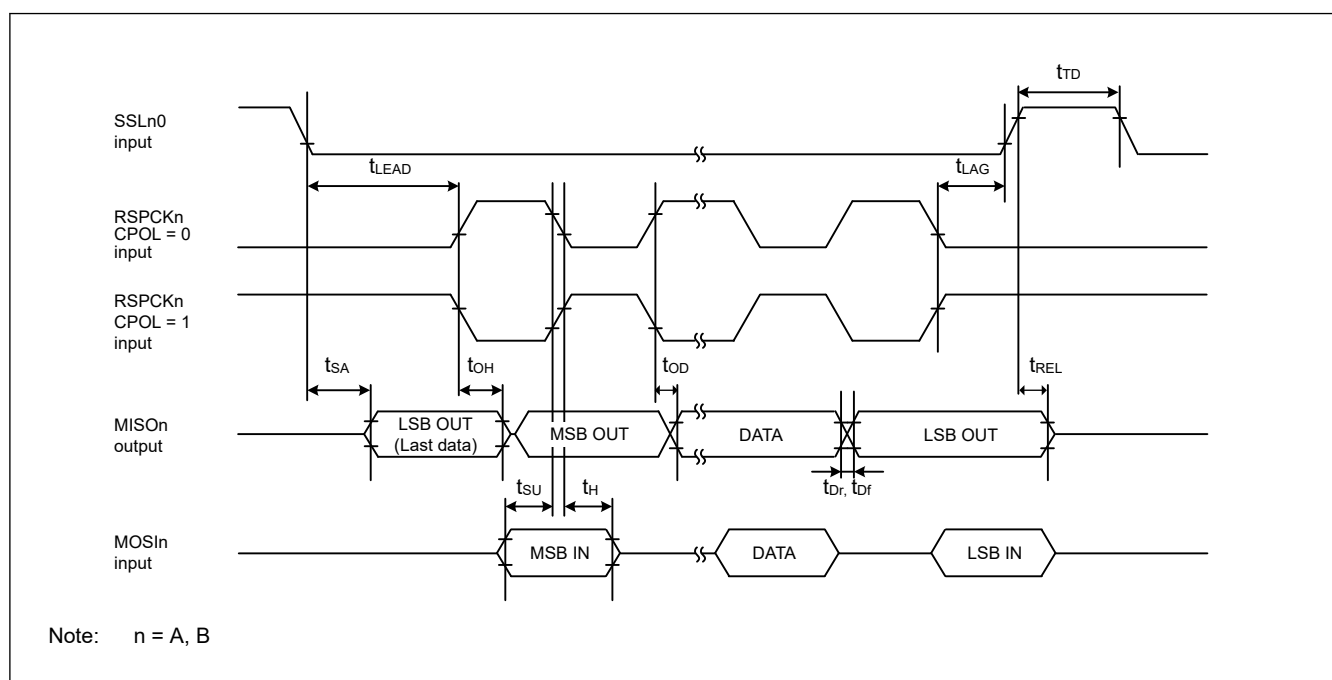


Figure 51.35 SPI timing (slave, CPHA = 1)

51.3.11 QSPI Timing

Table 51.28 QSPI timing

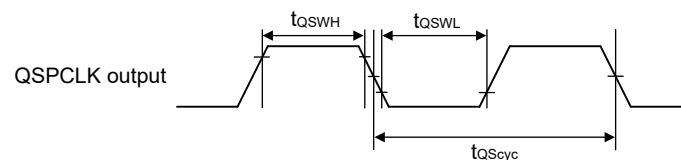
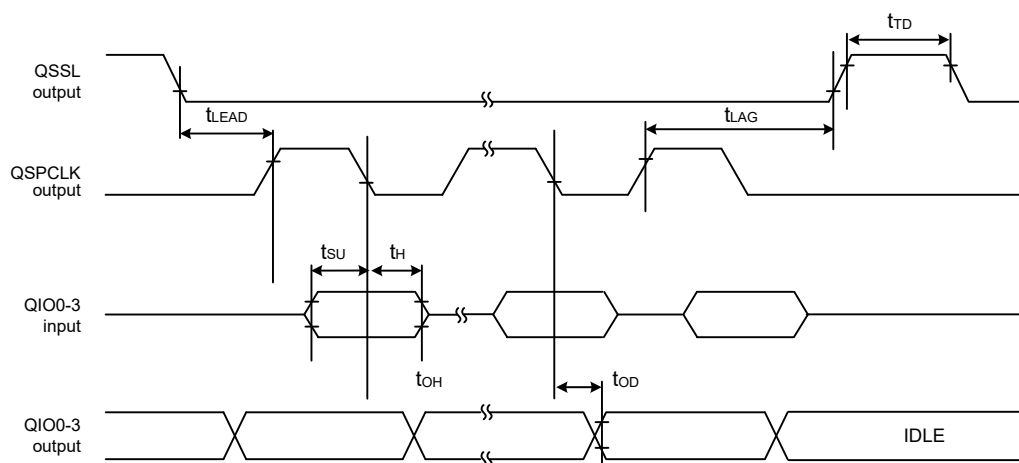
Condition: High drive output is selected in the drive capability control bits in PmnPFS register.

Item	Symbol	Min.	Max.	Unit ^{*1}	Measurement conditions
QSPCLK clock cycle (PCLKA > 48 MHz)	t_{QScyc}	3	4080	t_{Pcyc}	Figure 51.36
QSPCLK clock cycle (PCLKA ≤ 48 MHz)		2	4080		
QSPCLK clock high-level pulse width	t_{QSWH}	$t_{QScyc} \times 0.4$	—	ns	
QSPCLK clock low-level pulse width	t_{QSWL}	$t_{QScyc} \times 0.4$	—	ns	
Data input setup time	t_{SU}	25	—	ns	Figure 51.37 IOVCCn ≥ 2.7V
Data input hold time	t_H	12	—	ns	
QSSL setup time	t_{LEAD}	$(L + 0.5) \times t_{QScyc} - M^{*2}$	—	ns	
QSSL hold time	t_{LAG}	$(N + 0.5) \times t_{QScyc} - M^{*3}$	—	ns	
Data output delay	t_{OD}	-3.3	14	ns	
Successive transmission delay	t_{TD}	1	16	t_{QScyc}	

Note 1. t_{Pcyc} indicates the clock cycle of PCLKA.

Note 2. The value of L is the value set in the SFMSSC.SFMSLD bit. The value of M is 10 at the time of BOOST, and 15 at the time of NORMAL.

Note 3. The value of N is the value set in the SFMSSC.SFMSSHD bit. The value of M is 10 at the time of BOOST, and 15 at the time of NORMAL.

**Figure 51.36 QSPI clock timing****Figure 51.37 QSPI input/output timing**

51.3.12 IIC Timing

Table 51.29 IIC timingCondition: $V_{CC} = 3.0$ to 3.6 V, $V_{IH} = V_{CC} \times 0.7$, $V_{IL} = V_{CC} \times 0.3$, $V_{OH} = 0.6$ V, $I_{OL} = 6$ mA

Condition: Normal drive output is selected in the drive capability control bits in PmnPFS register. (PmnPFS.DSCR[1:0] = 10b)

Item		Symbol	Min.*1	Max.*1	Unit	Measurement conditions
IIC (Standard mode)	SCL input cycle time	t _{SCL}	6(12) × t _{IICcyc} + 1300	—	ns	Figure 51.38
	SCL input high-level pulse width	t _{SCLH}	3(6) × t _{IICcyc} + 300	—	ns	
	SCL input low-level pulse width	t _{SCLL}		—	ns	
	SCL, SDA input rise time	t _{Sr}	—	1000	ns	
	SCL, SDA input fall time	t _{Sf}	—	300	ns	
	SDA input bus free time	t _{BUF}	3(6) × t _{IICcyc} + 300	—	ns	
	Start condition input hold time	t _{STAH}	t _{IICcyc} + 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 *2	—	400	pF	
IIC (Fast mode)	SCL input cycle time	t _{SCL}	6(12) × t _{IICcyc} + 600	—	ns	Figure 51.38
	SCL input high-level pulse width	t _{SCLH}	3(6) × t _{IICcyc} + 300	—	ns	
	SCL input low-level pulse width	t _{SCLL}		—	ns	
	SCL, SDA input rise time	t _{Sr}	—	300	ns	
	SCL, SDA input fall time	t _{Sf}	—	300	ns	
	SDA input bus free time	t _{BUF}	3(6) × t _{IICcyc} + 300	—	ns	
	Start condition input hold time	t _{STAH}	t _{IICcyc} + 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 *2	—	400	pF	

Note: t_{IICcyc} indicates a clock cycle of IIC internal reference clock (IIC ϕ).

Note 1. If the digital filter is enabled by setting the ICFER.NFE bit to 1, when ICMR3.NF[1:0] bits are set to 11b, values in parentheses apply.

Note 2. C_b indicates the total capacity of the bus line.

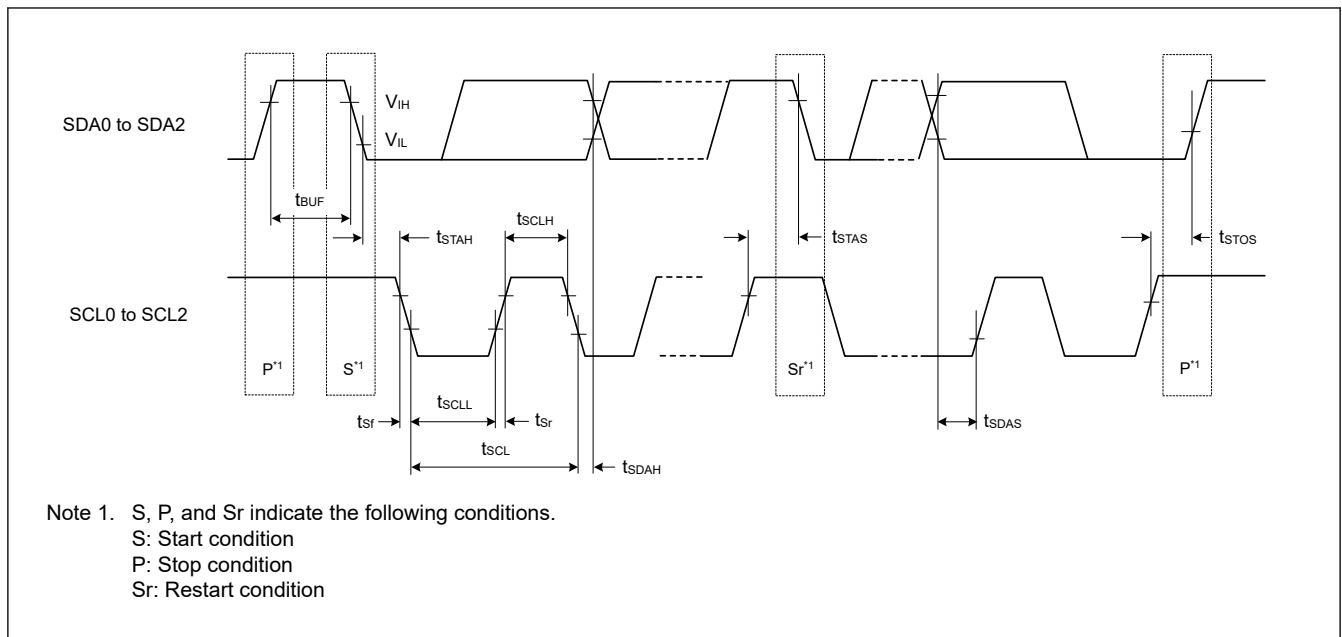


Figure 51.38 I2C bus interface input/output timing

51.3.13 MLCD Timing

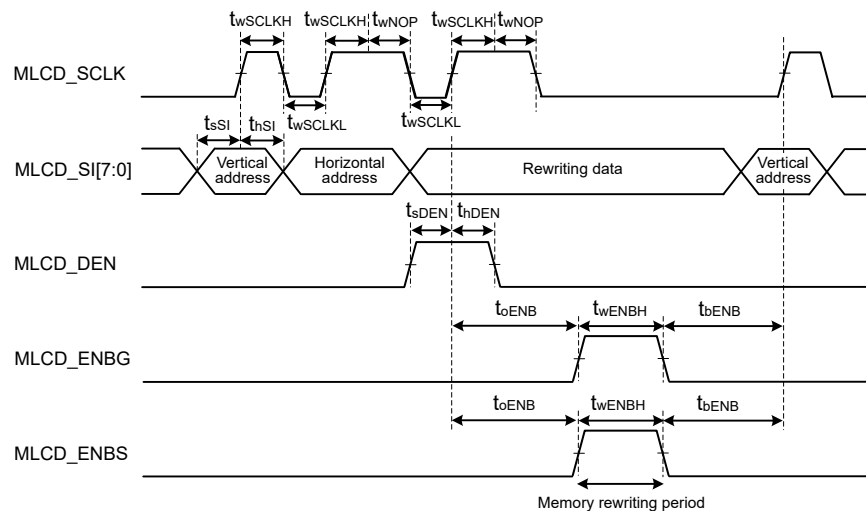
Table 51.30 MLCD timing

Condition: High drive output is selected in the drive capability control bits in PmnPFS register.

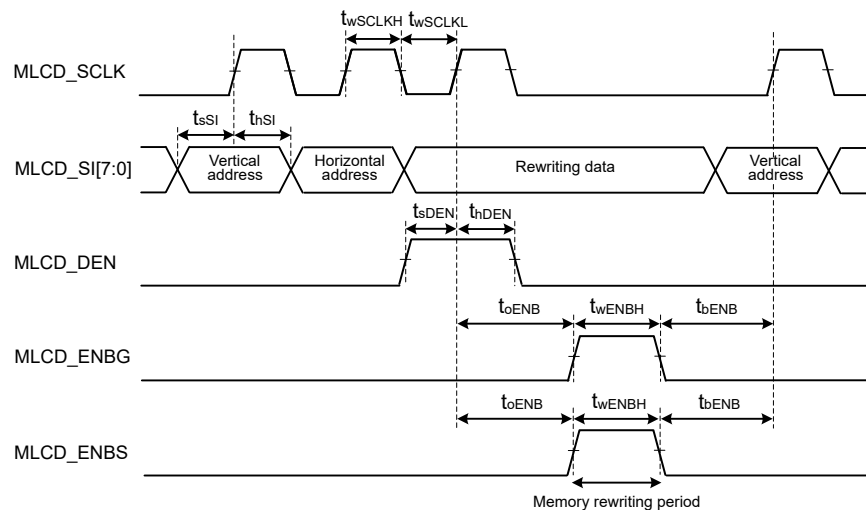
Item	Symbol	Min.	Typ.	Max.	Unit*1	Measurement conditions
MLCD_SCLK pin output high-level pulse width	t_{wSCLKH}	1	—	255	t_{Pcyc}	Figure 51.39
MLCD_SCLK pin output low-level pulse width	t_{wSCLKL}	1	—	255	t_{Pcyc}	
Data transmission wait time	t_{wNOP}	—	1	—	t_{Pcyc}	
MLCD_SI pin output setup time	t_{sSI}	1	—	255	t_{Pcyc}	
MLCD_SI pin output hold time	t_{hSI}	1	—	255	t_{Pcyc}	
MLCD_DEN pin output setup time	t_{sDEN}	1	—	255	t_{Pcyc}	
MLCD_DEN pin output hold time	t_{hDEN}	1	—	255	t_{Pcyc}	
MLCD_ENBG/S pin output high-level pulse width	t_{wENBH}	2	—	1023	t_{Pcyc}	
The time between the rise of MLCD_SCLK pin output and the rise of MLCD_ENBG/S pin output.	t_{oENB}	3	—	255	t_{Pcyc}	
The time between the fall of MLCD_ENBG/S pin output and the rise of MLCD_SCLK pin output.	t_{bENB}	3	—	255	t_{Pcyc}	
Duty ratio of MLCD_VCOM pin output	—	—	50	—	%	
MLCD_VCOM pin output high-level/low-level pulse time	t_{cVCOM}	500	—	5000	ms	

Note 1. t_{Pcyc} indicates the clock cycle of PCLKA.

(a) MLCD_CR.SCKCR[7:0] = 1



(b) MLCD_CR.SCKCR[7:0] = 2 to 255



(c) MLCD_VCOM pin output

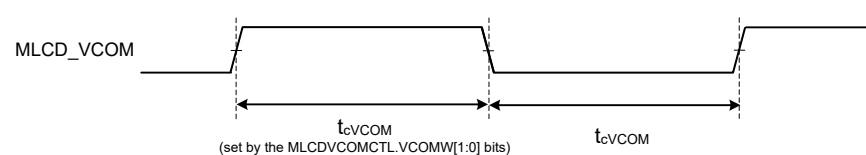


Figure 51.39 MLCD output timing

51.3.14 CLKOUT Timing

Table 51.31 CLKOUT timing

Item			Symbol	Min.	Max.	Unit	Measurement conditions
CLKOUT	CLKOUT pin output cycle*1	IOVCCn ≥ 2.7V	t _{Cyc}	31.25	—	ns	Figure 51.40
		IOVCCn < 2.7V		62.5	—		
CLKOUT32	CLKOUT pin output cycle		t _{Cyc}	30.5	—	μs	

Note 1. When the EXTAL external clock input or an sub-clock oscillator is used with division by 1 (the CKOCR.CKOSSEL[2:0] bits are 011b and the CKOCR.CKODIV[2:0] bits are 000b) to output from CLKOUT, the above should be satisfied with an input duty cycle of 45 to 55%.

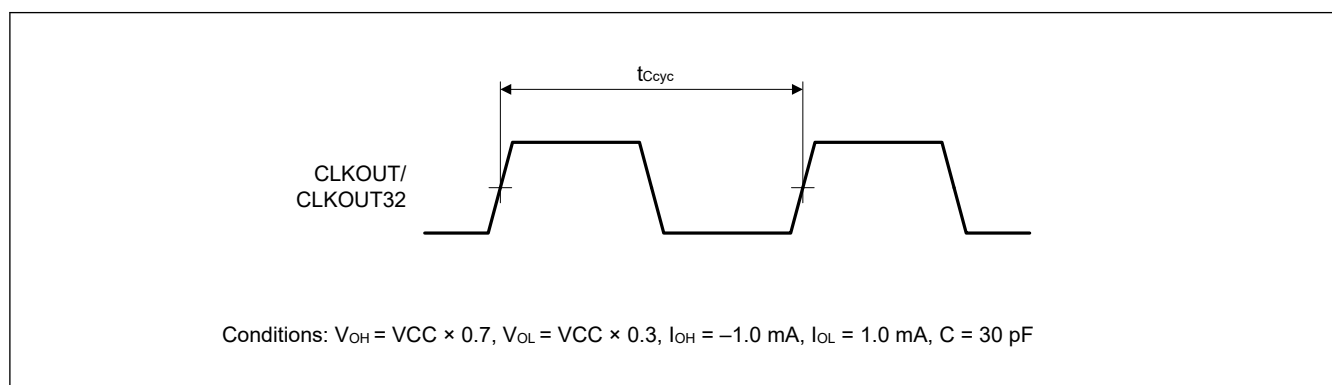


Figure 51.40 CLKOUT/CLKOUT32 pin output timing

51.3.15 TMR Timing

Table 51.32 TMR timing

Item			Symbol	Min.	Typ.	Max.	Unit*1	Measurement conditions
TMR	Timer clock pulse width	Single-edge setting	t_{TMCWH}	1.5	—	—	t_{Pcyc}	Figure 51.41
		Both-edge setting	t_{TMCWL}	2.5	—	—		

Note 1. t_{Pcyc} indicates the clock cycle of PCLKB.

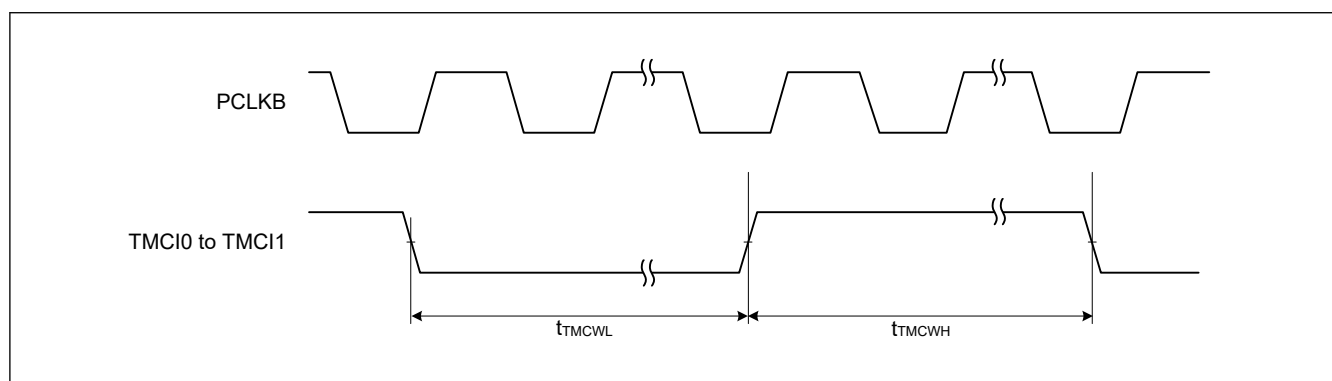


Figure 51.41 TMR clock input timing

51.4 A/D Conversion Characteristics

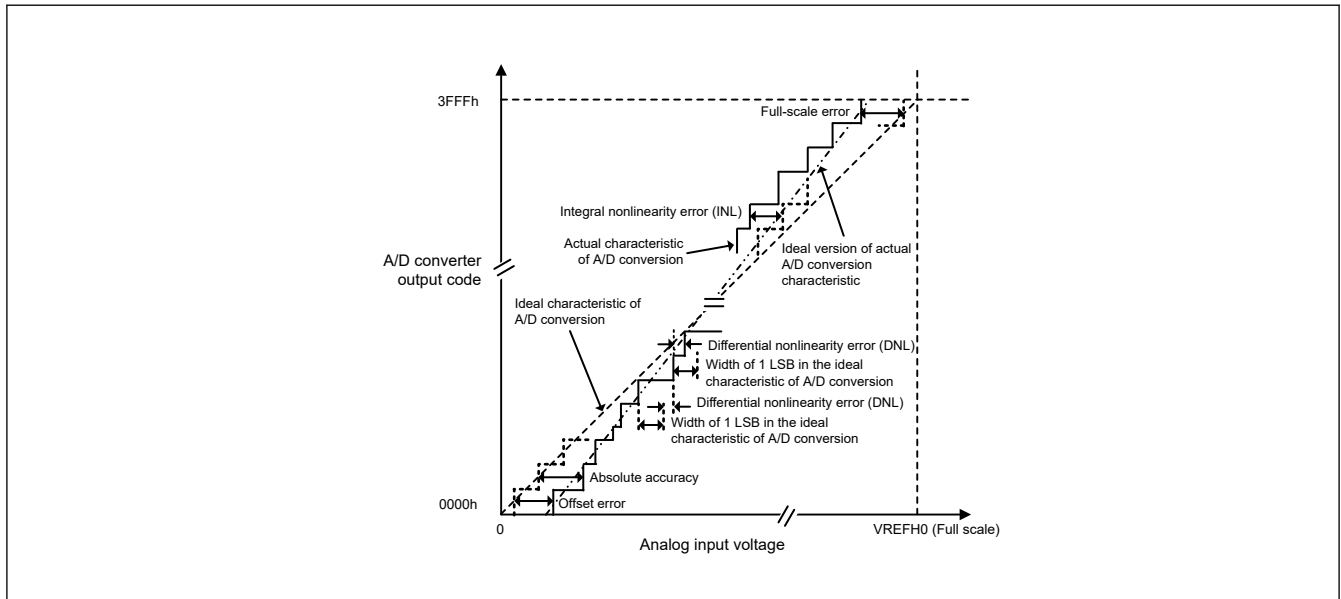


Figure 51.42 Illustration of A/D converter characteristic terms

Absolute accuracy

Absolute accuracy is the difference between output codes of the ideal A/D conversion characteristics and the actual A/D conversion result. When measuring absolute accuracy, the voltage at the midpoint of the width of the analog input voltage (1-LSB width), which can meet the expectation of outputting an equal code based on the theoretical A/D conversion characteristics, is used as the analog input voltage. In the case of 14-bit resolution and reference voltage of $V_{REFH0} = 3.276\text{ V}$, for example, because 1-LSB width is 0.2 mV, voltages such as 0 mV, 0.2 mV, and 0.4 mV are used as the analog input voltage.

If the analog input voltage is 1.6 mV, an absolute accuracy of $\pm 5\text{ LSB}$ means that the actual A/D conversion result ranges from 0x0003 to 0x000D though an output code of 0x0008 can be expected from the ideal A/D conversion characteristics.

Integral nonlinearity error (INL)

Integral nonlinearity error is the maximum deviation between the ideal line when the measured offset and full-scale errors are zeroed, and the actual output code.

Differential nonlinearity error (DNL)

Differential nonlinearity error is the difference between 1-LSB width of the ideal A/D conversion characteristics and the width of the output code actually output.

Offset error

Offset error is the difference between the transition point of the ideal first output code and the actual first output code.

Full-scale error

Full-scale error is the difference between the transition point of the ideal last output code and the actual last output code.

The conversion characteristics of A/D converter is not tested at the shipment unless otherwise specified. The values described are presented only as the design guidelines. The electrical characteristics presented are classified into the following six categories in accordance with the conditions such as voltages.

1. $AVCC0 = V_{REFH0} = 2.7\text{ to }3.6\text{ V}$ However, $\pm 3\sigma$ of the voltage in the normal distribution is within the range of maximum value.
2. $AVCC0 = V_{REFH0} = 2.7\text{ to }3.6\text{ V}$
3. $AVCC0 = V_{REFH0} = 1.62\text{ to }3.6\text{ V}$, 14-bit resolution
4. $AVCC0 = V_{REFH0} = 1.62\text{ to }3.6\text{ V}$, 12-bit resolution
5. $AVCC0 = 3.3\text{ V}$, $AVTRO = 2.5\text{ V}$ (The output value of the reference voltage generator circuit is used as the reference.)

6. $AVCC0 = 1.8\text{ V}$, $AVTRO = 1.25\text{ V}$ (The output value of the reference voltage generator circuit is used as the reference.)

Some points to note regarding the electrical characteristics of the A/D converter are listed below:

- The characteristics do not contain the quantization errors ($\pm 0.5\text{ LSB}$).
- The characteristics are the values after the offset calibrations.
- The characteristics only apply when the 14-bit A/D converter pins are in use for A/D conversion, and not for any other functions.
- The conversion time (t_{CONV}) is the sum of the sampling time (t_{SPL}) and time for conversion by successive approximation (t_{SAM}). The values in parentheses in the conversion time indicate the sampling time.

Table 51.33 A/D conversion characteristics (1)

Condition: $AVCC0 = VREFH0 = 2.7\text{ to }3.6\text{ V}$

Item		Min.	Typ.	Max.	Unit	Measurement conditions
Frequency		1	—	32^{*3}	MHz	ADCLKCR.SCLKEN = 0
		—	32.768	—	kHz	ADCLKCR.SCLKEN = 1
Dynamic range	A_{in}	0	—	VREFH0	V	—
Resolution		12	—	14	bit	—
Conversion time	Permissible signal source impedance Max. = $0.5\text{ k}\Omega$	1.0 (0.46875)	—	—	μs	High-precision channel ADCLKCR.SCLKEN = 0 ADSSTRn.SST = 0x0F
		1.5 (0.96875)	—	—	μs	Normal-precision channel ADCLKCR.SCLKEN = 0 ADSSTRn.SST = 0x1F
		593.75 (60.98)	—	—	μs	ADCLKCR.SCLKEN = 1 ADSSTRn.SST = 0x02
Offset error ^{*1}		-0.8	—	0.8	mV	High-precision channel
Full-scale error ^{*1}		-0.8	—	0.8		High-precision channel
Absolute accuracy ^{*1}		—	$\pm 4.0^{*2}$	± 7.0	LSB	High-precision channel
DNL differential nonlinearity error ^{*1}		—	$\pm 1.0^{*2}$	± 1.5	LSB	High-precision channel
INL integral nonlinearity error ^{*1}		—	$\pm 2.5^{*2}$	± 4.0	LSB	High-precision channel
ENOB (Effective number of bits) ^{*1*2*4}		—	13	—	bit	High-precision channel

Note 1. The values apply when the averaging mode is enabled, averaging of 16 results of conversion is selected (ADADC = 0x85), and the conversion resolution is set to 14 bits (ADCER.ADPRC[1:0] = 11b).

Note 2. The value applies when $AVCC0 = VREFH0 = 3.3\text{ V}$.

Note 3. If $AVCC0 \neq VREFH0$, the condition $AVCC0 \geq VREFH0 \geq 2.7\text{ V}$ applies.

Note 4. The value applies when the main oscillator is selected as PCLKB and a 50-Hz sine wave is input to the analog input pin.

Table 51.34 A/D conversion characteristics (2) (1 of 2)

Condition: $AVCC0 = VREFH0 = 2.7\text{ to }3.6\text{ V}$

Item		Min.	Typ.	Max.	Unit	Measurement conditions
Frequency		1	—	32^{*3}	MHz	ADCLKCR.SCLKEN = 0
		—	32.768	—	kHz	ADCLKCR.SCLKEN = 1
Dynamic range	A_{in}	0	—	VREFH0	V	—
Resolution		12	—	14	bit	—

Table 51.34 A/D conversion characteristics (2) (2 of 2)

Condition: AVCC0 = VREFH0 = 2.7 to 3.6 V

Item		Min.	Typ.	Max.	Unit	Measurement conditions
Conversion time	Permissible signal source impedance Max. = 0.5 kΩ	1.0 (0.46875)	—	—	μs	High-precision channel ADCLKCR.SCLKEN = 0 ADSSTRn.SST = 0x0F
		1.5 (0.96875)	—	—	μs	Normal-precision channel ADCLKCR.SCLKEN = 0 ADSSTRn.SST = 0x1F
		593.75 (60.98)	—	—	μs	ADCLKCR.SCLKEN = 1 ADSSTRn.SST = 0x02
Offset error*1		-1.2	—	1.2	mV	High-precision channel
Full-scale error*1		-1.2	—	1.2		High-precision channel
Absolute accuracy*1		—	±4.0*2	±9.0	LSB	High-precision channel
DNL differential nonlinearity error*1		—	±1.0*2	±1.7	LSB	High-precision channel
INL integral nonlinearity error*1		—	±2.5*2	±5.0	LSB	High-precision channel
ENOB (Effective number of bits) *1*2*4		—	13	—	bit	High-precision channel

Note 1. The values apply when the averaging mode is enabled, averaging of 16 results of conversion is selected (ADADC = 0x85), and the conversion resolution is set to 14 bits (ADCER.ADPRC[1:0] = 11b).

Note 2. The value applies when AVCC0 = VREFH0 = 3.3 V.

Note 3. If AVCC0 ≠ VREFH0, the condition AVCC0 ≥ VREFH0 ≥ 2.7V applies.

Note 4. The value applies when the main oscillator is selected as PCLKB and a 50-Hz sine wave is input to the analog input pin.

Table 51.35 A/D conversion characteristics (3)

Condition: AVCC0 = VREFH0 = 1.62 to 3.6 V

Item		Min.	Typ.	Max.	Unit	Measurement conditions
Frequency		1	—	16 ^{*3}	MHz	ADCLKCR.SCLKEN = 0
		—	32.768	—	kHz	ADCLKCR.SCLKEN = 1
Dynamic range	A _{in}	0	—	VREFH0	V	—
Resolution		12	—	14	bit	—
Conversion time	Permissible signal source impedance Max. = 0.5 kΩ	2.0 (0.9375)	—	—	μs	High-precision channel ADCLKCR.SCLKEN = 0 ADSSTRn.SST = 0x0F
		3.0 (1.9375)	—	—	μs	Normal-precision channel ADCLKCR.SCLKEN = 0 ADSSTRn.SST = 0x1F
		593.75 (60.98)	—	—	μs	ADCLKCR.SCLKEN = 1 ADSSTRn.SST = 0x02
Offset error ^{*1}		-1.2	—	1.2	mV	High-precision channel
Full-scale error ^{*1}		-1.2	—	1.2		High-precision channel
Absolute accuracy ^{*1}		—	±4.0 ^{*2}	±1.2 ^{*5}	LSB	High-precision channel
DNL differential nonlinearity error ^{*1}		—	±1.0 ^{*2}	±2.5 ^{*6}	LSB	High-precision channel
INL integral nonlinearity error ^{*1}		—	±2.5 ^{*2}	±5.0	LSB	High-precision channel
ENOB (Effective number of bits) ^{*1*2*4}		—	13	—	bit	High-precision channel

Note 1. The values apply when the averaging mode is enabled, averaging of 16 results of conversion is selected (ADADC = 0x85), and the conversion resolution is set to 14 bits (ADCER.ADPRC[1:0] = 11b).

Note 2. The value applies when AVCC0 = VREFH0 = 3.3 V.

Note 3. If AVCC0 ≠ VREFH0, the condition AVCC0 ≥ VREFH0 ≥ 2.4 V applies.

Note 4. The value applies when the main oscillator is selected as PCLKB and a 50-Hz sine wave is input to the analog input pin.

Note 5. When AVCC0 = VREFH0 = 2.4 to 3.6 V, the maximum value is ±9.0 LSB.

Note 6. When AVCC0 = VREFH0 = 2.4 to 3.6 V, the maximum value is ±1.7 LSB.

Table 51.36 A/D conversion characteristics (4)

Condition: AVCC0 = VREFH0 = 1.62 to 3.6 V

Item		Min.	Typ.	Max.	Unit	Measurement conditions
Frequency		1	—	16*2	MHz	ADCLKCR.SCLKEN = 0
		—	32.768	—	kHz	ADCLKCR.SCLKEN = 1
Dynamic range	A _{in}	0	—	VREFH0	V	—
Resolution		—	—	12	bit	—
Conversion time	Permissible signal source impedance Max. = 0.5 kΩ	2.0 (0.9375)	—	—	μs	High-precision channel ADCLKCR.SCLKEN = 0 ADSSTRn.SST = 0x0F
		3.0 (1.9375)	—	—	μs	Normal-precision channel ADCLKCR.SCLKEN = 0 ADSSTRn.SST = 0x1F
		593.75 (60.98)	—	—	μs	ADCLKCR.SCLKEN = 1 ADSSTRn.SST = 0x02
Offset error*1		-1.2	—	1.2	mV	High-precision channel
Full-scale error*1		-1.2	—	1.2		High-precision channel
Absolute accuracy*1		—	±2.0	±5.0	LSB	High-precision channel
DNL differential nonlinearity error*1		—	±1.0	±1.5	LSB	High-precision channel
INL integral nonlinearity error*1		—	±1.0	±2.0	LSB	High-precision channel

Note 1. The values apply when the averaging mode is disabled and the conversion resolution is set to 12 bits (ADCER.ADPRC[1:0] = 0x00)

Note 2. If AVCC0 ≠ VREFH0, the condition AVCC0 ≥ VREFH0 ≥ 1.62 V applies.

Table 51.37 A/D conversion characteristics when the output value of the reference voltage generator circuit is used as the reference voltage (1)

Condition: AVCC0 = 3.3 V, AVTRO = 2.50 V

Item		Min.	Typ.	Max.	Unit	Measurement conditions
Frequency		1	—	16	MHz	ADCLKCR.SCLKEN = 0
		—	32.768	—	kHz	ADCLKCR.SCLKEN = 1
Dynamic range	A _{in}	0	—	VREFH0	V	—
Resolution		12	—	14	bit	—
Conversion time	Permissible signal source impedance Max. = 0.5 kΩ	2.0 (0.46875)	—	—	μs	High-precision channel ADCLKCR.SCLKEN = 0 ADSSTRn.SST = 0x0F
		3.0 (1.9375)	—	—	μs	Normal-precision channel ADCLKCR.SCLKEN = 0 ADSSTRn.SST = 0x1F
		593.75 (60.98)	—	—	μs	ADCLKCR.SCLKEN = 1 ADSSTRn.SST = 0x02
Offset error*1		-1.2	—	1.2	mV	High-precision channel
DNL differential nonlinearity error*1		—	±1.5	—	LSB	High-precision channel
INL integral nonlinearity error*1		—	±3.0	—	LSB	High-precision channel

Note 1. The value in which 16 times conversion with enabling the averaging mode (ADADC = 0x85) is selected, and in which the conversion precision is set to 14-bit (ADCER.ADPRC[1:0] = 11b)

Table 51.38 A/D conversion characteristics when the output value of the reference voltage generator circuit is used as the reference voltage (2)

Condition: AVCC0 = 1.8 V, AVTRO = 1.25 V

Item		Min.	Typ.	Max.	Unit	Measurement conditions
Frequency		1	—	16	MHz	ADCLKCR.SCLKEN = 0
		—	32.768	—	kHz	ADCLKCR.SCLKEN = 1
Dynamic range	A _{in}	0	—	VREFH0	V	—
Resolution		—	—	12	bit	—
Conversion time	Permissible signal source impedance Max. = 0.5 kΩ	2.0 (0.46875)	—	—	μs	High-precision channel ADCLKCR.SCLKEN = 0 ADSSTRn.SST = 0x0F
		3.0 (1.9375)	—	—	μs	Normal-precision channel ADCLKCR.SCLKEN = 0 ADSSTRn.SST = 0x1F
		593.75 (60.98)	—	—	μs	ADCLKCR.SCLKEN = 1 ADSSTRn.SST = 0x02
Offset error*1		-1.2	—	1.2	mV	High-precision channel
DNL differential nonlinearity error*1		—	±1.0	—	LSB	High-precision channel
INL integral nonlinearity error*1		—	±1.0	—	LSB	High-precision channel

Note 1. The value in which the averaging mode is disabled and in which the conversion precision is set to 12-bit (ADCER.ADPRC[1:0] = 00b)

51.5 Temperature Sensor Characteristics

Table 51.39 Temperature sensor characteristics

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Relative accuracy	—	—	±5	—	°C	AVCC0 ≥ 2.6 V
	—	—	±6	—	°C	AVCC0 < 2.6 V
Temperature gradient	—	—	1.6	—	mV/°C	—
Temperature sensor start time	t _{TSTBL}	—	30	120	μs	—
Sampling time	—	—	2	7	μs	—

Note: Temperature sensor characteristics are a reference value because the shipment test is not performed.

51.6 VREF Characteristics

Table 51.40 VREF characteristics

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Output voltage	AVTRO	1.17	1.25	1.33	V	VREF.AVCR.AVSEL = 0 AVCC0 ≥ 2.8V
	AVTRO	2.34	2.50	2.66	V	VREF.AVCR.AVSEL = 1 AVCC0 ≥ 2.8V
	AVTRO	1.17	1.25	1.33	V	VREF.AVCR.AVSEL = 0 AVCC0 < 2.8V
Circuit startup stabilization wait time	t _{VRSTUP}	—	—	50	ms	—

Note: VREF characteristics are a reference value because the shipment test is not performed.

51.7 Oscillation Stop Detection Circuit Characteristics

Table 51.41 Oscillation stop detection circuit characteristics

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Detection time	t_{dr}	—	—	30	μs	Figure 51.43

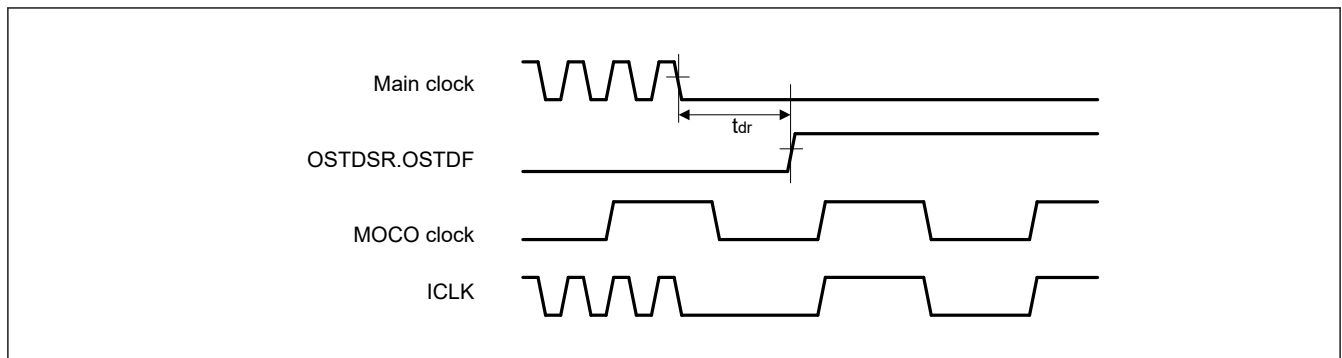


Figure 51.43 Oscillation stop detection timing

51.8 Power-on Reset Circuit and Low-voltage Detection Circuit Characteristics

Table 51.42 Power-on reset circuit and low-voltage detection circuit characteristics (1 of 2)

Item			Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Voltage detection level	Power-on reset circuit (POR)	Rise	V_{POR}	1.40	1.50	1.60	V	Figure 51.44
		Fall	V_{PORL}	1.30	1.40	1.50	V	
	Voltage monitoring 0 circuit (LVD0)		V_{det0_0}	2.34	2.42	2.50	V	Figure 51.45
			V_{det0_1}	2.10	2.17	2.24		
			V_{det0_2}	1.86	1.92	1.98		
			V_{det0_3}	1.62	1.67	1.72		
	Voltage monitoring 1 circuit (LVD1)		V_{det1_0}	2.74	2.83	2.92	V	Figure 51.46
			V_{det1_1}	2.58	2.66	2.74		
			V_{det1_3}	2.42	2.50	2.58		
			V_{det1_5}	2.26	2.33	2.40		
			V_{det1_7}	2.10	2.17	2.24		
			V_{det1_9}	1.94	2.00	2.06		
			V_{det1_B}	1.78	1.84	1.90		
			V_{det1_D}	1.62	1.67	1.72		
	Voltage monitoring BAT circuit (LVDBAT)		V_{detBAT_5}	2.26	2.33	2.40	V	Figure 51.47
			V_{detBAT_7}	2.10	2.17	2.24		
			V_{detBAT_9}	1.94	2.00	2.06		
			V_{detBAT_B}	1.78	1.84	1.90		
			V_{detBAT_D}	1.62	1.67	1.72		

Table 51.42 Power-on reset circuit and low-voltage detection circuit characteristics (2 of 2)

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Internal reset time	Power-on reset time* ⁸ (in the normal startup mode)	t _{PORNML}	—	—	44	ms	—
	LVD0 reset time	t _{LVD0}	—	2.5	—	ms	Figure 51.45
	LVD1 reset time	t _{LVD1}	—	0.8	—	ms	Figure 51.46
	LVDBAT reset time	t _{LVDBAT}	—	0.8	—	ms	Figure 51.47
Minimum VCC down time* ¹		t _{VOFFPOR}	4	—	—	ms	Figure 51.44
POR response delay time		t _{detpor}	—	—	500	μs	Figure 51.44
LVD0 response delay time		t _{det}	—	150	300	μs	Figure 51.45 to Figure 51.47
LVD0 response delay time			—	150	300	μs	
LVDBAT response delay time (VCC = VBAT_EHC, on connection)			—	150	300	μs	
LVDBAT response delay time (VCC ≠ VBAT_EHC, on independence)			—	400	800	μs	
LVD1 operation stabilization time (after LVD is enabled)		t _{d(E-A)}	—	—	400	μs	Figure 51.46, Figure 51.47
LVDBAT operation stabilization time (VCC = VBAT_EHC, on connection)			—		400	μs	
LVDBAT operation stabilization time (VCC ≠ VBAT_EHC, on independence)			—		1000	μs	
Hysteresis width (LVD1)		V _{LVH} * ²	—	60	—	mV	
		V _{LVH} * ³	—	55	—		
		V _{LVH} * ⁴	—	50	—		
		V _{LVH} * ⁵	—	45	—		
		V _{LVH} * ⁶	—	40	—		
		V _{LVH} * ⁷	—	35	—		

Note 1. The minimum VCC down time indicates the time when VCC is below the lowest value among voltage detection levels V_{POR} , V_{det0} , V_{det1} , and V_{detBAT} for POR and LVD.

Note 2. When V_{det1_0} is selected.

Note 3. When V_{det1_1} and V_{det1_3} are selected.

Note 4. When V_{det1_5} is selected.

Note 5. When V_{det1_7} is selected.

Note 6. When V_{det1_9} and V_{det1_B} are selected.

Note 7. When V_{det1_D} is selected.

Note 8. These values are based on simulation. They are not production tested.

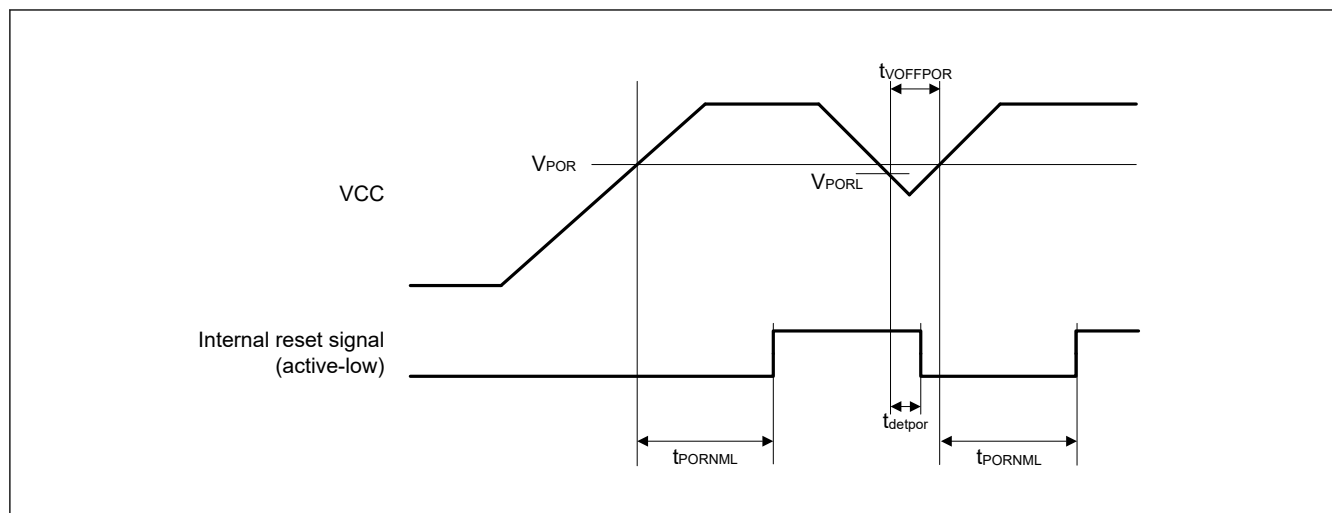
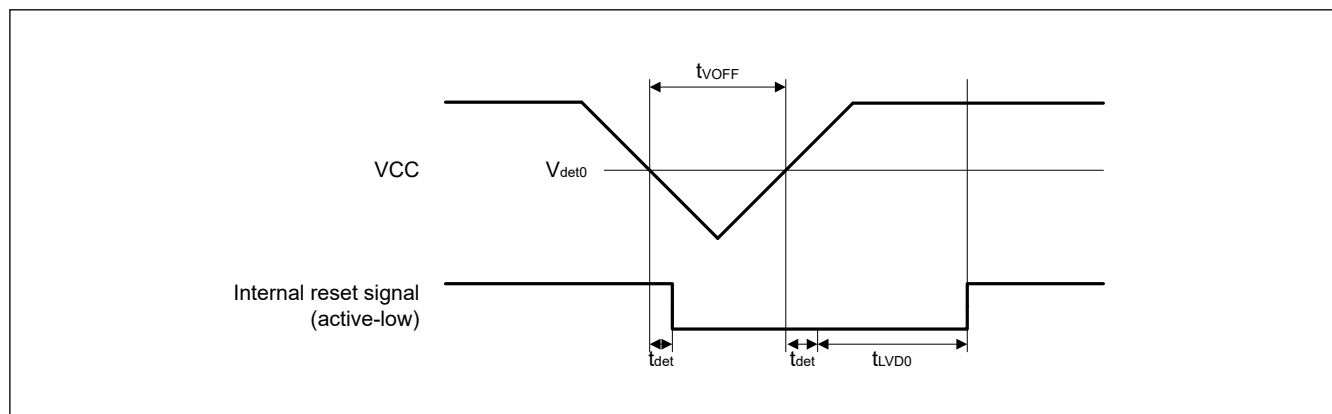


Figure 51.44 Power-on reset timing

Figure 51.45 Voltage monitoring 0 circuit detection voltage timing (V_{det0})

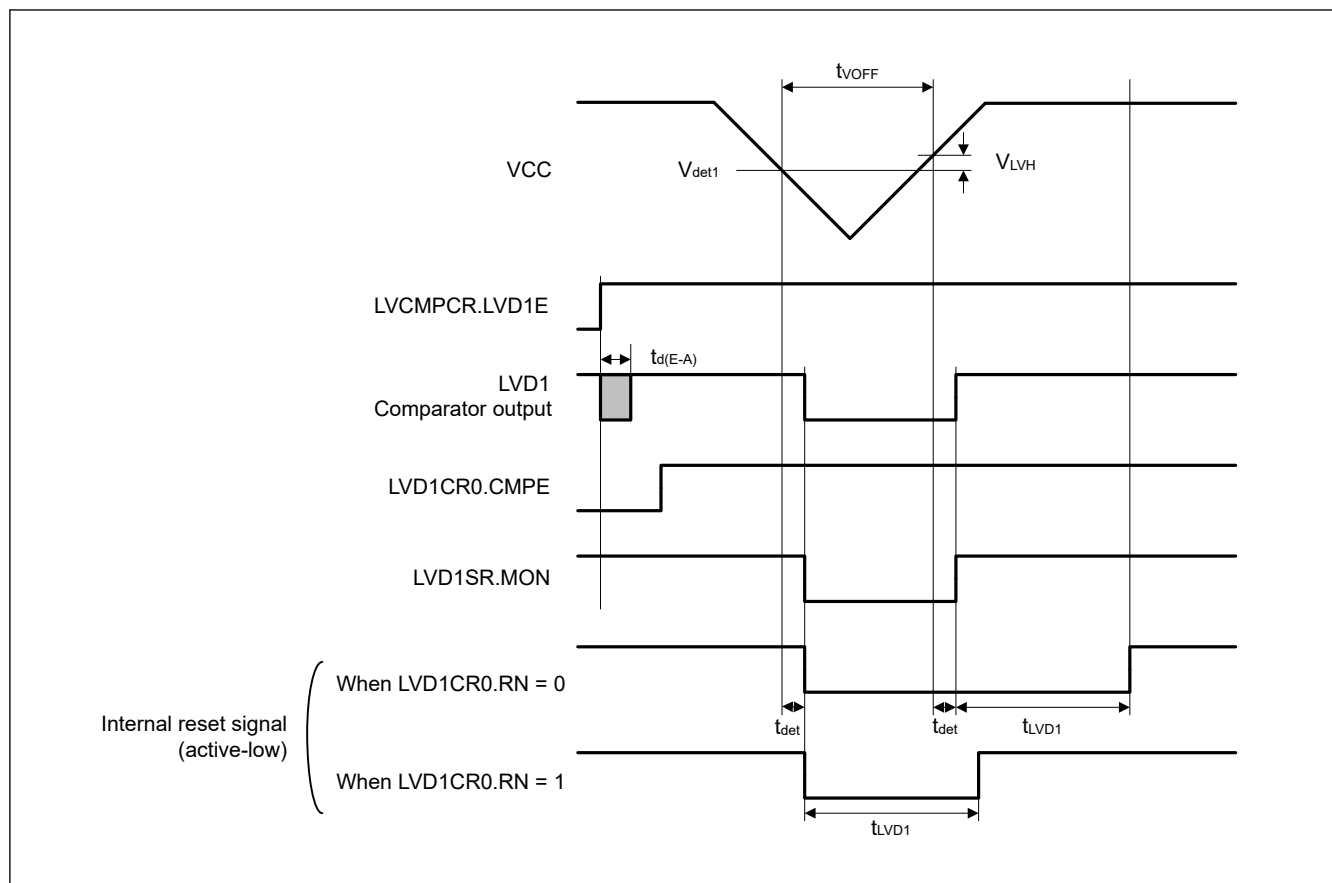
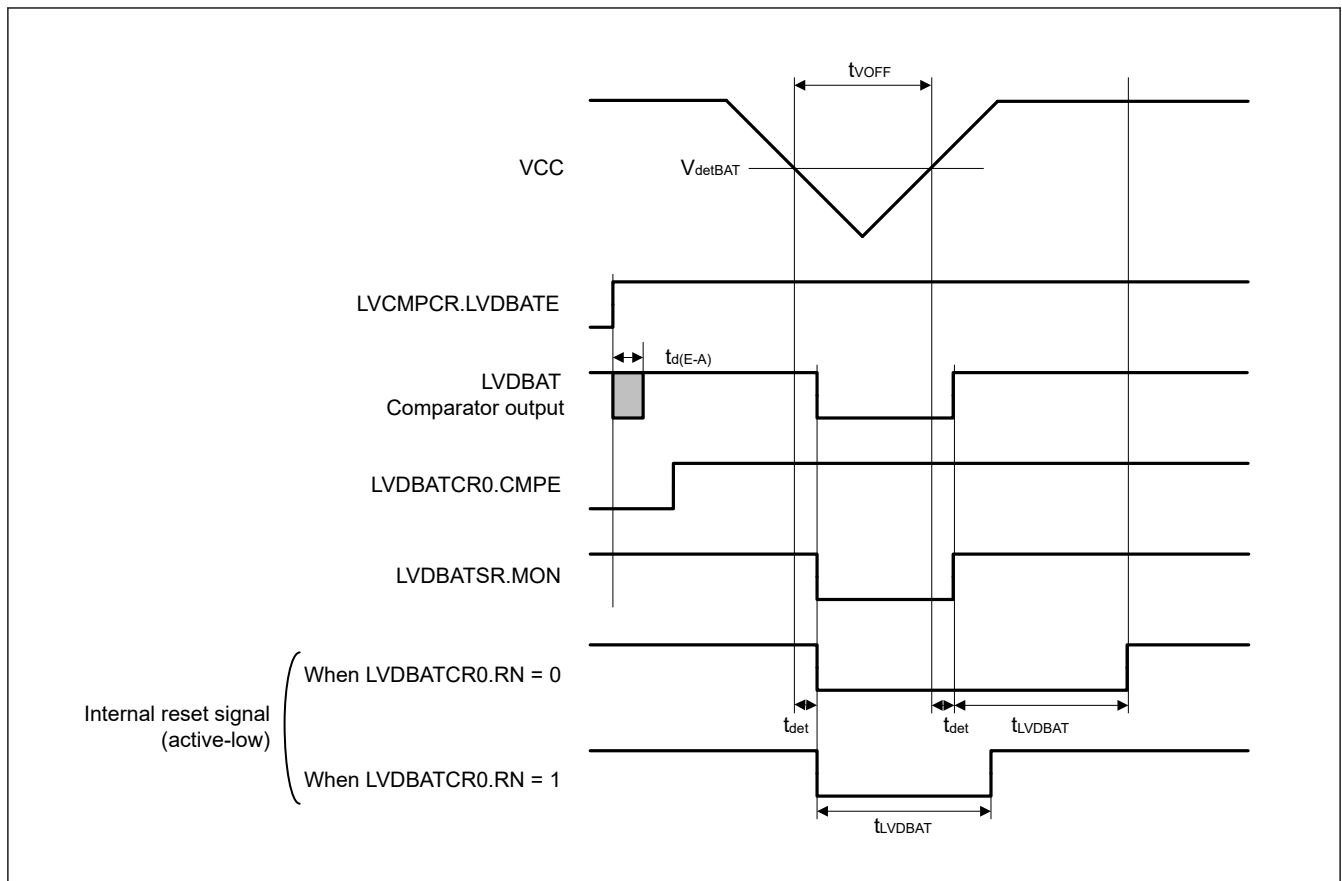


Figure 51.46 Voltage monitoring 1 circuit detection voltage timing (V_{det1})

Figure 51.47 Voltage monitoring BAT circuit detection voltage timing (V_{detBAT})

51.9 EHC Characteristics

Table 51.43 EHC characteristics (1 of 2)

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Current during reset	I_{CC}	—	0.02	—	μA	$T_a = 25^\circ C$ $V_{CC} = V_{SC\_VCC} = 0 V$, $V_{CC\_SU} = V_{BAT\_EHC} = 2.5 V$
Capacitance value of capacitor for electricity accumulation at VCC_SU side*1*3	C_{VCCSU}	—	100	—	μF	EHMD = 1 $T_a = -40$ to $60^\circ C$
		—	47	—		EHMD = 0 $T_a = -40$ to $50^\circ C$
		—	150	—		EHMD = 1 $T_a = -40$ to $85^\circ C$
Capacitance value of smoothing capacitor at VCC side*1	C_{VCC}	—	10	—	μF	$T_a = -40$ to $85^\circ C$
Current that can flow from VSC_VCC to the inside of MCU	I_{SC}	—	—	10	mA	$V_{SC\_VCC} \leq 3.6 V$
Current that can flow from VBAT_EHC to IOVCCn*2	I_{VBAT}	—	—	30	mA	—
Current that can flow from VCC/IOVCC to IOVCCn*2	I_{VCC}	—	—	30	mA	—
Permissible value of output impedance at VBAT_EHC side	R_{VBAT}	—	—	10	Ω	$V_{SC\_VCC} \leq 3.6 V$

Table 51.43 EHC characteristics (2 of 2)

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Threshold voltage for charging protection of secondary cells at VBAT side	VBAT_CHG	2.340	2.390	2.440	V	$I_{SC} = 3 \mu A$ to 10 mA, VSC_VCC = VBAT_EHC = 2.4 V
		2.438	2.488	2.538		$I_{SC} = 3 \mu A$ to 10 mA, VSC_VCC = VBAT_EHC = 2.5 V
		2.535	2.585	2.635		$I_{SC} = 3 \mu A$ to 10 mA, VSC_VCC = VBAT_EHC = 2.6 V
		2.633	2.683	2.733		$I_{SC} = 3 \mu A$ to 10 mA, VSC_VCC = VBAT_EHC = 2.7 V
		2.730	2.780	2.830		$I_{SC} = 3 \mu A$ to 10 mA, VSC_VCC = VBAT_EHC = 2.8 V
		2.827	2.877	2.927		$I_{SC} = 3 \mu A$ to 10 mA, VSC_VCC = VBAT_EHC = 2.9 V
		2.924	2.974	3.024		$I_{SC} = 3 \mu A$ to 10 mA, VSC_VCC = VBAT_EHC = 3.0 V
		3.020	3.070	3.120		$I_{SC} = 3 \mu A$ to 10 mA, VSC_VCC = VBAT_EHC = 3.1 V
Threshold voltage for charging protection of secondary cells at VCC side	VCC_CHG	3.021	3.071	3.121	V	$I_{SC} = 3 \mu A$ to 10 mA, VSC_VCC = VCC
Threshold voltage for high-speed startup of EHC capacitor charging at H side	VCC_SU_H	—	2.63	—	V	VSC_VCC = VCC and rise of VCC VBAT_EHC = 2.4 to 2.7 V
		—	2.92	—		VSC_VCC = VCC and rise of VCC VBAT_EHC = 2.8 to 3.1 V
Threshold voltage for high-speed startup of EHC capacitor charging at L side	VCC_SU_L	—	2.33	—		VSC_VCC = VCC and drop of VCC VBAT_EHC = 2.4 to 2.7 V
		—	2.61	—		VSC_VCC = VCC and drop of VCC VBAT_EHC = 2.8 to 3.1 V
Startup threshold voltage at the time of starting up the energy harvest mode	VCC_SU_H	—	2.60	—	V	$I_{SC} = 3 \mu A$ to 10 mA
Power generation status flag	V _{ENOUT}	—	0.42	—	V	VCC_SU = 2.5 V
Minimum startup current required for starting up the energy harvest mode	I _{SC}	—	3	—	μA	Ta = 25 °C, Connect capacitors of 100 μF to VCC_SU and 10 μF to VCC.

Note 1. Refer to the block diagram in the Energy Harvest Control Circuit (EHC) section.

Note 2. IOVCCn indicates IOVCC0 and IOVCC1.

Note 3. [Figure 51.49](#) shows the relationship between the upper limit of temperature and the capacitance value of capacitor for electricity accumulation at VCC_SU side.

If the capacitance value is insufficient for the temperature used, the startup current shown in [Figure 51.50](#) is required.

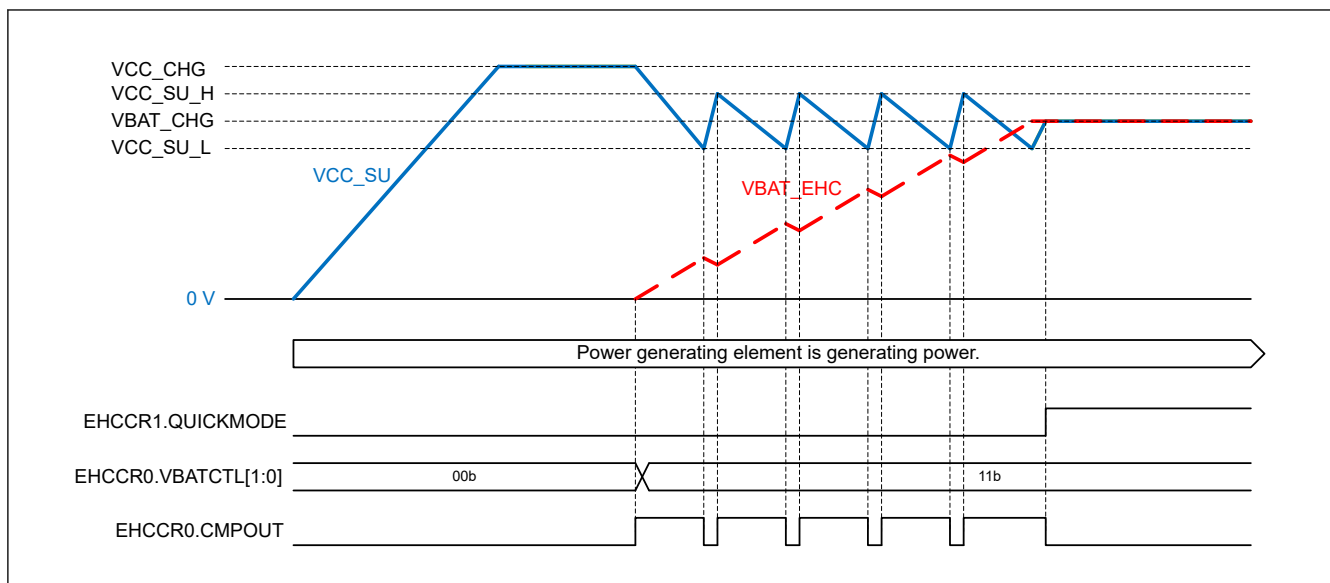


Figure 51.48 VBAT_EHC pin charging operation during high-speed startup function period of EHC capacitor charging

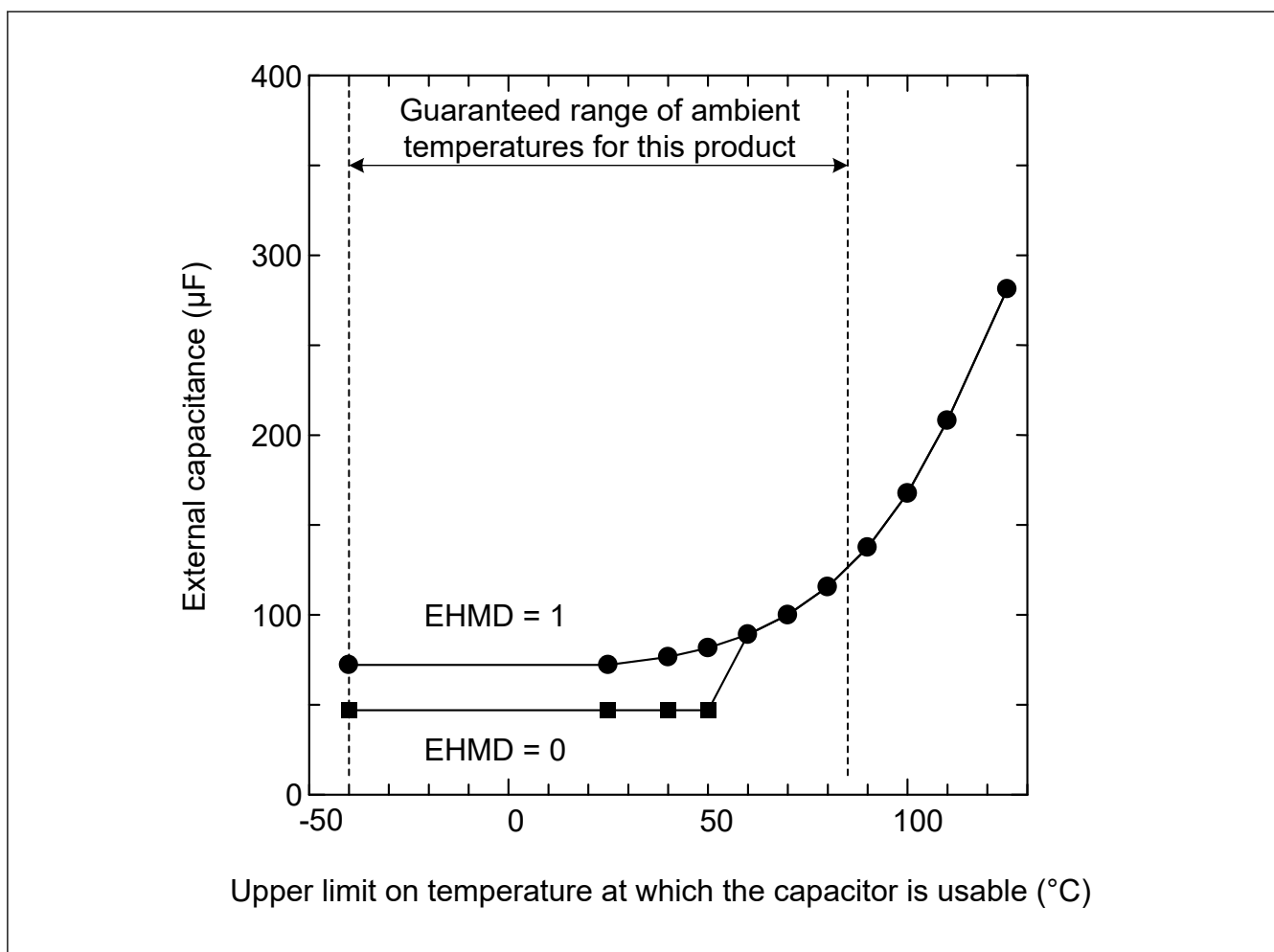


Figure 51.49 Relationship between the upper limit of temperature and the capacitance value of capacitor for electricity accumulation at VCC_SU side

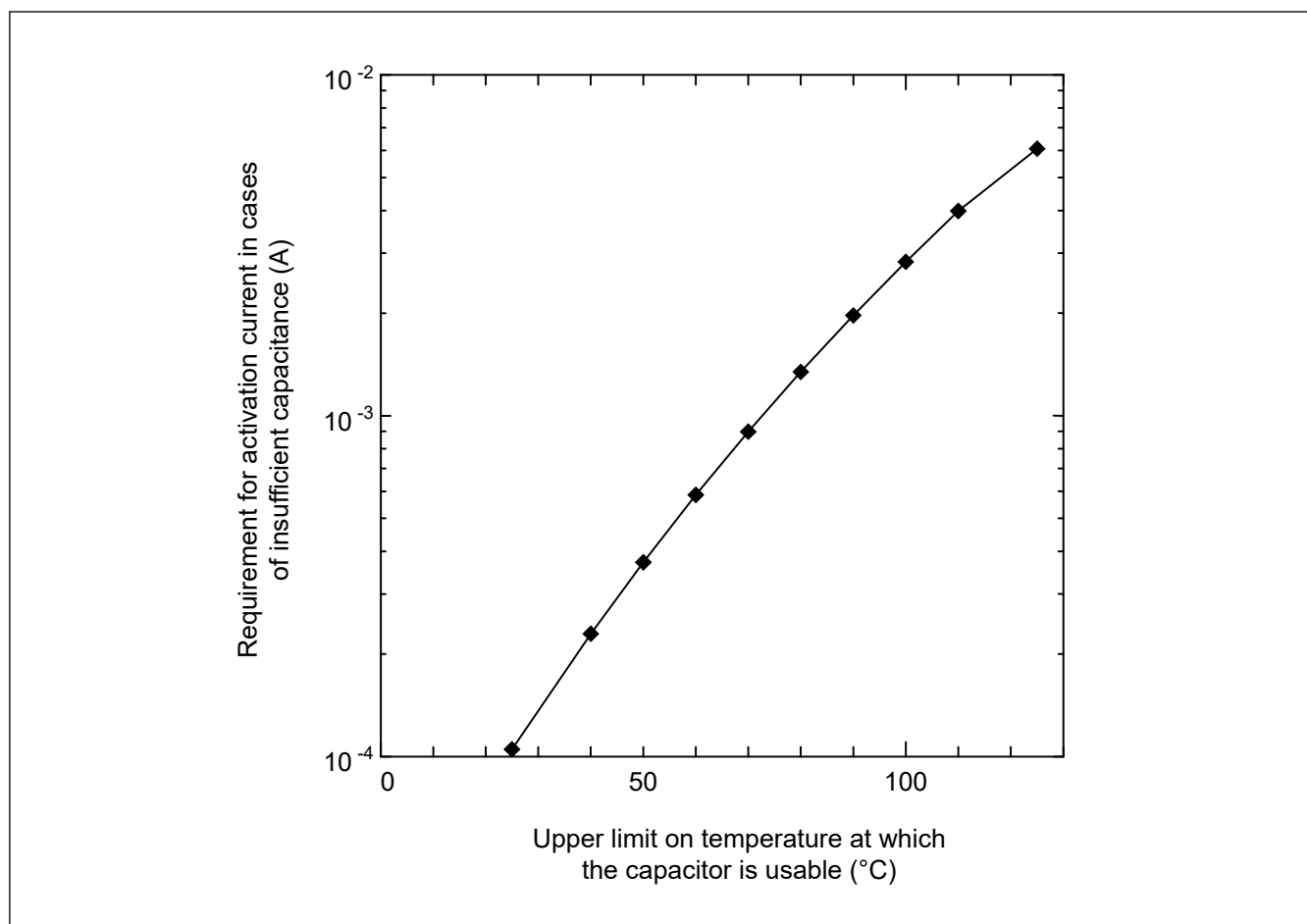


Figure 51.50 Relationship between the upper limit of temperature and the startup current in case that the capacitance is insufficient

51.10 Back Bias Voltage Control (VBBC) Circuit Characteristics

Table 51.44 Initial setup time of VBBC circuit

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
VBBC initial setup time*1	$t_{VBBCSTUP}$	—	100*2	400*2 *3	ms	Figure 51.51
Internal voltage discharge time	$t_{VBBCDIS}$	1	—	—	ms	Figure 51.52

Note 1. This is the time period between when 1 is written to VBBCR.VBBEN and when VBBST.VBBSTUP is changed to 1.

Note 2. This is the time when the value of the smoothing capacitor connected between the VBP and VBN pins is 0.56 $\mu\text{F} \pm 20\%$.

Note 3. We do not inspect the characteristics of the back-bias voltage control circuit before shipment. The values presented in this manual are only for reference.

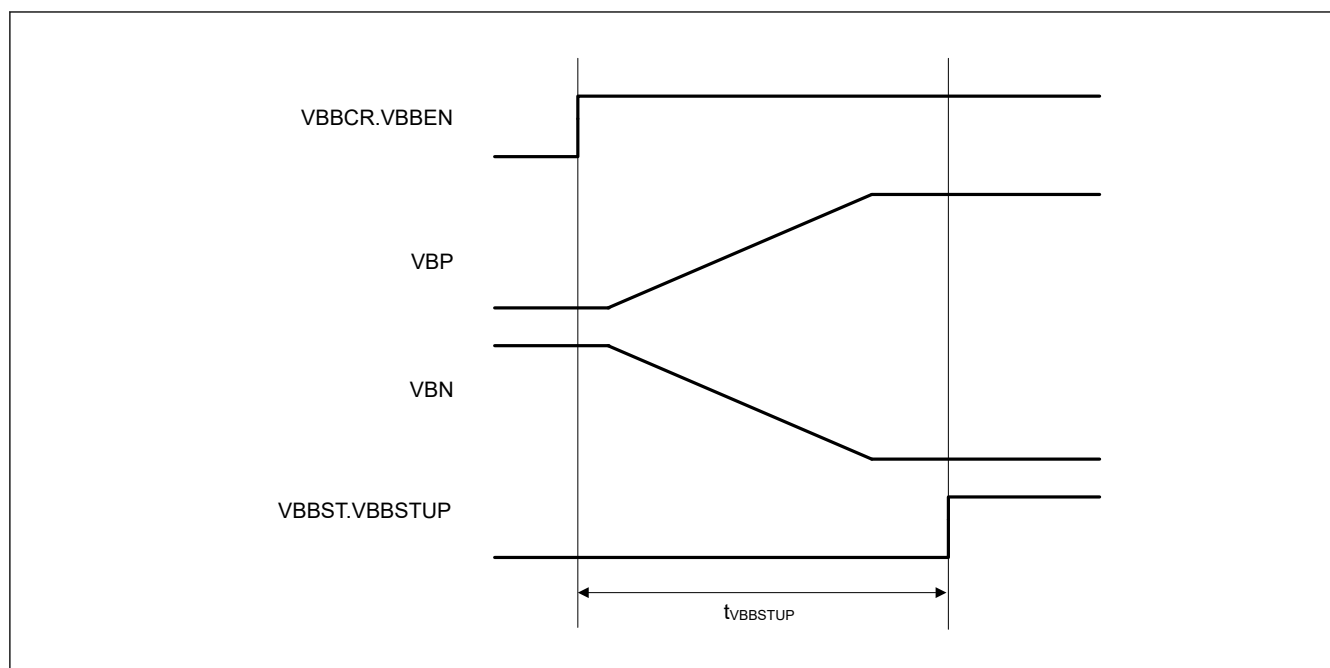


Figure 51.51 VBBC initial setup timing

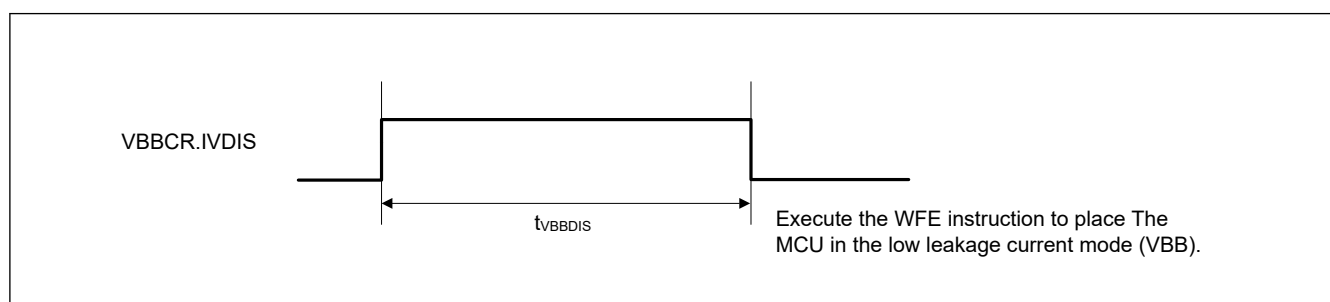


Figure 51.52 Internal voltage discharge time

51.11 Flash Memory Characteristics

51.11.1 Code Flash Memory Characteristics

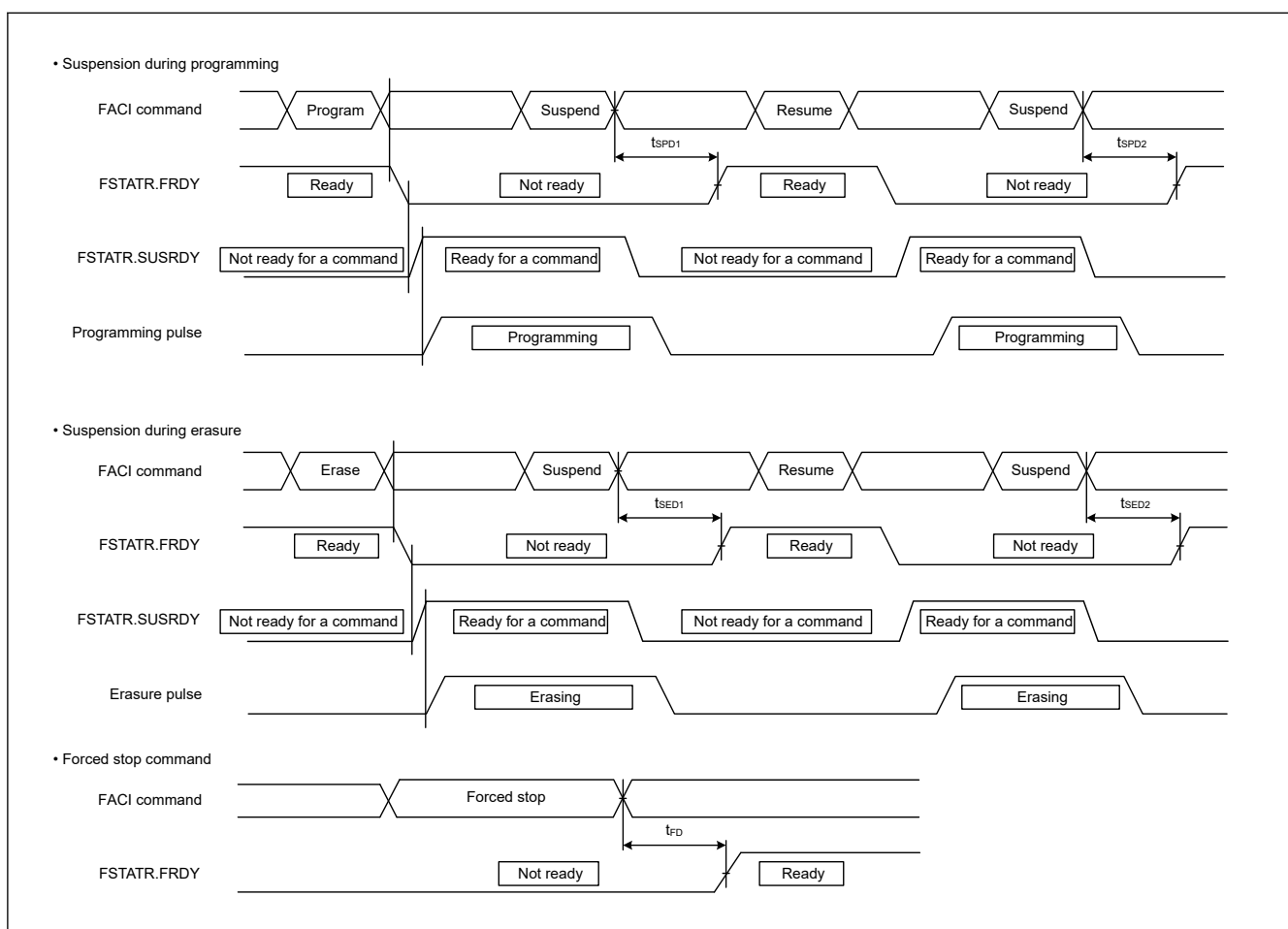
Table 51.45 Code flash memory characteristics (1)

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
Reprogramming/erasure cycle ^{*1}	N _{PEC}	10000	—	—	Times	JEDEC compliance
Data retention time	t _{DRP}	10	—	—	Year	JEDEC compliance

Note 1. The number of cycles of reprogramming and erasure defines the number of times a block can be erased. When the number of cycles of reprogramming and erasure is n, a block can be erased n times. For instance, if 8 bytes of data are written to the 256 different addresses on 8-byte boundaries within a 2-KB block, erasing the whole block is counted as a single cycle of reprogramming and erasure. Note that programming of the same address is only allowed once; that is, overwriting is prohibited.

Table 51.46 Code flash memory characteristics (2)

Item		Symbol	ICLK = 1 MHz			ICLK = 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	8 bytes	t_{P8}	—	5	6	—	5	6	ms
	256 bytes	t_{P256}	—	5	6	—	5	6	
Erasure time	4 KB	t_{E4K}	—	10	12	—	10	12	
Delay until first suspension during programming		t_{SPD1}	—	—	0.2	—	—	0.1	
Delay after second suspension during programming		t_{SPD2}	—	—	2.4	—	—	2	
Delay until first suspension during erasure		t_{SED1}	—	—	0.2	—	—	0.1	
Delay after second suspension during erasure		t_{SED2}	—	—	2.4	—	—	2	
Forced stop command		t_{FD}	—	—	0.2	—	—	0.1	

**Figure 51.53 Code flash memory command timings of program suspend, erase suspend, and forced stop**

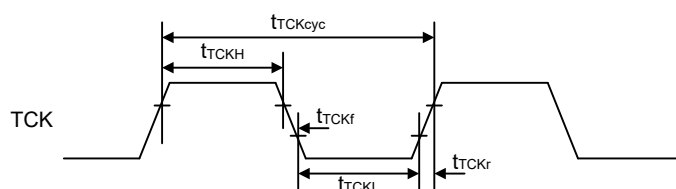
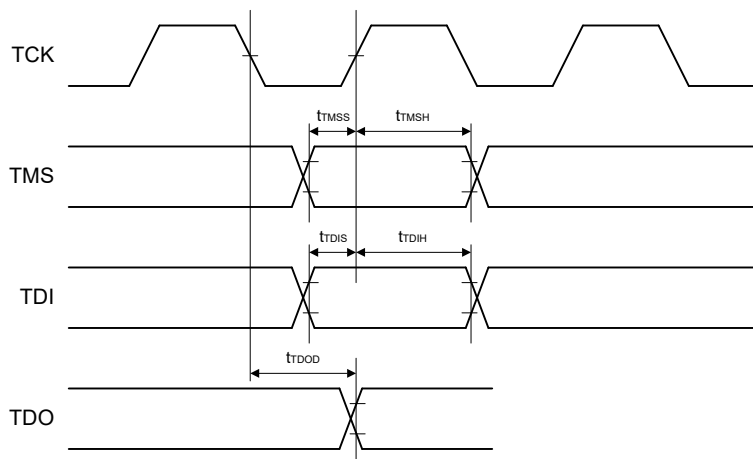
51.12 Boundary Scan Characteristics

Table 51.47 Boundary scan characteristics

Condition: High drive output is selected in the drive capability control bits in PmnPFS register. (PmnPFS.DSCR[1:0] = 11b)

Item	Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
TCK clock cycle time	t_{TCKcyc}	100	—	—	ns	Figure 51.54
TCK clock high-level pulse width	t_{TCKH}	43	—	—		
TCK clock low-level pulse width	t_{TCKL}	43	—	—		
TCK rise time	t_{TCKr}	—	—	7		
TCK fall time	t_{TCKf}	—	—	7		
TMS setup time	t_{TMSS}	15	—	—	ns	Figure 51.55
TMS hold time	t_{TMSH}	15	—	—		
TDI setup time	t_{TDIS}	15	—	—		
TDI hold time	t_{TDIH}	15	—	—		
TDO data delay time	t_{TDOD}	—	—	100		

Note: This is normal mode (high-speed mode).

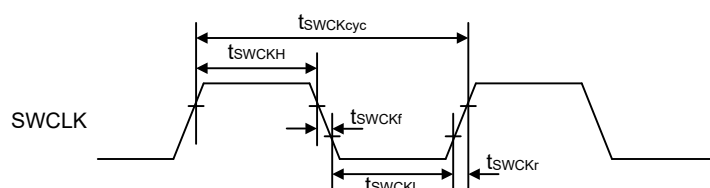
**Figure 51.54 Boundary scan TCK timing****Figure 51.55 Boundary scan input/output timing**

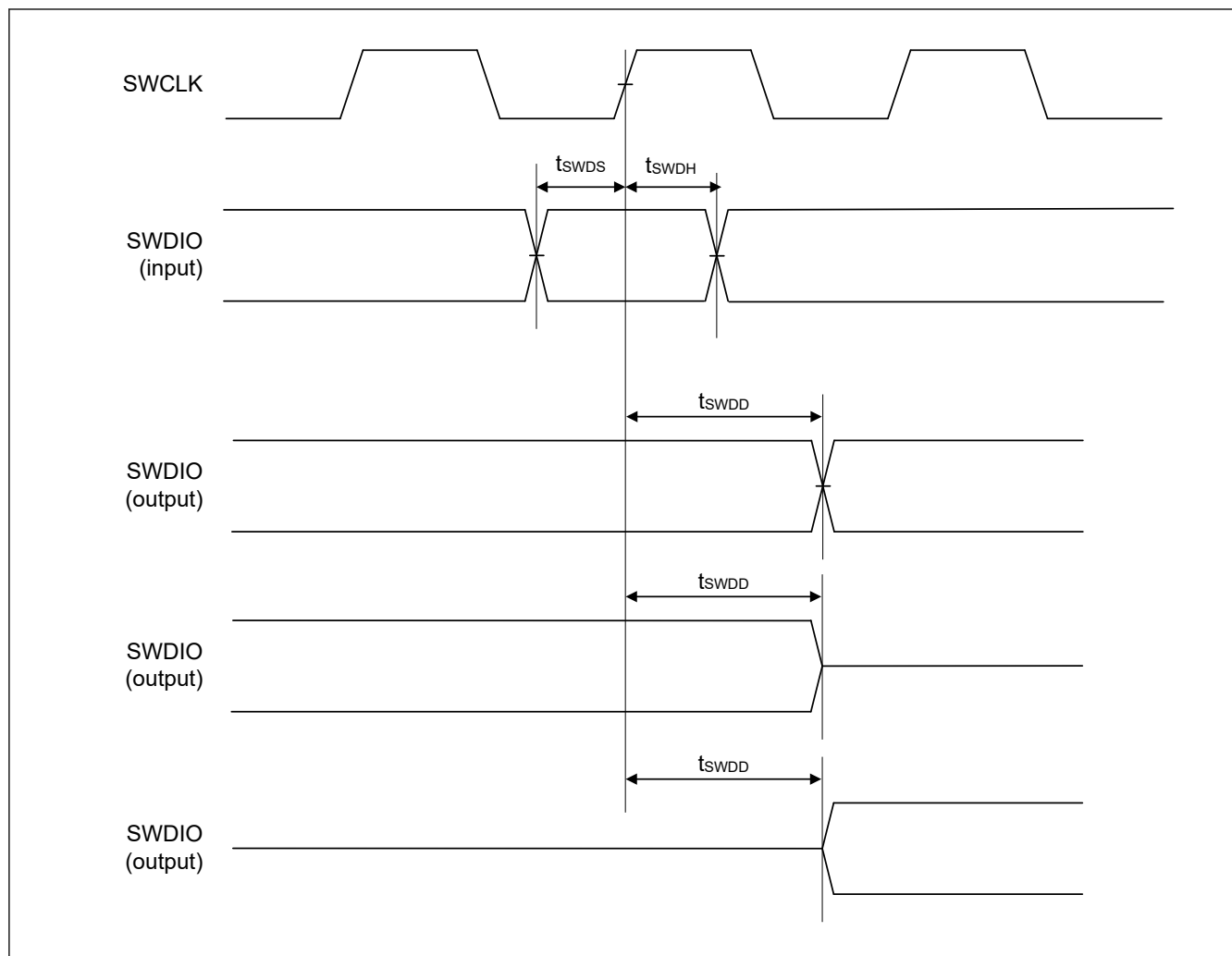
51.13 Serial Wire Debug (SWD) Characteristics

Table 51.48 SWD characteristics

Condition: VCC = AVCC0 = 1.62 to 3.6 V

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement conditions
NORMAL	SWCLK clock cycle time	t_{SWCKcyc}	80	—	—	ns	Figure 51.56
	SWCLK clock high-level pulse width	t_{SWCKH}	$t_{\text{SWCKcyc}} \times 0.5 - t_{\text{SWCKr}}$	—	—	ns	
	SWCLK clock low-level pulse width	t_{SWCKL}	$t_{\text{SWCKcyc}} \times 0.5 - t_{\text{SWCKf}}$	—	—	ns	
	SWCLK rise time	t_{SWCKr}	—	—	7	ns	
	SWCLK fall time	t_{SWCKf}	—	—	7	ns	
	SWDIO setup time	t_{SWDS}	$t_{\text{SWCKcyc}} \times 0.2$	—	—	ns	Figure 51.57
	SWDIO hold time	t_{SWDH}	$t_{\text{SWCKcyc}} \times 0.2$	—	—	ns	
	SWDIO data delay time	t_{SWDD}	2	—	50	ns	
VBB	SWCLK clock cycle time	t_{SWCKcyc}	30000	—	—	ns	Figure 51.56
	SWCLK clock high-level pulse width	t_{SWCKH}	$t_{\text{SWCKcyc}} \times 0.5 - t_{\text{SWCKr}}$	—	—	ns	
	SWCLK clock low-level pulse width	t_{SWCKL}	$t_{\text{SWCKcyc}} \times 0.5 - t_{\text{SWCKf}}$	—	—	ns	
	SWCLK rise time	t_{SWCKr}	—	—	7	ns	
	SWCLK fall time	t_{SWCKf}	—	—	7	ns	
	SWDIO setup time	t_{SWDS}	1000	—	—	ns	Figure 51.57
	SWDIO hold time	t_{SWDH}	1000	—	—	ns	
	SWDIO data delay time	t_{SWDD}	2	—	1000	ns	

**Figure 51.56 SWD SWCLK timing**

**Figure 51.57** SWD input/output timing

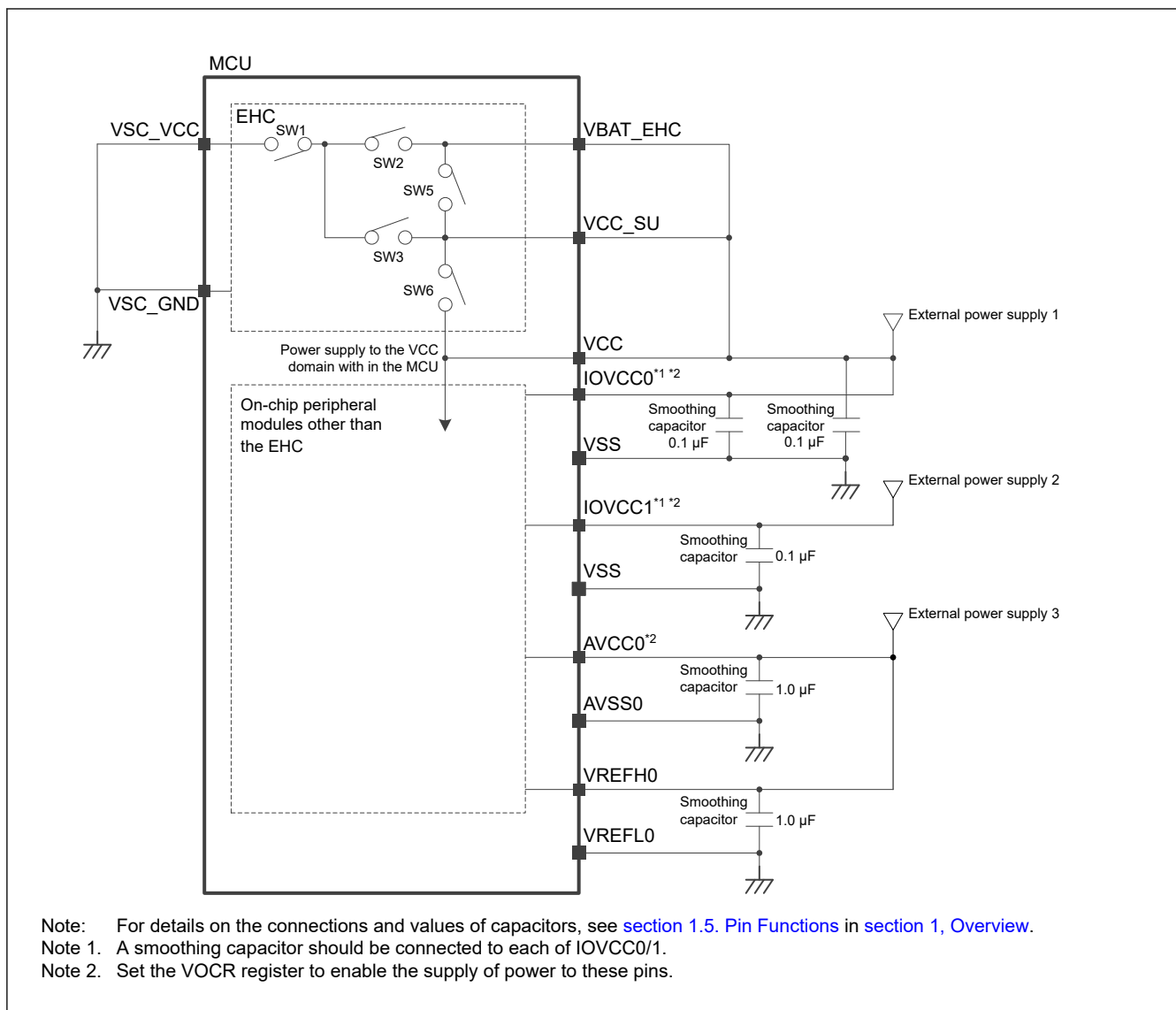


Figure 1.2 Normal startup mode with multiple power sources

1.2 Example of Connections in Energy Harvesting Startup Mode (1)

[Figure 1.3](#) shows an example of connections in energy harvesting startup mode with the EHC and VREF in use, and no external power supplies. [Figure 1.4](#) shows an example where AVCC0 is the reference voltage.

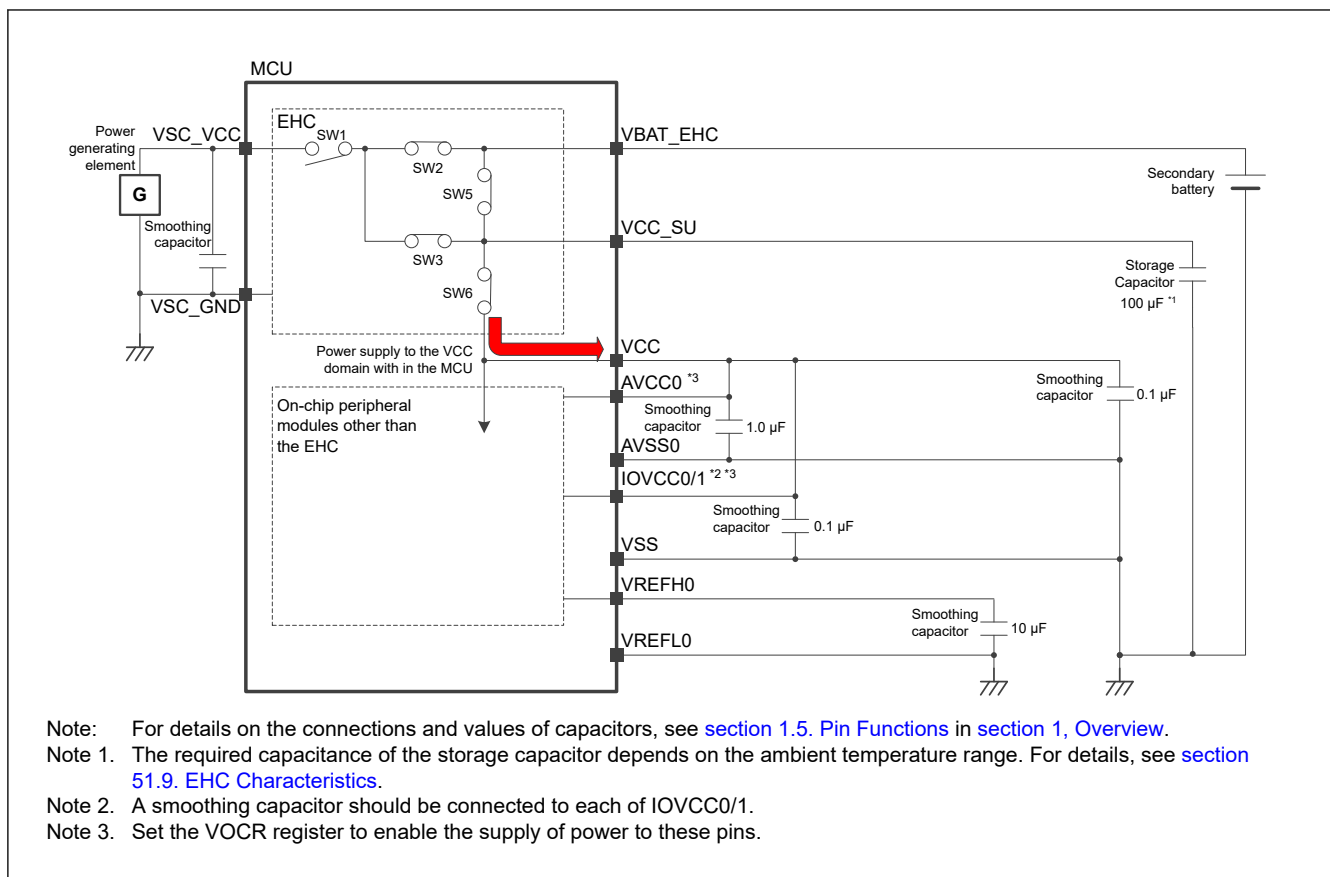


Figure 1.3 Energy harvesting startup mode with the VREF in use

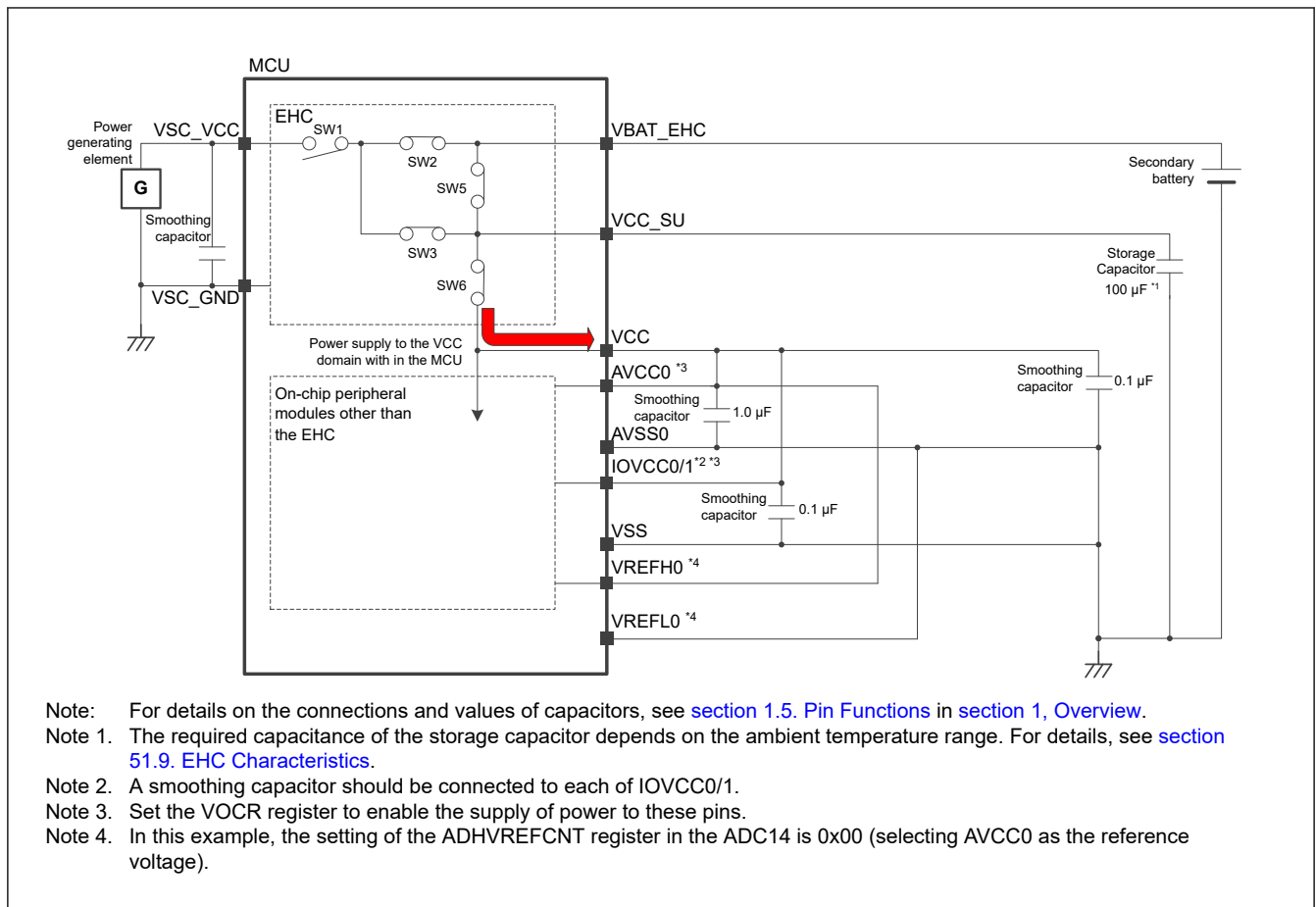


Figure 1.4 Energy harvesting startup mode with AVCC0 as the reference voltage

1.3 Example of Connections in Energy Harvesting Startup Mode (2)

[Figure 1.5](#) shows an example of connections in energy harvesting startup mode with the EHC in use and separate power sources for the analog circuits. [Figure 1.6](#) shows an example of a connection when an analog circuit is not used. [Figure 1.7](#) shows an example of minimum connections in energy harvesting startup mode.

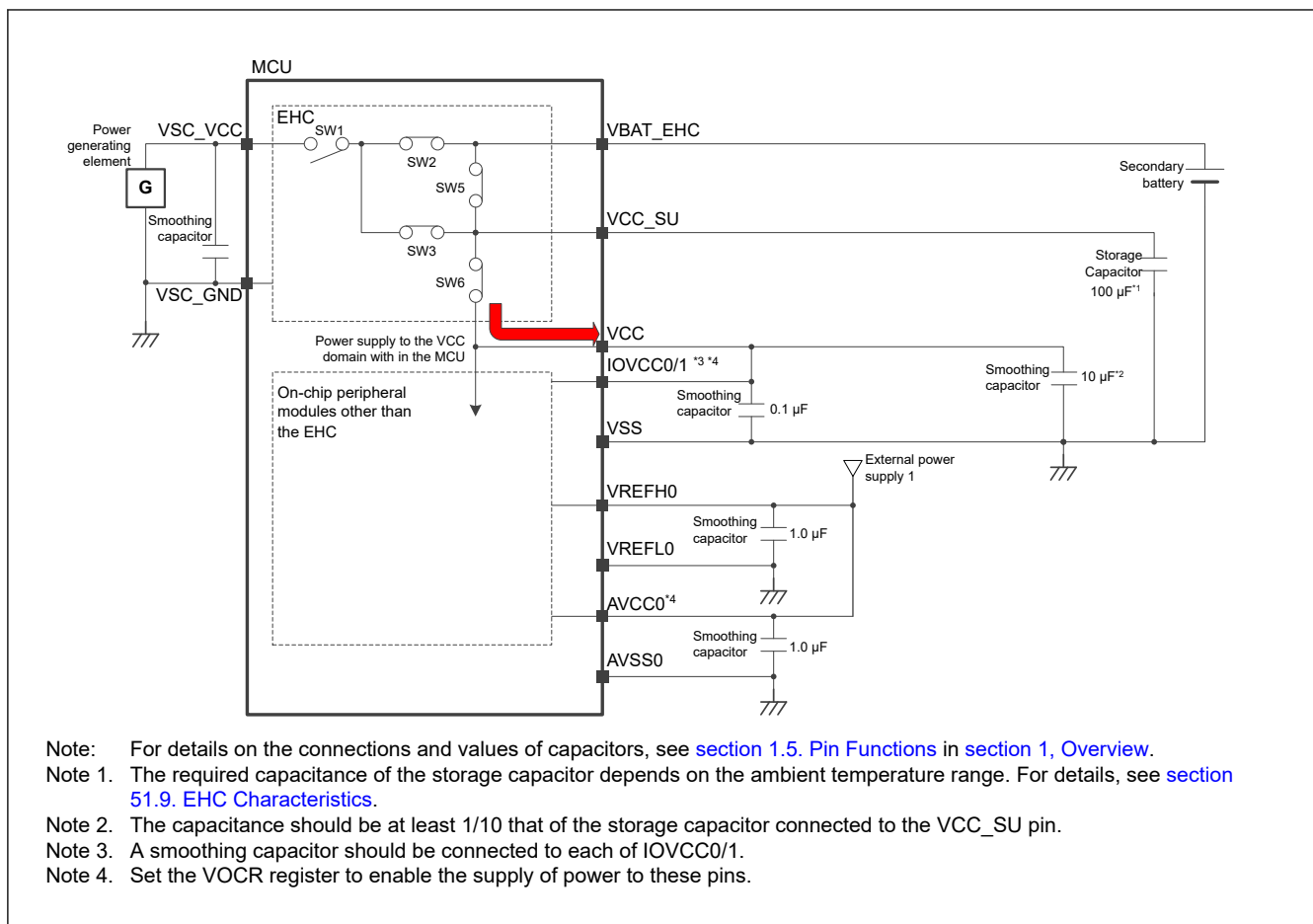
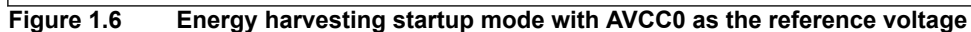


Figure 1.5 Energy harvesting startup mode with the VREF in use



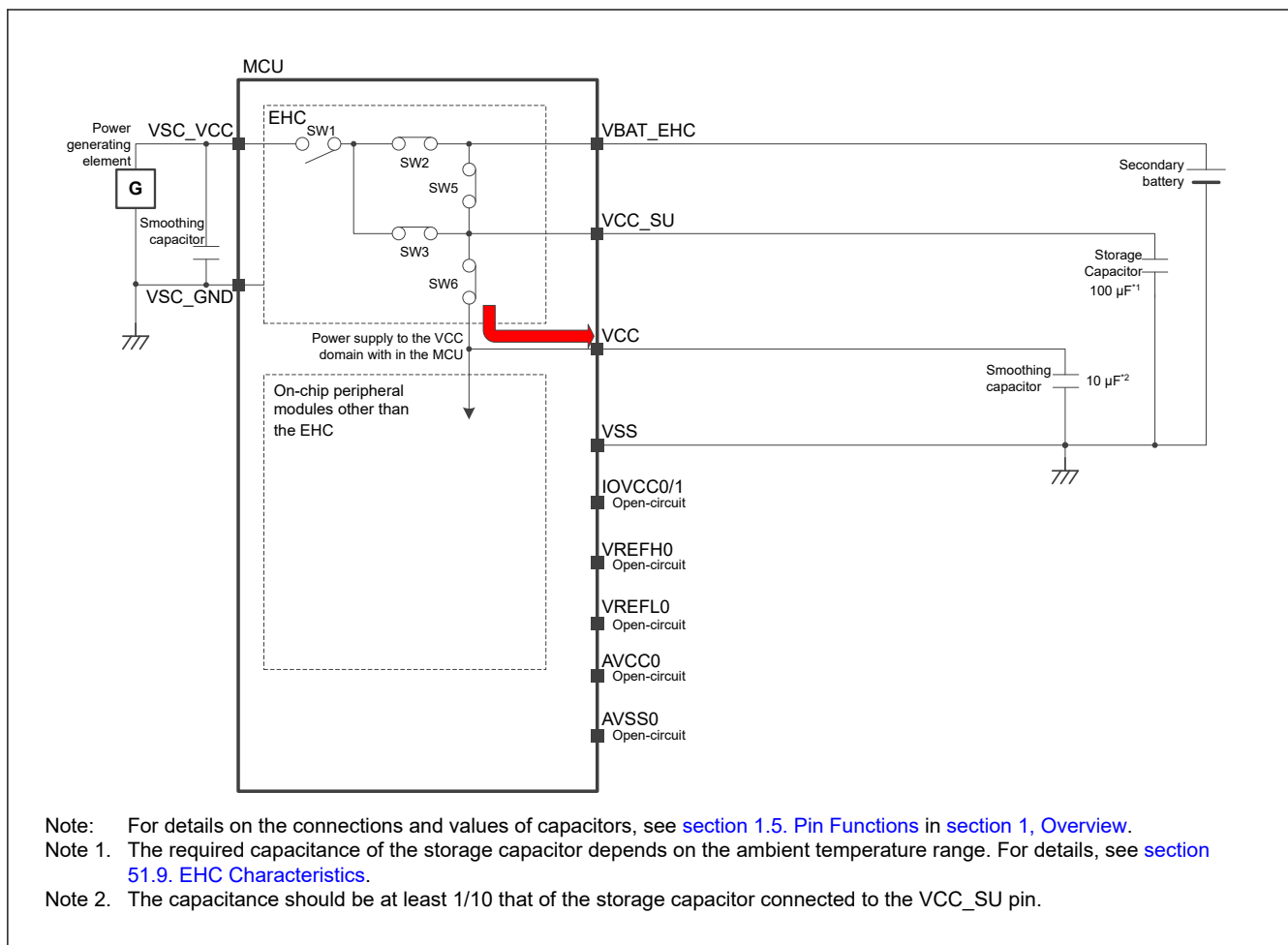
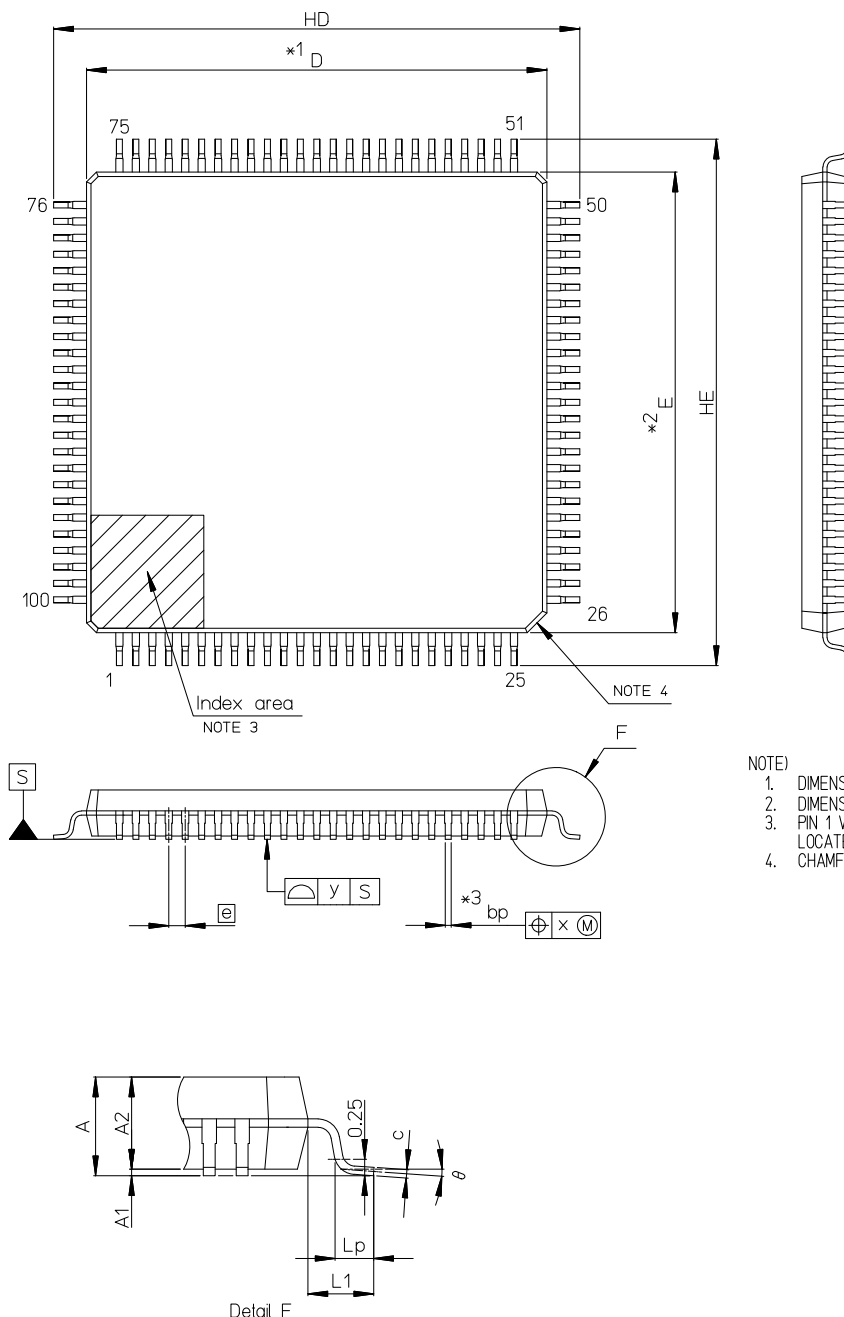


Figure 1.7 Energy harvesting startup mode as a minimum connection

Appendix 2. Package Dimensions

Information on the latest version of the package dimensions or mountings is displayed in “Packages” on the Renesas Electronics Corporation website.

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP100-14x14-0.50	PLQP0100KB-B	—	0.6g



- 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	Dimension in Millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A2	—	1.4	—
HD	15.8	16.0	16.2
HE	15.8	16.0	16.2
A	—	—	1.7
A1	0.05	—	0.15
bp	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
Lp	0.45	0.6	0.75
L1	—	1.0	—

Figure 2.1 LFQFP 100-pin



JEITA Package code	RENESAS code	MASS(TYP.)[g]
S-XBGA72-2.88x3.16-0.30	SXBG0072MA-A	T.B.D

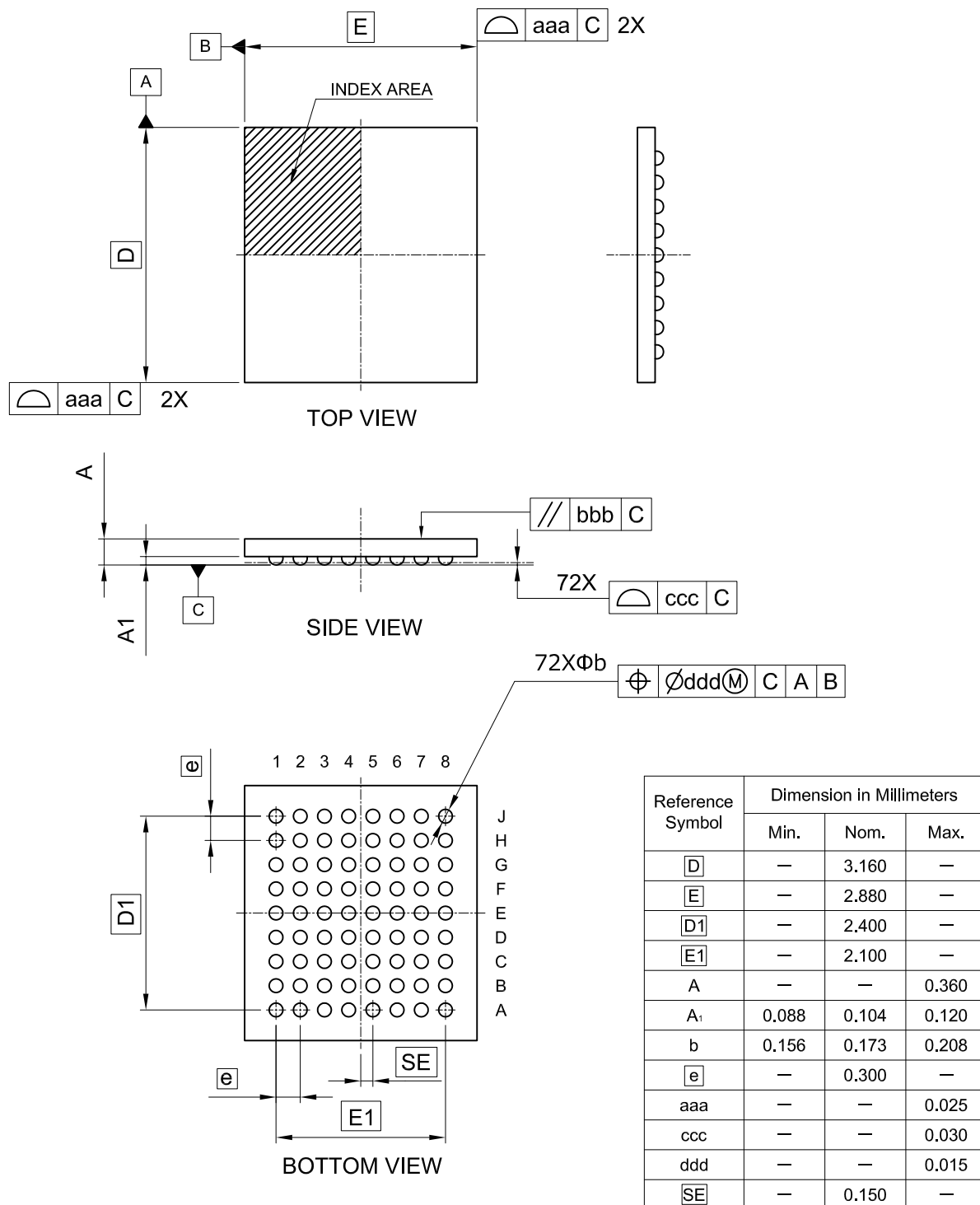


Figure 2.3 WLBGA 72-pin

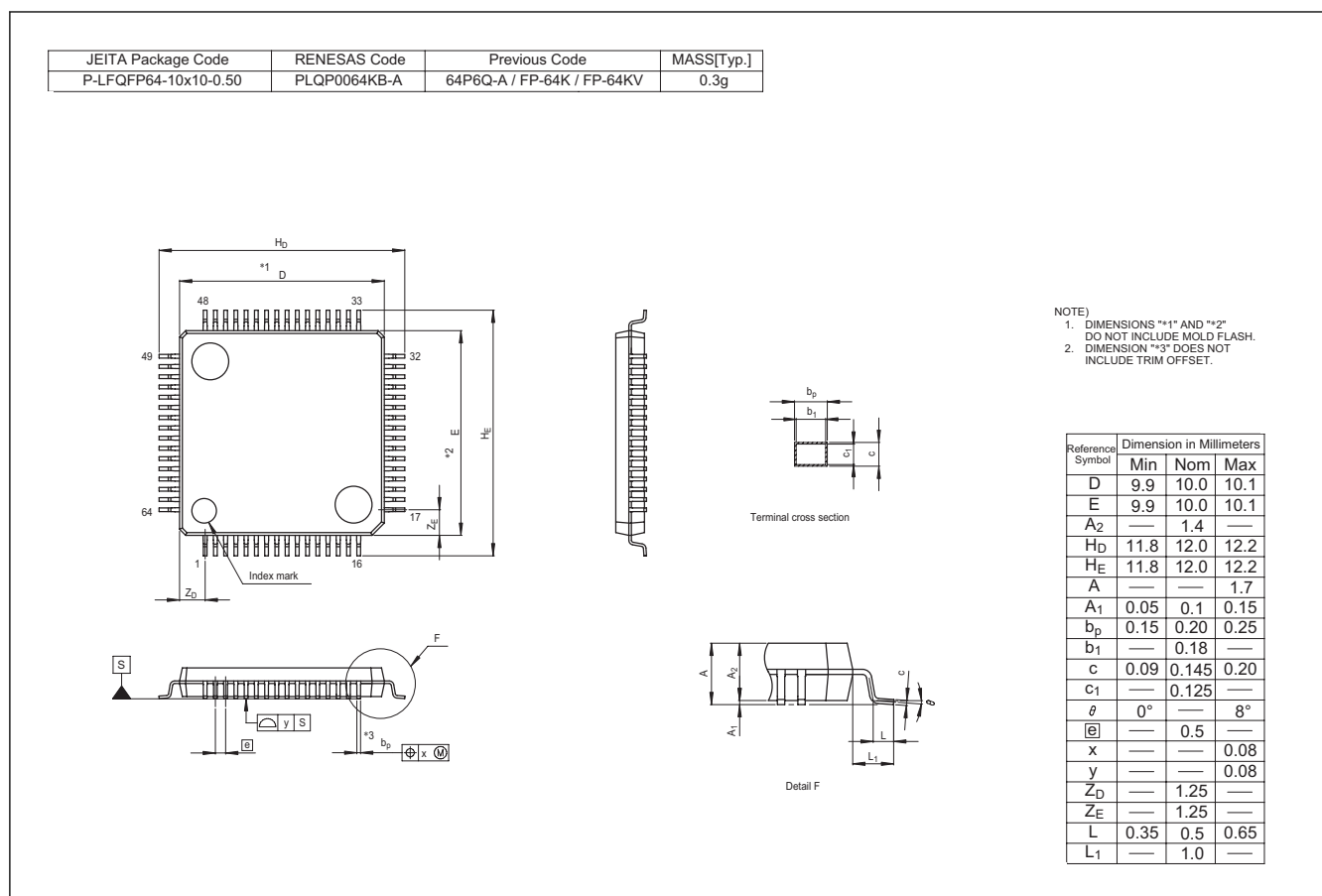


Figure 2.4 LFQFP 64-pin

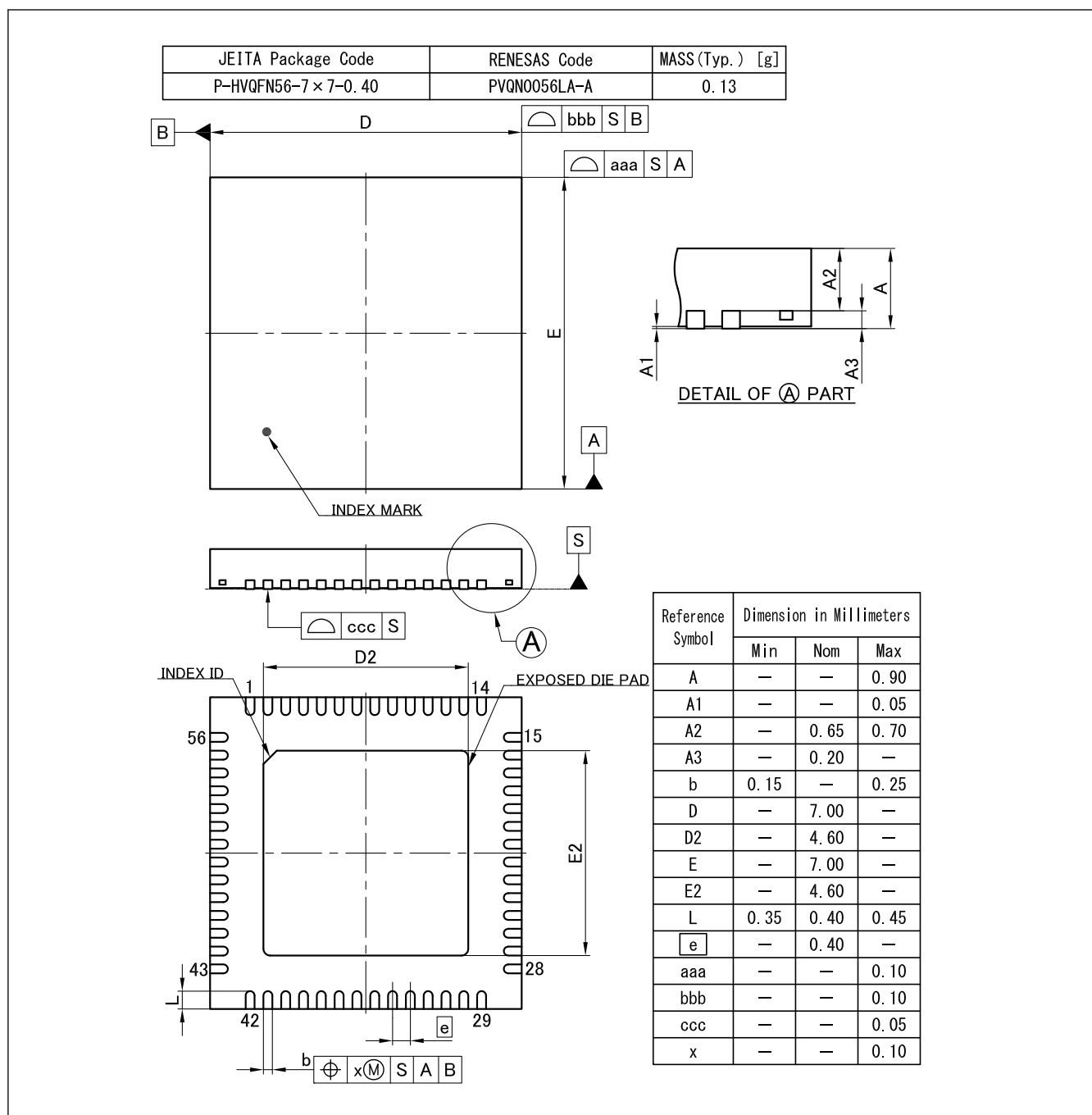


Figure 2.5 QFN 56-pin

Revision History

Revision 1.00 — March 31, 2020

First edition, issued

RE01 Group Products with 256-KB Flash Memory

Publication Date: Rev.1.00 Mar 31, 2020

Published by: Renesas Electronics Corporation

RE01 Group